



Юрій ЛАЗАРЄВ

Моделювання на ЕОМ

*Навчальний посібник
для студентів інженерних спеціальностей
вищих навчальних закладів*

Київ Політехніка 2007

Лазарев Юрій Федорович

Л17 Моделювання на ЕОМ. Навчальний посібник. - К.: Політехніка, 2007. - 290 с.

Викладено основні особливості проведення і організації обчислень на ЕОМ, основних чисельних методів і алгоритмів, що застосовуються при розв'язуванні інженерних задач на ЕОМ, особливості їхньої програмної реалізації. Значну увагу приділено засобам цифрової обробки даних і чисельному інтегруванню диференційних рівнянь. При викладенні широко застосовуються засоби комп'ютерної системи MatLAB інженерних обчислень на ЕОМ.

Для студентів вищих технічних навчальних закладів. Може бути корисним початківцям для ознайомлення з чисельними методами і засобами їх використання на ЕОМ і набуття навичок роботи з ними.

Зміст

1. Моделювання і математика	5
1.1. Моделювання і моделі	5
1.2. Математика і реальність	7
1.3. Теорія подібності і аналіз розмірностей	10
1.3.1. Подібність явищ і її ознаки	11
1.3.2. Основні положення теорії подібності	12
1.3.3. Аналіз розмірностей	13
1.3.4. Приклади застосування аналізу розмірностей	14
1.3.5. Застосування теорії подібності для побудови моделей	18
1.4. Етапи розв'язування інженерних задач на ЕОМ	21
1.4.1. Постановка задачі	22
1.4.2. Створення математичної моделі	22
1.4.3. Математичне моделювання	24
1.4.4. Побудова обчислювальної моделі	25
1.4.5. Алгоритм методу	26
1.4.6. Реалізація методу обчислень	28
1.5. Контрольні запитання	29
2. Організація наближених обчислень	30
2.1. Джерела й види похибок	30
2.2. Запис наближених чисел. Правило округлення	32
2.3. Похибки результату при діях із наближеними числами	34
2.3.1. Похибки підсумовування	34
2.3.2. Похибки добутку, ділення й обчислення довільної функції	36
2.4. Поширення похибок округлення при обчисленнях	37
2.5. Контрольні запитання	43
3. Обчислення значень функцій	45
3.1. Обчислення значень поліномів. Схема Горнера	45
3.2. Обчислення елементарних функцій за допомогою рядів	50
3.3. Обчислення функцій за допомогою ланцюгового дробу	54
3.4. Обчислення функцій на ЕОМ	56
3.5. Передатні функції	60
3.6. Обчислення значень частотних передатних функцій	71
3.7. Завдання	74
3.8. Контрольні запитання	85
4. Розв'язування систем лінійних алгебричних рівнянь (СЛАР)	87
4.1. Метод Крамера	87
4.2. Метод Гаусса та його модифікації	88
4.2.1. Схема єдиного ділення	89
4.2.2. Метод Гаусса з обранням роздільного елемента	91
4.2.3. Метод Гаусса-Жордана (метод повного виключення)	92
4.3. LU-розкладення матриці	93
4.4. Обчислення визначника й оберненої матриці	94
4.5. Розв'язування СЛАР на ЕОМ. Матричні операції на ЕОМ	95
4.6. Завдання	104
4.7. Контрольні запитання	106
5. Розв'язування нелінійних алгебричних рівнянь	108
5.1. Метод дихотомії (ділення навпіл)	109
5.2. Метод хорд	110
5.3. Метод дотичних (Ньютона)	112
5.4. Комбінований метод (хорд і дотичних)	114
5.5. Метод ітерацій	115
5.6. Реалізація процесу відшукування коренів на ЕОМ.	117
5.7. Розв'язування поліноміальних рівнянь	121
5.8. Завдання	126
5.9. Контрольні запитання	128
6. Наближення функцій	129
6.1. Постановка задачі наближення функції	129
6.2. Інтерполювання поліноміальне.	131
6.2.1. Інтерполювання за Лагранжем	132
6.2.2. Інтерполювання за Ньютоном	133

6.2.3. Інтерполювання за Ермітом	134
6.3. Інтерполювання сплайнами	135
6.4. Засоби інтерполювання на ЕОМ	139
6.5. Апроксимування. Метод найменших квадратів	141
6.6. Засоби апроксимування на ЕОМ	144
6.7. Приклад створення у середовищі MatLAB програми дослідження похибок апроксимації гармонічної функції	146
6.7.1. Мета створення програми і постановка задачі	146
6.7.2. Створення програми	148
6.8. Завдання	156
6.9. Контрольні запитання	156
7. Чисельне диференціювання і інтегрування функцій	157
7.1. Чисельне диференціювання	157
7.2. Чисельне інтегрування	160
7.3. Засоби чисельного диференціювання і інтегрування на ЕОМ	163
7.4. Контрольні запитання	165
8. Чисельне інтегрування диференціальних рівнянь	166
8.1. Однокрокові методи	166
8.1.1. Загальна постановка задачі чисельного інтегрування диференціальних рівнянь	166
8.1.2. Метод Ейлера	169
8.1.3. Модифікований метод Ейлера	172
8.1.4. Метод Хойне	174
8.1.5. Розповсюдження результатів на довільні системи диференціальних рівнянь	175
8.2. Багатокрокові методи	176
8.3. Загальна характеристика явних методів	182
8.4. Жорсткі системи. Неявні методи	184
8.5. Засоби чисельного інтегрування ЗДР системи MatLAB	185
8.6. Програма досліджування похибок чисельних методів інтегрування ЗДР	186
8.6.1. Постановка задачі створення програми	186
8.6.2. Розроблення програми	191
8.6.3. Деякі попередні результати роботи з програмою	195
8.7. Завдання	198
8.8. Контрольні запитання	199
9. Моделювання процесів і систем на ЕОМ	200
9.1. Особливості моделювання технічних систем	200
9.2. Засоби моделювання системи Matlab. Ознайомлення з пакетом Simulink	201
9.3. Генерування типових процесів	206
9.3.1. Формування імпульсних, неколивальних і нестационарних коливальних процесів	206
9.3.2. Генерування періодичних і стаціонарних коливальних процесів	212
9.4. Засоби математичної обробки сигналів	228
9.5. Спектральний і статистичний аналіз процесів	237
9.6. Моделювання процесів у динамічних системах	251
9.7. Приклади утворення S-моделей	259
9.7.1. Модель руху планет	260
9.7.2. Модель руху маятника з сухим тертям	270
9.8. Контрольні запитання	280
Література	282
Предметний покажчик	283
Покажчик операторів і функцій MatLAB	288

1. Моделювання і математика

1.1. Моделювання і моделі

Моделювання є основою пізнання людиною навколишнього світу. Проводячи експерименти, теоретичні досліджування, навіть обговорювання власних дій, намірів, висновків, ми практично здійснюємо моделювання. Цілі, задачі, засоби й методи моделювання у цих випадках значно відрізняються один від одного, але загальна спрямованість залишається єдиною - одержання нового знання шляхом випробування (досліджування) деякого замітника реального об'єкта дослідження - моделі.

Узагалі, спрощено, моделювання можна розглядати як певний експеримент, об'єктом якого у першому випадку є матеріальний аналог досліджуваного об'єкта, у другому випадку об'єктом випробувань є знакова (математична) модель, у третьому - відношення до моделі, яка обмірковується, з боку громади.

Загалом поняття моделі можна визначити у такий спосіб:

модель – це природний або штучний реальний об'єкт, який має певну відповідність до другого реального об'єкта (оригінала), поводження якого потрібно вивчити, або до деяких сторін оригінала, які потребують вивчення.

У процесі дослідження моделі (який називають моделюванням) вона постає у якості відносно самостійного квазі-об'єкта, який дозволяє шляхом його дослідження одержати деяке опосередковане знання про оригінал.

У випадку експериментальних досліджень моделлю є реальний об'єкт, який має ту саму фізичну природу, що й досліджуваний об'єкт. При теоретичних досліджуваннях модель має знакову форму - математичних формул, співвідношень, рівнянь, а задачею моделювання є встановлення нових знань про об'єкти, що описуються цими співвідношеннями. Обговорення встановлює слушність тих припущень і висновків, які були зроблені, шляхом моделювання відношення до них досвідчених співрозмовників.

Пізнання за допомогою моделювання, як це впливає з зазначеного, складається з наступних етапів:

- 1) побудова моделі або обрання її з існуючих;
- 2) дослідження моделі з потрібного боку – власне моделювання;
- 3) перенесення вивчених властивостей моделі на властивості оригінала, тобто виявлення поводження оригіналу за одержаними відомостями про вивчені властивості моделі.

До моделювання вдаються тоді, коли потребують пізнання деяких властивостей об'єкта вивчення, але при цьому сам об'єкт є недосяжним, або його вивчення наштовхується на значні труднощі і незручності. За таких обставин одним з найважливіших етапів здійснення моделювання є утворення спеціального реального об'єкта дослідження – моделі – з такими властивостями:

- він має властивості, що є подібними до відповідних властивостей об'єкта досліджування (оригінала), які потребують досліджування;

- він є більш досяжним, більш простим і зручним для досліджування, аніж оригінал, щоб можливо було безпосередньо дослідити бажані властивості.

Як бачимо, щоб утворити (побудувати) модель, з одного боку, потрібно щось знати про об'єкт досліджування (щоб бути впевненим, що в ній збережені досліджувані властивості), хоча, з іншого боку, знання про об'єкт досліджування обов'язково є неповними (інакше створення моделі не матиме сенсу). Тобто модель не потрібно утворювати, коли в об'єкті досліджування не має властивостей, які потрібно дослідити, і неможливо створити, коли про об'єкт досліджування не відомо нічого.

У подальшому обмежимося розглядом науково-технічного моделювання, тобто моделювання технічних (штучно створюваних) об'єктів на ґрунті наукових знань про поведінку таких об'єктів. Наукові знання базуються на математичному формулюванні головних властивостей об'єкта. Тому опис поведінки об'єкта у вигляді математичних співвідношень, рівнянь (алгебричних, диференціальних або інтегральних) є основою як для побудови моделі, так і для перенесення результатів моделювання на оригінал.

Зупинемося більш докладно на головних видах моделей.

Якщо модель відбудовується на основі реального об'єкта тієї самої фізичної природи, що й у оригінала, її називають *фізичною моделлю*.

Перевагою створення фізичної моделі є те, що, завдяки тому, що модель і оригінал підпорядковуються тим самим закономірностям, для створення моделі можна не знати конкретних математичних співвідношень і рівнянь, що описують поведінку оригіналу і моделі. Відповідність властивостей моделі і оригінала забезпечується автоматично.

Недоліком є необхідність:

- створення (проектування і виробництва) реального технічного об'єкта (моделі), хоча зазвичай більш простого, аніж оригінал;
- проведення експериментальних досліджень (випробувань) цього технічного об'єкта.

Це потребує значних витрат коштів, часу, кваліфікованих людських ресурсів, використання коштовного обладнання.

До фізичного моделювання вдаються, у головному, тоді, коли рівняння, що описують процес або явище, є невідомими або наближеними. Тому будується така модель, про яку можна заздалегідь можна сказати, що її рівняння такі самі, як оригінала, причому не знаючи цих рівнянь. Це цілком можливо у випадку, коли модель є зменшеною копією оригінала або є тим самим оригіналом, але в інших умовах або з деякими зміненими масштабами.

Інший підхід – створювати матеріальну модель іншої фізичної природи, закономірності якої є аналогічними до закономірностей фізики оригіналу. Відповідна модель називається *аналоговою*. При цьому зазвичай найбільш зручним є обрання моделі електричної природи, бо електроелементи є найбільш досяжними, невеликими за розміром, дешевими. За їх допомогою досить просто скласти вельми складні системи з різноманітними властивостями. Досить про-

сто і дешево забезпечити випробування електричних систем. Це складає головну перевагу аналогового моделювання. Неважко углядити і недоліки такого моделювання. У цьому випадку потрібно заздалегідь знати математичні співвідношення і рівняння поведження оригіналу, щоб забезпечити їх відворення при побудові аналогової моделі з використанням елементів іншої фізичної природи.

Найбільш дешевим і швидким є *математичне моделювання*, коли природа утвореної моделі є *знаковою*. *Математична* (теоретична) модель заміщує собою оригінал тільки в інформаційному сенсі, не маючи нічого спільного з оригіналом ані в матеріальній його природі, ані в енергетичному відношенні. Спільним є лише взаємозв'язок між реальними фізичними величинами оригінала, з одного боку, і математичним зв'язком між знаками, відображуваними ці величини в математичній моделі – з іншого боку.

Суто знакова (теоретична) модель оперує лише знаками, що відповідають фізичним величинам оригіналу. Задачею теоретичного моделювання є відшукати узагальнений розв'язок рівнянь, які подають математичну модель, не задаючи конкретні числові значення знакам в рівняннях, що відповідають незмінним величинам. Якщо вдається відшукати такий узагальнений теоретичний розв'язок, то стає можливим дослідити поведження оригіналу за довільних значень параметрів і, отже, обрати такі значення цих параметрів (якщо їх можна на практиці змінювати у широких межах), які забезпечували би найоптимальніше поведження оригіналу (технічного об'єкта). Тому, якщо вдається відшукати такий розв'язок, теоретичне моделювання є найбільш доцільним.

Але математика не є всесильною. Загальний теоретичний розв'язок вдається відшукати далеко не для усіх видів рівнянь. Наприклад, такий розв'язок мають лише лінійні звичайні диференціальні рівняння з постійними коефіцієнтами. Загальних методів розв'язування нелінійних диференціальних рівнянь і лінійних зі змінними коефіцієнтами наразі не існує. У цих випадках доцільніше відшукувати розв'язки рівнянь чисельними методами, застосовуючи обчислювальну техніку. Програмну модель, призначену для чисельного моделювання можна називати *чисельною математичною моделлю*. Саме про побудову таких моделей і проведення моделювання за ними йдеться далі.

1.2. Математика й реальність

Розповсюджений погляд, що математика - це специфічна мова. Ця думка має певне підґрунтя. Математика має усі ознаки мови. У зв'язку з цим постають деякі практичні питання, пов'язані із застосуванням математики у житті.

Завдяки певним рисам сучасного викладання математики у школі, іноді частина випускників сприймає математику як зібрання (зведення) деякої кількості правил, які мають до дійсності досить мале відношення, а у головному вигадані людьми - математиками. Це уявлення може бути досить стійким і підтримується в учнях завдяки тому, що головне наполягання у викладанні мате-

матиці здійснюється часто-густо не на задачі з життя, а на виконання математичних вправ, в яких головне - не відкрити для себе щось нове у оточуючому житті, а міцно закріпити математичні правила оперування з математичними об'єктами. Це те саме, що при вивченні мови замість опанування змістом нових слів, вивчати лише правила граматичного поєднання слів у речення. Таке уявлення про математику глибоко хибне й шкідливе. Варто нагадати, що саме завдяки досягненням математики, людство спромоглося піднятися на сучасний рівень цивілізації.

Зазначимо, що будь-яка мова складається не лише із правил побудови слів та речень. Найважливішою складовою кожної мови є її змістозна частина, тобто ділянка реальної дійсності, що описується за допомогою цієї мови. Без такої ділянки немає і самої мови. Без встановлення змістового зв'язку між словами мови й об'єктами дійсності, які вони позначають, немає сенсу і вести мову про мову. Власне мову і призначено задля відображення частини реальної дійсності, зберігання й передавання інформації про неї.

У математиці як мові є також ділянка дійсності, про яку математика розповідає. Наприклад, арифметика розмовляє з нами про деякі однорідні речі (предмети), надаючи можливість висновувати про їхні кількісні відношення і про їхнє змінювання при реальному оперуванні цими речами.

Коли ми пишемо $m+n$, то розуміємо, що маємо купу з m однакових предметів і іншу купу з n таких предметів і додаємо предмети з другої купи до першої. При цьому неявно припускається, що кожна річ із кожної купи існує окремо, незалежно від інших, має деяку стабільність (не змінюється з часом), займає деяку ділянку простору, може переміщуватися у просторі, не змінюючись, може приєднуватися до інших предметів, не змішуючись із ними. І всі ці особливості не вигадані, вони взяті зі спостережень за реальними речами, наприклад, за стадами тварин тощо. Саме із реальної дійсності узята й сама операція додавання, яка математично узагальнює реальні дії по переміщенню окремих речей з одного місця у друге, де вже розміщено інші аналогічні речі. Саме із практики, завдяки простому перерахуванню, було встановлено, що $2+2=4$. Подібна операція зворотного напрямку (коли з купи речей відбираються окремі речі і переносяться у інше місце) була названа у математиці відніманням. А через те, що практично усі математичні дії походять з операції додавання як головної, то можна висновувати, що уся математика спирається саме на описані властивості речей і дій з ними.

Таким чином, практично усі властивості математичних об'єктів узяті з реальної дійсності і лише децю узагальнені. При цьому варто дати собі раду у тому, що математичні дії й оператори мають відношення зовсім не до будь-якої сфери дійсності, а лише до таких її частин, які мають вищезазначені властивості.

Перш за все до таких властивостей відноситься існування реальної операції додавання, яка має такі властивості:

- асоціативності: $a + b + c = (a + b) + c = a + (b + c)$; ця властивість (результат рахунку не залежить від того, у якому порядку здійснюється додавання) має належати реальній операції додавання речей, які переліковуються;
- комутативності: $a + b = b + a$; результат додавання не залежить від того, до якої купи додаються речі з інших куп; ця властивість теж не є вигаданою, вона має належати реальній операції додавання речей;
- наявність нуля - є місце, а в ньому немає речей; і ця властивість має виконуватися у дійсних операціях із речами, що перераховуються;
- операція додавання має приводити до результату, який кількісно перевищує кожний з доданків.

Якщо хоча б одна із зазначених властивостей на практиці не властива фізичній операції додавання, до цих речей не можна прикладати математичні дії. А таких речей безліч у нашому оточенні.

Перш за все до них відносяться так звані якісні величини. Наприклад, важко уявити собі реальні операції з речами, внаслідок якої можна було б додавати одна до одній гладкість, гіркоту, або твердість. Деякі з величин можна деяким чином вимірювати, наприклад, твердість матеріалів, або гладкість поверхонь. Але якщо для них неможливо вказати операції їхнього фізичного додавання, яка б мала усі зазначені властивості, такі кількісні величини називають екстенсивними. До них, наприклад, можна віднести таку фізичну величину, як температура, а також вищезгадані твердість і гладкість.

Кількісні (тобто такі, які можуть бути тим чи іншим способом виміряні) величини, для яких установлено реальну (фізичну) операцію додавання, називають інтенсивними. До інтенсивних відноситься більшість фізичних величин. Переміщення, маса тіл, електричний струм, напруга, механічна напруга, сила, моменти сил, час - усе це приклади інтенсивних величин. Деякі з цих величин мають власну фізичну операцію додавання, інші - ні, але можуть бути подані як деякі прості функції від тих величин, що мають таку операцію.

Наприклад, операцією додавання для довжини (або переміщення) є така, коли початок однієї із двох довжин сполучається з кінцем другої. Результатом при цьому вважається довжина від початку другої довжини до кінця першої. Неважко впевнитися, що за умови розташування довжин вдовж однієї прямої в одному напрямку, така операція матиме усі ознаки операції додавання. У випадку просторового переміщення (або довільного розташування довжин у просторі) аналогічна операція є слушною по відношенню до будь-яких трьох ортогональних напрямків. У цілому в результаті одержуємо правило векторного додавання переміщень у просторі.

Для часу операція додавання може виглядати наступним чином: початок другого процесу сполучається з кінцем першого. Результатом є тривалість від початку першого процесу до кінця другого.

Додавання мас збігається з операцією жорсткого з'єднання мас в одну масу.

Додавання електричних зарядів полягає у об'єднанні зарядів при дотиканні заряджених тіл.

Величини, що є похідними від тих, що мають операцію додавання, також є інтенсивними. Наприклад, інтенсивною величиною є швидкість, яка визначається як результат ділення переміщення на проміжок часу, протягом якого це переміщення здійснюється, а також прискорення матеріальної точки. Аналогічно, електричний струм, що визначається як відношення приросту електричного заряду до проміжку часу, за який цей приріст відбувся, також є інтенсивною величиною.

Виходячи з того, що усі арифметичні дії (математичні операції віднімання, множення, ділення, піднесення до степеня, взяття похідної та інтегрування) є похідними від операції додавання, можна зробити висновок, що у повній мірі математичні висновки торкаються лише інтенсивних величин. Лише по відношенню до цих величин можна застосовувати усі здобутки математики як мови.

Математика (принаймні, це стосується диференціального й інтегрального зчислень, теорії диференціальних і інтегральних рівнянь) - це мова про інтенсивні величини, тобто, повторимо, про величини, які, з одного боку, є вимірюваними (кількісними), а, з іншого боку, мають реальну фізичну операцію додавання.

1.3. Теорія подібності і аналіз розмірностей

Побудова моделі (особливо фізичної і аналогової) потребує виконання наступних попередніх дій.

По-перше, необхідно уточнити задачу моделювання, тобто чітко визначити, які саме властивості оригіналу і в яких нових умовах потребують вивчення.

Далі слід виявити які саме фізичні величини і які параметри описують досліджувані властивості, і які фізичні величини та їхні параметри можуть суттєво впливати на ці властивості (до них обов'язково слід додати ті величини, вивчення впливу яких на поведження оригіналу потрібно вивчити). Усі інші величини і параметри, що характеризують оригінал, утворять множину характеристик оригіналу, які не впливають на досліджування і не входять у сферу цілей моделювання. Майбутню модель тепер слід будувати, встановлюючи останні величини з міркування, щоб модель була якомога більш досяжною, простою і зручною для дослідження. При цьому потрібно слідкувати, щоб суттєві характеристики оригіналу зберігалися у моделі.

Виконання усіх перелічених дій ще не гарантує створення моделі, оскільки невідомо, які значення мають одержати суттєві параметри моделі щоб результати моделювання можна було б перерахувати у параметри поведження

оригіналу без суттєвих похибок. Науково-обрунтовані методи визначення суттєвих параметрів моделі за заданими суттєвими параметрами оригіналу і переносу результатів моделювання на оригінал встановлюються *теорією подібності*.

1.3.1. Подібність явищ і її ознаки

При протіканні будь-якого досліджуваного процесу змінюються деякі величини, що характеризують стан системи з досліджуваного боку. Ці величини називаються *параметрами процесу*. Система, в якій здійснюються процеси, складається з елементів, які характеризуються власними (зазвичай постійними) параметрами, які називають *параметрами системи*.

При досліджуванні механічних процесів до параметрів процесу відносяться величини сил, переміщень, швидкостей, прискорень, а до параметрів системи – маси тіл, моменти інерції, коефіцієнти тертя, в'язкості і т.п. У випадку електричних систем параметрами процесу є величини напруг, струмів, електричних зарядів, а параметрами системи – величини індуктивностей, опорів, ємностей тощо.

Параметри системи або можна вважати незмінними протягом усього досліджуваного процесу, або можна враховувати їхнє змінювання у просторі і часі.

Якщо параметри системи змінюються при змінюванні одного чи кількох досліджуваних параметрів процесу (координат), то вони називаються *нелінійними* і, відповідно, система називається *нелінійною*,

Явище складається з сукупності процесів, що описуються рівняннями, які пов'язують параметри процесу з параметрами системи, записаними в обраній системі відліку. Система головних рівнянь, що визначають певне явище, математично описує механізм цілого класу фізичних явищ. Щоб з нескінченної множини явищ цього класу виділити поодинокі конкретні явища, тобто щоб одержати частковий розв'язок даної системи диференціальних рівнянь, необхідно задати відповідні додаткові умови, так звані *умови однозначності*. Умови однозначності – це ті умови, якими визначаються індивідуальні відмінності явищ певного класу. Вони встановлюються незалежно від механізму процесу, але суттєво впливають на збіг явищ.

До складу умов однозначності відносяться:

- 1) фізичні параметри середовища, що є суттєвими для плину процесу;
- 2) початкові (часові) умови (тільки для нестационарних процесів), тобто поля усіх змінних у початковий момент часу процесу;
- 3) межові умови, тобто умови взаємодії з оточуючим середовищем (зокрема, поля змінних на межах системи).

Сукупність системи основних рівнянь і умов однозначності принципово є достатньою для одержання часткового розв'язку, який визначає досліджуване конкретне явище. Однак точний аналітичний розв'язок фактично може бути одержаний тільки у виняткових випадках. Набагато більш продуктив-

ним є експериментальне досліджування поодиноких явищ з наступним по можливості більш широким узагальненням одержаних часткових результатів. Для цього потрібно мати науково обрнтовані методи такого узагальнення.

Вченням про науково-обрнтовані методи узагальнення результатів поодиноких експериментів і є теорія подібності.

1.3.2. Основні положення теорії подібності

1. Головна задача теорії подібності – виділення серед кожного класу явищ таких груп, у межах яких можливо узагальнення результатів поодиноких експериментів. Розв'язок цієї задачі показує, що науково-обрнтоване узагальнення результатів поодиноких експериментів можливо лише у межах групи подібних явищ.
2. Подібними називаються фізичні явища, в яких є подібними, тобто відповідно пропорційними усі характерні величини

$$\frac{R_{oi}}{R_{mi}} = k_R = const.$$

Тут позначено:

R_{oi} - деяка характерна скалярна або векторна фізична величина i -ої точки оригіналу;

R_{mi} - відповідна характерна фізична величина відповідної i -тої точки моделі.

При цьому для подібності векторів необхідною є також подібність їхніх компонент по осях систем відліку (тобто паралельність векторів):

$$\frac{R_{oiX}}{R_{miX}} = \frac{R_{oiY}}{R_{miY}} = \frac{R_{oiZ}}{R_{miZ}} = k_R = const.$$

3. Необхідними і достатніми умовами подібності фізичних явищ є подібність умов однозначності при тотожності (інваріантності) систем основних диференційних рівнянь (теорема М. В. Кирпичова і А. А. Гухмана).

Отже, необхідними і достатніми умовами подібності є:

- а) фізична подібність – подібність усіх фізичних параметрів середовища;
- б) подібність полів усіх змінних у початковий момент часу процесу (тільки для нестационарних процесів);
- в) подібність умов на межах системи;
- г) інваріантність системи основних диференційних рівнянь по відношенню до подібного перетворення змінних, тобто сумісне виконання рівнянь

$$F(u_1, u_2, \dots, u_n) = 0 \quad \text{та} \quad F(c_1 u_1, c_2 u_2, \dots, c_n u_n) = 0.$$

де u_1, u_2, \dots, u_n - змінні; c_1, c_2, \dots, c_n - множники подібного перетворення.

4. *Необхідною же і достатньою ознакою інваріантності системи основних диференціальних рівнянь по відношенню до подібного перетворення є інваріантність (рівність для відповідних точок полів і відповідних моментів часу) деяких безрозмірних (не маючих фізичної розмірності) степеневих комплексів з величин, що характеризують поведінку оригіналу і моделі. Ці безрозмірні степеневі комплекси прислуговуються у якості кількісних ознак подібності фізичних явищ і тому називаються критеріями подібності.*
5. *Для забезпечення подібності фізичних явищ (окрім подібності умов однозначності) достатньо рівності тільки критерієв, побудованих з величин, що входять в умови однозначності (зазвичай, - параметрів системи і параметрів зовнішніх дій). Тому такі критерії називають визначальними. Рівність решти критеріїв подібності є наслідком подібності фізичних явищ. Тому решта критерієв називаються визначаваними.*
6. *Уточнене формулювання головної теореми теорії подібності: необхідними і достатніми умовами подібності фізичних явищ є рівність визначальних критеріїв подібності при наявності подібності умов однозначності.*

1.3.3. Аналіз розмірностей

Як впливає з теорії подібності, для науково-обґрунтованої побудови моделі потрібно попередньо відшукати критерії подібності, тобто безрозмірні степеневі комплекси з фізичних величин, що характеризують поведінку досліджуваної системи і моделі. Основним методом відшукування критеріїв подібності є аналіз розмірностей. Він дозволяє одержати критерії подібності навіть у тому випадку, коли рівняння, що описують поведінку системи, невідомі, відомі лише усі параметри, які однозначно визначають досліджуваний процес, і фізичні розмірності цих параметрів. Таке знання дається зазвичай експериментом.

Відшукування критеріїв подібності методом аналізу розмірностей зводиться до наступної послідовності дій:

- 1) *обираються необхідні одиниці вимірювання величин, що є суттєвими для опису досліджуваного процесу; наприклад, для механічних систем такими одиницями можуть бути три - T одиниця виміру часу, L - одиниця виміру переміщень і M - одиниця виміру маси; для електричних систем можна обрати такі основні одиниці: T - одиниця виміру часу, Q - одиниця виміру електричного заряду і V - одиниця виміру електричної напруги;*
- 2) *розмірності усіх параметрів системи, включаючи змінні і умови однозначності, виражаються через обрані одиниці виміру;*
- 3) *з числа постійних параметрів системи і зовнішніх сил обираються кілька з незалежними розмірностями (тобто з такими розмірностями,*

щоб жодну з них не можна було б подати у вигляді степеневого комплексу, складеного з розмірностей решти); очевидно, кількість таких базових параметрів буде дорівнювати або меншим за кількість основних одиниць виміру (3 – для механічних систем, 3 – для електричних систем);

- 4) відшукуються безрозмірні степеневі комплекси (критерії подібності) у вигляді

$$\frac{R_i}{A^{\alpha_i} B^{\beta_i} C^{\gamma_i}}$$

де R_i - деякий i -й параметр системи, який не входить до числа базових; A , B і C - базові параметри; α_i , β_i , γ_i - показники степенів, в які потрібно піднести відповідно розмірності параметрів A , B і C , щоб у знаменнику одержати розмірність параметра R_i ;

- 5) далі, прирівнюючи показники при однакових одиницях виміру у чисельнику і знаменнику степеневого комплексу, одержують значення показників степенів α_i , β_i , γ_i , за яких розглядуваний степеневий комплекс стає безрозмірною величиною.

1.3.4. Приклади застосування аналізу розмірностей

Розглянемо коливальну механічну систему, яка складається з масивного тіла масою m , яке може переміщуватися вдовж горизонтальної осі X і з'єднане з нерухомою основою за допомогою пружини з жорсткістю c і демпфера з коефіцієнтом в'язкого тертя f . На тіла діє гармонічно змінювана у часі сила, амплітуда змінювання якої дорівнює F_m , частота змінювання ω , а початкова фаза - ε . Потрібно відшукати критерії подібності такої системи, враховуючи переходний режим.

До числа параметрів системи у цьому випадку відносяться: m - маса тіла, c - жорсткість пружини і f - коефіцієнт тертя.

Умови однозначності складаються з амплітуди сили F_m , частоти її змінювання ω , початкової фази ε і початкових умов: x_0 - початкового відхилення тіла від положення рівноваги і \dot{x}_0 - початкової швидкості.

До параметрів процесу можна віднести: поточний час t протікання процесу, поточне значення x відхилення тіла від положення рівноваги, поточне значення \dot{x} швидкості тіла, поточне значення \ddot{x} прискорення тіла.

Оберемо у якості основних одиниць виміру T одиницю виміру часу, L - одиницю виміру переміщень і M - одиницю виміру маси.

Виразимо розмірності усіх вищезгаданих величин через ці одиниці. Матимемо:

$$\{m\} = M; \quad \{t\} = T; \quad \{x\} = \{x_0\} = L; \quad \{\dot{x}\} = \{\dot{x}_0\} = LT^{-1}; \quad \{\ddot{x}\} = LT^{-2}$$

$$\{F_m\} = \{m\ddot{x}\} = \{m\}\{\ddot{x}\} = MLT^{-2}; \quad \{c\} = \frac{\{F_m\}}{\{x\}} = MT^{-2}; \quad \{f\} = \frac{\{F_m\}}{\{\dot{x}\}} = MT^{-1}.$$

Оберемо як базові три параметра:

m - масу тіла;

c - жорсткість пружини;

F_m - амплітуду змінювання зовнішньої сили.

Неважно впевнитися, що розмірності цих величин є взаємонезалежними.

Тоді безрозмірні степеневі комплекси слід відшукувати у формі:

$$\bar{R}_i = \frac{R_i}{m^{\alpha_i} c^{\beta_i} F_m^{\gamma_i}}.$$

Запишемо розмірність цього степеневого комплексу:

$$\begin{aligned} \left\{ \frac{R_i}{m^{\alpha_i} c^{\beta_i} F_m^{\gamma_i}} \right\} &= \frac{\{R_i\}}{\{m\}^{\alpha_i} \{c\}^{\beta_i} \{F_m\}^{\gamma_i}} = \frac{\{R_i\}}{M^{\alpha_i} (MT^{-2})^{\beta_i} (MLT^{-2})^{\gamma_i}} = \\ &= \frac{M^{r_M} T^{r_T} L^{r_L}}{M^{\alpha_i} (MT^{-2})^{\beta_i} (MLT^{-2})^{\gamma_i}} = \frac{M^{r_M} T^{r_T} L^{r_L}}{M^{(\alpha_i + \beta_i + \gamma_i)} T^{-2(\beta_i + \gamma_i)} L^{\gamma_i}}. \end{aligned}$$

Тут позначено r_M, r_T, r_L - показники степеня у розмірності параметра R_i у які підносяться відповідно одиниці маси, часу і переміщення.

Для забезпечення безрозмірності степеневого комплексу потрібно розв'язати систему лінійних алгебричних рівнянь

$$\begin{cases} \alpha_i + \beta_i + \gamma_i = r_M \\ -2(\beta_i + \gamma_i) = r_T \\ \gamma_i = r_L \end{cases}$$

і визначити у такий спосіб шукані показники $\alpha_i, \beta_i, \gamma_i$.

Неважно впевнитися, що застосування цієї процедури до базових параметрів приводить до того, що відповідні степеневі комплекси дорівнюватимуть одиниці:

$$\bar{m} = 1; \quad \bar{c} = 1; \quad \bar{F}_m = 1.$$

Здійснівши вищевказані дії по відношенню до решти параметрів, одержимо наступні критерії подібності:

$$\begin{aligned} \bar{t} &= t \sqrt{\frac{c}{m}}; & \bar{x} &= x \frac{c}{F_m}; & \bar{\dot{x}} &= \dot{x} \frac{\sqrt{cm}}{F_m}; & \bar{\ddot{x}} &= \ddot{x} \frac{m}{F_m}; \\ \bar{x}_0 &= x_0 \frac{c}{F_m}; & \bar{\dot{x}}_0 &= \dot{x}_0 \frac{\sqrt{cm}}{F_m}; & \bar{\omega} &= \omega \sqrt{\frac{m}{c}}; & \bar{f} &= \frac{f}{\sqrt{mc}}. \end{aligned}$$

Якщо рівняння руху досліджуваної системи має вигляд

$$m \cdot \ddot{x} + f \cdot \dot{x} + c \cdot x = F_m \sin(\omega \cdot t + \varepsilon)$$

і якщо ввести позначення

$$\omega_0 = \sqrt{\frac{c}{m}}; \quad \nu = \frac{\omega}{\omega_0}; \quad \zeta = \frac{f}{2\sqrt{mc}}; \quad \tau = \omega_0 t; \quad \bar{x}' = \frac{d\bar{x}}{d\tau}; \quad \bar{x}'' = \frac{d^2\bar{x}}{d\tau^2},$$

то це рівняння в безрозмірному вигляді набуде форми

$$\bar{x}'' + 2\zeta \cdot \bar{x}' + \bar{x} = \sin(\nu \cdot \tau + \varepsilon). \quad (\text{а})$$

Як бачимо, рівняння у безрозмірному вигляді значно простіше. Його визначають лише три числових параметри - ζ , ν , ε , замість шести у початковому рівнянні.

Здійснимо аналогічні процедури по відношенню до фізичного маятника, момент інерції якого відносно його осі підвісу дорівнює J , опорний маятниковий момент - mgl (m - маса маятника, g - прискорення сили тяжіння, l - зміщення центру мас відносно осі обертання маятника), R - коефіцієнт в'язкого тертя у підшипниках осі обертання маятника. На маятник навколо осі маятника діє момент сил, який змінюється з часом за гармонічним законом з частотою ω , амплітудою M_m і початковою фазою ε .

У цьому випадку параметрами системи є J , R , mgl , параметрами процесу - величини поточного кута φ відхилення маятника від вертикалі, кутової швидкості $\dot{\varphi}$ і кутового прискорення $\ddot{\varphi}$. До умов однозначності слід віднести

Оберемо ті самі основні одиниці виміру: T - часу, L - переміщення і M - маси. Тепер можна виразити розмірності усіх вищевказані параметри у такий спосіб:

φ , ε - безрозмірні величини;

$$\{J\} = ML^2; \quad \{mgl\} = ML^2T^{-2}; \quad \{R\} = ML^2T^{-1}; \quad \{t\} = T;$$

$$\{\dot{\varphi}\} = \{\dot{\varphi}_0\} = T^{-1}; \quad \{\ddot{\varphi}\} = T^{-2}; \quad \{M_m\} = ML^2T^{-2}; \quad \{\omega\} = T^{-1}.$$

Тут вже неможливо обрати три базові параметри, а лише два. Оберемо базовими параметрами момент інерції J і опорний момент mgl . Тоді відповідні критерії подібності дорівнюватимуть:

$$\bar{J} = 1; \quad \bar{mgl} = 1; \quad \tau = \bar{t} = t \sqrt{\frac{mgl}{J}}; \quad \bar{R} = \frac{R}{\sqrt{mgl \cdot J}}; \quad \bar{\omega} = \omega \sqrt{\frac{J}{mgl}};$$

$$\bar{\dot{\varphi}} = \dot{\varphi}' = \dot{\varphi} \sqrt{\frac{J}{mgl}}; \quad \bar{\dot{\varphi}}_0 = \dot{\varphi}'_0 = \dot{\varphi}_0 \sqrt{\frac{J}{mgl}}; \quad \bar{\ddot{\varphi}} = \ddot{\varphi}'' = \ddot{\varphi} \frac{J}{mgl}; \quad \bar{M}_m = \frac{M_m}{mgl}.$$

Введемо позначення:

$$\omega_0 = \sqrt{\frac{mgl}{J}}; \quad \tau = \omega_0 t; \quad \varphi' = \frac{d\varphi}{d\tau}; \quad \varphi'' = \frac{d^2\varphi}{d\tau^2}; \quad \nu = \frac{\omega}{\omega_0};$$

$$\zeta = \frac{\bar{R}}{2} = \frac{R}{2\sqrt{mgl \cdot J}}.$$

Тоді, якщо рівняння руху фізичного маятника має вигляд

$$J \cdot \ddot{\varphi} + R \cdot \dot{\varphi} + mgl \cdot \varphi = M_m \sin(\omega \cdot t + \varepsilon),$$

то відповідне рівняння у безрозмірній формі набуде вигляду

$$\varphi'' + 2\zeta \cdot \varphi' + \varphi = \bar{M}_m \sin(\nu \cdot \tau + \varepsilon). \quad (\text{б})$$

Нарешті розглянемо електричний коливальний контур, який складається з послідовно з'єднаних котушки індуктивності з індуктивністю L , резистора з опором R , конденсатора з ємністю C і джерела електрорушійної сили (е.р.с.), який виробляє гармонічно змінювану е.р.с. з амплітудою E_m , частотою ω і початковою фазою ε .

Параметрами цієї системи є поточне значення заряду q на конденсаторі, електричний струм $i = \frac{dq}{dt}$ у контурі і напруги на індуктивності, резисторі і конденсаторі.

У цьому випадку оберемо такі головні одиниці виміру: T - одиниця виміру часу, Q - одиниця виміру електричного заряду і V - одиниця виміру електричної напруги. Тоді розмірності визначальних параметрів можна подати у такий спосіб:

$$\{t\} = T; \quad \{\omega\} = T^{-1}; \quad \{E_m\} = V; \quad \{q\} = \{q_0\} = Q; \quad \{i\} = \{i_0\} = QT^{-1};$$

$$\left\{ \frac{di}{dt} \right\} = QT^{-2}; \quad \{L\} = VQ^{-1}T^2; \quad \{R\} = VQ^{-1}T; \quad \{C\} = V^{-1}Q.$$

Оберемо у якості базових параметрів індуктивність L , ємність C і амплітуду е.р.с. E_m . Проводячи відшукування критеріїв подібності, дійдемо:

$$\bar{L} = 1; \quad \bar{C} = 1; \quad \bar{E}_m = 1; \quad \bar{t} = \frac{t}{\sqrt{L \cdot C}}; \quad \bar{\omega} = \omega \sqrt{L \cdot C};$$

$$\bar{q} = \frac{q}{CE_m}; \quad \bar{i} = \frac{i}{E_m} \sqrt{\frac{L}{C}}; \quad \bar{q}_0 = \frac{q_0}{CE_m}; \quad \bar{i}_0 = \frac{i_0}{E_m} \sqrt{\frac{L}{C}}; \quad \bar{R} = R \sqrt{\frac{C}{L}}.$$

Використовуючи позначення:

$$\omega_0 = \frac{1}{\sqrt{LC}}; \quad \tau = \omega_0 t; \quad \bar{q}' = \frac{d\bar{q}}{d\tau}; \quad \bar{q}'' = \frac{d^2\bar{q}}{d\tau^2}; \quad \nu = \frac{\omega}{\omega_0};$$

$$\zeta = \frac{\bar{R}}{2} = \frac{R}{2} \sqrt{\frac{C}{L}},$$

можна рівняння електричного коливального контуру

$$L \frac{di}{dt} + R \cdot i + \frac{1}{C} q = E_m \sin(\omega \cdot t + \varepsilon)$$

подати у безрозмірному вигляді у такий спосіб:

$$\bar{q}'' + 2\zeta \cdot \bar{q}' + \bar{q} = \sin(\nu \cdot \tau + \varepsilon). \quad (\text{в})$$

Порівнюючи рівняння (а, б, в), можна зробити деякі попередні висновки:

- 1) рівняння руху деяких матеріальних систем, навіть різної фізичної природи, у безрозмірній формі можуть набувати практично ідентичного вигляду;
- 2) завдяки цьому виникають можливості:

- розповсюджувати результати поодинокого експерименту на поведіння цілого класу подібних явищ або тієї самої фізичної природи, або навіть зовсім іншої фізичної природи;

- проводити дослідження поведження оригіналу на основі експериментів з моделями або тієї самої фізичної природи, або іншої фізичної природи.

1.3.5. Застосування теорії подібності для побудови моделей

Подання параметрів досліджуваного процесу у вигляді безрозмірних критеріїв подібності надає широкі можливості у побудові моделей цього процесу. Наприклад, досліджування поведження механічної коливальної ланки, що описується рівнянням (а) можна здійснити:

1) на основі моделі того самого типу (маси з пружно-демпфіруючим зв'язком з основою), але з зовсім іншими (більш досяжними для дослідника) характеристиками маси, жорсткості пружини і амплітуди зовнішньої сили; до такого моделювання вдаються, коли потрібно вивчити поведження об'єкта або надто великих, або надто малих розмірів, виготовлення і експериментування з якими потребує великих втрат, чи взагалі неможливе;

2) на основі електричної моделі, що описується рівнянням (в); у цьому випадку замість виготовлення і проведення експериментів з механічною моделлю, що є досить складною справою, достатньо зібрати електричний контур з досить великого асортименту індуктивностей, резисторів і конденсаторів, підключити його до джерела змінної напруги і, проводячи експеримент, вимірювати струм і напруги за допомогою також досить великого і досяжного набору електричних вимірювачів (амперметрів і вольтметрів).

Розглянемо більш докладно побудову моделі, моделювання і розповсюдження результатів моделювання на поведження оригіналу у цих двох випадках.

Отже, нехай потрібно дослідити поведження механічної коливальної системи (оригіналу) з такими заданими значеннями її характеристик - $m_O, f_O, c_O, F_{mO}, \omega_O, \varepsilon_O, x_{0O}, \dot{x}_{0O}$, де індекс O позначає, що відповідний параметр відноситься до оригіналу.

Припустимо, що потрібно провести ці дослідження на механічному аналозі цієї системи. З того, що у якості базових параметрів при відшуканні критеріїв подібності обрані величини маси, жорсткості пружини і амплітуди діючої зовнішньої сили, виливає, що вказані величини у моделі можуть бути обрані дослідником довільними, виходячи з умов досяжності, простоти експериментування, дешевизни тощо. Позначимо обрані значення цих величин через m_m, c_m і F_{mm} , де індекс "m" позначає, що відповідні параметри характеризують модель.

Щоб забезпечити подібність явищ в моделі і оригіналі, достатньо обрати решту параметрів моделі з умови рівності критеріїв подібності, що відповідають умовам однозначності. У розглядуваному випадку цих критеріїв чотири - $\zeta, \nu, \bar{x}_0, \bar{\dot{x}}_0$. Отже, мають бути забезпечено виконання наступних співвідношень між параметрами оригіналу і моделі:

$$\zeta_O = \frac{f_O}{2\sqrt{m_O c_O}} = \zeta_M = \frac{f_M}{2\sqrt{m_M c_M}}; \quad \bar{\omega}_O = \omega_O \sqrt{\frac{m_O}{c_O}} = \bar{\omega}_M = \omega_M \sqrt{\frac{m_M}{c_M}};$$

$$\bar{x}_{0O} = x_{0O} \frac{c_O}{F_{mO}} = \bar{x}_{0M} = x_{0M} \frac{c_M}{F_{mM}}; \quad \dot{\bar{x}}_{0O} = \dot{x}_{0O} \frac{\sqrt{c_O m_O}}{F_{mO}} = \dot{\bar{x}}_{0M} = \dot{x}_{0M} \frac{\sqrt{c_M m_M}}{F_{mM}}.$$

Це означає, що коефіцієнт тертя, частота коливань діючої сили і початкові значення початкового відхилення і швидкості тепер не можуть бути довільними, а повинні дорівнювати значенням, підрахованим з попередніх співвідношень на основі заданих значень параметрів оригіналу і обраних значень параметрів моделі:

$$f_M = f_O \sqrt{\frac{m_M}{m_O}} \sqrt{\frac{c_M}{c_O}}; \quad \omega_M = \omega_O \sqrt{\frac{m_O}{m_M}} \sqrt{\frac{c_M}{c_O}}; \quad x_{0M} = x_{0O} \frac{F_{mM} c_O}{F_{mO} c_M};$$

$$\dot{x}_{0M} = \dot{x}_{0O} \sqrt{\frac{m_O}{m_M}} \sqrt{\frac{c_O}{c_M}} \frac{F_{mM}}{F_{mO}}.$$

Після встановлення розрахункових значень у моделі, можна проводити експериментальне її дослідження. Вимірюючи поточне переміщення, час і швидкість тіла у моделі і фіксуючи результати вимірювань, одержимо потрібні залежності $x_M(t_M)$, $\dot{x}_M(t_M)$.

Далі потрібно розповсюдити ці залежності на шукані аналогічні залежності в оригіналі. Це здійснюється також на основі рівності відповідних критеріїв подібності

$$x_O(t) = x_M(t_M) \frac{c_M}{c_O} \frac{F_{mO}}{F_{mM}}; \quad \dot{x}_O(t) = \dot{x}_M(t_M) \sqrt{\frac{m_M}{m_O}} \sqrt{\frac{c_M}{c_O}} \frac{F_{mO}}{F_{mM}},$$

причому зв'язок між часом t оригіналу і часом t_M моделі (модельним часом) визначається співвідношенням

$$t = t_M \sqrt{\frac{c_M}{c_O}} \sqrt{\frac{m_O}{m_M}}.$$

Цікавою є така обставина. Модельний час можна зробити значно більш швидкоплинним у порівнянні з часом оригіналу, за рахунок, наприклад, обрання таких параметрів моделі, щоб виконувалася нерівність

$$\sqrt{\frac{m_O}{m_M}} \sqrt{\frac{c_M}{c_O}} \gg 1.$$

Таке моделювання забезпечує значне скорочення часу дослідження. Цим часто користуються на практиці, наприклад, при моделюванні старіння будівельних конструкцій, аналізі поведіння насипів протягом значного часу (десятки років) тощо. Таке моделювання називають іноді "лупою часу".

Тепер перейдемо до аналогового моделювання тієї самої механічної системи на електричній моделі.

З того, що базовими параметрами електричного аналога є індуктивність, ємність конденсатора і амплітуда змінювання е.р.с., ці параметри дослідник може обрати довільно. Позначимо обрані значення через L_m , C_m і E_m .

Як і у попередньому випадку для визначення решти параметрів електричного аналога у відповідності до теорії подібності слід забезпечити рівність визначальних критеріїв подібності:

$$\zeta_O = \frac{f_O}{2\sqrt{m_O c_O}} = \zeta_m = \frac{R_m}{2} \sqrt{\frac{C_m}{L_m}}; \quad \bar{\omega}_O = \omega_O \sqrt{\frac{m_O}{c_O}} = \bar{\omega}_m = \omega_m \sqrt{L_m \cdot C_m};$$

$$\bar{x}_{0O} = x_{0O} \frac{c_O}{F_{mO}} = \bar{q}_{0,m} = \frac{q_{0,m}}{C_m E_{mm}}; \quad \bar{\dot{x}}_{0O} = \dot{x}_{0O} \frac{\sqrt{c_O m_O}}{F_{mO}} = \bar{i}_{0,m} = \frac{i_{0,m}}{E_{mm}} \sqrt{\frac{L_m}{C_m}}.$$

З них випливає, що величина опору резистора моделі має визначатися з співвідношення $R_m = \frac{f_O}{\sqrt{m_O c_O}} \sqrt{\frac{L_m}{C_m}}$, величина частоти змінювання е.р.с. визна-

читься рівністю $\omega_m = \frac{\omega_O}{\sqrt{L_m \cdot C_m}} \sqrt{\frac{m_O}{c_O}}$. При експерименті на моделі слід забезпе-

чити початкову величину заряду на конденсаторі $q_{0,m} = x_{0O} \frac{c_O}{F_{mO}} C_m E_{mm}$ і почат-

кове значення струму у контурі $i_{0,m} = \dot{x}_{0O} \frac{\sqrt{c_O m_O}}{F_{mO}} E_{mm} \sqrt{\frac{C_m}{L_m}}$. Одержані внаслідок

моделювання залежності величини струму в контурі від модельного часу можна перевести у залежність швидкості тіла за формулою

$$\dot{x}_O(t) = i_m(t_m) \frac{F_{mO}}{\sqrt{c_O m_O}} \frac{1}{E_{mm}} \sqrt{\frac{L_m}{C_m}}.$$

При цьому модельний час пов'язаний з часом оригіналу у такий спосіб:

$$t = \frac{t_m}{\sqrt{L_m \cdot C_m}} \sqrt{\frac{m_O}{c_O}}.$$

Примітки. 1. Звичайно, у розглянутих прикладах матеріальне моделювання не має сенсу, бо наведені рівняння досить просто розв'язуються теоретично. Але якщо рівняння, що описують поведінку оригіналу, є нелінійними і надто складними, моделювання може стати необхідним. Наведені приклади призначені лише для ілюстрації механізму застосування апарату теорії подібності задля побудови моделей і розповсюдження результатів моделювання на оригінал.

2. Проведені операції з забезпечення подібності оригіналу і моделі і перерахунку результатів моделювання, основані на застосуванні аналізу розмірностей, не спираються на знання диференціальних рівнянь руху. Тому забезпечення подібності можливо навіть тоді, коли рівняння руху взагалі є невідомими.

ми. Достатньо лише чітко визначити сукупність фізичних параметрів, які суттєво впливають на досліджуване явище, і встановити їхню фізичну розмірність.

1.4. Етапи розв'язування інженерних задач на ЕОМ

Результатом розв'язування інженерних (прикладних) задач будь-якого рівня є, як правило, чисельні оцінки (параметрів пристроїв, процесів, технічних і економічних характеристик, тощо), які є наслідком розрахунків, що здійснюються з наближеними первісними даними. Більшість прикладних задач зводяться до математичних задач, які розв'язуються різноманітними обчислювальними методами.

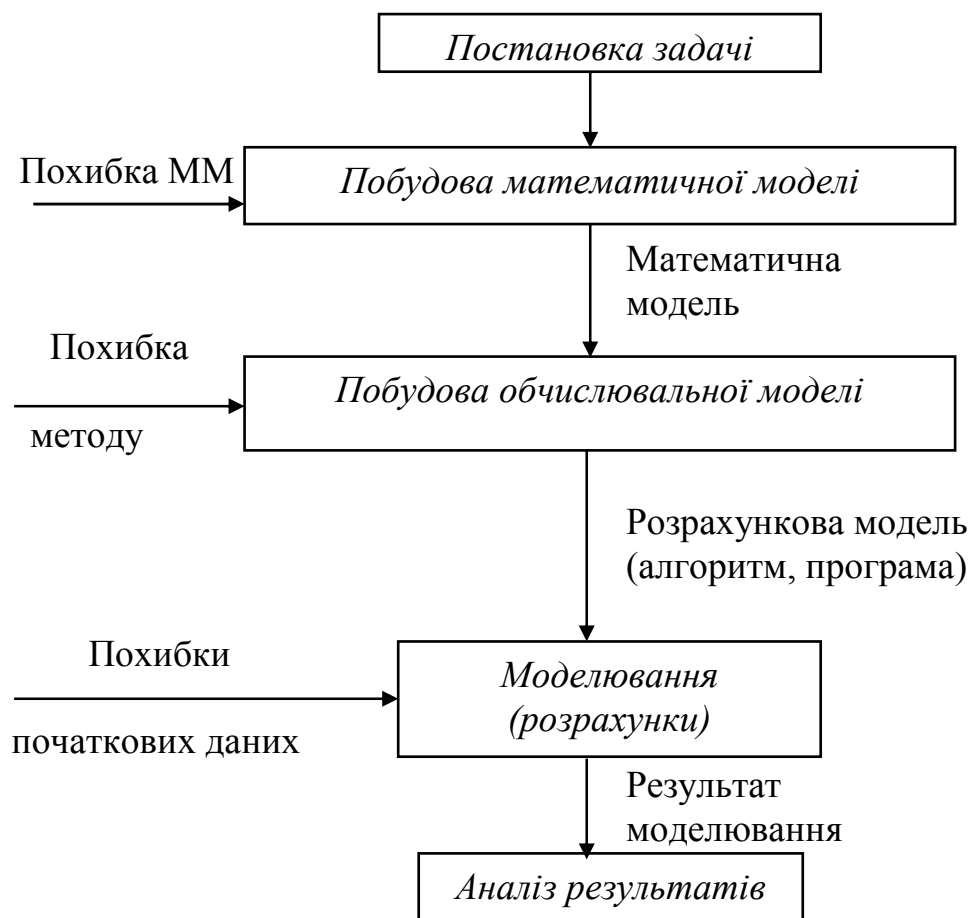


Рис. 1.1. Схема розв'язування інженерної задачі

Послідовність розв'язуванні таких задач можна подати у виді наступних етапів:

- 1) постановка задачі;
- 2) створення математичної моделі (формулювання задачі); перевірка моделі на адекватність;
- 3) побудова розрахункової (обчислювальної) моделі, яка відповідає прийнятій математичній моделі;

- 4) проведення розрахунків за обраною обчислювальною моделлю при заданих (відомих) значеннях первісних даних;
- 5) аналіз одержаних результатів.

У цілому процес розв'язування інженерної задачі може бути поданий у вигляді схеми, наведеної на рис. 1.1.

Розглянемо докладніше кожний з цих етапів.

1.4.1. Постановка задачі

Постановка задачі має передумовою словесне, змістовне формулювання задачі, умов, за яких вона ставиться, та вимог до її розв'язування. Слова "змістовне формулювання" слід розуміти так, що задача має бути сформульована у термінах опису реального об'єкта (технічного пристрою або процесу), поведження якого підлягає вивченню.

Як приклади розглядатимемо такі найпростіші інженерні задачі.

Задача 1. *Визначити характеристики власного руху фізичного маятника за умови малих його коливань.*

Задача 2. *Визначити змінювання швидкості тіла при його падінні, враховуючи опір оточуючого середовища.*

Задача 3. *Відшукати моменти інерції ротора гіроскопа.*

Задача 4. *Визначити характеристики власного руху гіроскопа у кардановому підвісі, а також характеристики його вимушеного руху під дією моментів зовнішніх сил, що діють по осях карданового підвісу і змінюються з часом за гармонічним законом.*

1.4.2. Створення математичної моделі

Математична модель - це математичний опис співвідношень постановки задачі. Такий опис можливий лише на ґрунті попередньо одержаних знань про поведження об'єкта, що вивчається, і про способи правильного й ефективного опису цього поведження у математичних термінах. В одних випадках утворення математичної моделі не складає труднощів (наприклад, модель є відомою заздалегідь за результатами раніше проведених досліджень), а в інших потрібно неодноразове уточнення постановки задачі, виділення головних визначальних чинників, відкидання чинників, які незначно впливають на результат і т.д..

Так, для задачі 1 математична модель може бути утворена, якщо врахувати наступні теоретичні відомості.

1. До характеристик власного руху коливальної ланки, яким є фізичний маятник, відносять:

- 1) частоту власних коливань;
- 2) коефіцієнт загасання цих коливань.

2. При малих відхиленнях від вертикалі рух маятника з достатньою точністю описується лінійним диференціальним рівнянням другого порядку зі сталими коефіцієнтами:

$$J \cdot \ddot{\varphi} + R \cdot \dot{\varphi} + mgl \cdot \varphi = 0, \quad (1.1)$$

де φ - кут відхилення маятника від вертикалі; J - момент інерції маятника відносно його осі обертання; R - коефіцієнт демпфірування; m - маса маятника; g - прискорення вільного падіння; l - зміщення центра мас маятника відносно осі його обертання; $\dot{\varphi} = \frac{d\varphi}{dt}$ - кутова швидкість повороту маятника навколо його осі обертання; $\ddot{\varphi} = \frac{d\dot{\varphi}}{dt}$ - кутове прискорення маятника.

3. Власний рух маятника описується співвідношенням

$$\varphi(t) = A \cdot e^{-ht} \cdot \sin(\omega_o \cdot t + \varepsilon), \quad (1.2)$$

де A - початкове значення амплітуди власних коливань і ε - початкова фаза власних коливань визначаються початковими умовами руху маятника, а ω_o - частота власних коливань та h - коефіцієнт загасання власних коливань - це параметри, які визначаються лише параметрами самого маятника і не залежать від інших чинників. Фактично ω_o і h є шуканими величинами.

4. Величини ω_o і h є відповідно уявною і дійсною частинами пари комплексно спряжених коренів характеристичного рівняння

$$J \cdot p^2 + R \cdot p + mgl = 0, \quad (1.3)$$

яке випливає з диференційного рівняння (1), тобто корінь рівняння (3) має вигляд:

$$p_{1,2} = -h \pm \omega_o. \quad (1.4)$$

У підсумку математично розв'язування задачі 1 зводиться до *відшукування комплексних коренів квадратного рівняння (3) і виділенню їхніх дійсної й уявної частин* за заданими первісними даними - значеннями параметрів J , R та mgl .

У задачі 2 треба припустити, що тіло є матеріальною точкою маси m , з'ясувати, під дією яких сил відбувається падіння тіла, визначити чинники, що впливають на силу опору, встановити залежність сили опору від цих факторів. Якщо вважати, що на тіло діють сила тяжіння $F_1 = mg$ та сила опору, що є пропорційною до швидкості v падіння, тобто $F_2 = -k \cdot v$, то, на основі законів механіки одержимо рівняння $m \cdot a = m \cdot g - k \cdot v$, або

$$\frac{dv}{dt} = g - \frac{k}{m} v. \quad (1.5)$$

Це диференційне рівняння із врахуванням початкової умови $v(t_0) = v_0$ і є математичною моделлю задачі.

У задачі 3 насамперед слід з'ясувати форму ротора, його розміри, розподіл мас, потім виділити у тілі ротора ряд частин, відшукування моментів інерції яких робиться досить просто (циліндри, кільця, конуси тощо). Тоді задача зводиться до обчислень моментів інерції окремих елементарних тіл і їхньому під-

сумовуванню. Формули обчислення моментів інерції окремих частин ротора і їх підсумовування і складуть математичну модель цієї задачі.

Постановка задачі 4 має містити опис власних параметрів системи "гіроскоп у кардановому підвісі", опис параметрів зовнішніх моментів сил, опис рівнянь руху. Наприклад, рівняння руху гіроскопа для цієї задачі можуть бути взяті у наступному вигляді

$$\begin{cases} A \cdot \ddot{\alpha} + H_o \cos \beta_o \cdot \dot{\beta} = N_m \cdot \sin(\omega \cdot t + \varepsilon_1) \\ B \cdot \ddot{\beta} - H_o \cos \beta_o \cdot \dot{\alpha} = L_m \cdot \sin(\omega \cdot t + \varepsilon_2) \end{cases} \quad (1.6)$$

Тут α і β - кути повороту гіроскопа навколо осей підвісу; A та B - його моменти інерції, H_o - власний кінетичний момент гіроскопа, β_o - початкове значення кута β ; $\dot{\alpha} = \frac{d\alpha}{dt}$; $\dot{\beta} = \frac{d\beta}{dt}$; $\ddot{\alpha} = \frac{d\dot{\alpha}}{dt}$; $\ddot{\beta} = \frac{d\dot{\beta}}{dt}$; N_m , L_m - амплітуди змінювання моментів зовнішніх сил; ω - частота (колова) цього змінювання; ε_1 , ε_2 - початкові фази коливань цих моментів.

За математичну модель у цьому випадку може правити сукупність розв'язків рівнянь (6), наведена нижче:

$$\begin{aligned} \alpha &= \alpha_o + C_1(1 - \cos \lambda t) / \lambda + (C_2 \sin \lambda t) / \lambda + F_{1s}(1 - \cos \omega t) / \omega + \\ &\quad + (F_{1c} \sin \omega t) / \omega; \\ \beta &= \beta_o + [C_2(1 - \cos \lambda t) - C_1 \sin \lambda t] \sqrt{\frac{A}{B}} / \lambda + F_{2s}(1 - \cos \omega t) / \omega + \\ &\quad + (F_{2c} \sin \omega t) / \omega, \end{aligned} \quad (1.7)$$

де α_o і β_o - початкові значення кутів визначаються α і β ; $\lambda = \frac{H_o \cos \beta_o}{\sqrt{AB}}$ - частота власних (нутаційних) коливань гіроскопа; F_{1s} , F_{1c} , F_{2s} , F_{2c} , C_1 , C_2 визначаються сукупністю співвідношень:

$$\begin{aligned} F_{1s} &= -\lambda \cdot (l_s + \sqrt{\frac{B}{A}} v \cdot n_c) / (1 - v^2); & F_{1c} &= \lambda \cdot (-l_c + \sqrt{\frac{B}{A}} v \cdot n_s) / (1 - v^2); \\ F_{2s} &= \lambda \cdot (n_s - \sqrt{\frac{A}{B}} v \cdot l_c) / (1 - v^2); & F_{2c} &= \lambda \cdot (n_c + \sqrt{\frac{A}{B}} v \cdot l_s) / (1 - v^2); \\ C_1 &= \lambda \cdot (\sqrt{\frac{B}{A}} n_c + v \cdot l_s) / (1 - v^2) - \sqrt{\frac{B}{A}} \cdot \dot{\beta}_o; \\ C_2 &= \lambda \cdot (l_c - \sqrt{\frac{B}{A}} v \cdot n_s) / (1 - v^2) + \dot{\alpha}_o; \end{aligned}$$

$$\begin{aligned} n_s &= \frac{N_s \sqrt{AB}}{H_o^2 \cdot \cos^2 \beta_o}; & n_c &= \frac{N_c \sqrt{AB}}{H_o^2 \cdot \cos^2 \beta_o}; & l_s &= \frac{L_s \sqrt{AB}}{H_o^2 \cdot \cos^2 \beta_o}; & l_c &= \frac{L_c \sqrt{AB}}{H_o^2 \cdot \cos^2 \beta_o}; \\ N_c &= N_m \cdot \sin \varepsilon_1; & N_s &= N_m \cdot \cos \varepsilon_1; & L_c &= L_m \cdot \sin \varepsilon_2; & L_s &= L_m \cdot \cos \varepsilon_2; \end{aligned}$$

$\nu = \frac{\omega}{\lambda}$ - відносна частота коливань моментів сил; $\dot{\alpha}_0$ і $\dot{\beta}_0$ - початкові значення кутових швидкостей $\dot{\alpha}$ і $\dot{\beta}$.

Рух гіроскопа за цими співвідношеннями може бути визначений у довільний момент часу.

Але як математичну модель можна також розглядати і первісну систему диференціальних рівнянь (6) за вказаних початкових умов.

Складання математичної моделі у прикладній задачі є найбільш складним і відповідальним етапом розв'язування і потребує, окрім істотних знань у спеціальній області, також і математичних і теоретичних знань.

Уже на цьому етапі розв'язування прикладної задачі доводиться нехтувати багатьма реальними процесами, як такими, що незначно впливають на процеси, які вивчаються, абстрагуватися від впливу багатьох чинників. Інакше кажучи, навіть коректно утворена математична модель завжди неповно, лише наближено. відображає реальні процеси. Але при цьому вона набуває риси більшої ясності, прозорості, більш доступна вичерпному дослідженню (з того боку, що підлягає вивченню).

1.4.3. Математичне моделювання

Модель утворюється задля подальшого її дослідження з метою одержати нові знання про відповідний реальний об'єкт. Таке дослідження вже готової моделі називають моделюванням. Дослідження математичної моделі називатимемо математичним моделюванням.

Математична задача є абстрагованою від конкретної сутності задачі. Для її розв'язування створюються спеціальні обчислювальні методи, причому до тої самої математичної моделі можуть зводитися зовсім різні прикладні задачі.

Так, задача 1 звелася до розв'язування квадратного рівняння, яке може відображувати характеристичне рівняння не тільки фізичного, але й математичного маятника, маси, яка з'єднана пружиною з корпусом (лінійного акселерометра), гіроскопічного тахометру і т. п.

Диференціальне рівняння (5) у задачі 2 може бути моделлю і для багатьох інших задач (вивчення змінювання швидкості тіла у в'язкому середовищі, змінювання електричного струму у найпростішому електричному ланцюзі, змінювання швидкості репродукції бактерій тощо).

Задля розв'язування задачі 3 потрібно обчислити низку визначених інтегралів. До обчислення визначених інтегралів приходять і при відшукуванні площ складних фігур, об'єму тіла або дуги плоскої кривої, розрахунках роботи змінної сили й у багатьох інших фізичних задачах.

Математична модель (7) задачі 4 може описувати не тільки поведіння гіроскопу, але й будь-якої іншої системи, якщо диференціальні рівняння руху останньої збігаються з рівняннями (6).

1.4.4. Побудова обчислювальної моделі

Побудова обчислювальної моделі може здійснюватися різними методами, які можна поділити на *точні* й *наближені*. Точні методи - це такі, які після скінченної кількості дій (обчислень) приводять до точного результату за умови, що обчислення здійснюються без похибок. Наближеними називають такі методи, які за тих же умов дозволяють одержати результат лише з деякою похибкою.

При використанні точних методів етап досліджування математичної моделі поділяється на такі підетапи: 1) відшукування точного розв'язку математичної моделі; 2) підставлення вихідних даних у знайдений точний розв'язок і реалізація передбачених ним обчислень.

Наприклад, для розв'язування задачі 1 краще використати точний метод, тобто формулу

$$p_{1,2} = -\frac{R}{2J} \pm j \cdot \sqrt{\frac{mgl}{J} - \left(\frac{R}{2J}\right)^2} \quad (1.8)$$

(припускається, що $mgl / J > [R / (2J)]^2$), але можна застосовувати й наближені способи відшукування коренів квадратного рівняння.

Диференційне рівняння (5) задачі 2 краще розв'язувати, розділяючи змінні, тобто приводячи його до вигляду

$$\frac{m}{mg - kv} dv = dt. \quad (1.9)$$

Однак, його можна розглядати і як лінійне диференційне рівняння зі сталими коефіцієнтами, або розв'язувати (інтегрувати) наближеними чисельними методами.

При розв'язуванні задачі 3 слід використовувати методи наближеного обчислення визначених інтегралів.

Задачу 4 також можна розв'язувати двома шляхами. Розглядаючи систему диференційних рівнянь (6) як вихідну математичну модель, можна, з одного боку, знайти точний її розв'язок (7), а потім здійснити підставлення значень вихідних даних і дійти явних залежностей $\alpha(t)$ і $\beta(t)$, а отже, й $\alpha(\beta)$. З іншого боку, до системи (6) можна безпосередньо застосувати методи чисельного інтегрування диференційних рівнянь (наближені методи).

Досліджування математичної моделі наближеними методами поділяється на такі етапи:

- 1) обрання обчислювального методу (зазвичай наближених чисельних методів буває декілька);
- 2) вивчення або складання алгоритму метода;
- 3) реалізація алгоритму за допомогою обчислювальних засобів.

При виборі чисельного методу суттєвими є обсяг обчислень, швидкість збіжності обчислень (як швидко здобувається результат) та інші чинники. Зокрема, обрання методу залежить і від вхідних даних.

Крім того, на вибір метода впливають засоби його реалізації (ручний розрахунок, наявність обчислювальної машини, наявність готової програми тощо). Так, якщо буде використані швидкодіюча ЕОМ і готова програма, то обсяг обчислень не повинен засмучувати виконавця і бути визначальним фактором при обранні метода. При ручному ж розрахунку слід віддати перевагу методу, який, можливо, потребує деяких певних попередніх досліджень і перетворень математичної моделі, але завдяки цьому потребує й значно меншу кількість обчислень.

1.4.5. Алгоритм методу

Алгоритмом методу називається система правил, яка задає точно визначену послідовність операцій, яка приводить до шуканого результату (точного або наближеного).

Алгоритм - одне із ґрунтовних понять математики. Хід розв'язування обчислювальної (і взагалі будь-якої) задачі має бути поданий через алгоритм.

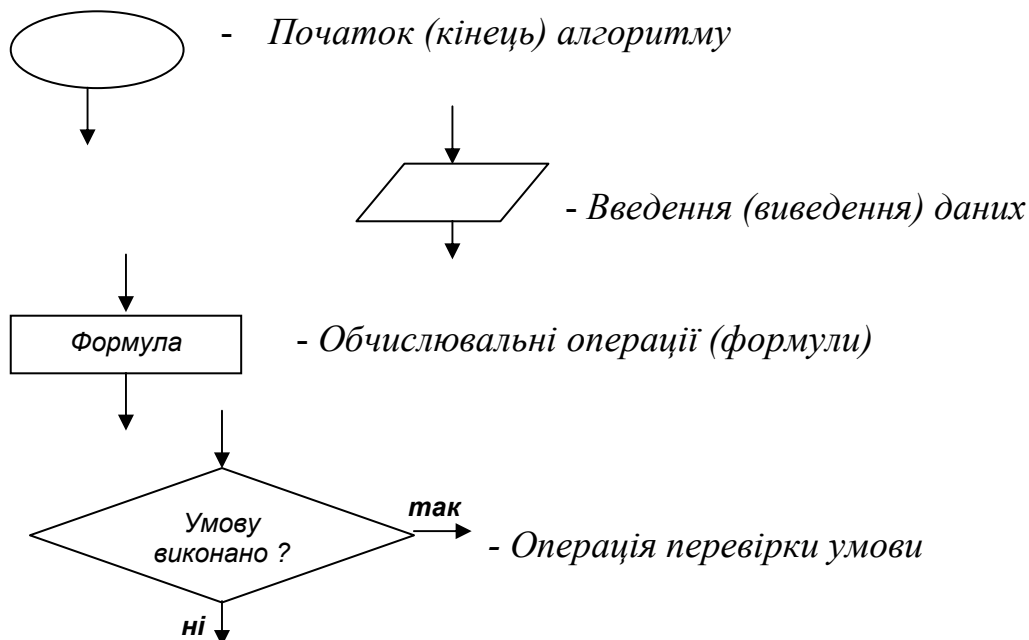


Рис. 1.2. Елементи блок-схеми алгоритму

Алгоритм можна записати словесно-формульно або у вигляді схеми. Так, словесно-формульний опис алгоритму розв'язування задачі 1 за формулою (8) має наступний вигляд:

1. Обчислити $h = \frac{R}{2J}$.
2. Обчислити $D = h^2 - \frac{mgl}{J}$.
3. Якщо $D < 0$, перейти до п. 7.
4. Обчислити $p_1 = -h + \sqrt{D}$ і $p_2 = -h - \sqrt{D}$.

5. Подати на пристрій виведення інформацію: "Рівняння має два дійсні корені:" і роздрукувати значення шуканих коренів p_1 і p_2 .
6. Перейти до п. 8.
7. Вивести на пристрій виведення інформацію "Коефіцієнт загасання дорівнює" і вивести значення h
"Частота власних коливань дорівнює" і роздрукувати значення $\sqrt{-D}$.
8. Кінець обчислень.

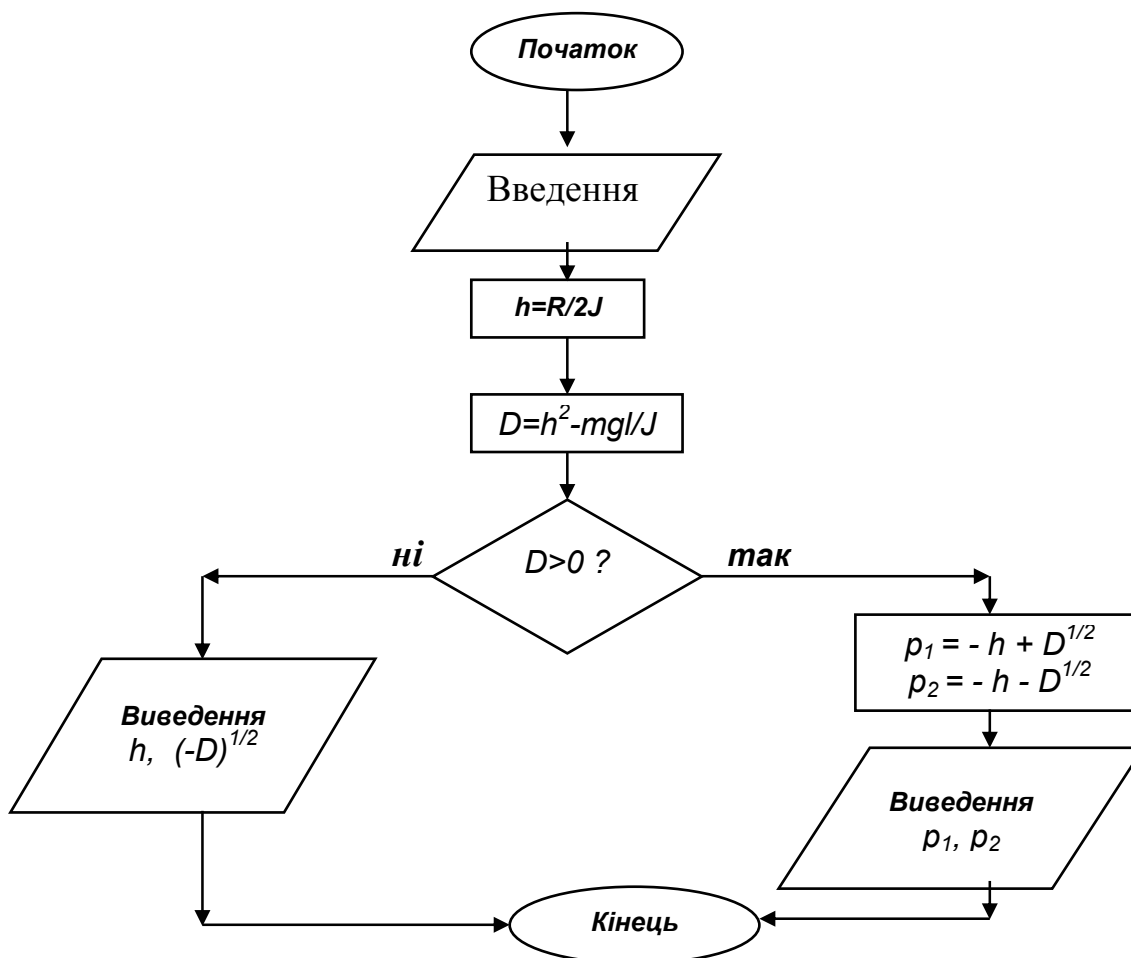


Рис. 1.3. Схема алгоритму відшукування коренів квадратного рівняння

При виконанні алгоритму перехід від однієї дії до іншої здійснюється строго у порядку їхнього запису. Якщо ж потрібно перервати природний хід дій за деякої умови, слід указувати на це (див. п. 3 наведеного алгоритму).

Структурною схемою алгоритму називають графічне зображення послідовності дій обчислювального процесу. У схемі кожна дія розміщується у певному геометричному символі (фігурі). Послідовність дій указується на схемі напрямком стрілок на лініях, якими з'єднують ці символи. Зазвичай прийнято початок і кінець обчислень зображувати овалами, введення даних і виведення результатів - у вигляді паралелограма. Обчислювальні операції розміщуються у прямокутниках, а операція перевірки деякої умови зображується у вигляді

ромбу. У середині кожної фігури розміщується стислий формульний опис відповідної операції.

Символи операцій перевірки умови мають два виходи: "так" і "ні". Стрілка на лінії, що виходить із виходу "так" вказує на операцію, до виконання якої потрібно перейти, якщо умову, яка перевіряється, виконано. Стрілка з написом "ні" вказує на операцію, до виконання якої слід перейти у випадку, коли умову не виконано.

На рис. 1.2. подані зображуючи елементи блок-схеми алгоритму обчислень. Фігури з'єднуються лініями зі стрілками, які вказують на операцію, до виконання якої слід перейти.

Для прикладу на рис 1.3 зображено схему алгоритму відшукування коренів квадратного рівняння.

1.4.6. Реалізація методу обчислень

Обчислення по алгоритмах відбувається за допомогою різних обчислювальних засобів.

При ручних (безпосередніх) розрахунках зазвичай використовуються найпростіші обчислювальні засоби: логарифмічна лінійка, таблиці, механічні, електричні, електронні клавішні обчислювальні машини. Проміжні результати дій алгоритму треба записувати у спеціальний розрахунковий бланк. Наявність програмувальних мікрокалькуляторів дозволяє реалізовувати обчислення автоматично, під керуванням програми.

Суттєвим є контроль обчислень, який проводять за так званим контрольним прикладом (тестом). Результат контрольного прикладу має бути заздалегідь відомим, тобто він або є очевидним, або його відшуковують яким-небудь іншим способом. При ручному рахунку контроль рекомендується проводити поетапно. При розрахунках на ЕОМ за складеною програмою контрольний приклад заздалегідь прораховують вручну, а потім звіряють поетапно результати розрахунків із здійснюваними машиною.

1.5. Контрольні запитання

1. Що таке "модель"? "моделювання"?
2. Які об'єктивні й суб'єктивні чинники можуть впливати на створення моделі?
3. Які види моделей трапляються в інженерній практиці?
4. Що таке "математична модель"? "обчислювальна модель"?
5. Як можна охарактеризувати постановку інженерної задачі?
6. Які етапи проходить у загальному випадку розв'язування інженерної задачі?
7. Що таке "алгоритм"? На які види поділяються алгоритми?
8. Що таке "блок-схема алгоритму"? Які позначення прийняті при побудові блок-схеми алгоритму?
9. Які є загальні правила побудови блок-схеми алгоритму?



2. Організація наближених обчислень

Як вже зазначалося, неминучим етапом розв'язування інженерних задач є проведення розрахунків із наближеними вихідними даними. Тому до основних умінь, необхідних інженеру в його професійної діяльності, слід віднести вміння грамотно (раціонально) організувати обчислення, під чим мається на увазі:

- знати можливі джерела похибок;
- вміння правильно записувати наближені дані й результати (у тому числі проміжні);
- вміння оцінювати похибку результату за заданими похибками компонент;
- вміння обирати найбільш раціональний *порядок обчислень*;
- вміння обирати алгоритм обчислення, найстійкіший до похибок обчислень;
- вміння контролювати хід і результати обчислень із метою виключення грубих похибок.

Не маючи достатніх умінь і навичок практичних обчислень можна при розв'язуванні задачі одержати результат, який не матиме нічого спільного з дійсним розв'язком задачі.

2.1. Джерела й види похибок

Практично на кожному етапі розв'язування прикладної задачі виникають власні джерела похибок.

Математична модель - це вже наближене подання реального об'єкта. Вихідні дані, що використовуються у розрахунках і виходять з експерименту, можна визначити лише наближено. Навіть точні числа, такі як π , $\sqrt{3}$, $6/7$ і т.п., при обчисленнях заміняють десятковими дробами, залишаючи лише певну кількість знаків після десяткової коми. Обчислювальні методи у більшості також є наближеними. Навіть при використанні найпростішої формули результат, як правило, одержують наближений.

Основні джерела виникнення похибок наближеного розв'язування прикладних задач такі.

1. Похибки математичної моделі. Їх пов'язано з використаними припущеннями, які дозволяють спростити математичну модель задачі. Вони не контролюються у процесі чисельного розв'язування задачі і можуть бути зменшені лише за рахунок більш точного математичного опису фізичної задачі.
2. Похибки первісних даних. Значення параметрів, що входять у математичний опис задачі, вимірюються експериментально з деякою похибкою. Похибки математичної моделі і вихідних даних у цілому утворюють так звані неусувні похибки. Назву обумовлено тим, що ці види похибок не можна усунути шляхом організації обчислень. Зменшення

їх лежить лише на шляху перебудови математичної моделі і точнішого виміру вихідних даних.

3. Похибки наближеного методу, або похибки усікання. При чисельному розв'язуванні задачі точний оператор, в якому кількість чисел або операцій перевищує допустимі межі, замінюється наближеним, який потребує скінченної кількості операцій. Наприклад, замінюють інтеграл сумою, функцію - поліномом (багаточленом) або будують нескінченний процес і обривають його після скінченної кількості операцій.
4. Обчислювальна похибка, що виникає в результаті вимушеного округлення чисел, наприклад, внаслідок скінченної кількості розрядів у запису числа в оперативній пам'яті ЕОМ.

Якщо розв'язок деякої задачі *неперервно* залежить від вхідних даних, тобто малому змінюванню вхідних даних відповідає мале змінювання розв'язку, то задача називається стійкою за вхідними даними, або грубою. У стійкому обчислювальному алгоритмі похибки округлення не накопичуються.

Точність наближеного числа характеризується поняттями абсолютної й відносної похибки.

Абсолютною похибкою наближеного числа 'a' називається абсолютне значення різниці між ним і точним його значенням:

$$\Delta = |A - a|,$$

де A - точне значення, a - наближене значення. Абсолютна похибка має суто теоретичний інтерес, оскільки точне значення A невідоме. Тому на практиці частіше використовують граничну абсолютну похибку Δ_a наближеного числа a , рівну по можливості найменшому числу, що є більшим за абсолютну похибку

$$\Delta = |A - a| \leq \Delta_a.$$

Значення a і Δ_a дозволяють вказати інтервал, що містить точне значення A :

$$a - \Delta_a \leq A \leq a + \Delta_a.$$

Частіше використовується компактніший запис

$$A = a \pm \Delta_a.$$

Очевидно, таке визначення абсолютної похибки не є однозначним. Так, якщо $A = \pi$, а як наближене значення узяти $a = 3,14$, то, враховуючи, що $3,140 < \pi < 3,142$, можна записати:

$$|\pi - a| < 0,002; \quad |\pi - a| < 0,01; \quad |\pi - a| < 0,1.$$

Кожне з чисел 0,002; 0,01; 0,1 буде граничною абсолютною похибкою числа a . Але чим ближче між собою числа $|A - a|$ і Δ_a , тим точніше абсолютна похибка оцінює фактичну похибку.

Основною характеристикою точності наближеного числа є його відносна похибка

$$\delta = \frac{\Delta}{|A|}.$$

Оскільки число A невідоме, то, як правило, вважають

$$\delta = \frac{\Delta}{|a|}.$$

Аналогічно з нерівності $\delta \leq \delta_a$ визначають граничну відносну похибку числа a , вважаючи

$$\delta_a = \frac{\Delta_a}{|a|}.$$

Величина δ характеризує якість наближення. Це безрозмірна величина, зазвичай її виражають у процентах. Так, відносна похибка числа π , прийнятого за наближене значення числа π , при $\Delta(3,14) = 0,002$ дорівнює

$$\delta(3,14) = \frac{0,002}{3,14} = 0,00064, (0,064\%).$$

2.2. Запис наближених чисел. Правило округлення

Записувати наближене число у вигляді $a \pm \Delta_a$ незручно. Тому в обчислювальній практиці часто вдаються до різних прийомів, що дозволяють тільки за записом наближеного числа a судити про його похибку.

Нехай наближене число a подане у вигляді скінченного десяткового дробу:

$$a = \alpha_m \cdot 10^m + \alpha_{m-1} \cdot 10^{m-1} + \dots + \alpha_{m-n+1} \cdot 10^{m-n+1}; \quad (\alpha_m \neq 0),$$

де α_i - цифри числа a ($\alpha_i = 0, 1, \dots, 9$).

Значущими цифрами числа називають усі цифри у запису числа, починаючи з першої ненульової зліва. Наприклад, у чисел $a = 0,0503$, $b = 0,00630500$ значущими є підкреслені цифри.

Значущу цифру називають вірною, якщо абсолютна похибка числа не перевищує половини одиниці розряду, що відповідає цій цифрі. У зворотному випадку цифра вважається сумнівною.

Наприклад, число 647,326 при $\Delta(a) = 0,03$ ($< 0,5 \cdot 10^{-1}$) має чотири вірні цифри 6, 4, 7, 3 і дві сумнівні 2, 6.

Згідно правилу, запропонованому А. Н. Криловим, наближене число потрібно записувати так, щоб усі значущі цифри у запису числа були вірними, а перша цифра з відкинутої частини відносилася до сумнівних. Так, число 0,884, відоме з похибкою 0,004, має бути записано так: 0,88. Раніше згадане число 647,326 при описаних умовах слід записати у вигляді 647,3.

У математичних таблицях значень функцій приводяться тільки вірні цифри. Відносна похибка табличних значень, наприклад, у тризначних таблицях не перевищує $0,5 \cdot 10^{-3}$, у семизначних - $0,5 \cdot 10^{-7}$.

Точність наближеного числа залежить не від кількості значущих цифр, а від кількості вірних цифр.

При проведенні розрахунків остаточний наближений результат зазвичай округлюють до його вірних цифр, залишаючи одну сумнівну, а у проміжних результатах зберігають одну, дві, а часом і три сумнівні цифри.

Приклад. Задані числа при вказаних абсолютних похибках округлити до вірних цифр. Визначити абсолютну похибку результату.

1. $a_1 = 2,6219$, $\Delta_{a_1} = 0,024$; 2. $a_2 = 47,35$, $\Delta_{a_2} = 1,3$;
2. $a_3 = 6,9971$, $\Delta_{a_3} = 0,0009$; 4. $a_4 = 0,648$, $\Delta_{a_4} = 0,04$.

РОЗВ'ЯЗУВАННЯ

1. Оскільки $0,024 < 0,05$, то число a_1 потрібно округлити до 0,1. Одержимо число $a_1 = 2,6$ із двома вірними цифрами. Визначимо абсолютну похибку результату $\Delta = 0,024 + |2,6219 - 2,6| = 0,0459 \approx 0,05$.
2. Через те що $1,3 < 0,5 \cdot 10^1$, то число a_2 слід округлити до десятків. Одержимо $a_2 \approx 5 \cdot 10$ з однією вірною цифрою. Варто відзначити, що не можна писати ані 50, ані $5,0 \cdot 10$, бо у обох випадках у відповідності із прийнятою системою запису це буде означати, що похибка записаного числа менша за 0,5, що суперечить умові. Абсолютна похибка результату дорівнює $\Delta(5 \cdot 10) = 1,3 + |50 - 47,35| = 1,3 + 2,65 = 3,95 \approx 4$.
3. Оскільки $0,0009 < 0,005$, то округлення числа a_3 здійснюємо до 0,01. Одержимо $a_3 \approx 7,00$ із трьома вірними цифрами. Абсолютна похибка результату буде наступною $\Delta(a_3) = 0,0009 + |7,00 - 6,9971| \approx 0,004$. У запису 7,00 нулі свідчать про три його вірні знаки, і цей запис відрізняється від 7 або 7,0.
4. Оскільки $0,04 < 0,05$, то число a_4 округлюємо до 0,1. Одержуємо $a_4 \approx 0,6$ з однією вірною цифрою. Абсолютна похибка результату дорівнюватиме $\Delta(0,6) = 0,04 + 0,048 = 0,088 > 0,05$, тобто цифра 6 вже є сумнівною. Тому при округленні рекомендується залишати одну-дві сумнівні цифри

$$a_4 = 0,(65); \quad \Delta(0,65) = 0,042 < 0,05.$$

Помітимо, що термін "вірні цифри" не слід розуміти буквально. Так, у числі 7,00, яке замінює 6,9971 у попередньому прикладі, жодна з цифр не збігається з цифрами числа, хоча усі цифри цього числа є "вірними" в описаному вище сенсі $|6,9971 - 7,00| < 0,005$. Однак вірні знаки наближеного числа часто збігаються з відповідними цифрами точного числа.

Відносна похибка наближеного числа безпосередньо залежить від кількості його вірних знаків. Наприклад, якщо наближене число a має три вірних знаки, то його відносна похибка δ_a перебуває у межах від 0,05 до 0,5% (зале-

жить від першої значущої цифри числа a). При збільшенні кількості вірних знаків на 1 відносна похибка зменшується у 10 разів.

На практиці зазвичай вважають, що число a є наближенням числа A з n вірними десятковими знаками, якщо $\Delta_a \leq 10^{-n}$. При такому визначенні в числі $a = 647,35$ при $\Delta_a = 0,095 < 0,1$ будуть вірними цифри 6, 4, 7, 3 і його слід записувати у вигляді 647,3. Згідно з колишнім визначенням, оскільки $\Delta(a) = 0,095 < 0,5$, то у цьому числі будуть вірними лише три цифри 6, 4 і 7 і його слід записувати як 647.

2.3. Похибки результату при діях із наближеними числами

Дії над наближеними числами приводять до поширення похибок. Для оцінки похибок результатів потрібно знати похибки вихідних чисел і правила обчислення похибки результату. Розглянемо ці правила.

2.3.1. Похибки підсумовування

Неважко впевнитися у слушності наступних тверджень.

Абсолютна похибка суми й різниці дорівнює сумі абсолютних похибок доданків

$$a = b \pm c \Rightarrow \Delta_a = \Delta_b + \Delta_c. \quad (2.1)$$

Відносна похибка суми двох величин однакового знаку перебуває в інтервалі між найменшою й найбільшою відносними похибками доданків

$$\delta_a = \frac{\Delta_a}{|a|} = \frac{\Delta_b + \Delta_c}{|b + c|}.$$

За умови $|b|=|c|$ матимемо
$$\delta_a = \frac{\Delta_b + \Delta_c}{2|b|} = \frac{\delta_b + \delta_c}{2}.$$

Якщо ж $|b| \gg |c|$, через що останнім доданком у знаменнику можна знехтувати, то

$$\delta_a = \frac{\Delta_b + \Delta_c}{|b|} = \delta_b + \frac{\Delta_c}{|b|} = \delta_b + \delta_c \frac{|c|}{|b|} \approx \delta_b.$$

Подамо вираз для відносної похибки у такий спосіб

$$\delta_a = \delta_b \frac{|b|}{|b + c|} + \delta_c \frac{|c|}{|b + c|}.$$

З його розгляду випливає наступне.

Додавання величин протилежного знаку (або віднімання величин однакового знаку) практично завжди приводить до збільшення відносної похибки результату у порівнянні з найбільшою з відносних похибок доданків.

Особливо небезпечним є віднімання дуже близьких величин. У цьому випадку відносна похибка результанта може сягати неприпустимих величин.

Приклад. Треба обчислити площу кругового тонкого кільця із внутрішнім радіусом $r = 1,750$ і товщиною $h = 5,0 \cdot 10^{-3}$ за формулами $S = \pi[(r+h)^2 - r^2]$ або $S = \pi \cdot h \cdot (2r+h)$. Відшукати похибки.

РОЗВ'ЯЗУВАННЯ

Перш за все потрібно зазначити, що, як впливає з угоди про форму запису наближеного числа, абсолютні похибки вихідних даних є такими:

$$\Delta_r = 5 \cdot 10^{-4}; \quad \Delta_h = 5 \cdot 10^{-5},$$

а відносні похибки

$$\delta_r = \frac{5 \cdot 10^{-4}}{1,75} \approx 2,9 \cdot 10^{-4}; \quad \delta_h = \frac{5 \cdot 10^{-5}}{5 \cdot 10^{-3}} \approx 10^{-2}.$$

Розрахунки згідно першої формули приводять до таких похибок:

$$\Delta_{r+h} = \Delta_r + \Delta_h = 5 \cdot 10^{-4} + 5 \cdot 10^{-5} = 5,5 \cdot 10^{-4};$$

$$\delta_{r+h} = \frac{\Delta_{r+h}}{r+h} = \frac{5,5 \cdot 10^{-4}}{1,755} = 3,13 \cdot 10^{-4};$$

$$\delta_{(r+h)^2} = 2\delta_{r+h} = 6,26 \cdot 10^{-4};$$

$$\Delta_{(r+h)^2} = \delta_{(r+h)^2} \cdot (r+h)^2 = 6,26 \cdot 10^{-4} \cdot (1,755)^2 = 19,3 \cdot 10^{-4};$$

$$\delta_{r^2} = 2 \cdot \delta_r = 5,8 \cdot 10^{-4};$$

$$\Delta_{r^2} = \delta_{r^2} \cdot r^2 = 5,8 \cdot 10^{-4} \cdot (1,75)^2 = 17,8 \cdot 10^{-4};$$

$$\Delta_{(r+h)^2 - r^2} = \Delta_{(r+h)^2} + \Delta_{r^2} = 19,3 \cdot 10^{-4} + 17,8 \cdot 10^{-4} \approx 3,7 \cdot 10^{-3};$$

$$\delta_{(r+h)^2 - r^2} = \frac{\Delta_{(r+h)^2 - r^2}}{(r+h)^2 - r^2} = \frac{3,7 \cdot 10^{-3}}{17,5 \cdot 10^{-3}} = 0,21.$$

Якщо ж скористатися другою формулою, то одержимо такі похибки:

$$\Delta_{2r+h} = \Delta_{2r} + \Delta_h = 2\Delta_r + \Delta_h = 10 \cdot 10^{-4} + 5 \cdot 10^{-5} = 10,5 \cdot 10^{-4};$$

$$\delta_{2r+h} = \frac{\Delta_{2r+h}}{2r+h} = \frac{10,5 \cdot 10^{-4}}{3,505} = 3 \cdot 10^{-4};$$

$$\delta_{(2r+h)h} = \delta_{2r+h} + \delta_h = 3 \cdot 10^{-4} + 10^{-2} \approx 1 \cdot 10^{-2};$$

$$\Delta_{(2r+h)h} = \delta_{(2r+h)h} \cdot (2r+h) \cdot h = 1 \cdot 10^{-2} \cdot (3,505) \cdot 0,005 = 1,75 \cdot 10^{-4};$$

Як бачимо

$$\frac{\Delta_{(2r+h)h}}{\Delta_{(r+h)^2 - r^2}} = \frac{1,75 \cdot 10^{-4}}{3,7 \cdot 10^{-3}} = 4,75 \cdot 10^{-2} \approx 0,05,$$

тобто підрахунок за другою формулою дає змогу одержати результат у 20 разів точніший, ніж розрахунок за першою формулою, де відбувається віднімання близьких за значенням величин.

Наведений приклад засвідчує, що похибка результату залежить від порядку проведення обчислень і це потрібно враховувати при розрахунках. Алгебрично наведені формули тотожні, але для проведення обчислень кращою є друга. У першій формулі при відніманні близьких величин $(r+h)^2$ і r^2 різко збільшується відносна похибка.

2.3.2. Похибки добутку, ділення й обчислення довільної функції

Доведемо, що відносна похибка добутку дорівнює сумі відносних похибок співмножників.

Дійсно:

$$a = b \cdot c \Rightarrow (a + \Delta_a) = (b + \Delta_b)(c + \Delta_c) = b \cdot c + \Delta_b \cdot c + \Delta_c \cdot b + \Delta_b \cdot \Delta_c,$$

звідки випливає

$$\Delta_a = \Delta_b \cdot c + \Delta_c \cdot b + \Delta_b \cdot \Delta_c \Rightarrow \delta_a = \delta_b + \delta_c + \delta_b \cdot \delta_c.$$

Через те що $\delta_b \ll 1$ і $\delta_c \ll 1$, то $\delta_b \delta_c \ll \delta_b$, а тому

$$\delta_a = \delta_b + \delta_c; \quad (2.2)$$

$$\Delta_a = \delta_a \cdot |a| = (\delta_b + \delta_c) \cdot |b \cdot c|,$$

що й треба було довести.

Відносна похибка обчислення величини, зворотної до даної, дорівнює відносній похибці вихідної величини.

Щоб довести це, врахуємо, що, якщо $d = \frac{1}{c} = c^{-1}$, то

$$d + \Delta d = \frac{1}{c - \Delta c} \approx \frac{1}{c} \left(1 + \frac{\Delta c}{c}\right) = \frac{1}{c} (1 + \delta_c).$$

З цього випливає $d = \frac{\delta_c}{c}$, а значить $\delta_d = \frac{\Delta d}{d} = \frac{\delta_c/c}{1/c} = \delta_c$, що й треба було довести.

ло довести.

Враховуючи це, можна дійти висновку, що відносна похибка частки дорівнює сумі відносних похибок діленого та дільника:

$$a = \frac{b}{c} \Rightarrow \delta_a = \delta_b + \delta_c \Rightarrow \Delta_a = (\delta_b + \delta_c) b / c. \quad (2.3)$$

Аналогічно можна встановити, що відносна похибка піднесення до степеня n наближеного числа (n - натуральне ціле) дорівнює добутку відносної похибки основи на абсолютну величину показника степеня

$$a = b^n \Rightarrow \delta_a = |n| \delta_b \Rightarrow \Delta_a = |n| \Delta_b \cdot b^{n-1}. \quad (2.4)$$

Абсолютна похибка обчислення функції дорівнює добутку абсолютної похибки аргументу на абсолютну величину похідної від функції:

$$a = f(b) \Rightarrow \Delta_a = \Delta_b \cdot \left| \frac{df(b)}{db} \right| \Rightarrow \delta_a = \delta_b \cdot \left| \frac{df(b)}{db} \cdot \frac{b}{f(b)} \right|. \quad (2.5)$$

Приклад 1. Похибка обчислення лінійної функції $y = k \cdot x$.

$$\text{Маємо } \frac{dy}{dx} = k \Rightarrow \Delta_y = \Delta_x \cdot k \Rightarrow \delta_y = \frac{\Delta_y}{y} = \frac{\Delta_x \cdot k}{k \cdot x} = \frac{\Delta_x}{x} = \delta_x.$$

Приклад 2. Похибки обчислення синуса $y = \sin(x)$.

$$\text{У цьому випадку } \frac{dy}{dx} = \cos(x), \text{ а тому } \Delta_y = \Delta_x \cdot |\cos(x)|, \text{ а } \delta_y = \delta_x \cdot \frac{|x|}{|\operatorname{tg}x|}.$$

Приклад 3. Похибки обчислення косинуса $y = \cos(x)$.

$$\text{У цьому випадку } \frac{dy}{dx} = -\sin(x), \text{ а тому } \Delta_y = \Delta_x \cdot |\sin(x)|, \text{ а } \delta_y = \delta_x \cdot |x \cdot \operatorname{tg}x|.$$

Аналіз останніх прикладів дозволяє висновувати, що

- 1) похибка обчислень суттєво залежить від значення аргументу;
- 2) при малих значеннях аргументу обчислення косинуса здійснюється зі значно меншою відносною похибкою, ніж похибка завдання аргументу;
- 3) обчислення косинуса при значеннях аргументу, близьких до $\pm \pi / 2$ приводить до вельми значних обчислювальних похибок; відносна похибка визначення косинуса у цьому випадку у багато разів перевищує відносну похибку завдання кута;
- 4) відносна ж похибка визначення синуса у діапазоні $[-\pi / 2, \pi / 2]$ завжди менша за відносну похибку аргументу.

2.4. Поширення похибок округлення при обчисленнях

Перед цим ми розглянули приклади розрахунків похибок результатів обчислень, обумовлених впливом похибок вихідних даних. Тепер зосередимо увагу на похибках результату обчислень за рахунок похибок округлення.

При виконанні арифметичних дій на будь-якому обчислювальному пристрої (логарифмічній лінійці, калькуляторі або ЕОМ) неминуче округлення проміжних результатів до певної кількості розрядів, а арифметичні операції з округленням мають інші властивості, аніж точні операції. Так, точні операції є комутативними, асоціативними і дистрибутивними. Ті ж операції, реалізовані на обчислювальному пристрої (ОП), вже не є такими.

Машинна арифметика має власні характерні особливості. Правильно враховуючи їх, можна досягти високої ефективності у розв'язуванні задач на ЕОМ. Неуважність до цих особливостей нерідко приводить до помилкових результатів.

Нехай обчислення відбуваються на ЕОМ, в якій кожне число подається п'ятьма значущими цифрами. Складемо два числа

$$9,2654 + 7,1625 = 16,4279.$$

Результат містить 6 значущих цифр і не вміщується у розрядну сітку машини. Його буде округлено до 16,428, і при цьому виникне (крім похибки внаслідок похибок вихідних даних) похибка округлення.

Оскільки ОП працює завжди з фіксованою кількістю розрядів, то округлення виникає весь час при запису проміжних результатів у пам'ять машини. Округлення полягає у корегуванні останнього (молодшого) розряду. Але й буває, що ніякого корегування не відбувається, а відкидається частина числа, яка не вміщується. Похибка округлення при цьому більша, але значне скорочення машинного часу й довжини робочої програми у цілому є економічно вигідними.

Число у машині подається у такому нормалізованому вигляді

$$x = \pm q^p \cdot \left(\sum_{k=1}^s \alpha_k \cdot q^{-k} \right), \quad (2.6)$$

де q - основа системи позиційного зчислення; p - порядок числа; s - кількість значущих розрядів (додатне ціле); α_k - цифри числа у заданій системі зчислення, причому вважається, що α_1 не дорівнює нулеві. Отже для десяткової системи зчислення $q = 10$, порядок числа - p , кількість значущих десяткових розрядів - s , цифри $\alpha_k = 0, 1, \dots, 9$. Наприклад, для числа $0,00123406 \cdot 10^{-5} = 0,123406 \cdot 10^{-7}$ десятковий порядок дорівнює $p = -7$, кількість значущих розрядів $s = 6$, а цифри

$$\alpha_1 = 1; \quad \alpha_2 = 2; \quad \alpha_3 = 3; \quad \alpha_4 = 4; \quad \alpha_5 = 0; \quad \alpha_6 = 6.$$

У нормалізованій формі (6) вираз у дужках подає так звану мантису числа. Мантиса у нормалізованому десятковому поданні завжди менша за одиницю і більша або дорівнює 0,1. У цілому будь-яке число однозначно описується трьома його характеристиками:

- 1) знаком числа;
- 2) значенням мантиси (вираз у дужках у (6));
- 3) цілим числом p - порядком числа.

ЕОМ працюють в основному у двійковій системі зчислення, тому для них $q = 2$. Обсяг оперативної пам'яті ЕОМ, який віддається на запис поодинокого окремого числа, вимірюється у байтах (8 двійкових розрядів - бітів) і залежить від мови програмування, на який записано програму, і типу числа, під яким його оголошено у програмі.

Наприклад, у мові Fortran під запис цілого числа типу INTEGER відводять 2 байти, тобто 16 двійкових розряди. Один із цих розрядів займає запис знака числа (додатне число чи від'ємне). У решті розрядів записується абсолютне значення цього числа. Неважко зрозуміти, що у такий обсяг можна записати число, за абсолютним значенням не більше за 32512.

Довільне число типу REAL записується у Фортрані у 4 байти оперативної пам'яті. З 32 бітів цього обсягу один біт займає запис знака числа, 7 розрядів - двійковий запис десяткового порядку числа (із них 1 розряд - запис знака порядку) і 24 розряди займає запис мантиси числа. З цього випливає, що абсолютне значення порядку десяткового числа у цьому випадку не перевищує 32. Тобто тип REAL дозволяє оперувати з числами за абсолютним значенням від

10^{-32} до 10^{+32} . При цьому мантиса числа, маючи для запису 24 розряди, зберігає $k = n \cdot \lg(2) = 24 \cdot 0.3 \approx 7$ десяткових розрядів числа. Тобто розглядуваний тип зберігає сім вірних десяткових цифр у кожному числі.

Розглянемо типи даних, передбачені мовою Паскаль.

Тут для подання цілих чисел існують 5 типів даних:

- `byte` - займає 1 байт пам'яті; за його допомогою можуть зберігатися цілі додатні числа від 0 до 255;
- `shortint` (коротке ціле) - займає теж 1 байт пам'яті; цей тип зберігає цілі числа від -128 до 127;
- `integer` (ціле) - для даних цього типу відводять 2 байти пам'яті; за його допомогою записуються цілі числа від -32768 до +32767;
- `word` - займає теж 2 байти; ним записуються лише додатні цілі від 0 до 65535;
- `longint` (довге ціле) - обіймає 4 байти пам'яті; зберігає числа від -2.147.483.648 до +2.147.483.647.

Дійсні дані у Паскалі подані наступними типами:

- `single` - на запис числа цього типу відводять 4 байти пам'яті, з них 24 біти займає запис мантиси, а 7 бітів - запис порядку числа; ним можуть бути подані дійсні числа з абсолютним значенням від $1,5 \cdot 10^{-45}$ до $3,4 \cdot 10^{+38}$ і з 7 вірними десятковими розрядами;
- `real` - займає 6 байтів (48 бітів); із них 7 бітів - запис порядку, а 40 бітів - мантиса числа; записуються числа з абсолютним значенням від $2,9 \cdot 10^{-39}$ до $1,7 \cdot 10^{+38}$ з 12 вірними десятковими розрядами;
- `double` - запис має обсяг у 8 байтів (64 бітів), запис порядку займає 10 бітів, запис мантиси - 54 бітів; ним зберігаються числа з абсолютним значенням від $5 \cdot 10^{-324}$ до $1,7 \cdot 10^{+308}$ і з 16 вірними десятковими розрядами;
- `extended` - тип, під який відводять 10 байтів оперативної пам'яті (14 бітів - під запис порядку і 65 - під запис мантиси); при цьому забезпечується збереження чисел від $1,9 \cdot 10^{-4951}$ до $1,1 \cdot 10^{+4932}$ з 19 вірними десятковими розрядами.

У середовищі комп'ютерної системи MatLAB усі числові дані мають тип ***double***, який по всіх показниках збігається з відповідним типом мови Паскаль.

Максимальна відносна похибка округлення при записі числа у пам'ять ЕОМ залежить не від його величини, а лише від кількості s десяткових розрядів у запису мантиси у його поданні на ЕОМ:

$$\delta \approx 1 \cdot 10^{-s+1}.$$

Наприклад, для типу REAL у Фортрані $\delta = 1 \cdot 10^{-6}$, для того ж типу мови Паскаль $\delta = 1 \cdot 10^{-11}$. Дані типу ***double*** у мові Паскаль і системі MatLAB мають елементарну похибку округлення

$$\delta = 1 \cdot 10^{-15}. \quad (2.7)$$

Розглянемо процес поширення похибок округлення при обчисленнях і впливу їх на похибку визначення результату обчислення.

Припустимо, потрібно скласти на ЕОМ дві величини

$$y = x_1 + x_2.$$

Цей процес розкладається на 4 етапи:

1) запис числа x_1 у пам'ять ЕОМ; при цьому, навіть коли вихідне значення x_1 відоме точно, при його записі може виникнути похибка округлення δ (це може бути, якщо точне значення величини містить більшу кількість десяткових розрядів, ніж s - кількість розрядів для запису числа в ЕОМ): $\delta_{x_1} = \delta$; $\Delta_{x_1} = \delta \cdot x_1$;

2) запис числа x_2 у пам'ять; це приводить до появи аналогічної похибки $\delta_{x_2} = \delta$; $\Delta_{x_2} = \delta \cdot x_2$;

3) підсумовування величин x_1 і x_2 ; при цьому перші дві похибки впливають на результат сумування $\delta_{x_1+x_2} = \frac{\Delta_{x_1} + \Delta_{x_2}}{x_1 + x_2} = \frac{\delta \cdot (x_1 + x_2)}{x_1 + x_2} = \delta$;

4) запис результату у пам'ять; при цьому може виникнути похибка округлення δ , якщо кількість розрядів в результаті перевищує кількість розрядів у запису числа у пам'ять; у цілому результуюча відносна похибка результату може сягати величини $\delta_y = \delta_{x_1+x_2} + \delta = 2\delta$, а абсолютну похибку сумування можна оцінити за формулою $\Delta_y = 2\delta \cdot (x_1 + x_2)$.

Якщо тепер перейти до сумування (послідовного) трьох величин

$$y = (x_1 + x_2) + x_3,$$

то, повторюючи попередній аналіз, дістанемо

■ похибка результату першого підсумовування $(x_1 + x_2)$ знайдена перед цим $\Delta_y = 2\delta \cdot (x_1 + x_2)$;

■ похибка запису числа x_3 $\Delta_{x_3} = \delta \cdot x_3$;

■ похибка сумування (проходження похибок вихідних даних)

$$\Delta_{x_1+x_2+x_3} = \Delta_{x_1+x_2} + \Delta_{x_3} = 2\delta \cdot (x_1 + x_2) + \delta \cdot x_3;$$

$$\delta_{x_1+x_2+x_3} = \frac{\Delta_{x_1+x_2+x_3}}{x_1 + x_2 + x_3} = \delta \frac{2(x_1 + x_2) + x_3}{x_1 + x_2 + x_3};$$

■ похибка запису результату у пам'ять ЕОМ - δ .

Разом, одержуємо

$$\delta_y = \delta_{x_1+x_2+x_3} + \delta = \delta \cdot \left[\frac{2(x_1 + x_2) + x_3}{x_1 + x_2 + x_3} + 1 \right] = \delta \cdot \frac{3(x_1 + x_2) + 2x_3}{x_1 + x_2 + x_3};$$

$$\Delta_y = \delta \cdot [3(x_1 + x_2) + 2x_3].$$

Поширення цього результату на підсумовування n чисел

$$y = x_1 + x_2 + \dots + x_n$$

приводить до наступних висновків

$$\Delta_y = \delta \cdot [n(x_1 + x_2) + (n-1) \cdot x_3 + \dots + 3x_{n-1} + 2x_n]; \quad (2.8)$$

$$\delta_y = \delta \cdot \frac{n(x_1 + x_2) + (n-1) \cdot x_3 + \dots + 3x_{n-1} + 2x_n}{x_1 + x_2 + x_3 + \dots + x_{n-1} + x_n}.$$

Одержаний результат є слушним у випадку, коли усі доданки додатні. Якщо вони усі від'ємні, можна користуватися одержаними результатами, розуміючи в них під x_i їхні абсолютні значення.

Становище різко змінюється при відніманні чисел (підсумовуванні чисел із протилежними знаками).

Отже, нехай $y = x_1 - x_2$; ($x_1 > x_2 > 0$). Повторюючи попередні міркування, одержимо

$$\delta_y = \delta + \frac{\Delta_{x_1} + \Delta_{x_2}}{|x_1 - x_2|} = \delta \cdot \left[1 + \frac{|x_1 + x_2|}{|x_1 - x_2|} \right] = 2\delta \cdot \frac{x_1}{x_1 - x_2}. \quad (2.9)$$

Як бачимо, похибка при відніманні чисел більше за похибку при їхньому додаванні у $\frac{x_1}{x_1 - x_2} > 1$ разів. Якщо віднімаються близькі величини, ця похибка може бути вельми значною.

Приклад 1. Нехай $x_1 = 3,14569$, $x_2 = 3,14551$. Оцінимо похибку віднімання цих чисел.

Вважаючи, що відносна похибка округлення $\delta = 1 \cdot 10^{-6}$, із формули (9) одержимо

$$\delta_y = 2\delta \cdot \frac{x_1}{x_1 - x_2} = 2 \cdot 10^{-6} \frac{3,14569}{0,00018} = 0,035.$$

Ця похибка більша за похибку підсумовування тих самих чисел у 17 500 разів.

Приклад 2. Порівняємо похибки одержання того самого результату за двома формулами

$$y_1 = \frac{x_1 - x_2}{x_3}; \quad y_2 = \frac{x_1}{x_3} - \frac{x_2}{x_3}.$$

Похибки округлення при розрахунку згідно першої формули:

1) занесення x_1 у пам'ять: $\delta_{x_1} = \delta$; $\Delta_{x_1} = \delta \cdot x_1$;

2) занесення x_2 у пам'ять: $\delta_{x_2} = \delta$; $\Delta_{x_2} = \delta \cdot x_2$;

3) віднімання x_1 і x_2 :

$$\Delta_{x_1 - x_2} = \Delta_{x_1} + \Delta_{x_2} = \delta \cdot (x_1 + x_2); \quad \delta_{x_1 - x_2} = \delta \cdot \frac{x_1 + x_2}{|x_1 - x_2|};$$

4) занесення різниці у пам'ять

$$\delta_y = \delta_{x_1 - x_2} + \delta = \delta \cdot \left(\frac{x_1 + x_2}{x_1 - x_2} + 1 \right) = \delta \cdot \frac{2x_1}{x_1 - x_2};$$

5) занесення x_3 у пам'ять: $\delta_{x_3} = \delta$;

$$6) \text{ ділення } \delta_{(x_1-x_2)/x_3} = \delta_y + \delta_{x_3} = \delta \cdot \frac{3x_1 - x_2}{x_1 - x_2};$$

7) занесення результату у пам'ять

$$\delta_{y1} = \delta_{(x_1-x_2)/x_3} + \delta = \delta_y + \delta_{x_3} = \delta \cdot \frac{4x_1 - 2x_2}{x_1 - x_2}.$$

Оцінка похибок розрахунку за другою формулою:

$$1) \text{ занесення } x_1 \text{ у пам'ять: } \delta_{x_1} = \delta; \quad \Delta_{x_1} = \delta \cdot x_1;$$

$$2) \text{ занесення } x_3 \text{ у пам'ять: } \delta_{x_3} = \delta;$$

$$3) \text{ ділення } x_1 \text{ на } x_3: \delta_{x_1/x_3} = \delta_{x_1} + \delta_{x_3} = 2\delta;$$

$$4) \text{ занесення результату у пам'ять } \delta_1 = \delta_{x_1/x_3} + \delta = 3\delta; \quad \Delta_1 = 3\delta \cdot x_1 / x_3;$$

$$5) \text{ аналогічно похибка від другого ділення } x_2 \text{ на } x_3: \quad \delta_2 = 3\delta;$$

$$\Delta_2 = 3\delta \cdot x_2 / x_3;$$

$$6) \text{ віднімання: } \Delta_{x_1/x_3 - x_2/x_3} = \Delta_1 + \Delta_2 = 3\delta \cdot (x_1 + x_2) / x_3;$$

$$\delta_{x_1/x_3 - x_2/x_3} = \Delta_1 + \Delta_2 = 3\delta \cdot (x_1 + x_2) / (x_1 - x_2);$$

7) занесення результату у пам'ять

$$\delta_{y2} = \delta_{x_1/x_3 - x_2/x_3} + \delta = \delta \cdot \frac{4x_1 + 2x_2}{x_1 - x_2}.$$

Тепер можна порівняти похибки розрахунків за цими двома формулами, поділивши похибку за другою формулою на похибку за першою:

$$\frac{\delta_{y2}}{\delta_{y1}} = \frac{4x_1 + 2x_2}{4x_1 - 2x_2}.$$

Якщо, наприклад, $x_1 \approx x_2$, то розрахунок за другою формулою приведе до похибки у 3 рази більшій, ніж розрахунок за першою формулою.

Приклад 3. Розглянемо відносну похибку результату обчислень, обумовлену округленням, для двох варіантів обчислення площі тонкого кільця

$$y_1 = (r + h)^2 - r^2; \quad y_2 = (2r + h) \cdot h.$$

Нехай $r = 1,750$; $h = 5,0 \cdot 10^{-3}$. Відносні похибки вихідних даних покладемо рівними нулеві. Елементарну відносну похибку при округленні приймемо рівною $\delta = 10^{-6}$.

РОЗВ'ЯЗУВАННЯ.

Розрахунки похибок за першою формулою:

$$1) \text{ при додаванні } r + h \text{ і запису результату у пам'ять відносна похибка становитиме } \delta_{r+h} = \delta;$$

$$2) \text{ при піднесенні у квадрат попередня відносна похибка подвоїться, а при запису результату додасться ще } \delta: \delta_{(r+h)^2} = 2\delta + \delta = 3\delta;$$

$$\Delta_{(r+h)^2} = 3\delta \cdot (r + h)^2;$$

3) при записі результату множення r^2 у пам'ять виникає похибка округлення $\delta_{r^2} = \delta$; $\Rightarrow \Delta_{r^2} = \delta \cdot r^2$;

4) при обчисленні різниці абсолютні похибки додаються

$$\Delta_{y1} = \Delta_{(r+h)^2} + \Delta_{r^2} = 3\delta \cdot (r+h)^2 + \delta \cdot r^2 \approx 2\delta r(2r+3h); \Rightarrow$$

$$\Rightarrow \delta_{y1} \approx \delta \frac{2r(2r+3h)}{h(2r+h)} \approx \delta \frac{2r}{h} = 10^{-6} \cdot \frac{3,5}{5 \cdot 10^{-3}} = 7 \cdot 10^{-4}.$$

Розрахунки похибок за другою формулою:

1) відносна похибка після занесення результату додавання $2r+h$ у пам'ять $\delta_{2r+h} = \delta$;

2) відносна похибка після множення результату на h і занесення результату у пам'ять $\delta_{y2} = \delta + \delta = 2\delta = 2 \cdot 10^{-6}$.

Відношення одержаних похибок дорівнює

$$\frac{\delta_{y1}}{\delta_{y2}} = \frac{7 \cdot 10^{-4}}{2 \cdot 10^{-6}} = 350.$$

Отже, похибка результату при обчисленні різниці двох близьких величин у 350 разів більша за похибку за формулою, яка виключає таке віднімання.

Резюмуючи, слід відзначити таку важливу особливість похибок округлення, яка відрізняє її від інших видів похибок: похибки внаслідок округлення проміжних результатів можуть накопичуватися, внаслідок чого підсумкова похибка збільшується із зростанням кількості здійснених операцій. З цього випливає, що основним засобом зменшення підсумкової похибки округлення є зменшення кількості обчислювальних операцій.

ВИСНОВКИ. Задля зменшення похибок результату обчислень внаслідок округлення проміжних результатів слід уживати наступних заходів:

- 1) потрібно зводити до мінімуму кількість арифметичних дій;
- 2) додавання чисел слід здійснювати у порядку зростання їхніх абсолютних величин;
- 3) по можливості потрібно уникати віднімання близьких величин;
- 4) якщо при обчисленнях зустрічаються різниці близьких величин, слід спочатку обчислити ці різниці, а лише потім здійснити решту операцій.

2.5. Контрольні запитання

1. У чому полягає вміння раціонально організувати обчислювальний процес?
2. Які існують джерела виникнення похибок при обчисленнях?
3. Що таке "неусувні" похибки? З яких саме похибок складаються вони?
4. Що таке похибки "усікання"? похибки округлення? похибки поширення?

5. Дайте визначення поняттю абсолютної похибки, граничної абсолютної похибки наближеного числа.
6. Що таке відносна похибка? Чим визначається відносна похибка при округленні числа?
7. Перелічіть основні правила, що дозволяють зменшити вплив на похибку результату обчислень похибок вихідних даних.
8. Перелічіть основні правила, що дозволяють зменшити вплив на похибку результату обчислень похибок округлення.

3. Обчислення значень функцій

Майже в усіх випадках при виконанні обчислень використовуються значення тих чи інших елементарних функцій (степеневих, багаточленів (поліномів), логарифмічної, показникової, тригонометричних і зворотних до них, а також різноманітних комбінацій перелічених функцій).

При ручних розрахунках завжди можна скористатися для цієї мети готовими таблицями, які є досить різноманітними, і, можливо, інтерполяцією по них. Виключення становлять лише найпростіші функції - поліноми, які доводиться обчислювати, бо усі поліноми затабулювати неможливо.

Інакше обстоїть справа при обчисленнях на ЕОМ, через те, що запис у пам'ять машини великих таблиць елементарних функцій виявляється зовсім неможливим через обмежений обсяг пам'яті. Тому для роботи на ЕОМ необхідно мати стандартні програми обчислення елементарних функцій. Такі програми зазвичай зводяться до обчислень значень поліномів, бо будь-яку аналітичну функцію можна подати у вигляді розкладу у ряд по степенях аргумента.

При обчисленні на ЕОМ значень функцій, заданих формулами, далеко не однаково, у якому виді записано відповідну формулу. Математично еквівалентні співвідношення часто виявляються нерівноцінними з точки зору практичних обчислень (як це було доведено перед цим). Основні операції ЕОМ : +, -, *, /. Тому бажано подати розглядувану математичну задачу у вигляді послідовності цих операцій.

3.1. Обчислення значень поліномів. Схема Горнера

При однократному обчисленні значень полінома (багаточлена) невисокого степеня послідовність виконання операцій не має особливого значення. Однак для обчислення поліномів досить високого степеня або для обчислення багатьох значень полінома при різних значеннях аргументу послідовність виконання операцій є суттєвим чинником. Попереднє обчислення усіх потрібних степенів аргументу $x^2, x^3 \dots$ зазвичай є не вигідним, бо потребує досить значної кількості операцій: при обчисленні значень полінома n -го степеня для одержання степенів до x^n включно потрібно $n-1$ множень. Окрім того, потрібні ще n множень на коефіцієнти, тобто усього $2n-1$ множень і n додавань.

Меншої кількості дій - n множень і n додавань - потребує обчислення полінома за так званою *схемою Горнера*.

Нехай даний поліном n -го степеня

$$P_n(x) = a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + \dots + a_1 \cdot x + a_0. \quad (3.1)$$

Запишемо його у вигляді

$$P_n(x) = (((a_n \cdot x + a_{n-1}) \cdot x + a_{n-2}) \cdot x + \dots + a_1) \cdot x + a_0. \quad (3.2)$$

Обчислення значення полінома у формі (2) і називають *схемою Горнера*.

Для поліномів загального виду неможливо побудувати схему більш економну у сенсі кількості операцій.

Зобразимо схему алгоритму обчислень значень полінома за схемою Горнера.

Попередньо введемо позначення

$$A(1) = a_n; A(2) = a_{n-1}; \dots A(k) = a_{n-k+1}; \dots A(n+1) = a_0, \quad (3.3)$$

тобто подамо коефіцієнти у вигляді деякого одновимірного масиву (вектора) із кількістю елементів. Позначимо через s значення послідовно виконуваних арифметичних виразів у (2).

Тоді схему алгоритму Горнера можна подати у вигляді, наведеному на рис. 3.1.

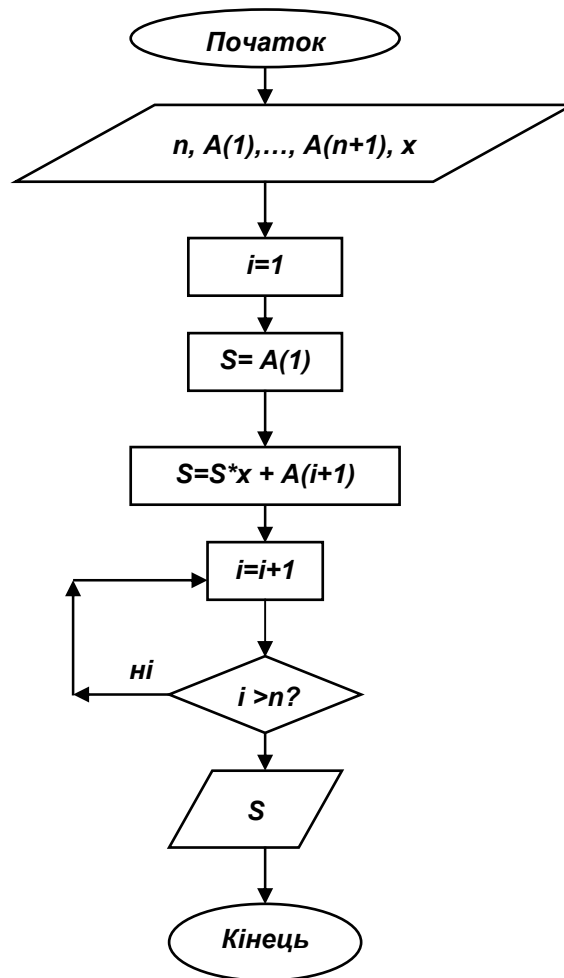


Рис. 3.1. Схема алгоритму Горнера

Складемо мовою Паскаль процедуру, яка здійснює розрахунок значення полінома за вказаним алгоритмом. Для цього перш за все введемо новий обчислювальний об'єкт у Паскалі - **polynom**, який матиме тип запису із двома полями - **pr** і **cf** (перше поле - ціле додатне число, що презентує порядок полінома, друге - масив із дійсних чисел, елементи якого дорівнюють відповідним коефіцієнтам полінома):

```
CONST maxprpolyn=16;
```

```

TYPE
  polynom=record
    pr: word;
    cf: array[0..maxprpolyn] of extended
  end;
VAR
  P1: polynom;

```

Тепер складемо текст процедури під назвою *polyval*, яка за заданим вектором коефіцієнтів полінома P і значенням аргументу x обчислює відповідне значення *pVal* цього полінома:

```

PROCEDURE polyval(P:polynom;x:extended; VAR pVal:extended);
{ Розрахунок значення полінома P(x) за заданим значенням
  аргументу x згідно схеми Горнера }
Var
  i,n: word;
Begin
  n:=P.pr;
  pVal:=P.cf[n];
  for i:=n-1 downto 0 do
    begin
      pVal:=pVal*x+P.cf[i];
    end;
  End;

```

Щоб увести поліном із клавіатури і проконтролювати правильність цього введення, потрібно ще скласти процедури введення полінома *pVvod* і виведення полінома на екран *pVyvod*.

У цілому текст програми, що реалізує алгоритм рис 3.1, може бути таким:

```

PROGRAM HORNER;
USES CRT;
CONST maxprpolyn=16;
TYPE
  polynom=record
    pr: word;
    cf: array[0..maxprpolyn] of extended
  end;
VAR
  P1: polynom;
  x,pVal: extended;
{=====}
PROCEDURE pVvod(var P:polynom);
{ Введення полінома P з клавіатури }
Var
  i,n: word;
Begin
  writeln('Vvod polynoma');
  writeln('a(n)*x^n+a(n-1)*x^(n-1)+...+ a(1)*x+a(0)');
  write('PORYADOK polynoma n=');
  readln(P.pr);
  n:=P.pr;
  for i:=n downto 0 do
    begin
      write('a(',i,')=');
      readln(P.cf[i]);
    end;
  end;

```

```

    end;
End;
{=====}
PROCEDURE pVvod(P:polynom;strArg:string);
{ Виведення полінома на екран дисплея зі знаком аргументу strArg }
Label p1;
Var
    i,n: word;
    s,zn: string;
Begin
    n:=P.pr;
    s:=strArg;
    if P.cf[n]=1 then write(s,'^',n) else write(P.cf[n]:9,s,'^',n);
    for i:= n-1 downto 0 do
        begin
            if P.cf[i]=0 then goto p1;
            if P.cf[i]=1
                then write(s,'^',i)
            else
                begin
                    if P.cf[i]>0
                        then write('+',abs(P.cf[i]):9,s,'^',i)
                    else write(P.cf[i]:9,s,'^',i);
                end;
        end;
    p1:
    end;
    writeln
End;
{=====}
PROCEDURE polyval(P:polynom;x:extended; var pVal:extended);
{ Розрахунок значення полінома P(x) за заданим значенням
    аргументу x згідно схемі Горнера }
Var
    i,n: word;
Begin
    n:=P.pr;
    pVal:=P.cf[n];
    for i:=n-1 downto 0 do
        begin
            pVal:=pVal*x+P.cf[i];
        end;
    End;
{===== ГОЛОВНА ПРОГРАМА =====}
BEGIN
    clrscr;
    pVvod(P1);
    pVvod(P1,'x');
    write('Уведіть значення аргументу x=');
    readln(x);
    polyval(P1,x, pVal);
    writeln('pVal=',pVal);
    readln
END.

```

Значно простіше обчислювати значення поліномів у середовищі MatLAB.

Наведемо деякі попередні відомості про комп'ютерну математичну систему MatLAB.

Система MatLAB є мовою-інтерпретатором високого рівня, розраховану на застосування в інженерних і науково-технічних розрахунках. Усі команди,

оператори вводяться у так званому командному вікні MatLAB, і в ньому ж відображуються результати їх виконання. в арифметичних виразах мови MatLAB використовуються наступні наступні знаки арифметичних операцій:

+	- додавання;
-	- віднімання;
*	- множення;
/	- ділення зліва направо;
\	- ділення справа наліво;
^	- піднесення до степеня.

У системі MatLAB можна здійснювати обчислення як у режимі калькулятора, коли у командний рядок записується послідовність арифметичних дій з числами (тобто звичайний арифметичний вираз), так і у програмному режимі, коли послідовність команд і операторів записується мовою MatLAB у окремий текстовий файл, який називається М-файлом і зберігається на диску. Щоб виконати дії, які записані у цьому файлі, достатньо ввести у командний рядок командного вікна MatLAB ім'я цього М-файла.

Основним обчислювальним об'єктом у MatLAB є числова матриця довільного розміру, елементами якої можуть бути як дійсні, так і комплексні числа. Тому усі вищевказані арифметичні операції застосовні як до звичайних дійсних чисел, так і до комплексних чисел, до векторів (одновимірних матриць) і до матриць.

Звичайні дійсні числа вводяться звичайно, як і в інших мовах високого рівня, у десятковому поданні, де десятковий порядок числа відокремлюється від мантиси за допомогою символу e .

Комплексні числа вводяться як сума дійсної і уявної частин, причому уявна частина виділяється множником i або j при ній.

Вектор-рядок вводиться у командному рядку у виді послідовності його елементів, яка міститься у квадратних дужках. Окремі елементи в рядку розділяються один від одного пропусками або комами. Вектор-стовпець вводиться аналогічно, за виключенням того, що його елементи відокремлюються один від одного знаком "крапка з комою".

Матриця вводиться послідовністю своїх елементів, яка також розміщується у квадратних дужках. Введення здійснюється порядково. Елементи кожного рядка відокремлюються один від одного пропусками або комами, а окремі рядки відокремлюються один від одного крапкою з комою.

Для обчислення значення полінома за заданим значенням його аргументу x достатньо ввести вектор A коефіцієнтів полінома (див. (3)), і застосувати процедуру *polyval*(A, x).

Наведемо приклад. Нехай потрібно обчислити значення полінома $P(x) = x^6 + 2x^5 - 3x^4$ при значенні аргументу $x = 2$. Для цього досить записати три команди

```
A=[1 2 -3 0 0 0 0]
A =
```

```

1 2 -3 0 0 0 0
x = 2
x = 2
polyval(A,x)
ans = 80

```

Значення поліному знайдено.

У MatLAB передбачені ще кілька процедур оперування з поліномами::

■ процедура **poly(V)** утворює вектор коефіцієнтів полінома, який має корені, подані у векторі *V*; для прикладу відшукаємо коефіцієнти полінома з такими коренями $x_1 = -3$; $x_2 = -1 + 5j$ і $x_3 = -1 - 5j$ (де *j* - уявна одиниця):

```

» x1=-3
x1 = -3
» x2=-1+5j
x2 = -1 + 5i
» x3=-1-5j
x3 = -1 - 5i
» V=[x1 x2 x3]
V = -3 -1 + 5i -1 - 5i
» P=poly(V)
P = 1 5 32 78

```

одержаний результат означає, що відповідний поліном має вигляд

$$P(x) = x^3 + 5x^2 + 32x + 78;$$

■ процедура **roots(P)** обчислює корені полінома, заданого вектором *P* своїх коефіцієнтів:

```

» x=roots(P)
x =
-1 + 5i
-1 - 5i
-3

```

■ процедура **conv(P1,P2)** формує вектор коефіцієнтів полінома, що є добутком поліномів, поданих заданими векторами *P1* і *P2*; при цьому прийнято говорити, що здійснюється *згортка векторів P1 і P2*; наприклад

```

» P1=[1 -x1]
P1 = 1 3
» P2=[1 -x2]
P2 = 1 1 - 5i
» P3=[1 -x3]
P3 = 1 1 + 5i
» P23=conv(P2,P3)
P23 = 1 2 26
» P123=conv(P1,P23)
P123 = 1 5 32 78

```

як і годиться, ми одержали той самий поліном;

■ процедура **deconv(P1,P2)** здійснює зворотну операцію - обчислює вектор коефіцієнтів полінома, що є часткою від ділення полінома *P1* на поліном *P2*, (або, по-іншому, здійснюється *обернена згортка векторів P1 і P2*); наприклад:

```

» dP=deconv(P123,P1)

```

dP = 1 2 26

■ нарешті, процедура *polyder(P)* відшукує вектор-масив коефіцієнтів полінома, що є похідною від полінома, вектор *P* коефіцієнтів якого заданий

» PP=polyder(P)

PP = 3 10 32

тобто похідна від наведеного вище полінома має вигляд

$$PP(x) = 3x^2 + 10x + 32.$$

3.2. Обчислення елементарних функцій за допомогою рядів

Степеневі ряди є вельми важливим апаратом для табулювання функцій, бо їхнє застосування дозволяє звести задачу обчислення значень функцій до задачі обчислення полінома, тобто до виконання арифметичних операцій.

Підґрунтям степеневого розкладання функції є так званий ряд Тейлора:

$$f(x) = f(a) + \frac{df_a}{dx}(x-a) + \frac{1}{2!} \frac{d^2 f_a}{dx^2}(x-a)^2 + \dots + \frac{1}{n!} \frac{d^{(n)} f_a}{dx^n}(x-a)^n + \dots, \quad (3.4)$$

де позначено

$$\frac{d^{(n)} f_a}{dx^n} = \left. \frac{d^{(n)} f(x)}{dx^n} \right|_{x=a}.$$

Зазвичай у формулі (4) вважають $a=0$, завдяки чому одержується звичайний степеневий ряд за степенями x , обриваючи який у потрібному місці, ми одержуємо поліном, який наближає функцію.

Наведемо приклади рядів для деяких функцій:

$$\left\{ \begin{array}{l} e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} + \dots \\ \cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + (-1)^n \frac{x^{2n}}{(2n)!} + \dots \\ \sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \dots \end{array} \right. \quad (3.5)$$

$$\left\{ \begin{array}{l} \frac{1}{1-x} = 1 + x + x^2 + x^3 + \dots + x^n + \dots \\ \frac{1}{1+x} = 1 - x + x^2 - x^3 + \dots + (-1)^n x^n + \dots \\ \sqrt{1 \pm x^2} = 1 \pm \frac{1}{2}x - \frac{1}{2 \cdot 4}x^2 \pm \frac{1 \cdot 3}{2 \cdot 4 \cdot 6}x^3 - \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6 \cdot 8}x^4 + \dots \\ \operatorname{tg} x = x + \frac{1}{3}x^3 + \frac{2}{15}x^5 \pm \frac{117}{315}x^7 + \dots \end{array} \right. \quad (3.6)$$

Для логарифмічної функції безпосередньо з формули (4) Тейлора можна одержати ряд:

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots + (-1)^{n-1} \frac{x^n}{n} + \dots, \quad (3.7)$$

який, однак, малопридатний для обчислення логарифмів, через те, що годиться лише для значень x , що задовільняють умові $-1 < x \leq 1$. Практично для обчислення значень натурального логарифма використовують ряд, який випливає з наведеної формули:

$$\ln \frac{1+x}{1-x} = 2 \left(x + \frac{x^3}{3} + \frac{x^5}{5} + \dots + \frac{x^{2n+1}}{2n+1} + \dots \right), \quad (3.8)$$

Цей ряд збігається лише за $|x| < 1$, але дроб $\frac{1+x}{1-x}$ може при цьому приймати будь-які додатні значення. Наприклад, для відшукування величини $\ln 5$, достатньо покласти $\frac{1+x}{1-x} = 5$, звідки $x = \frac{5-1}{5+1} = \frac{2}{3}$. Таким чином

$$\ln 5 = 2 \left[\frac{2}{3} + \frac{1}{3} \left(\frac{2}{3} \right)^3 + \frac{1}{5} \left(\frac{2}{3} \right)^5 + \dots \right].$$

Часто використовується також і біноміальний ряд, частковим випадком якого є відома формула бінома Ньютона:

$$(1 \pm x)^m = 1 \pm mx + \frac{m(m-1)}{2!} x^2 \pm \frac{m(m-1)(m-2)}{3!} x^3 + \dots \\ \dots + (-1)^n \frac{m(m-1)\dots(m-n+1)}{n!} x^n + \dots \quad (|x| \leq 1) \quad (3.9)$$

Дійсно, за натуральних m коефіцієнти ряду, починаючи з деякого місця, дорівнюватимуть нулю, тобто ряд обірветься.

Біноміальний ряд є зручним при піднесення до дробового степеня та добування коренів.

Просто виглядає і легко одержується також ряд для функції $\arctg(x)$:

$$\arctg(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \dots + (-1)^n \frac{x^{2n+1}}{2n+1}, \quad (3.10)$$

який, як і ряд для логарифмічної функції, збігається лише в зоні $-1 \leq x \leq 1$. Її для обчислення $\arctg(x)$ ряду (10) цілком достатньо, через те, що для $|x| > 1$ можна скористатися тотожністю $\arctg(x) = \pi/2 - \arctg(1/x)$.

При обчисленні функцій за допомогою степеневих рядів часто зручно користуватися рекурентними співвідношеннями, які дозволяють обчислювати черговий член ряду не безпосередньо, а через обчислення попередніх членів.

Рекурентним (або зворотним) співвідношенням називають рівність, яка пов'язує між собою два або кілька сусідніх членів послідовності або ряду. За допомогою такої рівності можна визначити наступний член ряду через попередні. У деяких випадках послідовність задається не виразом загального члена, а заданням перших її членів і рекурентного співвідношення, яке визначає решту членів. У такий спосіб, наприклад, задається відома послідовність чисел Фібо-

наччі (1, 1, 2, 3, 5, 8, 13, 21, 34, ...), в якій кожне наступне число дорівнює сумі двох попередніх:

$$F_0 = 1; \quad F_1 = 1; \quad \dots \quad F_{n+1} = F_n + F_{n-1}.$$

Це співвідношення набагато простіше за вираз для загального члена

$$F_n = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^{n+1} - \left(\frac{1-\sqrt{5}}{2} \right)^{n+1} \right]. \quad (3.11)$$

Для наведених вище рядів рекурентні співвідношення можуть бути виведені безпосередньо. Простіше за все взяти відношення двох сусідніх членів. Розглянемо, для прикладу, степеневий ряд (5) для функції e^x . Його загальний член має вигляд $a_n = x^n / n!$. Узявши відношення наступного члена до попере-

днього, одержимо $\frac{a_{n+1}}{a_n} = \frac{x^{n+1}}{(n+1)!} \frac{n!}{x^n} = \frac{x}{n+1}$. Таким чином, для двох сусідніх

членів ряду одержуємо рекурентне співвідношення $a_n = a_{n-1} \frac{x}{n}$, яке є дуже зручним для послідовного обчислення членів ряду.

При обчисленнях з рядами ми замінювали суму ряду його частковою сумою, тобто обмежувалися певною кількістю членів. Природно, цим ми привносимо похибку (похибку метода або похибку усікання), і виникає питання про оцінку величини цієї похибки. При цьому виникає і головне практичне питання: скільки членів ряду потрібно зберігти, щоб похибка, яка одержується, не перевищувала задану.

Якщо члени ряду убувають досить швидко і притому з самого початку, то вигідно мати справу зі знакоперемінним рядом, похибка якого легко оцінюється. Дійсно, з властивостей рядів є відомим, що сума знакоперемінного ряду менша за його перший член (за абсолютною величиною). Звідси випливає, що, замінюючи суму ряду його частковою сумою, ми припускаємо похибку, не перевищуючу за модулем першого з відкинутих членів.

Проте потрібно пам'ятати, що ця оцінка є корисною лише за вказаних умов. Якщо ж члени ряду убувають повільно або убувають хоча й швидко, але не з початку, а перші члени ряду досить великі, то хоча загальна теорема про суму ряду залишається слухною, але фактична похибка буде помітно більшою внаслідок похибки віднімання перших великих членів. У таких випадках набагато вигідніше мати справу з рядами, усі члени яких додатні. Наприклад, члени рядів (5) убувають досить швидко і ряди збігаються за будь-якого x . Проте за великих значень аргумента (наприклад, $x > 10$) перші члени цих рядів досить швидко зростають, і тому обчислення $\sin x$ і $\cos x$ за допомогою цих рядів є надзвичайно утрудненим через те, що при вирахуванні великих перших членів відбувається така втрата точності, яку неможливо відшкодувати обчисленням великої кількості доданків.

Для рядів з додатними членами оцінка похибки є більш складною, і ніяких загальних методів, придатних для усіх рядів, запропонувати неможна.

Приклад. Обчислимо значення $\ln 5$ за допомогою ряду (8), узявши п'ять членів ряду, і оцінимо одержану похибку.

Як зазначалося, у цьому випадку слід покласти $x = 2/3$. Тоді

$$\ln 5 = 2 \left[\frac{2}{3} + \frac{1}{3} \left(\frac{2}{3} \right)^3 + \frac{1}{5} \left(\frac{2}{3} \right)^5 + \frac{1}{7} \left(\frac{2}{3} \right)^7 + \frac{1}{9} \left(\frac{2}{3} \right)^9 \right].$$

Абсолютна похибка цієї рівності дорівнює сумі ряду

$$R = 2 \left[\frac{1}{11} \left(\frac{2}{3} \right)^{11} + \frac{1}{13} \left(\frac{2}{3} \right)^{13} + \dots \right].$$

Оцінку величини R можна одержати у такий спосіб. Якщо всередині квадратних дужок усі множники перед степенями $2/3$ замінити на $1/11$, то величина суми зможе тільки збільшитися. Отже $R < \frac{2}{11} \left[\left(\frac{2}{3} \right)^{11} + \left(\frac{2}{3} \right)^{13} + \dots \right]$.

Але сума праворуч тепер становить суму геометричної прогресії з першим членом $\left(\frac{2}{3} \right)^{11}$ і знаменником $\left(\frac{2}{3} \right)^2 = \frac{4}{9}$. Тому

$$R < \frac{2}{11} \frac{\left(\frac{2}{3} \right)^{11}}{1 - \frac{4}{9}} = \frac{18}{55} \left(\frac{2}{3} \right)^{11} < 0,004.$$

Наведений метод дозволяє розв'язати і зворотну задачу - визначити кількість членів ряду, які потрібно врахувати при обчисленні, щоб одержати значення $\ln 5$ з наперед заданою точністю.

Швидкості збіжності наведених вище рядів, тобто, у кінцевому рахунку, кількість членів, які потрібно утримувати, щоб одержати суму з потрібною точністю, є вельми різними. Більш того, навіть для того самого ряду потрібна кількість членів змінюється у залежності від значення x . Внаслідок цього швидкодія програм, які обчислюють значення елементарних функцій за допомогою рядів, виявляється різною для різних x .

Розглянуті степеневі ряди є зручним засобом для програмування обчислення елементарних функцій. Кожний з цих рядів легко програмується як звичайний цикл, який можна оформлювати і як арифметичний, і як ітераційний.

Для прикладу розглянемо показникову (експоненційну) функцію $y = e^x$.

Помітимо, що члени ряду $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} + \dots$ зв'язані співвідношенням $a_n = a_{n-1} \cdot x / n$. Якщо не обмежувати зону змінювання аргументу, то цикл потрібно писати як ітераційний. Для обчислень з машинною точністю можна

перевіряти загальний член ряду на збіг з нулем, тобто обчислювати доти, поки члени ряду не стануть машинним нулем. Втім, у цьому немає ніякої потреби. Набагато простіше і швидше порівнювати дві сусідні суми і припиняти обчислення тоді, коли вони збіжуться, тобто коли додавання чергового доданку не буде змінювати суму. Звичайно цей момент настає раніше, ніж члени ряду обертаються на машинний нуль.

Такий підхід є незручним для від'ємних аргументів, великих за абсолютним значенням, бо у цьому випадку ряд одержується знакоперемінним, і його перші члени є великими за абсолютним значенням, тоді як значення функції є малим. У цьому випадку може вийти велика втрата точності. Зручніше у цьому випадку обчислювати e^{-x} , тобто обчислювати значення функції всеж для додатного показника степеня (тобто працюючи зі знакосталим рядом), а потім користуватися тотожністю $e^x = 1/e^{-x}$.

Оформимо обчислення експоненти у вигляді процедури на Паскалі.

```
function Ex(x:extended):extended;
var i: word;
    u,ex,s: extended;
begin
  u:=1;   ex:=1;   i:=1;
  if x<>0 do
    begin
      repeat
        s:=ex;
        u:=u*abs(x)/i;
        ex:=s+u;
        i:=i+1;
      until (ex.ne.s);
      if x<0 do ex:=1/ex;
    end
  end;
end;
```

3.3. Обчислення функцій за допомогою ланцюгового дробу

Окрім степеневих рядів, при обчисленні елементарних функцій на ЕОМ використовуються також ланцюгові дроби.

Ланцюговим, або неперервним, дробом називають вираз виду

$$\frac{a_0}{b_0 + \frac{a_1}{b_1 + \frac{a_2}{b_2 + \frac{a_3}{b_3 + \dots}}}} \quad (3.12)$$

Цей дріб може бути як скінченним, так і нескінченним. Числа a_0, a_1, a_2, \dots у виразі (12) називають частковими чисельниками, а b_0, b_1, b_2, \dots - частковими знаменниками ланцюгового дробу.

Якщо обірвати ланцюговий дріб у будь-якому місці, то звичайний дріб, який одержується після перетворення того дробу, що залишився, називають придатним дробом.

Звичайно замість (12) застосовують більш компактний запис, записуючи ланки ланцюгового дробу, з'єднані знаками плюс, поставленими на рівні знаменників:

$$\frac{a_0}{b_0 + \frac{a_1}{b_1 + \frac{a_2}{b_2 + \frac{a_3}{b_3 + \dots}}}} \quad (3.13)$$

Ланцюговий дріб можна привести до вигляду:

$$\frac{1}{\alpha_0 + \frac{1}{\alpha_1 + \frac{1}{\alpha_2 + \frac{1}{\alpha_3 + \dots}}}} = \frac{1}{\alpha_0 + \frac{1}{\alpha_1 + \frac{1}{\alpha_2 + \frac{1}{\alpha_3 + \dots}}}} \quad (3.14)$$

Такі ланцюгові дроби називають звичайними, а його часткові знаменники - неповними частками звичайного ланцюгового дробу. Для звичайного ланцюгового дробу використовується також скорочене позначення

$$[\alpha_0, \alpha_1, \alpha_2, \alpha_3, \dots]. \quad (3.15)$$

Теорема. Якщо усі неповні частки звичайного ланцюгового дробу

$$[\alpha_0, \alpha_1, \alpha_2, \alpha_3, \dots]$$

є додатними, то для збіжності цього ланцюгового дробу необхідно і достатньо, щоб ряд

$$\sum_{n=1}^{\infty} \alpha_n = \alpha_1 + \alpha_2 + \alpha_3 + \dots \quad (3.16)$$

розходився.

Щоб перетворити степеневий ряд у ланцюговий дріб застосовують формулу Ейлера:

$$\sum_{n=0}^{\infty} c_n \cdot x^n = \frac{c_0}{1 - c_0 + c_1 \cdot x - c_1 + c_2 \cdot x - c_2 + c_3 \cdot x - \dots c_{n-1} + c_n \cdot x - \dots} \quad (3.17)$$

Питання про збіжність (17) розв'язується окремо і не пов'язане зі збіжністю степеневого ряду, який перетворюється.

Перетворення за допомогою формули Ейлера степеневого ряду для експоненційної функції приводить до такого ланцюгового дробу:

$$e^x = \frac{1}{1 - \frac{x}{1 + \frac{x}{2 - \frac{x}{3 + \frac{x}{2 - \frac{x}{5 + \dots 2 - 2k + 1 + \dots}}}}} \quad (3.18)$$

Наведений ланцюговий дріб збігається за будь-яких значень x , хоча швидкість збіжності невелика. Частіше використовують інший розклад:

$$e^x = 1 + \frac{2x}{2 - x} + \frac{x^2}{6} + \frac{x^2}{10} + \frac{x^2}{14} + \dots \frac{x^2}{2(2k + 1)} + \dots \quad (3.19)$$

Логарифмічну функцію можна подати у вигляді такого ланцюгового дробу:

$$\ln(1+x) = \frac{x}{1} - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \frac{x^5}{5} - \dots \quad (3.20)$$

Тригонометричні функції можна подати у вигляді:

$$\sin x = x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040} + \frac{x^9}{362880} - \dots \quad (3.21)$$

$$\cos x = 1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{720} + \frac{x^8}{40320} - \dots$$

але більш зручним є обчислення $\operatorname{tg}x$ за ланцюговим дробом

$$\operatorname{tg}x = \frac{x}{1} - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \frac{x^9}{9} - \dots \quad (3.22)$$

$$\operatorname{arctg}x = \frac{x}{1} + \frac{x^3}{3} - \frac{x^5}{5} + \frac{x^7}{7} - \frac{x^9}{9} + \dots$$

а потім скористатися співвідношеннями

$$\arcsin x = \operatorname{arctg} \frac{x}{\sqrt{1-x^2}}; \quad \cos x = \frac{1 - \operatorname{tg}^2\left(\frac{x}{2}\right)}{1 + \operatorname{tg}^2\left(\frac{x}{2}\right)}; \quad \sin x = \frac{2 \operatorname{tg}\left(\frac{x}{2}\right)}{1 + \operatorname{tg}^2\left(\frac{x}{2}\right)}$$

3.4. Обчислення функцій на ЕОМ

Обчислення значень більшості із зазначених арифметичних і алгебричних функцій на ЕОМ не становить труднощів. Кількість обчислювальних функцій і якість обчислення залежать від наявної програми-компілятора відповідної мови програмування. Таким чином, можливості ЕОМ у цьому відношенні залежать від мови або середовища програмування, що використовується. При цьому ми будемо говорити про мову програмування, коли відповідна програма-компілятор є *транслятором*, тобто *утворює* окремо, поза цього компілятора, *виконувани програму*, що записується на диск, або інший носій інформації. До таких мов-трансляторів відносяться Pascal, Fortran, C, C++, Delphi. Мову-середовище утворюють мови, споряджені компілятором типу *інтерпретатора*, який компілює утворений текст програми порядково, і ЕОМ одразу виконує зкомпільований рядок програми. При цьому не утворюється виконуваний файл і не записується на диск. Через це програми виконується лише у власному середовищі відповідної мови. До мов-середовищ належать Basic, MatLAB. В останні часи з'явилися засоби утворення за допомогою вказаних мов-інтерпретаторів виконуваних файлів. Так, версія Quick-Basic 45 має можливість, використовуючи власні бібліотеки і транслятор, створювати з програм, написаних мовою QB45, окремі самостійно виконувани програми. Аналогічні засоби є й у середовищі MatLAB.

Зазначені мови програмування далеко не є еквівалентними у можливостях обчислення функцій.

Найменш розвиненою у цьому відношенні мовою є Pascal. Це, мабуть, єдина мова програмування, в якій не передбачений знак (і відповідно, операція) піднесення до степеня. Вбудовані арифметичні функції тут подані такими підпрограмами-функціями:

Abs(x) - обчислює модуль дійсного числа x ;

ArcTan(x) - обчислює кут у радіанах (у межах від $-\pi/2$ до $+\pi/2$), тангенс якого дорівнює числу x ;

Cos(x) - обчислює косинус кута в x радіан;

Exp(x) - обчислює експоненту числа x ;

Frac(x) - обчислює дробову частину дійсного числа x ;

Int(x) - обчислює цілу частину дійсного числа x ;

Ln(x) - обчислює натуральний логарифм дійсного числа x ;

Sin(x) - обчислює синус кута в x радіан;

Sqr(x) - обчислює квадрат дійсного числа x ;

Sqrt(x) - обчислює корінь квадратний з дійсного числа x ;

Random - генерує випадкове число, рівномірно розподілене у діапазоні від 0 до 0.99.

У мові Basic передбачені як операція піднесення до степеня (знак \wedge), так і такі арифметичні вбудовані функції:

ABS (X) - обчислює модуль дійсного числа X ;

ATN (X) - обчислює кут у радіанах (у межах від $-\pi/2$ до $+\pi/2$), тангенс якого дорівнює числу X ;

COS (X) - обчислює косинус кута в X радіан;

EXP (X) - обчислює експоненту числа X ;

INT (X) - обчислює цілу частину дійсного числа X ;

LOG (X) - обчислює натуральний логарифм дійсного числа X ;

LOG10 (X) - обчислює логарифм дійсного числа X за основою 10;

SIN (X) - обчислює синус кута в X радіан;

SGN (X) - обчислює сигнум-функцію за алгоритмом:

+1, якщо $X > 0$;

-1, якщо $X < 0$;

0, якщо $X = 0$;

SQR (X) - обчислює корінь квадратний з дійсного числа X .

Зверніть увагу на появу нових важливих для інженера функцій десятичного логарифму та сигнум-функції.

Значно ширший діапазон арифметичних вбудованих функцій, потрібних інженеру, подано у мові Fortran, яку спеціально утворено для проведення чисельних розрахунків високого математичного рівня. Зокрема, у ній існує ціла низка операцій з комплексними числами і функцій від них. Але, на жаль, графічні та інші допоміжні функції Fortran настільки відстають від сучасного рівня, що цією мовою практично не користуються у персональних комп'ютерах. Але на

ґрунті цієї мови було створено найпотужнішу, як на цей час, обчислювальну математичну систему проведення інженерних обчислень - MatLAB.

У мові MatLAB передбачені такі елементарні арифметичні функції:

Тригонометричні і гіперболічні функції

- sin***(Z) - синус числа Z;
sinh(Z) - гіперболічний синус;
asin(Z) - арксинус (у радіанах, у діапазоні від $-\pi/2$ до $+\pi/2$);
asinh(Z) - обернений гіперболічний синус;
cos(Z) - косинус;
cosh(Z) - гіперболічний косинус;
acos(Z) - арккосинус (у діапазоні від 0 до π);
acosh(Z) - обернений гіперболічний косинус;
tan(Z) - тангенс;
tanh(Z) - гіперболічний тангенс;
atan(Z) - арктангенс (у діапазоні від $-\pi/2$ до $+\pi/2$);
atan2(X,Y) - чотириквadrантний арктангенс (кут у діапазоні від $-\pi$ до $+\pi$ між горизонтальним правим променем і променем, що проходить через точку з координатами X і Y);

- atanh***(Z) - обернений гіперболічний тангенс;
sec(Z) - секанс;
sech(Z) - гіперболічний секанс;
asec(Z) - арксеканс;
asech(Z) - обернений гіперболічний секанс;
csc(Z) - косеканс;
csch(Z) - гіперболічний косеканс;
acsc(Z) - арккосеканс;
acsch(Z) - обернений гіперболічний косеканс;
cot(Z) - котангенс;
coth(Z) - гіперболічний котангенс;
acot(Z) - арккотангенс;
acoth(Z) - обернений гіперболічний котангенс.

Експоненціальні функції

- exp***(Z) - експонента числа Z;
log(Z) - натуральний логарифм;
log10(Z) - десятковий логарифм;
sqrt(Z) - квадратний корінь із числа Z;
abs(Z) - модуль числа Z.

Цілочислові функції

- fix***(Z) - округлення до найближчого цілого в бік нуля;
floor(Z) - округлення до найближчого цілого в бік від'ємної нескінченності;
ceil(Z) - округлення до найближчого цілого в бік додатної нескінченності;

- round**(Z) - звичайне округлення числа Z до найближчого цілого;
rem(X,Y) - обчислення остачі від ділення X на Y;
sign(Z) - обчислення сигнум-функції числа Z
 (0 при Z=0, -1 при Z<0, 1 при Z>0).

Окрім елементарних у мові MatLAB передбачений цілий ряд спеціальних математичних функцій. Нижче наведений перелік і стислий зміст цих функцій. Правила звернення до них і використання користувач може знайти в описах цих функцій, що виводяться на екран, якщо набрати команду **help** і вказати в тому ж рядку ім'я функції.

Функції перетворення координат

- cart2sph** - перетворення декартових координат у сферичні;
cart2pol - перетворення декартових координат у полярні;
pol2cart - перетворення полярних координат у декартові;
sph2cart - перетворення сферичних координат у декартові.

Функції Бесселя

- besselj** - функція Бесселя першого роду;
bessely - функція Бесселя другого роду;
besseli - модифікована функція Бесселя першого роду;
besselk - модифікована функція Бесселя другого роду.

Бета-функції

- beta** - бета-функція;
betainc - неповна бета-функція;
betaln - логарифм бета-функції.

Гамма-функції

- gamma** - гамма-функція;
gammainc - неповна гамма-функція;
gammaln - логарифм гамма-функції.

Еліптичні функції й інтеграли

- ellipj** - еліптичні функції Якоби;
ellipke - повний еліптичний інтеграл;
expint - функція експоненціального інтегралу.

Функції похибок

- erf** - функція похибок;
erfc - додаткова функція похибок;
erfcx - масштабована додаткова функція похибок;
erfinv - обернена функція похибок.

Інші функції

- gcd** - найбільший загальний дільник;
lcm - найменше загальне кратне;
legendre - узагальнена функція Лежандра;
log2 - логарифм за основою 2;
pow2 - піднесення 2 до зазначеного степеня;
rat - подання числа у виді раціонального дробу;

rats - подання чисел у виді раціонального дробу.

Особливо важливим у мові MatLAB є те, що практично усі подані вбудовані елементарні математичні функції обчислюються і за комплексних значень аргумента Z. Завдяки цьому, наприклад, функція *sqrt* обчислює, на відміну від інших мов програмування, квадратний корінь із від'ємного аргументу, а функція *abs* при комплексному значенні аргументу обчислює модуль комплексного числа. У MatLAB є декілька додаткових функцій, розрахованих тільки на комплексний аргумент:

Функції комплексного аргумента

real(Z) - виділяє дійсну частину комплексного аргументу Z;

imag(Z) - виділяє уявну частину комплексного аргументу;

angle(Z) - обчислює значення аргументу комплексного числа Z (у радіанах від $-\pi$ до $+\pi$);

conj(Z) - видає число, комплексно спряжене щодо Z.

Найпростіші дії з комплексними числами - додавання, віднімання, множення, ділення і піднесення до степеня - здійснюються за допомогою звичайних арифметичних знаків - "+", "-", "*", "/", "\" і "^" - відповідно.

3.5. Передатні функції

У теоретичних інженерних дослідженнях великого поширення здобуло подання динамічних систем у вигляді передатних функцій. Таке подання можливе у випадку, коли поведження системи описується системою лінійних диференційних рівнянь зі сталими коефіцієнтами.

Наприклад, розглянемо диференційне рівняння n -го порядку з однією змінною:

$$a_n \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \dots + a_1 \frac{dy}{dt} + a_0 \cdot y = b_m \frac{d^m x}{dt^m} + b_{m-1} \frac{d^{m-1} x}{dt^{m-1}} + \dots + b_1 \frac{dx}{dt} + b_0 \cdot x, \quad (3.23)$$

де t - аргумент, y - вихідна змінна, змінювання якої у залежності від аргумента відшукується, x - вхідна змінна, змінювання якої у часі задається, a_k ($k = 0, 1, 2, \dots, n$) і b_j ($j = 0, 1, 2, \dots, m$) сталі коефіцієнти.

Рівняння (23) - *неоднорідне*. Неоднорідність рівняння утворюють ті члени рівняння, що перенесені у (23) праворуч від знака рівності. У математиці їх часто називають *вільними членами рівняння* (бо вони не залежать від шуканих змінних), а у теорії динамічних систем - *зовнішніми силами* (тут будь-який окремий член диференційного рівняння називають "силою" за аналогією з механікою). Величину $x(t)$ інженери найчастіше зовуть *зовнішньою дією* і розглядають її як *вхідну величину*, а шукану змінну $y(t)$ - як *вихідну величину*.

Якщо зовнішня дія відсутня, відповідна система називається *вільною*. Вільна система описується *однорідним* рівнянням

$$a_n \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \dots + a_1 \frac{dy}{dt} + a_0 \cdot y = 0. \quad (3.24)$$

Як відомо з математики, окремий незалежний розв'язок однорідного рівняння (24) має у загальному випадку вигляд

$$y(t) = C \cdot e^{pt}, \quad (3.25)$$

Підставляючи (25) у (24), матимемо:

$$C \cdot e^{pt} \cdot (a_n \cdot p^n + a_{n-1} \cdot p^{n-1} + \dots + a_1 \cdot p + a_0) = 0. \quad (3.26)$$

Рівність (26) обертається у тотожність при будь-якому значенні аргумента t (а саме за цієї умови вираз (25) можна назвати розв'язком рівняння (24)) у двох випадках:

- коли $C=0$;
- коли вираз у дужках у (26) дорівнює нулю:

$$a_n \cdot p^n + a_{n-1} \cdot p^{n-1} + \dots + a_1 \cdot p + a_0 = 0. \quad (3.27)$$

Перша умова відповідає так званому *тривіальному* розв'язку, коли рух у системі відсутній:

$$y(t) \equiv 0. \quad (3.28)$$

Звичайно цей специфічний частковий випадок не є цікавим для практичного використання, бо відповідає стану покою системи у положенні рівноваги, який є об'єктом вивчення у статичі. Тому завжди розглядають лише другий випадок.

Поліном

$$P_n(p) = a_n \cdot p^n + a_{n-1} \cdot p^{n-1} + \dots + a_1 \cdot p + a_0 \quad (3.29)$$

одержав назву характеристичного поліному рівнянь (24) і (23).

З (27) випливає, що вираз (25) є розв'язком рівняння (24) у випадку, коли величина p у ньому дорівнює одному з коренів характеристичного полінома (29).

Цей висновок є слушним тоді, коли серед коренів характеристичного рівняння немає кратних. Якщо ж серед коренів є корінь $p_i = \alpha$ кратності s , то розв'язок, що відповідає цьому кореню, має вигляд

$$y_i(t) = \left(\sum_{m=0}^{s-1} C_{i+m} \cdot t^m \right) \cdot e^{\alpha t}. \quad (3.30)$$

Загальна кількість коренів характеристичного рівняння і, відповідно, кількість незалежних розв'язків рівняння (24) і кількість сталих C_k інтегрування завжди дорівнює порядку диференційного рівняння. У цілому загальний розв'язок рівняння (24) у випадку відсутності кратних коренів має вигляд

$$y(t) = \sum_{i=1}^n C_i \cdot e^{p_i t}. \quad (3.31)$$

Знання значень коренів характеристичного полінома диференційного рівняння дозволяє одержати важливу інформацію про можливе поведіння системи, що описується цим рівнянням. Наприклад, якщо деякий корінь характе-

ристичного полінома є дійсним, то відповідна частина (25) розв'язку характеризує експоненційне змінювання вихідної величини з часом, якщо під аргументом t диференційного рівняння розуміється саме час. При цьому, якщо цей дійсний корінь є додатним, то з часом вихідна величина збільшується до нескінченності, що відповідає нестійкому рухові.

Таким чином, якщо серед коренів характеристичного полінома є хоча б один дійсний додатний, то рух такої системи буде нестійким.

Корені полінома з дійсними коефіцієнтами, як відомо, можуть бути тільки або дійсними, або мати вигляд комплексно-спряжених пар.

Нехай два корені утворюють комплексно спряжену пару, тобто мають вигляд

$$p_1 = \alpha + j \cdot \omega_o; \quad p_2 = \alpha - j \cdot \omega_o, \quad (3.32)$$

де $j = \sqrt{-1}$ - позначення уявної одиниці, α - дійсна частина цієї комплексно спряженої пари, ω_o - її уявна частина (не зменшуючи узагальненності, можна вважати $\omega_o > 0$).

Частина розв'язку диференційного рівняння, що відповідає цим двом кореням має вигляд

$$y(t) = e^{\alpha t} \cdot (C_1 \cdot e^{j\omega_o t} + C_2 \cdot e^{-j\omega_o t}). \quad (3.33)$$

З того, що вихідна величина $y(t)$ є дійсною величиною, і з (32) випливає, що це є можливим лише за умови, що сталі C_1 і C_2 є комплексно спряженими. Тому подамо останні у вигляді:

$$C_1 = \frac{A}{2} - j \frac{B}{2}; \quad C_2 = \frac{A}{2} + j \frac{B}{2}, \quad (3.34)$$

де вже A і B - дійсні величини. Підставляючи (34) у (33), одержимо:

$$y(t) = e^{\alpha t} \cdot \left(A \cdot \frac{e^{j\omega_o t} + e^{-j\omega_o t}}{2} - jB \cdot \frac{e^{j\omega_o t} - e^{-j\omega_o t}}{2} \right) = e^{\alpha t} \cdot (A \cdot \cos \omega_o t + B \cdot \sin \omega_o t) \quad (3.35)$$

Отже, при наявності серед коренів характеристичного полінома системи пари комплексно спряжених вигляду (32) відповідна частина розв'язку описує гармонічні коливання вихідної величини з коловою частотою, рівною уявній частині коренів, і з амплітудою, яка змінюється з часом за експоненційним законом з показником, що дорівнює дійсній частині коренів.

Якщо дійсна частина пари комплексно спряжених коренів є додатною, то відповідні коливання вихідної величини здійснюються з амплітудою, яка збільшується до нескінченності (розбіжні коливання). У цьому випадку відповідна реальна система, рух якої описується рівнянням (23) або (24), є нестійкою.

Нестійки системи є саморуйнівними, або їхній рух збігається до іншого, стійкого положення рівноваги. Тому реальні системи, з якими має справу інженер, завжди є (або мають бути зроблені) стійкими, тобто такими, характеристичні поліноми диференційних рівнянь яких містять лише корені з від'ємними (або рівними нулю) дійсними частинами.

Отже, якщо ми знайшли характеристичний поліном системи і відшукали його корені, то ми маємо змогу висновувати

- чи є досліджувана система стійкою чи нестійкою,
- чи є власний рух системи до положення рівноваги плавним (здійснюється за експоненціальним законом у часі), чи коливним (здійснюється за гармонічним законом),
- з якими коловими частотами здійснюється власний рух системи до положення рівноваги,
- як інтенсивно згасають відповідні власні коливання, або як швидко наближається система до положення рівноваги.

Поняття *передатної функції* системи є застосовним для систем, поведіння яких описується рівняннями виду (24) і тісно пов'язане з інтегральним перетворенням Лапласу.

Зображенням деякої заданої функції $f(t)$ за Лапласом називають деяку алгебричну функцію $F(p)$ комплексної змінної p , яка пов'язана з заданою функцією інтегральним перетворенням

$$\mathcal{L}\{f(t)\} = F(p) = p \int_0^{\infty} f(t) \cdot e^{-pt} dt. \quad (3.36)$$

З цього визначення випливають такі співвідношення між Лаплас-зображеннями похідних від функції з Лаплас-зображенням самої функції:

$$\mathcal{L}\left\{\frac{df(t)}{dt}\right\} = p \cdot [F(p) - f(0)]; \quad \mathcal{L}\left\{\frac{d^2 f(t)}{dt^2}\right\} = p^2 \cdot [F(p) - f(0)] - p \cdot f'(0), \quad (3.37)$$

де $f(0)$ - позначення для значення функції, а $f'(0)$ - значення похідної $\frac{df}{dt}$ при початковому значенні аргументу t (зазвичай його приймають рівним нулю).

Застосовуючи до рівняння (23) перетворення (36) Лапласа і використовуючи при цьому співвідношення (37) при нульових початкових умовах

$$f^{(n-1)}(0) = f^{(n-2)}(0) = \dots = f^{(1)}(0) = f(0) = 0, \quad (3.38)$$

одержимо

$$\begin{aligned} (a_n \cdot p^n + a_{n-1} \cdot p^{n-1} + \dots + a_1 \cdot p + a_0) \cdot Y(p) = \\ = (b_m \cdot p^m + b_{m-1} \cdot p^{m-1} + \dots + b_1 p + b_0) \cdot X(p), \end{aligned} \quad (3.39)$$

де $X(p)$ і $Y(p)$ - Лаплас-зображення відповідно вхідного процесу $x(t)$ і вихідного процесу $y(t)$.

З (39) випливає

$$Y(p) = \frac{b_m \cdot p^m + b_{m-1} \cdot p^{m-1} + \dots + b_1 p + b_0}{a_n \cdot p^n + a_{n-1} \cdot p^{n-1} + \dots + a_1 \cdot p + a_0} \cdot X(p) = W(p) \cdot X(p). \quad (3.40)$$

Величина

$$W(p) = \frac{Y(p)}{X(p)} = \frac{b_m \cdot p^m + b_{m-1} \cdot p^{m-1} + \dots + b_1 p + b_0}{a_n \cdot p^n + a_{n-1} \cdot p^{n-1} + \dots + a_1 \cdot p + a_0} \quad (3.41)$$

називається *передатною функцією* системи, що описується рівнянням (23).

Узагальнюючи, можна визначити: передатною функцією системи є відношення Лаплас-зображення вихідної величини до Лаплас-зображення вхідної за нульових початкових умов системи.

Це відношення не залежить ані від часу, ані від закону змінювання з часом вхідної величини лише для лінійних диференціальних рівнянь зі сталими коефіцієнтами. Тому апарат передатних функцій може бути застосований тільки до так званих лінійних стаціонарних систем (ЛСС), тобто таких, поводження яких можна описати звичайними лінійними диференціальними рівняннями зі сталими коефіцієнтами.

Як бачимо з (41), передатна функція є дробово-раціональною функцією комплексної змінної p . Знаменник передатної функції завжди є характеристичним поліномом системи. Корені характеристичного полінома системи (знаменника передатної функції) утворюють *полюси системи*, а корені чисельника - *нули передатної функції*.

Важливим наслідком поняття передатної функції $W(p)$ є поняття частотної передатної функції (ЧПФ) $W(j\omega)$, яка визначається як значення передатної функції за суто уявних значень комплексного аргументу:

$$W(j\omega) = W(p) \Big|_{p=j\omega} = \frac{b_m \cdot (j\omega)^m + b_{m-1} \cdot (j\omega)^{m-1} + \dots + b_1(j\omega) + b_0}{a_n \cdot (j\omega)^n + a_{n-1} \cdot (j\omega)^{n-1} + \dots + a_1 \cdot (j\omega) + a_0}. \quad (3.42)$$

Покажемо, що частотна передатна функція (ЧПФ) є вичерпним описувачем усталених вимушених коливань вихідної величини системи, якщо вхідна величина змінюється у часі за гармонічним законом.

Перш за все згадаємо, що гармонічною функцією часу називають таку, яка змінюється у часі t за синусоїдальним

$$x(t) = x_m \cdot \sin(\omega t + \mathcal{G}), \quad (3.43)$$

або косинусоїдальним

$$x(t) = x_m \cdot \cos(\omega t + \varepsilon) \quad (3.44)$$

законом. Тут через x_m позначено амплітуду гармонійних коливань, ω - колову частоту цих коливань, а \mathcal{G} і ε - початкові фази відповідно синусоїдальних і косинусоїдальних коливань.

Обидва ці закони мажуть бути зведеними і до більш загальної форми, яка не містить початкової фази

$$x(t) = x_s \cdot \sin \omega t + x_c \cdot \cos \omega t, \quad (3.45)$$

де x_s і x_c назвемо відповідно "амплітудами" синусоїдальної і косинусоїдальної частин гармонічного коливання. Нехай ця форма подання зветься *безфазною*.

Найбільш загальною і зручною формою подання гармонічного коливання є комплексна еспоненційна форма, яка ґрунтується на тій обставині, що комплексна функція

$$\begin{aligned}
 x(t) &= X_m \cdot e^{j\omega t} = \frac{x_m}{2} \cdot (\cos \varepsilon + j \cdot \sin \varepsilon) \cdot (\cos \omega t + j \sin \omega t) = \\
 &= \frac{x_m}{2} \cdot [\cos(\omega t + \varepsilon) + j \cdot \sin(\omega t + \varepsilon)] = \frac{x_m}{2} \cdot e^{j(\omega t + \varepsilon)}
 \end{aligned} \tag{3.46}$$

теж є гармонічною з тим же періодом $T = 2\pi / \omega$. Сталу комплексну величину

$$X_m = \frac{x_m}{2} \cdot e^{j\varepsilon} \tag{3.47}$$

називають комплексною амплітудою гармонічного коливання. Взаємозв'язок між дійсною і комплексною формами гармонічного коливання дається формулою Ейлера

$$\cos(\omega t + \varepsilon) = \frac{1}{2} (e^{j(\omega t + \varepsilon)} + e^{-j(\omega t + \varepsilon)}),$$

звідки випливає

$$x_m \cdot \cos(\omega t + \varepsilon) = \frac{x_m}{2} (e^{j(\omega t + \varepsilon)} + e^{-j(\omega t + \varepsilon)}) = X_m \cdot e^{j\omega t} + \bar{X}_m \cdot e^{-j\omega t}. \tag{3.48}$$

Тут \bar{X}_m - величина, комплексно спряжена щодо X_m .

Математика стверджує, що у випадку, коли члени диференційного рівняння, що утворюють його неоднорідність, змінюються з часом за гармонічним законом, то частинний розв'язок рівняння, що відповідає цій неоднорідності теж є гармонічною функцією часу з тим самим періодом. Тобто, якщо вхідна величина у рівнянні (23) змінюється за законом (46), то усталені вимушені коливання вихідної величини будуть змінюватися за аналогічним законом

$$y(t) = Y_m \cdot e^{j\omega t}. \tag{3.49}$$

Підставимо (49) і (46) у рівняння (23). Одержимо

$$Y_m \cdot e^{j\omega t} \cdot \{a_n(j\omega)^n + \dots + a_1(j\omega) + a_0\} = X_m \cdot e^{j\omega t} \cdot \{b_m(j\omega)^m + \dots + b_1(j\omega) + b_0\},$$

звідки випливає

$$\frac{Y_m}{X_m} = \frac{b_m(j\omega)^m + \dots + b_1(j\omega) + b_0}{a_n(j\omega)^n + \dots + a_1(j\omega) + a_0} = W(j\omega), \tag{3.50}$$

з чого можна зробити висновок:

значення ЧПФ за певного значення частоти ω зовнішньої дії дорівнює відношенню комплексних амплітуд вихідної і вхідної гармонічно змінюваних з цією частотою величин.

Тому, знаючи ЧПФ як функцію частоти, завжди можна визначити усі параметри вимушених коливань вихідної величини. Для цього зручно використати співвідношення (47) у (50):

$$\frac{Y_m}{X_m} = \frac{y_m \cdot e^{j\varphi}}{x_m \cdot e^{j\varepsilon}} = \frac{y_m}{x_m} \cdot e^{j(\varphi - \varepsilon)} = W(j\omega) = |W(j\omega)| \cdot e^{j \cdot \arg(W(j\omega))}, \tag{3.51}$$

з чого виходить
$$\frac{y_m}{x_m} = |W(j\omega)| \tag{3.52}$$

відношення амплітуди вихідного сигналу до амплітуди вхідного дорівнює модулю ЧПФ;

$$\varphi - \varepsilon = \arg(W(j\omega)) \quad (3.53)$$

зсув початкової фази вихідної величини по відношенню до початкової фази вхідної величини дорівнює аргументу комплексної ЧПФ.

Через це знання ЧПФ дає змогу за відомими параметрами вхідної дії визначити параметри вихідної величини, наприклад:

$$y_m = |W(j\omega)| \cdot x_m$$

щоб знайти амплітуду усталених вимушених коливань вихідної величини достатньо величину амплітуди вхідної величини домножити на модуль ЧПФ;

$$\varphi = \varepsilon + \arg(W(j\omega))$$

щоб знайти початкову фазу коливань вихідної величини потрібно до початкової фази вхідної величини додати аргумент ЧПФ на відомій частоті.

Крім того, з (51) випливають також співвідношення, які дозволяють визначити параметри "безфазного" подання коливань:

$$y_c = x_c \cdot U(\omega) + x_s \cdot V(\omega) \quad (3.54)$$

щоб знайти "амплітуду" косинусоїдальної частини гармонічної вихідної величини достатньо до добутку "амплітуди" косинусоїдальної частини вхідної величини на дійсну частину $U(\omega)$ ЧПФ додати добуток "амплітуди" синусоїдальної частини вхідної величини на уявну частину $V(\omega)$ ЧПФ;

$$y_s = x_c \cdot V(\omega) - x_s \cdot U(\omega) \quad (3.55)$$

щоб знайти "амплітуду" синусоїдальної частини гармонічної вихідної величини достатньо від добутку "амплітуди" косинусоїдальної частини вхідної величини на уявну частину ЧПФ відняти добуток "амплітуди" синусоїдальної частини вхідної величини на дійсну частину $V(\omega)$ ЧПФ.

Усе вищезазначене відноситься до так званих одновимірних систем, тобто систем з єдиним входом і єдиним виходом (див. рис. 3.2).

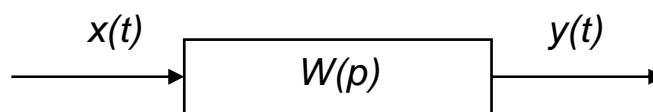
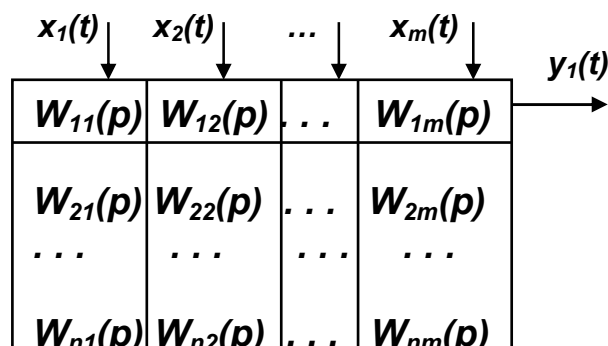


Рис. 3.2. Схема одновимірної системи

Але найбільш зручно і вигідно застосовувати апарат передатних функцій у випадку багатовимірних систем, які мають кілька входів і кілька виходів. Такі системи вже описуються не однією передатною функцією, а матрицею передатних функцій (див. рис. 3.3).



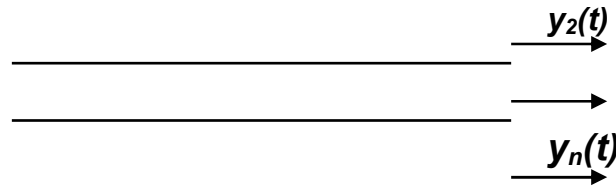


Рис. 3.3. Схема багатовимірної системи

Позначимо

$$W_{ik}(p) = \frac{Y_i(p)}{X_k(p)}. \quad (3.56)$$

Це є передатна функція системи з i -го виходу за k -тим входом.

Багатовимірні системи описуються кількома диференціальними рівняннями з відповідною кількістю вихідних величин.

Для прикладу, розглянемо систему диференціальних рівнянь, що описує рух гіроскопа у кардановому підвісі:

$$\begin{cases} A\ddot{\alpha} + H\dot{\beta} = N(t) \\ B\ddot{\beta} - H\dot{\alpha} = L(t) \end{cases} \quad (3.57)$$

Вважатимемо вхідними величинами моменти сил $N(t)$ і $L(t)$, що діють вдовж осей карданового підвісу. Як вихідні величини розглядатимемо кутові швидкості $\dot{\alpha}$ і $\dot{\beta}$ повороту гіроскопа. У цьому випадку маємо 2 входи і 2 виходи і матрицю 2-на-2 передатних функцій

$$W_{11}(p) = \frac{\dot{\alpha}(p)}{N(p)}; \quad W_{12}(p) = \frac{\dot{\alpha}(p)}{L(p)}; \quad W_{21}(p) = \frac{\dot{\beta}(p)}{N(p)}; \quad W_{22}(p) = \frac{\dot{\beta}(p)}{L(p)}. \quad (3.58)$$

Щоб відшукати конкретні вирази цих передатних функцій потрібно здійснити наступні операції:

- 1) здійснити перетворення Лапласа по відношенню до заданих диференціальних рівнянь; наприклад, застосовуючи його до системи рівнянь (57), одержимо

$$\begin{cases} A \cdot p \cdot \dot{\alpha}(p) + H \cdot \dot{\beta}(p) = N(p) \\ B \cdot p \cdot \dot{\beta}(p) - H \cdot \dot{\alpha}(p) = L(p) \end{cases}$$

- 2) одержану систему алгебричних рівнянь привести до матричної форми, об'єднуючи всі вихідні величини у єдиний вектор вихідних величин, наприклад

$$\begin{bmatrix} Ap & H \\ -H & Bp \end{bmatrix} \cdot \begin{bmatrix} \dot{\alpha}(p) \\ \dot{\beta}(p) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot N(p) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot L(p)$$

- 3) виділити квадратну матрицю, що є множником при векторі вихідних змінних

$$P(p) = \begin{bmatrix} Ap & H \\ -H & Bp \end{bmatrix}$$

ця матриця називається характеристичною матрицею системи;

4) обчислити визначник характеристичної матриці

$$\Delta(p) = \det(P(p)) = \begin{vmatrix} Ap & H \\ -H & Bp \end{vmatrix} = ABp^2 + H^2;$$

5) використовуючи метод Крамера, відшукати передатні функції системи, згідно формули $W_{ik}(p) = \frac{\Delta_{ik}(p)}{\Delta(p)}$,

де $\Delta_{ik}(p)$ - визначник матриці, яка одержується з характеристичної шляхом заміни i -го стовпця її на стовпець при k -тій вхідній величині; наприклад, для розглядуваної системи

$$\Delta_{11}(p) = \det \begin{bmatrix} 1 & H \\ 0 & Bp \end{bmatrix} = Bp; \quad \Delta_{12}(p) = \det \begin{bmatrix} 0 & H \\ 1 & Bp \end{bmatrix} = -H;$$

$$\Delta_{21}(p) = \det \begin{bmatrix} Ap & 1 \\ -H & 0 \end{bmatrix} = H; \quad \Delta_{22}(p) = \det \begin{bmatrix} Ap & 0 \\ -H & 1 \end{bmatrix} = Ap;$$

в результаті одержуються такі передатні функції гіроскопа у кардановому підвісі:

$$W_{11}(p) = \frac{Bp}{ABp^2 + H^2}; \quad W_{12}(p) = \frac{-H}{ABp^2 + H^2};$$

$$W_{21}(p) = \frac{H}{ABp^2 + H^2}; \quad W_{22}(p) = \frac{Ap}{ABp^2 + H^2}. \quad (3.59)$$

Знайшовши конкретні вирази для усіх передатних функцій багатовимірної системи, можна досить легко розв'язати задачу відшукання усталених вимушених коливань, якщо усі вхідні величини змінюються у часі за гармонічним законом з єдиною частотою. Для цього, як було попередньо зазначено, зручніше за все скористатися формулами (54) і (55):

$$y_{ic} = \sum_{k=1}^{s1} [x_{kc} \cdot U_{ik}(\omega) + x_{ks} \cdot V_{ik}(\omega)]; \quad y_{is} = \sum_{k=1}^{s1} [x_{kc} \cdot V_{ik}(\omega) - x_{ks} \cdot U_{ik}(\omega)]. \quad (3.60)$$

Тут i - номер вихідної величини, k - номер вхідної величини, $s1$ - кількість вхідних величин, що змінюються з часом за гармонічним законом з частотою ω , тобто припускається

$$x_k(t) = x_{ks} \cdot \sin \omega t + x_{kc} \cdot \cos \omega t,$$

$U_{ik}(\omega)$ - дійсна частина ЧПФ з i -того виходу по k -му входу, $V_{ik}(\omega)$ - уявна частина відповідної ЧПФ.

Як приклад розглянемо усталені коливання гіроскопа у кардановому підвісі з кутової швидкості $\dot{\alpha}(t)$ його обертання навколо зовнішньої осі підвісу при дії на нього гармонічних моментів сил вдовж зовнішньої осі

$$N(t) = N_s \cdot \sin \omega t + N_c \cdot \cos \omega t$$

і вдовж внутрішньої осі підвісу

$$L(t) = L_s \cdot \sin \omega t + L_c \cdot \cos \omega t.$$

Для цього спочатку відшукаємо відповідні ЧПФ на основі знання ПФ (59):

$$W_{11}(j\omega) = j \frac{B\omega}{H^2 - AB\omega^2}; \quad W_{12}(j\omega) = \frac{-H}{H^2 - AB\omega^2};$$

$$W_{21}(j\omega) = \frac{H}{H^2 - AB\omega^2}; \quad W_{22}(j\omega) = j \frac{A\omega}{H^2 - AB\omega^2}.$$

Як бачимо, діагональні ЧПФ є суто уявними, а решта - суто дійсними. Враховуючи знайдені ЧПФ і користуючись співвідношеннями (60), матимемо:

$$\dot{\alpha}(t) = \frac{1}{H^2 - AB\omega^2} \left[(B\omega N_c + HL_s) \cdot \sin \omega t + (B\omega N_s - HL_c) \cdot \cos \omega t \right].$$

Використання передатних і частотних передатних функцій в інженерній практиці обумовлено наступними міркуваннями:

- 1) вичерпністю ПФ і ЧПФ для опису поведінки системи за будь-яких збурень;
- 2) простотою і зручністю відшукування ПФ навіть для складних систем;
- 3) прозорістю фізичного змісту ЧПФ і окремих її параметрів;
- 4) можливістю відшукування ПФ за схемою з'єднання окремих ланок системи, що описуються заданими їх ПФ.

Зупинемося докладніше на останньому.

Нехай маємо дві одновимірні ланки, що описуються передатними функціями $W_1(p)$ і $W_2(p)$. Позначимо їхні входи відповідно через x_1 і x_2 , а виходи - y_1 і y_2 (див. рис. 3.4).

Послідовним з'єднанням двох ланок називають таке, коли вихід першої ланки становиться входом другої (рис. 3.5). В результаті одержується нова, більш складна ланка, входом якої є вхід першої ланки, а виходом - вихід другої.

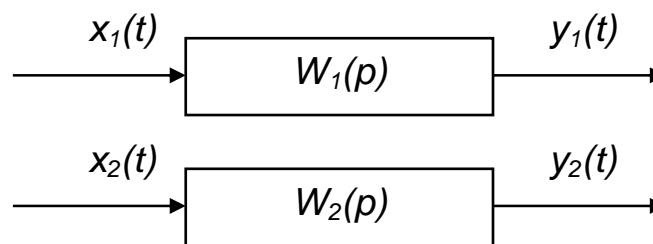


Рис. 3.4. Схема ланок

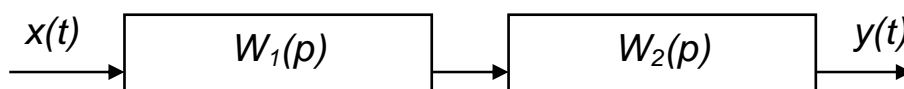


Рис. 3.5. Послідовне з'єднання ланок

Передатна функція нової ланки, як неважко впевнитися, дорівнює добутку передатних функцій складових ланок

$$W(p) = W_1(p) \cdot W_2(p). \quad (3.61)$$

Паралельним з'єднанням двох ланок називають таке, коли вихід у обох ланок є спільним, а виходи підсумовуються (рис. 3.6).

Передатна функція паралельного з'єднання дорівнює сумі передатних функцій складових ланок

$$W(p) = W_1(p) + W_2(p). \quad (3.62)$$

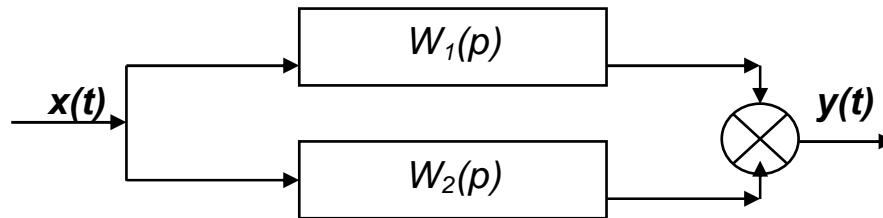


Рис. 3.6. Паралельне з'єднання ланок

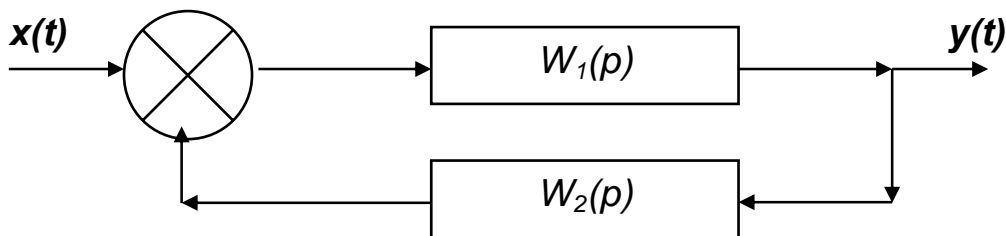


Рис. 3.7. Зворотний від'ємний зв'язок

З'єднання типу зворотного від'ємного зв'язку здійснюється так, як подано на рис. 3.7. При цьому перша ланка утворює прямий ланцюг, а друга - ланцюг зворотного зв'язку. Вхід першої ланки утворює різниця між загальним входом результуючої ланки і виходом другої ланки. Вихід першої ланки є одночасно виходом результуючої ланки і входом другої.

Передатна функція такого з'єднання визначається формулою

$$W(p) = \frac{W_1(p)}{1 + W_1(p)W_2(p)}. \quad (3.63)$$

У більшості випадків інженери складають системи автоматичного керування саме шляхом з'єднання між собою окремих пристроїв одним з поданих способів. Для дослідження утворюваних при цьому систем нема потреби складати диференційні рівняння результуючої системи. Достатньо відшукати її передатну функцію за відомими ПФ складових ланок, користуючись викладеними правилами.

3. 6. Обчислення значень частотних передатних функцій

Головними способами дослідження лінійних стаціонарних систем (ЛСС) за їхніми ЧПФ є наступні:

- обчислення значень полюсів і нулів ЧПФ;

- побудова графіків модуля ЧПФ при змінюванні частоти ω у широкому діапазоні; через те, що модуль ЧПФ у відповідності з (52) дорівнює відношенню амплітуд вихода і входу, графік залежності модуля ЧПФ від частоти називають амплітудно-частотною характеристикою (АЧХ) системи;
- побудова графіка залежності аргумента ЧПФ від частоти (у відповідності з (53) він називається фазо-частотною характеристикою (ФЧХ) системи).

В усіх випадках при обчисленнях необхідно мати справу з комплексними числами. Тому використання апарату ЧПФ необхідно пов'язане з діями над комплексними числами.

Такі дії безпосередньо можливі лише при користуванні мовою Fortran. При застосуванні мови Pascal потрібно або самостійно утворити бібліотеку (Units) процедур і функцій операцій з комплексними числами, або придбати існуючу бібліотеку цих процедур.

Більшості проблем цього плану можна позбутися, якщо використовувати систему MatLAB, в якій передбачені вбудовані дії з комплексними числами, операції з поліномами (див. п. 3.1). Особливо зручним є наявність функції *roots* обчислення коренів полінома, яка дозволяє одразу визначати нулі і полюси передатної функції.

Нехай нам потрібно визначити нулі і полюси передатної функції

$$W(p) = \frac{0,7p^2 + p}{4p^4 + 0,1p^3 + 7p^2 + p + 125}$$

Тоді у середовищі MatLAB достатньо здійснити такі дії:

1) ввести вектор Ch коефіцієнтів полінома чисельника ПФ:

» **Ch=[0.7 1 0]**

Ch =

0.7 1 0

2) ввести вектор Zn коефіцієнтів полінома знаменника ПФ:

» **Zn=[4 0.1 7 1 125]**

Zn =

4 0.1 7 1 125

3) визначити полюси ПФ, застосовуючи процедуру *roots* до вектора її знаменника

» **Poles=roots(Zn)**

Poles =

1.5292 + 1.8082i

1.5292 - 1.8082i

-1.5417 + 1.7876i

-1.5417 - 1.7876i

як бачимо, система є коливальною і нестійкою;

4) застосувати ту ж процедуру до вектора чисельника і цим відшукати значення нулів системи

» **Noles=roots(Ch)**

Noles =

0

-1.4286

Покажемо як будується АЧХ і ФЧХ на прикладі ПФ виду

$$W(p) = \frac{p^2 + 100}{p^4 + 2,2p^3 + 1617,4p^2 + 352,22p + 25632}$$

Для цього введемо вектор коефіцієнтів полінома чисельника

```
» ch=[1 0 100]
ch = 1 0 100
```

і вектор полінома знаменника

```
» zn=[ 1 2.2 1617.4 352.22 25632]
zn = 1 2.2 1617.4 352.22 25632
```

Задля контролю знайдемо нулі

```
» roots(ch)
ans =
    0 + 10i
    0 - 10i
```

і полюси системи

```
» roots(zn)
ans =
   -1 + 40i
   -1 - 40i
  -0.1 + 4i
  -0.1 - 4i
```

Таким чином, розглядувана система є коливною, стійкою, з частотами власних коливань 4 і 40 рад/с.

Щоб побудувати АЧХ і ФЧХ, спочатку утворимо вектор значень частот

```
» om=0:0.2:100;
```

потім вектор суто уявних відповідних значень комплексного аргументу p

```
» p=j*om;
```

Далі розраховуються комплексні значення чисельника

```
» chis=polyval(ch,p);
```

і знаменника ЧПФ

```
» znam=polyval(zn,p);
```

Тепер можна обчислити модуль ЧПФ

```
» ACH=sqrt(abs(chis))./sqrt(abs(znam));
```

і побудувати графік АЧХ

```
» plot(om,ACH)
» grid
» title('Амплітудно - частотна характеристика')
» ylabel('Відношення амплітуди вихода до амплітуди входу')
» xlabel('Частота (рад/с)')
```

Результат поданий на рис. 3.8.

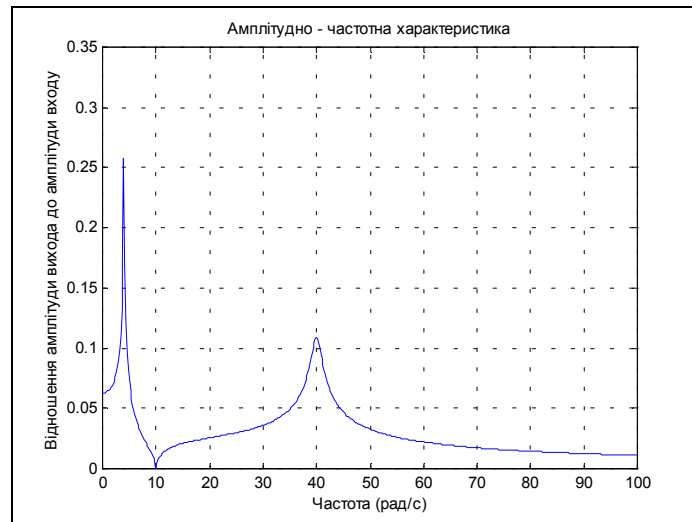


Рис. 3.8. АЧХ, побудована засобами MatLAB

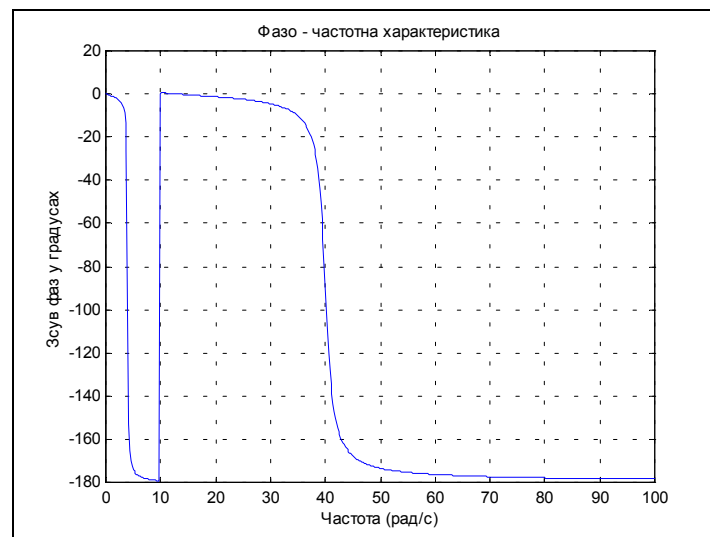


Рис. 3.9. ФЧХ, побудована засобами MatLAB

Нарешті розрахуємо значення ФЧХ

» `FCH=angle(chis./znam)*180/pi;`

і будуємо її графік

» `plot(om,FCH)`

» `grid`

» `title('Фазо - частотна характеристика')`

» `xlabel('Частота (рад/с)')`

» `ylabel('Зсув фаз у градусах')`

В результаті одержуємо графік, поданий на рис. 3.9.

Як бачимо, система MatLAB містить усі необхідні засоби дослідження лінійних стаціонарних систем за їхніми АЧХ. Окрім розглянутих функцій в ній передбачено спеціальний пакет функцій і процедур CONTROL аналізу і синтезу лінійних систем.

3.6. Завдання

Завдання 1. Користуючись середовищем MATLAB як калькулятором, обчисліть у ньому зазначений арифметичний вираз. Укажіть послідовність натискання клавіш. Порівняйте отриманий результат із наведеною відповіддю.

	Відповідь
1. $\frac{\left(12\frac{1}{6} - 6\frac{1}{27} - 5,25\right)13,5 + 0,111}{0,02}$	599,3
2. $\frac{\left(1\frac{1}{12} + 2\frac{5}{32} + \frac{1}{24}\right) : 9,6 + 2,13}{0,0004}$	179,5
3. $\frac{\left(6,6 - 3\frac{3}{14}\right)5\frac{5}{6}}{(21 - 1,25) : 2,5}$	2,5
4. $\frac{2,625 - \frac{2}{3} \cdot 2\frac{5}{14}}{\left(3\frac{1}{12} + 4,375\right) : 19\frac{8}{9}}$	2,8095
5. $\frac{0,134 + 0,05}{18\frac{1}{6} - 1\frac{11}{14} - \frac{2}{15} \cdot 2\frac{6}{7}}$	0,00115
6. $\frac{\left(58\frac{4}{15} - 56\frac{7}{24}\right) : 0,8 + 2\frac{1}{9} \cdot 0,225}{8,75 \cdot 0,6}$	0,56071
7. $\frac{\left(\frac{0,216}{0,15} + 0,56\right) : 0,5}{\left(7,7 : 24,75 + \frac{2}{15}\right)4,5}$	2
8. $\frac{1\frac{4}{11} \cdot 0,22 : 0,3 - 0,96}{\left(0,2 - \frac{3}{40}\right)1,6}$	-4,35
9. $\frac{\left(\frac{3}{5} + 0,425 - 0,005\right) : 0,12}{30,5 + \frac{1}{6} + 3\frac{1}{3}}$	0,25

10. $\frac{3\frac{1}{3} + 2,5}{2,5 - 1\frac{1}{3}} \cdot \frac{4,6 - 2\frac{1}{3}}{4,6 + 2\frac{1}{3}} : \left(\frac{0,05}{\frac{1}{7} - 0,125} + 5,7 \right)$. 0,19231
11. $\frac{0,725 + 0,42(6)}{0,128 - 6,25 - (0,0345 : 0,12)} \cdot 0,25$. -0,04492
12. $\frac{\left(4,5 \cdot 1\frac{2}{3} - 6,75 \right) \cdot 0,(6)}{\left(3,333 \cdot 0,3 + 0,222 \cdot \frac{4}{9} \right) 2\frac{2}{3}}$. 0,17068
13. $\frac{\left(5\frac{4}{45} - 4\frac{1}{6} \right) : 5\frac{8}{15}}{\left(4\frac{2}{3} + 0,75 \right) 3\frac{9}{13}} \cdot 34\frac{2}{7}$. 0,28571
14. $\frac{1\frac{4}{11} \cdot 0,22 : 0,3 - 0,96}{\left(0,2 - \frac{3}{40} \right) 1,68}$. 0,19048
15. $\frac{\left(40\frac{7}{30} - 38\frac{5}{12} \right) : 10,9 + \left(0,875 - \frac{7}{30} \right) \cdot \frac{20}{11}}{0,008}$. 166,67
16. $\frac{(68,023 - 66,028) : 6\frac{1}{9} + \frac{7}{40} \cdot 4,5}{0,042 + 0,086}$. 8,7028
17. $\frac{(2,1 - 1,965) : (1,2 \cdot 0,045)}{0,00325 : 0,013} - \frac{4}{0,2 \cdot 0,73}$. -17,397
18. $\frac{(1,88 + 2,127) \cdot 0,01875}{0,625 - \frac{13}{18} : 3,13} + 8,29$. 8,2441
19. $\frac{3 : 0,4 - 0,009 : (0,15 : 2,5)}{0,32 \cdot 6 + 0,033 - (5,3 - 3,88)}$. 13,79
20. $\frac{(34,06 - 33,81) \cdot 4}{6,84 : (28,57 - 25,15)} + 1,33 : \frac{4}{21}$. 7,4825
21. $\frac{8,8077}{20 - (28,2 : (13,333 \cdot 0,3 + 0,0125)) 2,004}$. 1,4889

22. $\frac{\left(1,75 : \frac{2}{3} - 1,75 \cdot 1,125\right) : \frac{7}{12}}{(0,2012 - 0,0325) : 400}$. 2667,5
23. $\frac{\left(26\frac{1}{3} - 18,02 \cdot 0,75\right) \cdot 2,4 : 0,88}{1,37 - 23\frac{2}{3} : 1,82}$. -3,005
24. $26 : \frac{3 : (0,48 - 0,27)}{2,52(1,38 + 2,45)} + 1,27$. 8,836
25. $\left(16,5 - 13\frac{7}{9}\right)\frac{6}{11} + 2,2 : (0,241 - 0,91)$. -1,8036

Завдання 2. Проведіть обчислення по заданій формулі при заданих значеннях параметрів. Зазначте необхідну послідовність дій. Порівняйте отриманий результат із наведеною відповіддю.

Вказівка. У системі *MatLAB* декілька останніх команд запам'ятовуються. Виклик цих команд у командне вікно здійснюється натисканням клавіш $\langle \downarrow \rangle$ і $\langle \uparrow \rangle$. Використовуйте цю можливість для повторного звернення до набраної функції.

1. $3m^2 + \sqrt[3]{2n^2} : m$; а) $m = -\frac{14}{5}$, $n = \operatorname{tg} \frac{\pi}{8}$; б) $m = 2,2 \cdot 10^{-2}$, $n = \frac{1}{3,1}$.

ВІДПОВІДЬ: а) 23,27; б) 26,938.

2. $\frac{4}{3}l^3 \sin^2 \frac{\alpha}{2} \sqrt{\cos \alpha}$; а) $l = 1,7 \cdot 10^3$, $\alpha = 18^\circ$; б) $l = \frac{16}{21}$, $\alpha = \frac{\pi}{5}$.

ВІДПОВІДЬ: а) 1. 5633e+008; б) 5. 0651e-002.

3. $\sqrt{\frac{a\sqrt{b}}{\sqrt[3]{\operatorname{tg} \alpha}}}$; а) $a = 1,5$, $b = 0,8$, $\alpha = 61^\circ$; б) $a = 3 \cdot 10^{-2}$, $b = 0,71$, $\alpha = \frac{3}{7}\pi$.

ВІДПОВІДЬ: а) 1. 0498e+000; б) 1. 2429e-001.

4. $\frac{3a^2 \sqrt{6,8 \cdot (a-b)}}{4(a+b)^3}$; а) $a = 4,13 \cdot 10^{-1}$, $b = \frac{1}{261}$;

б) $a = \sin \frac{5\pi}{8}$, $b = -\operatorname{tg} 12^\circ$

ВІДПОВІДЬ: а) 2. 9464e+000; б) 4. 9445e+000.

5. $\frac{c^3}{6} \cos \frac{\alpha}{2} \sqrt{\sin \alpha}$; а) $c = \lg 2,38$, $\alpha = \frac{\pi}{5}$; б) $c = e^{-0,3}$, $\alpha = 65^\circ$.

ВІДПОВІДЬ: а) 3. 4657e-004; б) 2. 2120e-002.

6. $\sqrt{\frac{n^3}{16,3 \sin \alpha \sin 2\alpha}}$; а) $n = 3,1516 \cdot 10^{-2}$, $\alpha = 5^\circ$; б) $n = e^{3,5}$, $\alpha = \frac{2\pi}{13}$.

ВІДПОВІДЬ: а) 1. 1265e-002; б) 7. 6324e+001.

$$7. \quad 5 \sin 35^\circ \sqrt{\frac{S^3 \cos 36^\circ}{\pi^3 \operatorname{tg} \alpha}}; \text{ а) } S = \ln 3, \alpha = 44^\circ; \text{ б) } S = \frac{18}{25}, \alpha = \frac{7}{12} \pi.$$

ВІДПОВІДЬ: а) 5. 4283e-001; б) 8. 9703e-018+ 1. 4650e-001i.

$$8. \quad |\lg(1 + \sin \alpha) + \ln(1 - \sin \beta)|; \text{ а) } \alpha = \frac{3\pi}{7}, \beta = 83^\circ; \text{ б) } \alpha = \frac{2}{3} \pi, \beta = 16^\circ.$$

ВІДПОВІДЬ: а) 4. 6035e+000; б) 5. 1546e-002.

$$9. \quad \sqrt[3]{\sin^2(\alpha + \beta) - \sin^2(\alpha - \beta)}; \text{ а) } \alpha = \frac{5}{7} \pi, \beta = 0,3\pi; \text{ б) } \alpha = 12^\circ, \beta = 220^\circ$$

ВІДПОВІДЬ: а) 4. 8756e-001+ 8. 4448e-001i; б) 7. 3715e-001.

$$10. \quad (\log_a(b + 1,4))^{-3/4}; \text{ а) } a = 3,56, b = e^{0,316}; \text{ б) } a = 2, b = 2,1649 \cdot 10^{-2}.$$

ВІДПОВІДЬ: а) 1. 1790e+000; б) 1. 6630e+000.

$$11. \quad 3 \left(p^{-2/3} + q^{-1/2} \right) \sqrt[3]{pq}; \text{ а) } p = \ln 3, q = \lg 3; \text{ б) } p = 0,013, q = 1,4 \cdot 10^2.$$

ВІДПОВІДЬ: а) 5. 7737e+000; б) 6. 6559e+001.

$$12. \quad \frac{2}{3} m \sqrt{m^3 \sqrt{m^4 m}}; \text{ а) } m = 3,6485 \cdot 10^2; \text{ б) } m = \frac{24}{37}.$$

ВІДПОВІДЬ: а) 1. 5880e+004; б) 5. 4516e-001.

$$13. \quad \frac{8}{3} S \sqrt{\frac{S}{\pi}} \sin^6 \frac{\alpha}{2}; \text{ а) } S = e^{1,11}, \alpha = \frac{7}{11} \pi; \text{ б) } S = 5,403, \alpha = 28^\circ.$$

ВІДПОВІДЬ: а) 2. 8187e+000; б) 3. 7879e-003.

$$14. \quad 2 \sqrt{\frac{F}{\pi}} \operatorname{tg} \alpha \sin^2 \frac{\alpha}{2}; \text{ а) } F = \frac{1}{0,03}, \alpha = \frac{5}{7} \pi; \text{ б) } F = \ln 7, \alpha = 1,34^\circ.$$

ВІДПОВІДЬ: а) -6. 6313e+000; б) 5. 0346e-006.

$$15. \quad \frac{1}{12} \cdot \frac{m^3 \cos \alpha}{(\sin \alpha + \cos \alpha)^3}; \text{ а) } m = -20,1, \alpha = 20^\circ; \text{ б) } m = \lg 13,6, \alpha = 1,48.$$

ВІДПОВІДЬ: а) -3. 0201e+002; б) 8. 5792e-003.

$$16. \quad \frac{\sqrt{3h^3}}{\cos^2 \alpha} \sin(\alpha + 30^\circ) \sin(\alpha - 30^\circ);$$

$$\text{ а) } h = 0,28, \alpha = 41^\circ; \text{ б) } h = e^{0,415}, \alpha = 237^\circ.$$

ВІДПОВІДЬ: а) 8. 1284e-002; б) 4. 9334e+000.

$$17. \quad \frac{\alpha}{3} (\lg(d + 2) - \operatorname{tg} \alpha)^2;$$

$$\text{ а) } d = 6,178, \alpha = 20^\circ; \text{ б) } d = -2,2461 \cdot 10^{-2}, \alpha = 1,146.$$

ВІДПОВІДЬ: а) 3. 5028e-002; б) 1. 4003e+000.

$$18. \quad d^3 \operatorname{ctg} \alpha \sqrt{\sin^4 \alpha - \cos^4 \alpha}; \text{ а) } d = 10,6, \alpha = 50^\circ; \text{ б) } d = e^{2,3}, \alpha = 1.$$

ВІДПОВІДЬ: а) 4. 1645e+002; б) 4. 1101e+002.

$$19. \frac{a^2 \sqrt{3}}{4} (\sec \alpha + \operatorname{cosec} \alpha)^4; \text{ а) } a = 5,08, \alpha = 25^\circ; \text{ б) } a = \ln 1,37, \alpha = \frac{12}{25} \pi$$

ВІДПОВІДЬ: а) 1. 6193e+003; б) 3. 5238e+003.

$$20. \frac{\sqrt{\pi}}{3} \cdot \frac{1}{(\operatorname{ctg} A + \operatorname{ctg} B)^2}; \text{ а) } A = 51^\circ, B = 39^\circ; \text{ б) } A = 0,643, B = \frac{\pi}{7}.$$

ВІДПОВІДЬ: а) 1. 4132e-001; б) 5. 0772e-002.

$$21. \lg \left(3^{x^2 - x - 9} + \frac{8}{27} \right); \text{ а) } x = e^{1,648}; \text{ б) } x = \operatorname{tg} 1,21.$$

ВІДПОВІДЬ: а) 6. 1109e+000; б) -5. 1927e-001.

$$22. \frac{\sqrt[5]{5e^{4a}(a+12,36)^2}}{\ln(a+7)}; \text{ а) } a = 2,1754 \cdot 10^2; \text{ б) } a = \cos 17^\circ.$$

ВІДПОВІДЬ: а) 8. 5511e+075; б) 4. 0272e+000;

$$23. \lg^2 x - \left(\frac{27}{8} \right)^{x-1} \sin \sqrt{x}; \text{ а) } x = e^{2,145}; \text{ б) } x = 2,468 \cdot 10^{-1}.$$

ВІДПОВІДЬ: а) -2. 0936e+003; б) 1. 7858e-001.

$$24. \sqrt[5]{(x-y)^2} \sqrt[3]{\frac{1}{y-x}}; \text{ а) } x = e^{-0,37}, y = \ln 2,1517; \text{ б) } x = 37^\circ, y = \cos \frac{7}{24} \pi.$$

ВІДПОВІДЬ: а) 3. 4445e-001; б) 2. 6745e-001.

$$25. \frac{\sin A + \operatorname{tg} B}{\sqrt[5]{(A-3B)^2}}; \text{ а) } A = 5,6, B = \lg 25; \text{ б) } A = \frac{8}{9} \pi, B = \frac{\pi}{10}.$$

ВІДПОВІДЬ: а) 4. 4466e+000; б) 5. 2145e-001.

Завдання 3. Виконайте такі дії (див. таблицю 3.1):

а) число z_1 , задане в алгебричній (експоненціальній) формі, перекладіть в експоненціальну (алгебричну) форму і запишіть результат;

б) число z_2 , задане в експоненціальній (алгебричній) формі, перекладіть в алгебричну (експоненціальну) форму і запишіть результат;

в) обчисліть заданий вираз; запишіть результат в алгебричній і експоненціальній формах.

Таблиця 3.1

Варіант	Комплексне число				Вираз
	z_1	z_2	z_3	z_4	
1	$4 + 3i$	$2,71e^{i\pi/12}$	$1,82e^{-1,2i}$	$\sqrt{3} - 2i$	$z_1^2 \cdot z_2 : z_3 + z_4$
2	$0,8 - 2i$	$3,08 e^{i\pi/12}$	$8,01e^{2i}$	$-5 + \sqrt{2}i$	$z_1^2 : z_2 + z_3 - z_4$
3	$-0,7 + 4i$	$1,74e^{i0,3\pi}$	$3 + 4i$	$2,1e^{-2,3i}$	$\sqrt{z_1 : z_2} \cdot z_3 + z_4$

4	$-3-2i$	$3,21e^{15^\circ i}$	$1,2+3i$	$2,71e^{-78^\circ i}$	$\sqrt{z_1 \cdot z_2} : z_3 + z_4$
5	$2,71e^{i\pi/12}$	$-0,7+4i$	$1,31e^{-i5\pi/12}$	$-8-3i$	$\sqrt{z_1 : z_2} \cdot z_3 - z_4$
6	$3,08e^{i7\pi/12}$	$-3-2i$	$2,3e^{i\pi/3}$	$\sqrt{3} + \sqrt{2}i$	$(z_1 + z_2)^4 \cdot z_3 : z_4$
7	$1,74e^{0,3\pi i}$	$0,8-2i$	$3,28e^{-1,2i}$	$\sqrt{3} - \sqrt{2}i$	$(\sqrt{z_1 + z_2}) \cdot z_3 : z_4$
8	$3,21e^{15i}$	$4+3i$	$\sqrt{3} - 4i$	$1,23e^{111^\circ i}$	$(z_1 - z_2)^3 \cdot z_3 + z_4$
9	$1+i\pi/2$	$1,2e^{107^\circ i}$	$0,8-2i$	$2,5e^{14^\circ i}$	$(z_1 : z_2 + z_3)^3 \cdot z_4$
10	$\sqrt{5} - i$	$0,7e^{1,7i}$	$1,2e^{0,9i}$	$-3-2i$	$(z_1 : z_2 + z_3)^2 - z_4$
11	$0,187-3,94i$	$0,3e^{-107^\circ i}$	$-0,7+4i$	$1,5e^{23^\circ i}$	$\sqrt[3]{z_1 + z_2 - z_3} \cdot z_4$
12	$-1+\sqrt{5}i$	$2,1e^{211^\circ i}$	$0,4e^{32^\circ i}$	$4+3i$	$\sqrt[3]{z_1 \cdot z_2} : z_3 + z_4$
13	$-\sqrt{3} - 4i$	$1,25e^{-0,8i}$	$-3-2i$	$0,75e^{0,7i}$	$(\sqrt[3]{z_1 \cdot z_2} + z_3) : z_4$
14	$1,2e^{1,7i}$	$0,18-3,9i$	$0,71e^{4i}$	$0,8-2i$	$(\sqrt[3]{z_1 : z_2} + z_3) \cdot z_4$
15	$0,3e^{-97^\circ i}$	$-1+\sqrt{5}i$	$-0,7+4i$	$5,2e^{71^\circ i}$	$(\sqrt{z_1 \cdot z_2} - z_3) : z_4$
16	$1,25e^{0,6i}$	$-\sqrt{3} - 4i$	$4+3i$	$2,5e^{3,8i}$	$(\sqrt{z_1 : z_2} - z_3) \cdot z_4$
17	$1,05e^{-0,4i}$	$\sqrt{5} - i$	$2,7e^{0,8i}$	$-0,7+4i$	$\sqrt{(z_1 : z_2 + z_3) \cdot z_4}$
18	$2,1e^{73^\circ i}$	$1+i\pi/2$	$\sqrt{2} + \sqrt{3}i$	$1,93e^{192^\circ i}$	$\sqrt{(z_1 \cdot z_2 - z_3) : z_4}$
19	$2,7+0,8i$	$2e^{-\sqrt{3}i}$	$0,81e^{i\pi/7}$	$-\sqrt{2} - \sqrt{3}i$	$\sqrt{(z_1 + z_2) : z_3 \cdot z_4}$
20	$-0,8+2,7i$	$-2e^{\sqrt{3}i}$	$0,9e^{i5\pi/7}$	$3,1-2,1i$	$\sqrt{(z_1 + z_2) \cdot z_3 : z_4}$
21	$-1,1-3,2i$	$0,33e^{-1,9i}$	$2e^{\sqrt{2}i}$	$2,08+i$	$\sqrt{z_1 - z_2} \cdot z_3 : z_4$
22	$2,1-3,2i$	$0,68e^{148^\circ i}$	$\sqrt{5} + \sqrt{2}i$	$2,73e^{23^\circ i}$	$\sqrt{z_1 - z_2} : z_3 \cdot z_4$
23	$1,1e^{-0,8i}$	$\sqrt{5} - 2i$	$-1,7+i$	$0,97e^{\sqrt{2}i}$	$((z_1 + z_2)^2 - z_3) : z_4$
24	$2,1e^{0,8i}$	$-\sqrt{5} + 2i$	$1,7e^{\sqrt{3}i}$	$0,8e^{2,5i}$	$((z_1 - z_2)^2 + z_3) : z_4$
25	$1,1e^{-2,1i}$	$\pi/8 - 2,1i$	$2,71+0,4i$	$1,71e^{-\sqrt{3}i}$	$(z_1 - z_2)^3 : z_3 + z_4$

Завдання 4. Відшукайте корені квадратного рівняння

$$a \cdot x^2 + b \cdot x + c = 0$$

при заданих значеннях коефіцієнтів a , b і c (табл. 3.2) за точними формулами і користуючись функцією **roots**. Порівняйте результати.

Таблиця 3.2

Варіант	a	b	c
1	0.56	1. 2e-4	4.08
2	1	0.1	100
3	4. 2e-3	8. 03e-4	1.06
4	7. 1e3	9. 4e4	8. 3e10
5	5.09	4.32	256
6	8.3	5.34	693
7	27	27	1276
8	3.08	0.2	30
9	5.3	10.6	876
10	0.45	0. 034	121
11	4.3	10.7	3. 4e3
12	13	0.8	287
13	6. 035	5.2	875
14	2.3	7.9	324
15	1	0.02	16.57
16	1.3	0.56	18.8
17	0.13	0. 056	18.8
18	17	12	956
19	0. 085	1	1. 3e3
20	1.2	0.32	15
21	7.1	6.4	256
22	0.2	0. 002	2.9
23	1. 4e-3	3.9	2. 6e2
24	0.86	3.2	5. 4e2
25	7. 3e3	8. 2e2	3. 5e8

Завдання 5. Обчисліть значення функції $f(x)$ на відрізку $[a;b]$ із кроком h у відповідності з табл. 3.3.

Таблиця 3.3

Варіант	$f(x)$	a	b	h
1	$\frac{x^2}{1+0,25\sqrt{x}}$	1,1	3,1	0,02
2	$\frac{x^3 - 0,3x}{\sqrt{1+2x}}$	2,05	3,05	0,01
3	$\frac{2e^{-x}}{2\pi + x^3}$	0	1,6	0,016

4	$\frac{\cos \pi x^2}{\sqrt{1-3x}}$	-1	0	0,01
5	$\sqrt{1+4x} \sin \pi x$	0,1	0,8	0,002
6	$\frac{e^{x/3}}{1+x^2}$	1,4	2,4	0,01
7	$e^{-2x} + x^2 - 1$	0,25	2,25	0,02
8	$(e+x) \sin(\pi \sqrt{x-1})$	1,8	2,8	0,01
9	$\sqrt{3+2x} \cdot \operatorname{tg} \frac{\pi x^3}{2}$	0,1	0,9	0,08
10	$\sqrt{2+3x} \cdot \ln(1+3x^2)$	-0,1	0,9	0,01
11	$\sqrt[3]{x^2+3} \cdot \cos \frac{\pi x}{2}$	1	2,5	0,015
12	$(4+7x) \sin(\pi^3 \sqrt{1+x})$	0	7	0,07
13	$e^{-x^2} (1+3x-x^2)$	0	2	0,02
14	$x^3 - 3x + \frac{8}{\sqrt{1+x^2}}$	0	1,7	0,17
15	$\sqrt{\operatorname{sh} \sqrt{2\pi x}}, \left(\operatorname{sh} x = \frac{e^x - e^{-x}}{2} \right)$	0	1,2	0,12
16	$\sqrt{\operatorname{ch} \frac{x}{\sqrt{2\pi}}}, \left(\operatorname{ch} x = \frac{e^x + e^{-x}}{2} \right)$	0,5	1,5	0,01
17	$\frac{x^3 + 2x}{\sqrt{1+e^x}}$	-0,2	0,8	0,01
18	$\sqrt{1+2x^2} \cdot \sin \frac{3x}{2}$	2	4	0,2
19	$\sqrt{3x^2+5} \cdot \cos \frac{\pi x}{2}$	0,5	1,5	0,1
20	$\arccos e^{-\sqrt[3]{3x}}$	0,2	0,5	0,03
21	$\arcsin e^{-x^2/5}$	8	13	0,05
22	$x + \ln(x + \sqrt{1+x^2})$	-0,5	0,5	0,01
23	$\frac{1+e^{-x/2}}{\sqrt{3x^2+1}}$	3	5	0,02

24	$3x^3 + \frac{1}{x} + e^{-2x^2}$	1,2	2,2	0,01
25	$x^{2,1} + x^3 - 2x$	1	5	0,04

Побудуйте в графічному вікні MatLAB графік функції. Роздрукуйте цей графік на аркуші паперу.

Завдання 6. Обчисліть вектори:

- модуля частотної передатної функції (ЧПФ);
- аргументу ЧПФ;
- дійсної частини ЧПФ;
- уявної частини ЧПФ

по заданих чисельнику і знаменнику передатної функції (табл. 3.4).

Попередньо відшукайте корені знаменника передатної функції, визначити найбільшу власну частоту ω_{\max} системи. Забезпечте обчислення ЧПФ при 100 значеннях частоти ω у діапазоні від 0 до $5\omega_{\max}$.

Побудуйте в одному графічному вікні MatLAB графіки амплітудно-частотної (модуля ЧПФ) і фазо-частотної (аргументу ЧПФ) характеристик (значення аргументу відкладати у градусах).

Таблиця 3.4

Варіант	Чисельник	Знаменник
1	$1.82p+67.56$	$p^4+2.65p^3+3.09p^2+7.04p+34.05$
2	$4.61p^2+1.82p+67.56$	$p^4+3.65p^3+45p^2+7.04p+125$
3	$p^2+4p+23$	$p^4+2p^3+39p^2+2p+45$
4	$3p^2+1.82p+67.56$	$p^2+7.04p+34.05$
5	$p+6$	$p^2+0.7p+48$
6	$p^3+4.61p^2+1.82p$	$2.65p^3+3p^2+4p+87$
7	$p^3+4.61p^2+1.82p+67.56$	$p^4+2.65p^3+68p^2+5p+34$
8	$4.61p^2+68$	$p^4+2.65p^3+3.09p^2+7.04p+34.05$
9	7.56	$p^4+2.65p^3+3.09p^2+7.04p+34.05$
10	$p^3+1.8p+7$	$p^4+6.5p^3+39p^2+7p+45$
11	$p^3+4.61p^2+1.82p+67.56$	$p^3+3.09p^2+70p+34$
12	$p^2+1.8p+78$	$2.65p^3+3.09p^2+7.04p+34.05$
13	$p^3+1.82p+67.56$	$p^4+2.6p^3+3p^2+4p+34$
14	$p^3+4.61p^2+1.82p+67.56$	$7p^2+7p+34$
15	$4.61p^2+1.82p+67.56$	$p^2+7.04p+560$
16	$1.82p+67.56$	$3.09p^2+7.04p+34.05$
17	p^3	$3.09p^2+7.8p+125$
18	$1.82p$	$p^3+3.09p^2+7.04p+34.05$
19	$4.61p^2$	$p^2+7.04p+34.05$
20	$p^3+67.56$	$p^4+2.65p^3+3.09p^2+7.04p+34.05$
21	p^3	$p^4+2p^3+3p^2+12p+100$
22	$p^3+4.61p^2+1.82p+67.56$	$p^4+5p^3+30p^2+7p+305$

23	$p^2+1.82p+67.56$	$p^4+2p^3+9p^2+4p+35$
24	$p^3+61p^2+182p+67$	$p^4+3p^3+9p^2+0.04p+39$
25	$p^2+1.82p+67.56$	$p^4+5p^3+20p^2+7p+34$

Завдання 7.

1. Відповідно до таблиці 3.5 виконайте:

- обчислення точних (по стандартних функціях MatLAB) значень відповідної функції в діапазоні змінювання аргументу від x_1 до x_2 в m рівновіддалених точках цього діапазону, включаючи його кінці;

- обчислення по зазначених степеневих рядах наближених значень функції в тих же точках, обмежуючись r першими членами ряду;

- розрахунок похибки наближеного визначення функції в кожній точці, порівнюючи наближене значення з точним, і побудову графіка залежності похибки від аргументу;

- обчислення наближених значень функції в тих же точках із абсолютною похибкою не вище $\varepsilon=0.001$; побудову графіка отриманих абсолютних похибок.

Таблиця 3.5

Вар.	x1	x2	m	r	f(x)	Наближена формула
1	0.2	5	20	4	sin(x)	$\sum_{k=1}^{\infty} \frac{x^{2k-1}}{(2k-1)!}$
2	1	10	30	5	cos(x)	$1 - \sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^{2k}}{(2k)!}$
3	0.3	3	40	5	exp(x)	$1 + \sum_{k=1}^{\infty} \frac{x^k}{k!}$
4	0.4	4	50	4	ln(1+x)	$\sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^k}{k}$
5	0.6	6	30	3	ln(x)	$\sum_{k=1}^{\infty} \frac{a^k}{k}$, де $a = \frac{x-1}{x}$
6	0.1	1.9	40	4	ln(x)	$\sum_{k=1}^{\infty} \frac{(-1)^{k+1} a^k}{k}$, де $a = x-1$
7	0.7	7	50	5	ln(x)	$2 \sum_{k=1}^{\infty} \frac{a^{2k-1}}{2k-1}$, де $a = \frac{x-1}{x+1}$
8	0.8	8	45	6	ln(x+a)	$\ln(a) + 2 \sum_{k=1}^{\infty} \frac{b^{2k-1}}{2k-1}$, де $b = \frac{x}{2a+x}$
9	1.1	11	40	3	ctg(x)	$\frac{1}{x} + 2x \sum_{k=1}^{\infty} \frac{1}{x^2 - k^2 \pi^2}$

10	1.2	12	50	4	cosec(x)	$\frac{1}{x} + \frac{x}{6} + \frac{2x^2}{\pi^2} \sum_{k=1}^{\infty} \frac{(-1)^k}{k^2(x^2 - k^2\pi^2)}$
11	1.3	13	50	5	cosec(x)	$\frac{1}{x} + \frac{x}{2} \sum_{k=1}^{\infty} \frac{(-1)^k}{x^2 - k^2\pi^2}$
12	1.4	14	60	6	arctg(x)	$\sum_{k=1}^{\infty} (-1)^{k-1} \frac{x^{2k-1}}{2k-1}$
13	1.5	15	45	5	arctg(x)	$\frac{\pi}{2} + \sum_{k=1}^{\infty} (-1)^k \frac{1}{(2k-1)x^{2k-1}}$
14	1.6	16	40	4	ln(x)	$\sum_{k=1}^{\infty} (-1)^{k-1} \frac{a^k}{k}$, де $a = x - 1$
15	0.9	9	50	6	sin(x)	$x \prod_{k=1}^{\infty} \left(1 - \frac{x^2}{(k\pi)^2}\right)$
16	1	10	50	4	cos(x)	$\prod_{k=1}^{\infty} \left(1 - \frac{(2x)^2}{(2k-1)^2\pi^2}\right)$
17	0.6	5	50	3	ln(x)	$\sum_{k=1}^{\infty} \frac{a^k}{k}$, де $a = (x-1)/x$
18	-0.9	0.9	45	4	arctg(x)	$\frac{\pi}{2} - \sum_{k=1}^{\infty} (-1)^{k-1} \frac{x^{2k-1}}{(2k-1)}$
19	1	20	50	5	sh(x)	$\sum_{k=1}^{\infty} \frac{x^{2k-1}}{(2k-1)!}$
20	1	20	50	5	ch(x)	$1 + \sum_{k=1}^{\infty} \frac{x^{2k}}{(2k)!}$
21	-0.9	0.9	50	5	arth(x)	$\sum_{k=1}^{\infty} \frac{x^{2k-1}}{(2k-1)!}$
22	1	20	50	5	arth(x)	$\sum_{k=1}^{\infty} \frac{1}{(2k-1)! x^{2k-1}}$
23	-0.8	0.8	50	4	arcsin(x)	$x + \sum_{k=1}^{\infty} \frac{1 \cdot 3 \cdot 5 \dots (2k-1) \cdot x^{2k+1}}{2 \cdot 4 \cdot 6 \dots (2k) \cdot (2k+1)}$
24	-0.8	0.8	50	4	arccos(x)	$\frac{\pi}{2} - \left\{ x + \sum_{k=1}^{\infty} \frac{1 \cdot 3 \cdot 5 \dots (2k-1) \cdot x^{2k+1}}{2 \cdot 4 \cdot 6 \dots (2k) \cdot (2k+1)} \right\}$
25	-5	5	50	6	exp(x)	$1 + \sum_{k=1}^{\infty} \frac{x^k}{k!}$

3.7. Контрольні запитання

1. Як подаються дійсні числа при обчисленнях у системі MatLAB?
2. Як змінити формат подання дійсних чисел у командному вікні?
3. Яким чином оголошуються змінні в мові MatLAB?
4. Як зробити так, щоб результат дій, записаних у черговому рядку
 - а) виводився в командне вікно; б) не виводився на екран?
5. Як повернути в командний рядок раніше введену команду ?
6. Як увести значення комплексного числа й у якому виді воно виведеться на екран ?
7. Як мовою MatLAB забезпечити додавання, віднімання, множення, ділення й піднесення до степеня комплексних чисел ?
8. Які функції роботи з комплексними числами є в мові MatLAB ?
9. Що таке "схема Горнера"?
10. Який об'єкт у MatLAB називається поліномом ?
11. Як у MatLAB здійснюється перемноження і ділення поліномів ?
12. За допомогою яких функцій можна знайти корені заданого полінома, значення полінома за відомим значенням аргументу ?
13. Які функції дозволяють відшукати поліном-похідну від заданого полінома ?
14. Що таке степеневий ряд, який відповідає певній функції?
15. Що називають рекурентним співвідношенням?
16. Які степеневі ряди раціонально використовувати при обчисленнях трансцендентних функцій на ЕОМ?
17. Що таке ланцюговий дріб? Як його використовувати при обчисленнях функцій?
18. Як можна відшукати комплексні корені поліному?
19. Що таке передатна функція? частотна передатна функція?
20. Які засоби роботи з комплексними числами є у сучасних мовах програмування?
21. Що таке АЧХ і ФЧХ? як вони обчислюються?

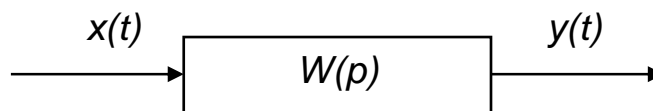


Рис. 3.2. Схема одновимірної системи

Але найбільш зручно і вигідно застосовувати апарат передатних функцій у випадку *багатовимірних систем*, які мають *кілька входів і кілька виходів*. Такі системи вже описуються не однією передатною функцією, а матрицею передатних функцій (див. рис. 3.3).

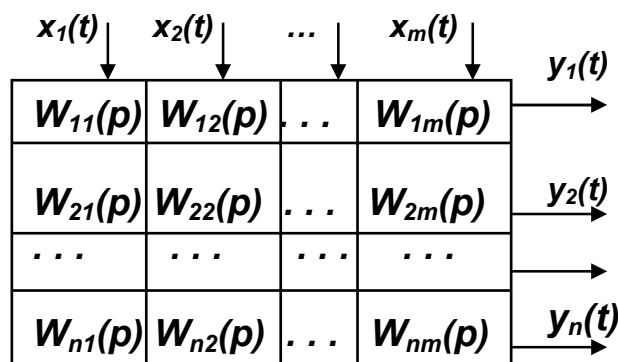


Рис. 3.3. Схема багатовимірної системи

Позначимо

$$W_{ik}(p) = \frac{Y_i(p)}{X_k(p)}. \quad (3.56)$$

Це є передатна функція системи з i -го виходу за k -тим входом.

Багатовимірні системи описуються кількома диференційними рівняннями з відповідною кількістю вихідних величин.

Для прикладу, розглянемо систему диференційних рівнянь, що описує рух гіроскопа у кардановому підвісі:

$$\begin{cases} A\ddot{\alpha} + H\dot{\beta} = N(t) \\ B\ddot{\beta} - H\dot{\alpha} = L(t) \end{cases} \quad (3.57)$$

Вважатимемо вхідними величинами моменти сил $N(t)$ і $L(t)$, що діють вдовж осей карданового підвісу. Як вихідні величини розглядатимемо кутові швидкості $\dot{\alpha}$ і $\dot{\beta}$ повороту гіроскопа. У цьому випадку маємо 2 входи і 2 виходи і матрицю 2-на-2 передатних функцій

$$W_{11}(p) = \frac{\dot{\alpha}(p)}{N(p)}; \quad W_{12}(p) = \frac{\dot{\alpha}(p)}{L(p)}; \quad W_{21}(p) = \frac{\dot{\beta}(p)}{N(p)}; \quad W_{22}(p) = \frac{\dot{\beta}(p)}{L(p)}. \quad (3.58)$$

Щоб відшукати конкретні вирази цих передатних функцій потрібно здійснити наступні операції:

1) здійснити перетворення Лапласа по відношенню до заданих диференціальних рівнянь; наприклад, застосовуючи його до системи рівнянь (57), одержимо

$$\begin{cases} A \cdot p \cdot \dot{\alpha}(p) + H \cdot \dot{\beta}(p) = N(p) \\ B \cdot p \cdot \dot{\beta}(p) - H \cdot \dot{\alpha}(p) = L(p) \end{cases}$$

2) одержану систему алгебричних рівнянь привести до матричної форми, об'єднуючи всі вихідні величини у єдиний вектор вихідних величин, наприклад

$$\begin{bmatrix} Ap & H \\ -H & Bp \end{bmatrix} \cdot \begin{bmatrix} \dot{\alpha}(p) \\ \dot{\beta}(p) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot N(p) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot L(p)$$

3) виділити квадратну матрицю, що є множником при векторі вихідних змінних

$$P(p) = \begin{bmatrix} Ap & H \\ -H & Bp \end{bmatrix}$$

ця матриця називається *характеристичною матрицею системи*;

4) обчислити визначник характеристичної матриці

$$\Delta(p) = \det(P(p)) = \begin{vmatrix} Ap & H \\ -H & Bp \end{vmatrix} = ABp^2 + H^2;$$

5) використовуючи метод Крамера, відшукати передатні функції системи, згідно формули $W_{ik}(p) = \frac{\Delta_{ik}(p)}{\Delta(p)}$,

де $\Delta_{ik}(p)$ - визначник матриці, яка одержується з характеристичної шляхом заміни i -го стовпця її на стовпець при k -тій вхідній величині; наприклад, для розглядуваної системи

$$\Delta_{11}(p) = \det \begin{bmatrix} 1 & H \\ 0 & Bp \end{bmatrix} = Bp; \quad \Delta_{12}(p) = \det \begin{bmatrix} 0 & H \\ 1 & Bp \end{bmatrix} = -H;$$

$$\Delta_{21}(p) = \det \begin{bmatrix} Ap & 1 \\ -H & 0 \end{bmatrix} = H; \quad \Delta_{22}(p) = \det \begin{bmatrix} Ap & 0 \\ -H & 1 \end{bmatrix} = Ap;$$

в результаті одержуються такі передатні функції гіроскопа у кардановому підвісі:

$$\begin{aligned} W_{11}(p) &= \frac{Bp}{ABp^2 + H^2}; & W_{12}(p) &= \frac{-H}{ABp^2 + H^2}; \\ W_{21}(p) &= \frac{H}{ABp^2 + H^2}; & W_{22}(p) &= \frac{Ap}{ABp^2 + H^2}. \end{aligned} \quad (3.59)$$

Знайшовши конкретні вирази для усіх передатних функцій багатовимірної системи, можна досить легко розв'язати задачу відшукання усталених вимушених коливань, якщо усі вхідні величини змінюються у часі за гармонічним

законом з єдиною частотою. Для цього, як було попередньо зазначено, зручніше за все скористатися формулами (54) і (55):

$$y_{ic} = \sum_{k=1}^{s1} [x_{kc} \cdot U_{ik}(\omega) + x_{ks} \cdot V_{ik}(\omega)]; \quad y_{is} = \sum_{k=1}^{s1} [x_{kc} \cdot V_{ik}(\omega) - x_{ks} \cdot U_{ik}(\omega)]. \quad (3.60)$$

Тут i - номер вихідної величини, k - номер вхідної величини, $s1$ - кількість вхідних величин, що змінюються з часом за гармонічним законом з частотою ω , тобто припускається

$$x_k(t) = x_{ks} \cdot \sin \omega t + x_{kc} \cdot \cos \omega t,$$

$U_{ik}(\omega)$ - дійсна частина ЧПФ з i -того виходу по k -му входу, $V_{ik}(\omega)$ - уявна частина відповідної ЧПФ.

Як приклад розглянемо усталені коливання гіроскопа у кардановому підвісі з кутової швидкості $\dot{\alpha}(t)$ його обертання навколо зовнішньої осі підвісу при дії на нього гармонічних моментів сил вдовж зовнішньої осі

$$N(t) = N_s \cdot \sin \omega t + N_c \cdot \cos \omega t$$

і вдовж внутрішньої осі підвісу

$$L(t) = L_s \cdot \sin \omega t + L_c \cdot \cos \omega t.$$

Для цього спочатку відшукаємо відповідні ЧПФ на основі знання ПФ (59):

$$W_{11}(j\omega) = j \frac{B\omega}{H^2 - AB\omega^2}; \quad W_{12}(j\omega) = \frac{-H}{H^2 - AB\omega^2};$$

$$W_{21}(j\omega) = \frac{H}{H^2 - AB\omega^2}; \quad W_{22}(j\omega) = j \frac{A\omega}{H^2 - AB\omega^2}.$$

Як бачимо, діагональні ЧПФ є суто уявними, а решта - суто дійсними. Враховуючи знайдені ЧПФ і користуючись співвідношеннями (60), матимемо:

$$\dot{\alpha}(t) = \frac{1}{H^2 - AB\omega^2} [(B\omega N_c + HL_s) \cdot \sin \omega t + (B\omega N_s - HL_c) \cdot \cos \omega t].$$

Використання передатних і частотних передатних функцій в інженерній практиці обумовлено наступними міркуваннями:

- 1) вичерпністю ПФ і ЧПФ для опису поведінки системи за будь-яких збурень;
- 2) простотою і зручністю відшукування ПФ навіть для складних систем;
- 3) прозорістю фізичного змісту ЧПФ і окремих її параметрів;
- 4) можливістю відшукування ПФ за схемою з'єднання окремих ланок системи, що описуються заданими їх ПФ.

Зупинемося докладніше на останньому.

Нехай маємо дві одновимірні ланки, що описуються передатними функціями $W_1(p)$ і $W_2(p)$. Позначимо їхні входи відповідно через x_1 і x_2 , а виходи - y_1 і y_2 (див. рис. 3.4).

Послідовним з'єднанням двох ланок називають таке, коли вихід першої ланки становиться входом другої (рис. 3.5). В результаті одержується нова, більш складна ланка, входом якої є вхід першої ланки, а виходом - вихід другої.

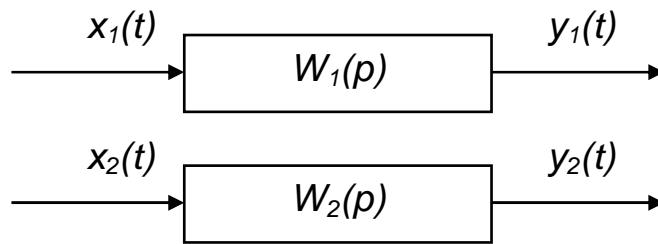


Рис. 3.4. Схема ланок

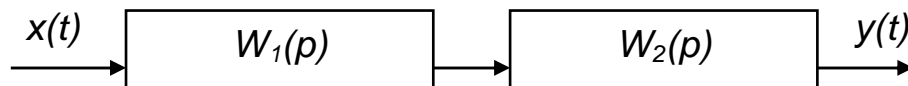


Рис. 3.5. Послідовне з'єднання ланок

Передатна функція нової ланки, як неважко впевнитися, дорівнює добутку передатних функцій складових ланок

$$W(p) = W_1(p) \cdot W_2(p). \quad (3.61)$$

Паралельним з'єднанням двох ланок називають таке, коли вихід у обох ланок є спільним, а виходи підсумовуються (рис. 3.6).

Передатна функція паралельного з'єднання дорівнює сумі передатних функцій складових ланок

$$W(p) = W_1(p) + W_2(p). \quad (3.62)$$

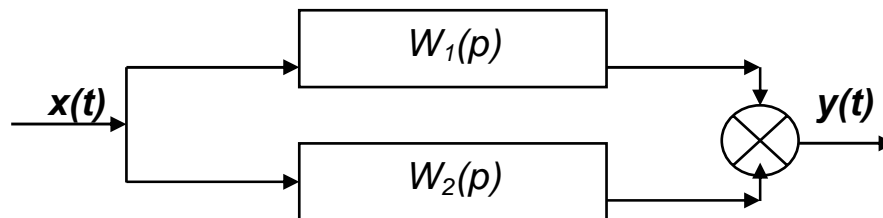


Рис. 3.6. Паралельне з'єднання ланок

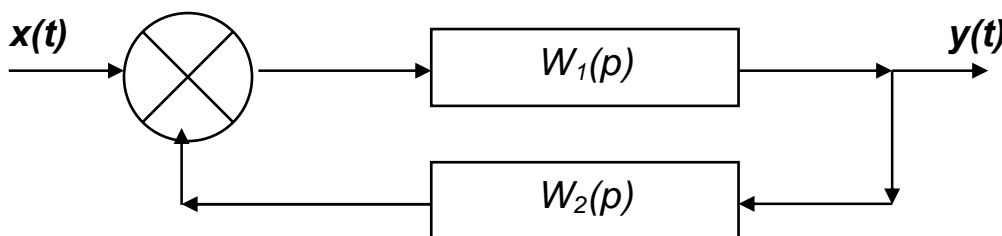


Рис. 3.7. Зворотний від'ємний зв'язок

З'єднання типу зворотного від'ємного зв'язку здійснюється так, як подано на рис. 3.7. При цьому перша ланка утворює прямий ланцюг, а друга - ланцюг зворотного зв'язку. Вхід першої ланки утворює різниця між загальним входом

результуючої ланки і виходом другої ланки. Вихід першої ланки є одночасно виходом результуючої ланки і входом другої.

Передатна функція такого з'єднання визначається формулою

$$W(p) = \frac{W_1(p)}{1 + W_1(p)W_2(p)}. \quad (3.63)$$

У більшості випадків інженери складають системи автоматичного керування саме шляхом з'єднання між собою окремих пристроїв одним з поданих способів. Для дослідження утворюваних при цьому систем нема потреби складати диференціальні рівняння результуючої системи. Достатньо відшукати її передатну функцію за відомими ПФ складових ланок, користуючись викладеними правилами.

3. 6. Обчислення значень частотних передатних функцій

Головними способами дослідження лінійних стаціонарних систем (ЛСС) за їхніми ЧПФ є наступні:

- обчислення значень полюсів і нулів ЧПФ;
- побудова графіків модуля ЧПФ при змінюванні частоти ω у широкому діапазоні; через те, що модуль ЧПФ у відповідності з (52) дорівнює відношенню амплітуд вихода і входу, графік залежності модуля ЧПФ від частоти називають амплітудно-частотною характеристикою (АЧХ) системи;
- побудова графіка залежності аргумента ЧПФ від частоти (у відповідності з (53) він називається фазо-частотною характеристикою (ФЧХ) системи.

В усіх випадках при обчисленнях необхідно мати справу з комплексними числами. Тому використання апарату ЧПФ необхідно пов'язане з діями над комплексними числами.

Такі дії безпосередньо можливі лише при користуванні мовою Fortran. При застосуванні мови Pascal потрібно або самостійно утворити бібліотеку (Units) процедур і функцій операцій з комплексними числами, або придбати існуючу бібліотеку цих процедур.

Більшості проблем цього плану можна позбутися, якщо використовувати систему MatLAB, в якій передбачені вбудовані дії з комплексними числами, операції з поліномами (див. п. 3.1). Особливо зручним є наявність функції *roots* обчислення коренів полінома, яка дозволяє одразу визначати нулі і полюси передатної функції.

Нехай нам потрібно визначити нулі і полюси передатної функції

$$W(p) = \frac{0,7p^2 + p}{4p^4 + 0,1p^3 + 7p^2 + p + 125}.$$

Тоді у середовищі MatLAB достатньо здійснити такі дії:

- 1) ввести вектор Ch коефіцієнтів полінома чисельника ПФ:
» Ch=[0.7 1 0]

Ch =
0.7 1 0

2) ввести вектор Zn коефіцієнтів полінома знаменника ПФ:

» Zn=[4 0.1 7 1 125]

Zn =
4 0.1 7 1 125

3) визначити полюси ПФ, застосовуючи процедуру *roots* до вектора її знаменника

» Poles=roots(Zn)

Poles =
1.5292 + 1.8082i
1.5292 - 1.8082i
-1.5417 + 1.7876i
-1.5417 - 1.7876i

як бачимо, система є коливальною і нестійкою;

4) застосувати ту ж процедуру до вектора чисельника і цим відшукати значення нулів системи

» Noles=roots(Ch)

Noles =
0
-1.4286

Покажемо як будується АЧХ і ФЧХ на прикладі ПФ виду

$$W(p) = \frac{p^2 + 100}{p^4 + 2,2p^3 + 1617,4p^2 + 352,22p + 25632}$$

Для цього введемо вектор коефіцієнтів полінома чисельника

» ch=[1 0 100]

ch = 1 0 100

і вектор полінома знаменника

» zn=[1 2.2 1617.4 352.22 25632]

zn = 1 2.2 1617.4 352.22 25632

Задля контролю знайдемо нулі

» roots(ch)

ans =
0 + 10i
0 - 10i

і полюси системи

» roots(zn)

ans =
-1 + 40i
-1 - 40i
-0.1 + 4i
-0.1 - 4i

Таким чином, розглядувана система є коливною, стійкою, з частотами власних коливань 4 і 40 рад/с.

Щоб побудувати АЧХ і ФЧХ, спочатку утворимо вектор значень частот

» om=0:0.2:100;

потім вектор суто уявних відповідних значень комплексного аргументу p

» p=j*om;

Далі розраховуються комплексні значення чисельника

» chis=polyval(ch,p);

і знаменника ЧПФ

» znam=polyval(zn,p);

Тепер можна обчислити модуль ЧПФ

```
» ACH=sqrt(abs(chis))./sqrt(abs(znam));
```

і побудувати графік АЧХ

```
» plot(om,ACH)
```

```
» grid
```

```
» title('Амплітудно - частотна характеристика')
```

```
» ylabel('Відношення амплітуди вихода до амплітуди входу')
```

```
» xlabel('Частота (рад/с)')
```

Результат поданий на рис. 3.8.

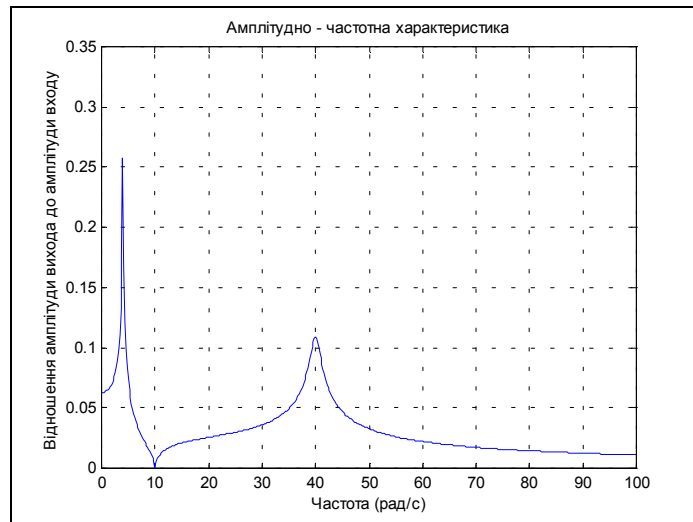


Рис. 3.8. АЧХ, побудована засобами MatLAB

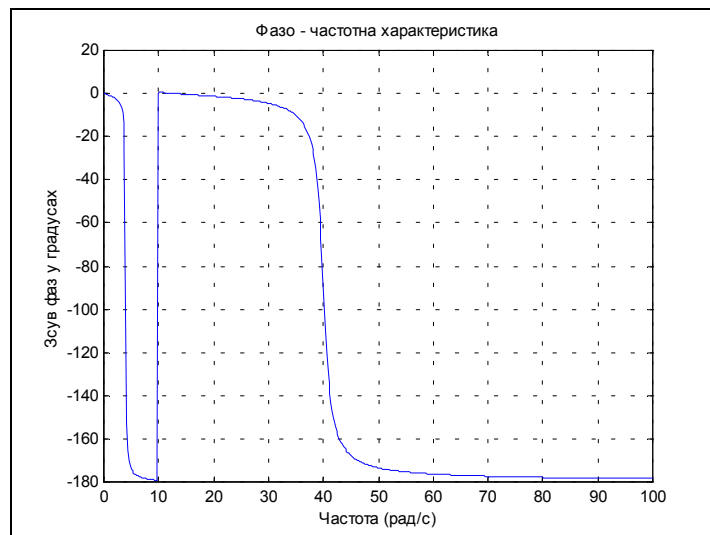


Рис. 3.9. ФЧХ, побудована засобами MatLAB

Нарешті розрахуємо значення ФЧХ

```
» FCH=angle(chis./znam)*180/pi;
```

і будуємо її графік

```
» plot(om,FCH)
```

```
» grid
```

```
» title('Фазо - частотна характеристика')
```

```
» xlabel('Частота (рад/с)')
```

```
» ylabel('Зсув фаз у градусах')
```

В результаті одержуємо графік, поданий на рис. 3.9.

Як бачимо, система MatLAB містить усі необхідні засоби дослідження лінійних стаціонарних систем за їхніми АЧХ. Окрім розглянутих функцій в ній передбачено спеціальний пакет функцій і процедур CONTROL аналізу і синтезу лінійних систем.

3.6. Завдання

Завдання 1. Користуючись середовищем MATLAB як калькулятором, обчисліть у ньому зазначений арифметичний вираз. Укажіть послідовність натискання клавіш. Порівняйте отриманий результат із наведеною відповіддю.

	Відповідь
1. $\frac{\left(12\frac{1}{6} - 6\frac{1}{27} - 5,25\right)13,5 + 0,111}{0,02}$	599,3
2. $\frac{\left(1\frac{1}{12} + 2\frac{5}{32} + \frac{1}{24}\right) : 9,6 + 2,13}{0,0004}$	179,5
3. $\frac{\left(6,6 - 3\frac{3}{14}\right)5\frac{5}{6}}{(21 - 1,25) : 2,5}$	2,5
4. $\frac{2,625 - \frac{2}{3} \cdot 2\frac{5}{14}}{\left(3\frac{1}{12} + 4,375\right) : 19\frac{8}{9}}$	2,8095
5. $\frac{0,134 + 0,05}{18\frac{1}{6} - 1\frac{11}{14} - \frac{2}{15} \cdot 2\frac{6}{7}}$	0,00115
6. $\frac{\left(58\frac{4}{15} - 56\frac{7}{24}\right) : 0,8 + 2\frac{1}{9} \cdot 0,225}{8,75 \cdot 0,6}$	0,56071
7. $\frac{\left(\frac{0,216}{0,15} + 0,56\right) : 0,5}{\left(7,7 : 24,75 + \frac{2}{15}\right)4,5}$	2
8. $\frac{1\frac{4}{11} \cdot 0,22 : 0,3 - 0,96}{\left(0,2 - \frac{3}{40}\right)1,6}$	-4,35

9. $\frac{\left(\frac{3}{5} + 0,425 - 0,005\right) : 0,12}{30,5 + \frac{1}{6} + 3\frac{1}{3}} \cdot 0,25$ 0,25
10. $\frac{3\frac{1}{3} + 2,5}{2,5 - 1\frac{1}{3}} \cdot \frac{4,6 - 2\frac{1}{3}}{4,6 + 2\frac{1}{3}} : \left(\frac{0,05}{\frac{1}{7} - 0,125} + 5,7\right)$ 0,19231
11. $\frac{0,725 + 0,42(6)}{0,128 - 6,25 - (0,0345 : 0,12)} \cdot 0,25$ -0,04492
12. $\frac{\left(4,5 \cdot 1\frac{2}{3} - 6,75\right) \cdot 0,(6)}{\left(3,333 \cdot 0,3 + 0,222 \cdot \frac{4}{9}\right) 2\frac{2}{3}}$ 0,17068
13. $\frac{\left(5\frac{4}{45} - 4\frac{1}{6}\right) : 5\frac{8}{15}}{\left(4\frac{2}{3} + 0,75\right) 3\frac{9}{13}} \cdot 34\frac{2}{7}$ 0,28571
14. $\frac{1\frac{4}{11} \cdot 0,22 : 0,3 - 0,96}{\left(0,2 - \frac{3}{40}\right) 1,68}$ 0,19048
15. $\frac{\left(40\frac{7}{30} - 38\frac{5}{12}\right) : 10,9 + \left(0,875 - \frac{7}{30}\right) \cdot \frac{20}{11}}{0,008}$ 166,67
16. $\frac{(68,023 - 66,028) : 6\frac{1}{9} + \frac{7}{40} \cdot 4,5}{0,042 + 0,086}$ 8,7028
17. $\frac{(2,1 - 1,965) : (1,2 \cdot 0,045)}{0,00325 : 0,013} - \frac{4}{0,2 \cdot 0,73}$ -17,397
18. $\frac{(1,88 + 2,127) \cdot 0,01875}{0,625 - \frac{13}{18} : 3,13} + 8,29$ 8,2441
19. $\frac{3 : 0,4 - 0,009 : (0,15 : 2,5)}{0,32 \cdot 6 + 0,033 - (5,3 - 3,88)}$ 13,79
20. $\frac{(34,06 - 33,81) \cdot 4}{6,84 : (28,57 - 25,15)} + 1,33 : \frac{4}{21}$ 7,4825

21. $\frac{8,8077}{20 - (28,2 : (13,333 \cdot 0,3 + 0,0125))2,004}$ 1,4889
22. $\frac{\left(1,75 : \frac{2}{3} - 1,75 \cdot 1,125\right) : \frac{7}{12}}{(0,2012 - 0,0325) : 400}$ 2667,5
23. $\frac{\left(26\frac{1}{3} - 18,02 \cdot 0,75\right) \cdot 2,4 : 0,88}{1,37 - 23\frac{2}{3} : 1,82}$ -3,005
24. $26 : \frac{3 : (0,48 - 0,27)}{2,52(1,38 + 2,45)} + 1,27$ 8,836
25. $\left(16,5 - 13\frac{7}{9}\right)\frac{6}{11} + 2,2 : (0,241 - 0,91)$ -1,8036

Завдання 2. Проведіть обчислення по заданій формулі при заданих значеннях параметрів. Зазначте необхідну послідовність дій. Порівняйте отриманий результат із наведеною відповіддю.

Вказівка. У системі *MatLAB* декілька останніх команд запам'ятовуються. Виклик цих команд у командне вікно здійснюється натисканням клавіш $\langle \downarrow \rangle$ і $\langle \uparrow \rangle$. Використовуйте цю можливість для повторного звернення до набраної функції.

1. $3m^2 + \sqrt[3]{2n^2} : m$; а) $m = -\frac{14}{5}$, $n = \operatorname{tg} \frac{\pi}{8}$; б) $m = 2,2 \cdot 10^{-2}$, $n = \frac{1}{3,1}$.

ВІДПОВІДЬ: а) 23,27; б) 26,938.

2. $\frac{4}{3}l^3 \sin^2 \frac{\alpha}{2} \sqrt{\cos \alpha}$; а) $l = 1,7 \cdot 10^3$, $\alpha = 18^\circ$; б) $l = \frac{16}{21}$, $\alpha = \frac{\pi}{5}$.

ВІДПОВІДЬ: а) 1. 5633e+008; б) 5. 0651e-002.

3. $\sqrt{\frac{a\sqrt{b}}{\sqrt[3]{\operatorname{tg} \alpha}}}$; а) $a = 1,5$, $b = 0,8$, $\alpha = 61^\circ$; б) $a = 3 \cdot 10^{-2}$, $b = 0,71$, $\alpha = \frac{3}{7}\pi$.

ВІДПОВІДЬ: а) 1. 0498e+000; б) 1. 2429e-001.

4. $\frac{3a^2 \sqrt{6,8 \cdot (a-b)}}{4(a+b)^3}$; а) $a = 4,13 \cdot 10^{-1}$, $b = \frac{1}{261}$;

б) $a = \sin \frac{5\pi}{8}$, $b = -\operatorname{tg} 12^\circ$

ВІДПОВІДЬ: а) 2. 9464e+000; б) 4. 9445e+000.

5. $\frac{c^3}{6} \cos \frac{\alpha}{2} \sqrt{\sin \alpha}$; а) $c = \lg 2,38$, $\alpha = \frac{\pi}{5}$; б) $c = e^{-0,3}$, $\alpha = 65^\circ$.

ВІДПОВІДЬ: а) 3. 4657e-004; б) 2. 2120e-002.

$$6. \sqrt{\frac{n^3}{16,3 \sin \alpha \sin 2\alpha}}; \text{ а) } n = 3,1516 \cdot 10^{-2}, \alpha = 5^\circ; \text{ б) } n = e^{3,5}, \alpha = \frac{2\pi}{13}.$$

ВІДПОВІДЬ: а) 1. 1265e-002; б) 7. 6324e+001.

$$7. 5 \sin 35^\circ \sqrt{\frac{S^3 \cos 36^\circ}{\pi^3 \operatorname{tg} \alpha}}; \text{ а) } S = \ln 3, \alpha = 44^\circ; \text{ б) } S = \frac{18}{25}, \alpha = \frac{7}{12}\pi.$$

ВІДПОВІДЬ: а) 5. 4283e-001; б) 8. 9703e-018+ 1. 4650e-001i.

$$8. |\lg(1 + \sin \alpha) + \ln(1 - \sin \beta)|; \text{ а) } \alpha = \frac{3\pi}{7}, \beta = 83^\circ; \text{ б) } \alpha = \frac{2}{3}\pi, \beta = 16^\circ.$$

ВІДПОВІДЬ: а) 4. 6035e+000; б) 5. 1546e-002.

$$9. \sqrt[3]{\sin^2(\alpha + \beta) - \sin^2(\alpha - \beta)}; \text{ а) } \alpha = \frac{5}{7}\pi, \beta = 0,3\pi; \text{ б) } \alpha = 12^\circ, \beta = 220^\circ$$

ВІДПОВІДЬ: а) 4. 8756e-001+ 8. 4448e-001i; б) 7. 3715e-001.

$$10. (\log_a(b + 1,4))^{-3/4}; \text{ а) } a = 3,56, b = e^{0,316}; \text{ б) } a = 2, b = 2,1649 \cdot 10^{-2}.$$

ВІДПОВІДЬ: а) 1. 1790e+000; б) 1. 6630e+000.

$$11. 3 \left(p^{-2/3} + q^{-1/2} \right) \sqrt[3]{pq}; \text{ а) } p = \ln 3, q = \lg 3; \text{ б) } p = 0,013, q = 1,4 \cdot 10^2.$$

ВІДПОВІДЬ: а) 5. 7737e+000; б) 6. 6559e+001.

$$12. \frac{2}{3} m \sqrt{m^3 \sqrt{m^4 m}}; \text{ а) } m = 3,6485 \cdot 10^2; \text{ б) } m = \frac{24}{37}.$$

ВІДПОВІДЬ: а) 1. 5880e+004; б) 5. 4516e-001.

$$13. \frac{8}{3} S \sqrt{\frac{S}{\pi}} \sin^6 \frac{\alpha}{2}; \text{ а) } S = e^{1,11}, \alpha = \frac{7}{11}\pi; \text{ б) } S = 5,403, \alpha = 28^\circ.$$

ВІДПОВІДЬ: а) 2. 8187e+000; б) 3. 7879e-003.

$$14. 2 \sqrt{\frac{F}{\pi}} \operatorname{tg} \alpha \sin^2 \frac{\alpha}{2}; \text{ а) } F = \frac{1}{0,03}, \alpha = \frac{5}{7}\pi; \text{ б) } F = \ln 7, \alpha = 1,34^\circ.$$

ВІДПОВІДЬ: а) -6. 6313e+000; б) 5. 0346e-006.

$$15. \frac{1}{12} \cdot \frac{m^3 \cos \alpha}{(\sin \alpha + \cos \alpha)^3}; \text{ а) } m = -20,1, \alpha = 20^\circ; \text{ б) } m = \lg 13,6, \alpha = 1,48.$$

ВІДПОВІДЬ: а) -3. 0201e+002; б) 8. 5792e-003.

$$16. \frac{\sqrt{3h^3}}{\cos^2 \alpha} \sin(\alpha + 30^\circ) \sin(\alpha - 30^\circ);$$

$$\text{ а) } h = 0,28, \alpha = 41^\circ; \text{ б) } h = e^{0,415}, \alpha = 237^\circ.$$

ВІДПОВІДЬ: а) 8. 1284e-002; б) 4. 9334e+000.

$$17. \frac{\alpha}{3} (\lg(d + 2) - \operatorname{tg} \alpha)^2;$$

$$\text{ а) } d = 6,178, \alpha = 20^\circ; \text{ б) } d = -2,2461 \cdot 10^{-2}, \alpha = 1,146.$$

ВІДПОВІДЬ: а) 3. 5028e-002; б) 1. 4003e+000.

18. $d^3 \operatorname{ctg} \alpha \sqrt{\sin^4 \alpha - \cos^4 \alpha}$; а) $d = 10,6$, $\alpha = 50^\circ$; б) $d = e^{2,3}$, $\alpha = 1$.

ВІДПОВІДЬ: а) 4. 1645e+002; б) 4. 1101e+002.

19. $\frac{a^2 \sqrt{3}}{4} (\sec \alpha + \operatorname{cosec} \alpha)^4$; а) $a = 5,08$, $\alpha = 25^\circ$; б) $a = \ln 1,37$, $\alpha = \frac{12}{25} \pi$

ВІДПОВІДЬ: а) 1. 6193e+003; б) 3. 5238e+003.

20. $\frac{\sqrt{\pi}}{3} \cdot \frac{1}{(\operatorname{ctg} A + \operatorname{ctg} B)^2}$; а) $A = 51^\circ$, $B = 39^\circ$; б) $A = 0,643$, $B = \frac{\pi}{7}$.

ВІДПОВІДЬ: а) 1. 4132e-001; б) 5. 0772e-002.

21. $\lg\left(3^{x^2-x-9} + \frac{8}{27}\right)$; а) $x = e^{1,648}$; б) $x = \operatorname{tg} 1,21$.

ВІДПОВІДЬ: а) 6. 1109e+000; б) -5. 1927e-001.

22. $\frac{\sqrt[5]{5e^{4a}} (a + 12,36)^2}{\ln(a + 7)}$; а) $a = 2,1754 \cdot 10^2$; б) $a = \cos 17^\circ$.

ВІДПОВІДЬ: а) 8. 5511e+075; б) 4. 0272e+000;

23. $\lg^2 x - \left(\frac{27}{8}\right)^{x-1} \sin \sqrt{x}$; а) $x = e^{2,145}$; б) $x = 2,468 \cdot 10^{-1}$.

ВІДПОВІДЬ: а) -2. 0936e+003; б) 1. 7858e-001.

24. $\sqrt[5]{(x-y)^2} \sqrt[3]{\frac{1}{y-x}}$; а) $x = e^{-0,37}$, $y = \ln 2,1517$; б) $x = 37^\circ$, $y = \cos \frac{7}{24} \pi$.

ВІДПОВІДЬ: а) 3. 4445e-001; б) 2. 6745e-001.

25. $\frac{\sin A + \operatorname{tg} B}{\sqrt[5]{(A-3B)^2}}$; а) $A = 5,6$, $B = \lg 25$; б) $A = \frac{8}{9} \pi$, $B = \frac{\pi}{10}$.

ВІДПОВІДЬ: а) 4. 4466e+000; б) 5. 2145e-001.

Завдання 3. Виконайте такі дії (див. таблицю 3.1):

а) число z_1 , задане в алгебричній (експоненціальній) формі, перекладіть в експоненціальну (алгебричну) форму і запишіть результат;

б) число z_2 , задане в експоненціальній (алгебричній) формі, перекладіть в алгебричну (експоненціальну) форму і запишіть результат;

в) обчисліть заданий вираз; запишіть результат в алгебричній і експоненціальній формах.

Таблиця 3.1

Варіант	Комплексне число				Вираз
	z_1	z_2	z_3	z_4	
1	$4 + 3i$	$2,71e^{i\pi/12}$	$1,82e^{-1,2i}$	$\sqrt{3} - 2i$	$z_1^2 \cdot z_2 \cdot z_3 + z_4$
2	$0,8 - 2i$	$3,08e^{i7\pi/12}$	$8,01e^{2i}$	$-5 + \sqrt{2}i$	$z_1^2 : z_2 + z_3 - z_4$

3	$-0,7 + 4i$	$1,74e^{i0,3\pi}$	$3 + 4i$	$2,1e^{-2,3i}$	$\sqrt{z_1:z_2} \cdot z_3 + z_4$
4	$-3 - 2i$	$3,21e^{15^\circ i}$	$1,2 + 3i$	$2,71e^{-78^\circ i}$	$\sqrt{z_1 \cdot z_2}:z_3 + z_4$
5	$2,71e^{i\pi/12}$	$-0,7 + 4i$	$1,31e^{-i5\pi/12}$	$-8 - 3i$	$\sqrt{z_1}:z_2 \cdot z_3 - z_4$
6	$3,08e^{i7\pi/12}$	$-3 - 2i$	$2,03e^{i4/13}$	$\sqrt{3} + \sqrt{2}i$	$(z_1 + z_2)^4 \cdot z_3:z_4$
7	$1,74e^{0,3\pi i}$	$0,8 - 2i$	$3,28e^{-1,2i}$	$\sqrt{3} - \sqrt{2}i$	$(\sqrt{z_1} + z_2) \cdot z_3:z_4$
8	$3,21e^{15i}$	$4 + 3i$	$\sqrt{3} - 4i$	$1,23e^{111^\circ i}$	$(z_1 - z_2)^3 \cdot z_3 + z_4$
9	$1 + i\pi/2$	$1,2e^{107^\circ i}$	$0,8 - 2i$	$2,5e^{14^\circ i}$	$(z_1:z_2 + z_3)^3 \cdot z_4$
10	$\sqrt{5} - i$	$0,7e^{1,7i}$	$1,2e^{0,9i}$	$-3 - 2i$	$(z_1:z_2 + z_3)^2 - z_4$
11	$0,187 - 3,94i$	$0,3e^{-107^\circ i}$	$-0,7 + 4i$	$1,5e^{23^\circ i}$	$\sqrt[3]{z_1 + z_2 - z_3} \cdot z_4$
12	$-1 + \sqrt{5}i$	$2,1e^{211^\circ i}$	$0,4e^{32^\circ i}$	$4 + 3i$	$\sqrt[3]{z_1 \cdot z_2}:z_3 + z_4$
13	$-\sqrt{3} - 4i$	$1,25e^{-0,8i}$	$-3 - 2i$	$0,75e^{0,7i}$	$(\sqrt[3]{z_1 \cdot z_2} + z_3):z_4$
14	$1,2e^{1,7i}$	$0,18 - 3,9i$	$0,71e^{4i}$	$0,8 - 2i$	$(\sqrt[3]{z_1:z_2} + z_3) \cdot z_4$
15	$0,3e^{-97^\circ i}$	$-1 + \sqrt{5}i$	$-0,7 + 4i$	$5,2e^{71^\circ i}$	$(\sqrt{z_1 \cdot z_2} - z_3):z_4$
16	$1,25e^{0,6i}$	$-\sqrt{3} - 4i$	$4 + 3i$	$2,5e^{3,8i}$	$(\sqrt{z_1:z_2} - z_3) \cdot z_4$
17	$1,05e^{-0,4i}$	$\sqrt{5} - i$	$2,7e^{0,8i}$	$-0,7 + 4i$	$\sqrt{(z_1:z_2 + z_3) \cdot z_4}$
18	$2,1e^{73^\circ i}$	$1 + i\pi/2$	$\sqrt{2} + \sqrt{3}i$	$1,93e^{192^\circ i}$	$\sqrt{(z_1 \cdot z_2 - z_3):z_4}$
19	$2,7 + 0,8i$	$2e^{-\sqrt{3}i}$	$0,81e^{i\pi/7}$	$\sqrt{2} \cdot \sqrt{3}i$	$\sqrt{(z_1 + z_2):z_3 \cdot z_4}$
20	$-0,8 + 2,7i$	$-2e^{\sqrt{3}i}$	$0,9e^{i5\pi/7}$	$3,1 - 2,1i$	$\sqrt{(z_1 + z_2) \cdot z_3:z_4}$
21	$-1,1 - 3,2i$	$0,33e^{-1,9i}$	$2e^{\sqrt{2}i}$	$2,08 + i$	$\sqrt{z_1 - z_2} \cdot z_3:z_4$
22	$2,1 - 3,2i$	$0,68e^{148^\circ i}$	$\sqrt{5} + \sqrt{2}i$	$2,73e^{23^\circ i}$	$\sqrt{z_1 - z_2}:z_3 \cdot z_4$
23	$1,1e^{-0,8i}$	$\sqrt{5} - 2i$	$-1,7 + i$	$0,97e^{\sqrt{2}i}$	$((z_1 + z_2)^2 - z_3):z_4$
24	$2,1e^{0,8i}$	$-\sqrt{5} + 2i$	$1,7e^{\sqrt{3}i}$	$0,8e^{2,5i}$	$((z_1 - z_2)^2 + z_3):z_4$
25	$1,1e^{-2,1i}$	$\pi/8 - 2,1i$	$2,71 + 0,4i$	$1,71e^{-\sqrt{3}i}$	$(z_1 - z_2)^3:z_3 + z_4$

Завдання 4. Відшукайте корені квадратного рівняння

$$a \cdot x^2 + b \cdot x + c = 0$$

при заданих значеннях коефіцієнтів a , b і c (табл. 3.2) за точними формулами і користуючись функцією **roots**. Порівняйте результати.

Таблиця 3.2

Варіант	a	b	c
1	0.56	1. 2e-4	4.08
2	1	0.1	100
3	4. 2e-3	8. 03e-4	1.06
4	7. 1e3	9. 4e4	8. 3e10
5	5.09	4.32	256
6	8.3	5.34	693
7	27	27	1276
8	3.08	0.2	30
9	5.3	10.6	876
10	0.45	0. 034	121
11	4.3	10.7	3. 4e3
12	13	0.8	287
13	6. 035	5.2	875
14	2.3	7.9	324
15	1	0.02	16.57
16	1.3	0.56	18.8
17	0.13	0. 056	18.8
18	17	12	956
19	0. 085	1	1. 3e3
20	1.2	0.32	15
21	7.1	6.4	256
22	0.2	0. 002	2.9
23	1. 4e-3	3.9	2. 6e2
24	0.86	3.2	5. 4e2
25	7. 3e3	8. 2e2	3. 5e8

Завдання 5. Обчисліть значення функції $f(x)$ на відрізку $[a;b]$ із кроком h у відповідності з табл. 3.3.

Таблиця 3.3

Варіант	$f(x)$	a	b	h
1	$\frac{x^2}{1+0,25\sqrt{x}}$	1,1	3,1	0,02
2	$\frac{x^3 - 0,3x}{\sqrt{1+2x}}$	2,05	3,05	0,01
3	$\frac{2e^{-x}}{2\pi + x^3}$	0	1,6	0,016

4	$\frac{\cos \pi x^2}{\sqrt{1-3x}}$	-1	0	0,01
5	$\sqrt{1+4x} \sin \pi x$	0,1	0,8	0,002
6	$\frac{e^{x/3}}{1+x^2}$	1,4	2,4	0,01
7	$e^{-2x} + x^2 - 1$	0,25	2,25	0,02
8	$(\cdot + \cdot) \alpha v(\pi \sqrt{\cdot} - 1)$	1,8	2,8	0,01
9	$\sqrt{3+2x} \cdot \operatorname{tg} \frac{\pi x^3}{2}$	0,1	0,9	0,08
10	$\sqrt{2+3x} \cdot \ln(1+3x^2)$	-0,1	0,9	0,01
11	$\sqrt[3]{x^2+3} \cdot \cos \frac{\pi x}{2}$	1	2,5	0,015
12	$(4+7x) \sin(\pi^3 \sqrt{1+x})$	0	7	0,07
13	$e^{-x^2} (1+3x-x^2)$	0	2	0,02
14	$x^3 - 3x + \frac{8}{\sqrt{1+x^2}}$	0	1,7	0,17
15	$\sqrt{\operatorname{sh} \sqrt{2} \pi}, \left(\operatorname{sh} x = \frac{e^x - e^{-x}}{2} \right)$	0	1,2	0,12
16	$\sqrt{\operatorname{ch} \frac{x}{\sqrt{2\pi}}}, \left(\operatorname{ch} x = \frac{e^x + e^{-x}}{2} \right)$	0,5	1,5	0,01
17	$\frac{x^3 + 2x}{\sqrt{1+e^x}}$	-0,2	0,8	0,01
18	$\sqrt{1+2x^2} \cdot \sin \frac{3x}{2}$	2	4	0,2
19	$\sqrt{3x^2+5} \cdot \cos \frac{\pi x}{2}$	0,5	1,5	0,1
20	$\arccos e^{-\sqrt[3]{3x}}$	0,2	0,5	0,03
21	$\arcsin e^{-x^2/5}$	8	13	0,05
22	$x + \ln(x + \sqrt{1+x^2})$	-0,5	0,5	0,01
23	$\frac{1+e^{-x/2}}{\sqrt{3x^2+1}}$	3	5	0,02

24	$3x^3 + \frac{1}{x} + e^{-2x^2}$	1,2	2,2	0,01
25	$x^{2x+1} + x^3 - 2x$	1	5	0,04

Побудуйте в графічному вікні MatLAB графік функції. Роздрукуйте цей графік на аркуші паперу.

Завдання 6. Обчисліть вектори:

- модуля частотної передатної функції (ЧПФ);
- аргументу ЧПФ;
- дійсної частини ЧПФ;
- уявної частини ЧПФ

по заданих чисельнику і знаменнику передатної функції (табл. 3.4).

Попередньо відшукайте корені знаменника передатної функції, визначити найбільшу власну частоту ω_{\max} системи. Забезпечте обчислення ЧПФ при 100 значеннях частоти ω у діапазоні від 0 до $5\omega_{\max}$.

Побудуйте в одному графічному вікні MatLAB графіки амплітудно-частотної (модуля ЧПФ) і фазо-частотної (аргументу ЧПФ) характеристик (значення аргументу відкладати у градусах).

Таблиця 3.4

Варіант	Чисельник	Знаменник
1	$1.82p+67.56$	$p^4+2.65p^3+3.09p^2+7.04p+34.05$
2	$4.61p^2+1.82p+67.56$	$p^4+3.65p^3+45p^2+7.04p+125$
3	$p^2+4p+23$	$p^4+2p^3+39p^2+2p+45$
4	$3p^2+1.82p+67.56$	$p^2+7.04p+34.05$
5	$p+6$	$p^2+0.7p+48$
6	$p^3+4.61p^2+1.82p$	$2.65p^3+3p^2+4p+87$
7	$p^3+4.61p^2+1.82p+67.56$	$p^4+2.65p^3+68p^2+5p+34$
8	$4.61p^2+68$	$p^4+2.65p^3+3.09p^2+7.04p+34.05$
9	7.56	$p^4+2.65p^3+3.09p^2+7.04p+34.05$
10	$p^3+1.8p+7$	$p^4+6.5p^3+39p^2+7p+45$
11	$p^3+4.61p^2+1.82p+67.56$	$p^3+3.09p^2+70p+34$
12	$p^2+1.8p+78$	$2.65p^3+3.09p^2+7.04p+34.05$
13	$p^3+1.82p+67.56$	$p^4+2.6p^3+3p^2+4p+34$
14	$p^3+4.61p^2+1.82p+67.56$	$7p^2+7p+34$
15	$4.61p^2+1.82p+67.56$	$p^2+7.04p+560$
16	$1.82p+67.56$	$3.09p^2+7.04p+34.05$
17	p^3	$3.09p^2+7.8p+125$
18	$1.82p$	$p^3+3.09p^2+7.04p+34.05$
19	$4.61p^2$	$p^2+7.04p+34.05$
20	$p^3+67.56$	$p^4+2.65p^3+3.09p^2+7.04p+34.05$
21	p^3	$p^4+2p^3+3p^2+12p+100$
22	$p^3+4.61p^2+1.82p+67.56$	$p^4+5p^3+30p^2+7p+305$

23	$p^2+1.82p+67.56$	$p^4+2p^3+9p^2+4p+35$
24	$p^3+61p^2+182p+67$	$p^4+3p^3+9p^2+0.04p+39$
25	$p^2+1.82p+67.56$	$p^4+5p^3+20p^2+7p+34$

Завдання 7.

1. Відповідно до таблиці 3.5 виконайте:

- обчислення точних (по стандартних функціях MatLAB) значень відповідної функції в діапазоні змінювання аргументу від x_1 до x_2 в m рівновіддалених точках цього діапазону, включаючи його кінці;

- обчислення по зазначених степеневих рядах наближених значень функції в тих же точках, обмежуючись r першими членами ряду;

- розрахунок похибки наближеного визначення функції в кожній точці, порівнюючи наближене значення з точним, і побудову графіка залежності похибки від аргументу;

- обчислення наближених значень функції в тих же точках із абсолютною похибкою не вище $\varepsilon=0.001$; побудову графіка отриманих абсолютних похибок.

Таблиця 3.5

Вар.	x1	x2	m	r	f(x)	Наближена формула
1	0.2	5	20	4	sin(x)	$\sum_{k=1}^{\infty} (-1)^{k+1} \frac{x^{2k-1}}{(2k-1)!}$
2	1	10	30	5	cos(x)	$1 - \sum_{k=1}^{\infty} (-1)^{k+1} \frac{x^{2k}}{(2k)!}$
3	0.3	3	40	5	exp(x)	$1 + \sum_{k=1}^{\infty} \frac{x^k}{k!}$
4	0.4	4	50	4	ln(1+x)	$\sum_{k=1}^{\infty} (-1)^{k+1} \frac{x^k}{k}$
5	0.6	6	30	3	ln(x)	$\sum_{k=1}^{\infty} \frac{a^k}{k}, \text{ де } a = \frac{x-1}{x}$
6	0.1	1.9	40	4	ln(x)	$\sum_{k=1}^{\infty} (-1)^{k+1} \frac{a^k}{k}, \text{ де } a = x-1$
7	0.7	7	50	5	ln(x)	$2 \sum_{k=1}^{\infty} \frac{a^{2k-1}}{2k-1}, \text{ де } a = \frac{x-1}{x+1}$
8	0.8	8	45	6	ln(x+a)	$\ln(a) + 2 \sum_{k=1}^{\infty} \frac{b^{2k-1}}{2k-1}, \text{ де } b = \frac{x}{2a+x}$
9	1.1	11	40	3	ctg(x)	$\frac{1}{x} + 2x \sum_{k=1}^{\infty} \frac{1}{x^2 - k^2 \pi^2}$

10	1.2	12	50	4	cosec(x)	$\frac{1}{x} + \frac{x}{6} + \frac{2x^2}{\pi^2} \sum_{k=1}^{\infty} \frac{(-1)^k}{k^2(x^2 - k^2\pi^2)}$
11	1.3	13	50	5	cosec(x)	$\frac{1}{x} + 2x \sum_{k=1}^{\infty} \frac{(-1)^k}{x^2 - k^2\pi^2}$
12	1.4	14	60	6	arctg(x)	$\sum_{k=1}^{\infty} (-1)^{k-1} \frac{x^{2k-1}}{2k-1}$
13	1.5	15	45	5	arctg(x)	$\frac{\pi}{2} + \sum_{k=1}^{\infty} (-1)^k \frac{1}{(2k-1)x^{2k-1}}$
14	1.6	16	40	4	ln(x)	$\sum_{k=1}^{\infty} (-1)^{k-1} \frac{a^k}{k}$, де $a = x - 1$
15	0.9	9	50	6	sin(x)	$x \prod_{k=1}^{\infty} \left(1 - \frac{x^2}{(k\pi)^2}\right)$
16	1	10	50	4	cos(x)	$\prod_{k=1}^{\infty} \left(1 - \frac{(2x)^2}{(2k-1)^2\pi^2}\right)$
17	0.6	5	50	3	ln(x)	$\sum_{k=1}^{\infty} \frac{a^k}{k}$, де $a = (x-1)/x$
18	-0.9	0.9	45	4	arcctg(x)	$\frac{\pi}{2} - \sum_{k=1}^{\infty} (-1)^{k-1} \frac{x^{2k-1}}{(2k-1)}$
19	1	20	50	5	sh(x)	$\sum_{k=1}^{\infty} \frac{x^{2k-1}}{(2k-1)!}$
20	1	20	50	5	ch(x)	$1 + \sum_{k=1}^{\infty} \frac{x^{2k}}{(2k)!}$
21	-0.9	0.9	50	5	arth(x)	$\sum_{k=1}^{\infty} \frac{x^{2k-1}}{(2k-1)!}$
22	1	20	50	5	arth(x)	$\sum_{k=1}^{\infty} \frac{1}{(2k-1)! x^{2k-1}}$
23	-0.8	0.8	50	4	arcsin(x)	$x + \sum_{k=1}^{\infty} \frac{1 \cdot 3 \cdot 5 \dots (2k-1) \cdot x^{2k+1}}{2 \cdot 4 \cdot 6 \dots (2k) \cdot (2k+1)}$
24	-0.8	0.8	50	4	arccos(x)	$\frac{\pi}{2} - \left\{ x + \sum_{k=1}^{\infty} \frac{1 \cdot 3 \cdot 5 \dots (2k-1) \cdot x^{2k+1}}{2 \cdot 4 \cdot 6 \dots (2k) \cdot (2k+1)} \right\}$
25	-5	5	50	6	exp(x)	$1 + \sum_{k=1}^{\infty} \frac{x^k}{k!}$

3.7. Контрольні запитання

1. Як подаються дійсні числа при обчисленнях у системі MatLAB?
2. Як змінити формат подання дійсних чисел у командному вікні?
3. Яким чином оголошуються змінні в мові MatLAB?
4. Як зробити так, щоб результат дій, записаних у черговому рядку
 - а) виводився в командне вікно; б) не виводився на екран?
5. Як повернути в командний рядок раніше введену команду ?
6. Як увести значення комплексного числа й у якому виді воно виведеться на екран ?
7. Як мовою MatLAB забезпечити додавання, віднімання, множення, ділення й піднесення до степеня комплексних чисел ?
8. Які функції роботи з комплексними числами є в мові MatLAB ?
9. Що таке "схема Горнера"?
10. Який об'єкт у MatLAB називається поліномом ?
11. Як у MatLAB здійснюється перемноження і ділення поліномів ?
12. За допомогою яких функцій можна знайти корені заданого полінома, значення полінома за відомим значенням аргументу ?
13. Які функції дозволяють відшукати поліном-похідну від заданого полінома ?
14. Що таке степеневий ряд, який відповідає певній функції?
15. Що називають рекурентним співвідношенням?
16. Які степеневі ряди раціонально використовувати при обчисленнях трансцендентних функцій на ЕОМ?
17. Що таке ланцюговий дріб? Як його використовувати при обчисленнях функцій?
18. Як можна відшукати комплексні корені поліному?
19. Що таке передатна функція? частотна передатна функція?
20. Які засоби роботи з комплексними числами є у сучасних мовах програмування?
21. Що таке АЧХ і ФЧХ? як вони обчислюються?

4. Розв'язування систем лінійних алгебричних рівнянь (СЛАР)

Система лінійних алгебричних рівнянь (СЛАР) n -го порядку має вигляд:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \quad \dots \quad \dots \quad \dots = \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases} \quad (4.1)$$

Тут позначено: x_i ($i = 1, 2, \dots, n$) - деякі змінні, a_{ij} ($i, j = 1, 2, \dots, n$) - коефіцієнти при змінних, b_i ($i = 1, 2, \dots, n$) - так звані "вільні" члени.

Під розв'язуванням СЛАР розуміється відшукування таких значень змінних x_i , підстановка яких у кожне з n рівнянь, перетворює їх одночасно у тотожності.

Якщо використати матричні позначення:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}; \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}; \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}, \quad (4.2)$$

то система рівнянь (1) може бути поданою у матричній формі у такий спосіб:

$$A \cdot X = B. \quad (4.3)$$

4.1. Метод Крамера

Найбільш відомий у математиці метод розв'язування СЛАР - метод Крамера - дозволяє побудувати повний аналітичний розв'язок. Цей розв'язок має вигляд

$$x_i = \frac{\det(A_i)}{\det(A)}; \quad (i = 1, 2, \dots, n), \quad (4.4)$$

де $\det(A)$ - позначення визначника матриці A , а $\det(A_i)$ - визначник матриці

$$A_i = \begin{bmatrix} a_{11} & \dots & a_{1(i-1)} & b_1 & a_{1(i+1)} & \dots & a_{1n} \\ a_{21} & \dots & a_{2(i-1)} & b_2 & a_{2(i+1)} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & \dots & a_{n(i-1)} & b_n & a_{n(i+1)} & \dots & a_{nn} \end{bmatrix}, \quad (4.5)$$

яка виходить з матриці A коефіцієнтів шляхом заміни в ній i -го стовпця на стовпець B з вільних членів.

Як бачимо, розв'язування тісно пов'язане з обчисленням визначників. Усього для відшукування повного розв'язку потрібно обчислити $n + 1$ визначни-

ків n -го порядку і додатково здійснити ще n ділень відшуканих значень визначників. Обчислення визначника за відомим правилом розкладання на суму алгебричних доповнень є вельми марнотратним у розумінні кількості операцій, бо потребує $2 \cdot n!$ операцій множення. Усього ж для відшукування розв'язку потрібно, таким чином, $n + 2 \cdot (n + 1)!$ операцій множення-ділення. Наприклад, для розв'язування цим методом системи 3-го порядку потрібно 51, 5-го порядку - 445, 10-го порядку - 79.833.610 операцій множення-ділення. Тому у обчислювальному плані метод Крамера є найбільш невиконаним і недоцільним.

4.2. Метод Гаусса та його модифікації

Усі чисельні методи розв'язування СЛАР спираються на деякі перетворення вихідної системи рівнянь, які, з одного боку, не змінюють шуканого розв'язку, а, з іншого, - значно спрощують подальше його відшукування. Звичайно, при цьому домагаються, щоб загальна кількість операцій разом із попередніми перетвореннями була якомога меншою у порівнянні з методом Крамера.

Розглянемо, які ж саме перетворення рівнянь (1) не змінюють розв'язків цієї системи. Незавжди зрозуміти, що до таких перетворень належать:

- 1) ділення будь-якого з рівнянь на будь-яке число s , що не дорівнює нулю; така операція не змінює розв'язків СЛАР, але змінює визначник матриці, який зменшується у стільки ж разів, тобто визначник початкової матриці A дорівнюватиме значенню визначника нової матриці коефіцієнтів, домноженому на s ;
- 2) додавання до будь-якого рівняння системи (1) будь-якого іншого рівняння тієї ж системи, помноженого на будь-яке число, що не дорівнює нулю;
- 3) переставлення місцями двох яких-небудь рівнянь; ця операція призводить до змінювання знака визначника системи.

Якщо ввести у розгляд так звану "розширену матрицю" коефіцієнтів рівнянь системи

$$A^* = [A \quad B] = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{bmatrix}, \quad (4.6)$$

яка матиме розміри $n \cdot (n + 1)$, то вищезгадані перетворення можна інтерпретувати як перетворення саме розширеної матриці коефіцієнтів СЛАР. До таких допустимих перетворень цієї матриці можна, як це було зазначено, віднести:

- 1) множення будь-якого рядка матриці на число, що не дорівнює нулю;
- 2) сумування будь-якого рядка матриці з іншим рядком цієї матриці, помноженим на число, що не дорівнює нулю;

- 3) перестановку місцями двох будь-яких рядків розширеної матриці; при цьому знак головного визначника матриці A змінюється на протилежний;
- 4) переставляння місцями двох будь-яких стовпців матриці A ; ця операція є еквівалентною лише змінюванню позначень шуканих змінних; при цьому також змінюється знак визначника матриці A .

4.2.1. Схема єдиного ділення

Найпростіший варіант методу Гаусса називається схемою єдиного ділення. Розглянемо його докладніше.

Схема єдиного ділення складається із двох етапів. На першому з них (його називають *прямим ходом*) вихідні рівняння (1) перетворюються таким чином, що з наступних рівнянь вилучаються усі попередні змінні, тобто із другого і подальших рівнянь вилучається змінна x_1 , з третього і подальших - змінна x_2 і так далі. В результаті таких дій виходить, що останнє рівняння міститиме лише одну змінну - x_n , передостаннє - дві змінні - x_n і x_{n-1} і так далі у порядку зростання кількості змінних.

На другому етапі (який називається *зворотним ходом*) визначаються шукані розв'язки СЛАР. Значення змінної x_n визначається безпосередньо з останнього одержаного рівняння, значення x_{n-1} з передостаннього (із врахуванням відшуканого значення x_n) і так далі.

Розглянемо докладніше прямий хід.

Припускаючи, що a_{11} не дорівнює нулю, поділимо на нього перше рівняння (1). Одержимо перше рівняння у вигляді:

$$x_1 + a_{12}^{(1)} \cdot x_2 + a_{13}^{(1)} \cdot x_3 + a_{14}^{(1)} \cdot x_4 + \dots + a_{1n}^{(1)} \cdot x_n = b_1^{(1)},$$

де використане позначення

$$a_{li}^{(1)} = \frac{a_{li}}{a_{11}}, \quad (i = 2, 3, \dots, n); \quad b_1^{(1)} = \frac{b_1}{a_{11}},$$

а індекс угорі позначає номер кроку прямого ходу.

Тепер виключимо із другого рівняння (1) змінну x_1 . Для цього помножимо перетворене перше рівняння (7) на коефіцієнт a_{21} і віднімемо його із другого рівняння (1). Одержимо друге рівняння у виді:

$$a_{22}^{(1)} \cdot x_2 + a_{23}^{(1)} \cdot x_3 + a_{24}^{(1)} \cdot x_4 + \dots + a_{2n}^{(1)} \cdot x_n = b_2^{(1)},$$

де позначено

$$a_{2i}^{(1)} = a_{2i} - a_{21} \cdot a_{1i}^{(1)}, \quad (i = 2, 3, \dots, n); \quad b_2^{(1)} = b_2 - a_{21} \cdot b_1^{(1)}.$$

У такий само спосіб перетворюються усі подальші рівняння. Після цього вони набудуть вигляду (k - номер рівняння):

$$a_{k2}^{(1)} \cdot x_2 + a_{k3}^{(1)} \cdot x_3 + a_{k4}^{(1)} \cdot x_4 + \dots + a_{kn}^{(1)} \cdot x_n = b_k^{(1)},$$

причому

$$a_{ki}^{(1)} = a_{ki} - a_{k1} \cdot a_{1i}^{(1)}, \quad (i = 2, 3, \dots, n); \quad b_k^{(1)} = b_k - a_{k1} \cdot b_1^{(1)}. \quad (4.7)$$

В результаті першого кроку прямого ходу система рівнянь (1) здобуває вигляду (8). При цьому усі рівняння системи, починаючи із другого, матимуть на одну змінну менше за вихідну систему (1), тобто у сукупності утворюють СЛАР $(n-1)$ -го порядку.

$$\begin{cases} x_1 + a_{12}^{(1)} x_2 + \dots + a_{1n}^{(1)} x_n = b_1^{(1)} \\ a_{22}^{(1)} x_2 + \dots + a_{2n}^{(1)} x_n = b_2^{(1)} \\ \dots \quad \dots \quad \dots \quad \dots = \dots \\ a_{n2}^{(1)} x_2 + \dots + a_{nn}^{(1)} x_n = b_n^{(1)} \end{cases} \quad (4.8)$$

Другий крок прямого ходу методу Гаусса полягає у аналогічному перетворенні СЛАР $(n-1)$ -го порядку, яку складають одержані рівняння (8) з другого до останнього. В результаті виходить така система рівнянь:

$$\begin{cases} x_1 + a_{12}^{(1)} x_2 + a_{13}^{(1)} x_3 + \dots + a_{1n}^{(1)} x_n = b_1^{(1)} \\ x_2 + a_{23}^{(2)} x_3 + \dots + a_{2n}^{(2)} x_n = b_2^{(2)} \\ \dots \quad \dots \quad \dots = \dots \\ a_{n3}^{(2)} x_3 + \dots + a_{nn}^{(2)} x_n = b_n^{(2)} \end{cases} \quad (4.9)$$

Коефіцієнти визначаються аналогічно:

$$a_{2i}^{(2)} = \frac{a_{2i}^{(1)}}{a_{22}^{(1)}}, \quad b_2^{(2)} = \frac{b_2^{(1)}}{a_{22}^{(1)}},$$

$$a_{ki}^{(2)} = a_{ki}^{(1)} - a_{k2}^{(1)} \cdot a_{2i}^{(2)}, \quad b_k^{(2)} = b_k^{(1)} - a_{k2}^{(1)} \cdot b_2^{(2)}; \quad (k, i = 3, 4, \dots, n). \quad (4.10)$$

У такий само спосіб здійснюються подальші кроки прямого ходу. Формули перетворення на m -му кроці визначаються співвідношеннями:

$$a_{mi}^{(m)} = \frac{a_{mi}^{(m-1)}}{a_{mm}^{(m-1)}}, \quad b_m^{(m)} = \frac{b_m^{(m-1)}}{a_{mm}^{(m-1)}}, \quad a_{ki}^{(m)} = a_{ki}^{(m-1)} - a_{k2}^{(m-1)} \cdot a_{mi}^{(m)},$$

$$b_k^{(m)} = b_k^{(m-1)} - a_{km}^{(m-1)} \cdot b_m^{(m)}; \quad (k, i = m+1, m+2, \dots, n). \quad (4.11)$$

У підсумку, за $m-1$ крок утворюється така система рівнянь:

$$\begin{cases} x_1 + a_{12}^{(1)} x_2 + a_{13}^{(1)} x_3 + a_{14}^{(1)} x_4 + \dots + a_{1n}^{(1)} x_n = b_1^{(1)} \\ x_2 + a_{23}^{(2)} x_3 + a_{24}^{(2)} x_4 + \dots + a_{2n}^{(2)} x_n = b_2^{(2)} \\ x_3 + a_{34}^{(3)} x_4 + \dots + a_{3n}^{(3)} x_n = b_3^{(3)} \\ \dots \quad \dots \quad \dots = \dots \\ x_n = b_n^{(n)} \end{cases} \quad (4.12)$$

Матриця $A_{(n-1)}$ коефіцієнтів цієї системи є верхньою трикутною матрицею з одиничними елементами вдовж головної діагоналі:

$$A_{(n-1)} = \begin{bmatrix} 1 & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} & \dots & a_{1n}^{(1)} \\ 0 & 1 & a_{23}^{(2)} & a_{24}^{(2)} & \dots & a_{2n}^{(2)} \\ 0 & 0 & 1 & a_{34}^{(3)} & \dots & a_{3n}^{(3)} \\ 0 & 0 & 0 & 1 & \dots & a_{4n}^{(4)} \\ 0 & 0 & 0 & 0 & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 1 \end{bmatrix}. \quad (4.13)$$

Відповідна до цієї системи розширена матриця коефіцієнтів має вигляд:

$$A_{(n-1)}^* = \begin{bmatrix} 1 & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} & \dots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & 1 & a_{23}^{(2)} & a_{24}^{(2)} & \dots & a_{2n}^{(2)} & b_2^{(2)} \\ 0 & 0 & 1 & a_{34}^{(3)} & \dots & a_{3n}^{(3)} & b_3^{(3)} \\ 0 & 0 & 0 & 1 & \dots & a_{4n}^{(4)} & b_4^{(4)} \\ 0 & 0 & 0 & 0 & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 1 & b_n^{(n)} \end{bmatrix}. \quad (4.14)$$

Резюмуючи, можна сказати, що основною метою прямого ходу метода Гаусса є перетворення розширеної матриці системи до трикутної форми (14), після чого відшукання розв'язків СЛАР легко здійснюється зворотним ходом за співвідношеннями:

$$x_n = b_n^{(n)}; \quad x_m = b_m^{(m)} - \sum_{j=m+1}^n a_{mj}^{(m)} \cdot x_j, \quad (m = n-1, n-2, \dots, 1). \quad (4.15)$$

Таким чином прямий хід метода Гауса зводиться до побудови розширеної матриці системи (6) і подальшого її перетворення до верхньої трикутної форми за допомогою таких операцій:

- ділення елементів першого рядка матриці на першій елемент цього рядка (який міститься на головній діагоналі); цей елемент називається роздільним;
- віднімання з подальших рядків матриці першого рядка, помноженого на елемент відповідного рядка, що знаходиться у тому зі стовпців, що й роздільний елемент; обнуління елементів, які містяться у стовпці роздільного елемента;
- повторення цих дій по відношенню до другого рядка, а потім і для усіх подальших рядків нових одержуваних матриць.

4.2.2. Метод Гаусса з обранням роздільного елемента

Неважко углядіти суттєвий недолік розглянутого метода єдиного ділення, який впливає з того, що однією з необхідних операцій є ділення на роздільний елемент. Якщо на головній діагоналі перед цим виявиться, що роздільний елемент дорівнює нулю, відповідна операція стає неможливою. Якщо ж роздільний елемент виявиться досить малою за абсолютним значенням величи-

ною, то хоч операція ділення при цьому є здійсненою, але вона приводить до значної похибки у визначенні подальших коефіцієнтів - членів розширеної матриці. Тому необхідно здійснювати заходи з уникання цього становища.

У методі Гаусса з обранням роздільного елемента цього досягають у такий спосіб. Перед діленням рядка на роздільний елемент порівнюють за модулем усі елементи відповідного стовпця матриці, відшуковують той рядок (із подальших), де міститься елемент, найбільший за модулем, і міняють місцями поточний рядок із знайденим рядком із максимальним елементом. Таким чином домагаються, щоб роздільний елемент був якомога більшим за модулем.

При обміні місцями рядків, визначник матриці A коефіцієнтів змінює свій знак на протилежний. Тому, якщо завданням є обчислення у подальшому визначника цієї матриці, слід лічити кількість p таких обмінів місцями рядків. Тоді, визначник початкової матриці A може бути обчислений як значення визначника перетвореної матриці $A_{(n-1)}$, помножене на $(-1)^p$.

4.2.3. Метод Гаусса-Жордана (метод повного виключення)

Недоліком метода Гаусса є наявність зворотного ходу. Виникає питання, чи не можна позбутися цього етапу, і на першому ж етапі виключення змінних визначити шукані розв'язки СЛАР.

Виявляється, це є можливим завдяки поширенню виключення змінних не тільки на подальші рівняння, а й на попередні. Цей процес і називають повним виключенням, або методом Гаусса-Жордана.

Перший крок цього метода є таким само, як у методі Гаусса. На другому кроці слід ту саму операцію виключення розповсюдити і на перше рівняння, тобто з першого рівняння (9) відняти друге, помножене на $a_{12}^{(1)}$. Це є рівносильним до того, що обчислення за формулами (10) поширюються й на $k = 1$:

$$\begin{aligned} a_{2i}^{(2)} &= \frac{a_{2i}^{(1)}}{a_{22}^{(1)}}, & b_2^{(2)} &= \frac{b_2^{(1)}}{a_{22}^{(1)}}, \\ a_{ki}^{(2)} &= a_{ki}^{(1)} - a_{k2}^{(1)} \cdot a_{2i}^{(2)}, & b_k^{(2)} &= b_k^{(1)} - a_{k2}^{(1)} \cdot b_2^{(2)}; \\ & & & (i = 3, 4, \dots, n; \quad k = 1, 3, 4, \dots, n). \end{aligned} \quad (4.16)$$

Унаслідок цього після другого кроку матимемо замість системи (9) іншу:

$$\left\{ \begin{array}{l} x_1 + 0 + a_{13}^{(2)} x_3 + \dots + a_{1n}^{(2)} x_n = b_1^{(2)} \\ x_2 + a_{23}^{(2)} x_3 + \dots + a_{2n}^{(2)} x_n = b_2^{(2)} \\ \dots \quad \dots \quad \dots = \dots \\ a_{n3}^{(2)} x_3 + \dots + a_{nn}^{(2)} x_n = b_n^{(2)} \end{array} \right. \quad (4.17)$$

Аналогічно, на m -ому кроці треба здійснювати обчислення за формулами (11), поширюючи їх на усі попередні рівняння:

$$a_{mi}^{(m)} = \frac{a_{mi}^{(m-1)}}{a_{mm}^{(m-1)}}, \quad b_m^{(m)} = \frac{b_m^{(m-1)}}{a_{mm}^{(m-1)}},$$

$$a_{ki}^{(m)} = a_{ki}^{(m-1)} - a_{k2}^{(m-1)} \cdot a_{mi}^{(m)}, \quad b_k^{(m)} = b_k^{(m-1)} - a_{km}^{(m-1)} \cdot b_m^{(m)}; \\ (i = m+1, m+2, \dots, n; \quad k = 1, 2, \dots, m-1, m+1, m+2, \dots, n). \quad (4.18)$$

У підсумку виходять "рівняння" виду:

$$\begin{cases} x_1 = b_1^{(n)} \\ x_2 = b_2^{(n)} \\ \dots\dots\dots \\ x_n = b_n^{(n)} \end{cases} \quad (4.19)$$

які вже, очевидно, є просто розв'язками початкової СЛАР.

Неважко зрозуміти, що виграш у вигляді зникнення операцій зворотного ходу у цьому методі відбувається за рахунок збільшення кількості операцій у прямому ході.

4.3. LU-розкладення матриці

У багатьох випадках виникає необхідність розв'язувати низку СЛАР типу (3), матриця коефіцієнтів A яких є єдиною, а вектори B вільних членів - різні, причому окремі вектори B стають відомими не всі одразу, а поступово, або по мірі надходження інформації від джерела, або в міру обчислення значень цього вектора згідно деякого алгоритму. У цьому випадку можливо суттєво зменшити кількість операцій з відшукування розв'язків усіх цих систем ЛАР, якщо ці операції розділити на дві частини, одна з яких не залежить від конкретного значення вектора B .

Припустимо, що матрицю A можна подати у вигляді добутку двох матриць

$$A = L \cdot U, \quad (4.20)$$

де L (від low - нижній) - нижня трикутна матриця, а U (від up - верхній) - верхня трикутна матриця з одиничними діагональними елементами. Тоді рівняння (3) замінюється на

$$L \cdot U \cdot X = B. \quad (4.21)$$

Позначимо

$$U \cdot X = Y. \quad (4.22)$$

Тоді рівняння (21) можна подати у вигляді

$$L \cdot Y = B. \quad (4.23)$$

Розв'язування системи (23) (тобто відшукування вектора Y) є досить простою задачею і не потребує додаткових перетворень, через те, що L - трикутна матриця. А знаючи вектор Y , не складніше відшукати вектор X із рівняння (22).

Таким чином, якщо попередньо зроблено LU-розкладення матриці A коефіцієнтів, відшукування розв'язків системи (3) значно спрощується. Найбільш складне перетворення цієї матриці A робиться один раз. Повторювані ж дії з послідовного розв'язування СЛАР (23) і (22) є досить простими.

Для відшукування елементів l_{ik} ($i \geq k$) матриці L й елементів u_{kj} матриці U можна використовувати наступні співвідношення:

$$l_{i1} = a_{i1}; \quad l_{ik} = a_{ik} - \sum_{p=1}^{k-1} l_{ip} \cdot u_{pk}, \quad (k = 2, 3, \dots, n; i = k, k+1, \dots, n);$$

$$l_{kk} = 1 / l_{kk};$$

$$u_{kj} = l_{kk} (a_{kj} - \sum_{p=1}^{k-1} l_{kp} \cdot u_{pj}), \quad (j = k+1, \dots, n); \quad u_{kk} = 1.$$
(4.24)

4.4. Обчислення визначника й оберненої матриці

У прямому ході методу Гаусса над елементами матриці A здійснюються елементарні перетворення, які не змінюють значення визначника цієї матриці, окрім операції ділення на роздільний елемент, яка відповідно зменшує значення визначника одержуваної матриці у порівнянні з початковою.

Матриця A внаслідок таких перетворень стає верхньою трикутною матрицею з одиничними елементами. Її визначник дорівнює одиниці. Якщо у прямому ході рядки матриці A не переставлялися, то знак визначника не змінювався. Таким чином, значення визначника невиродженої матриці дорівнює добуткові роздільних елементів у прямому ході виключення Гаусса:

$$\det(A) = \Delta = a_{11} \cdot a_{22}^{(1)} \cdot a_{33}^{(2)} \cdot \dots \cdot a_{nn}^{(n-1)}. \quad (4.25)$$

Якщо ж застосований метод Гаусса з обранням провідного роздільного елемента, то значення (25) слід помножити на $(-1)^s$, де s - кількість перестановок місцями рядків і стовпців матриці A при її перетворенні до трикутної форми:

$$\det(A) = \Delta = (-1)^s \cdot a_{11} \cdot a_{22}^{(1)} \cdot a_{33}^{(2)} \cdot \dots \cdot a_{nn}^{(n-1)}. \quad (4.26)$$

Усі схеми методу Гаусса дозволяють здійснювати одночасно розв'язування кількох СЛАР із різними правими частинами (векторами вільних членів) і з єдиною матрицею A коефіцієнтів, якщо усі вектори вільних членів об'єднати у єдину матрицю B :

$$A \cdot X = B. \quad (4.27)$$

Якщо ця об'єднана матриця B має розміри $(n * m)$, де n - кількість рівнянь СЛАР, а m позначає кількість стовпців матриці B , то й результатом розв'язання системи (27) буде вже не вектор, а матриця X , що складається з такої ж кількості m стовпців. Причому кожний зі стовпців є розв'язком СЛАР, що відповідає такому ж стовпцю матриці B . Якщо матриця B є квадратною, результат X теж буде квадратною матрицею.

А тепер припустимо, що матриця B є одиничною. Тоді рівняння (27) набуде вигляду

$$A \cdot X = E. \quad (4.28)$$

Однак, як відомо, матриця X , така, що, якщо її помножити на задану матрицю A , дає у результаті одиничну, називається згідно визначення матрицею,

оберненою до заданої. З цього випливає висновок: щоб відшукати (обчислити) матрицю X , обернену до заданої матриці A , досить розв'язати матричне рівняння (28).

Звідси випливає такий спосіб обчислення оберненої матриці:

- 1) формуємо розширену матрицю, додаючи до заданої матриці справа одиничну матрицю того самого розміру:

$$A^* = [A \quad E] = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & 1 & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & a_{2n} & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} & 0 & 0 & \dots & 1 \end{bmatrix};$$

- 2) перетворюємо цю розширену матрицю згідно методу повного виключення Гаусса-Жордана; в результаті одержимо матрицю:

$$A_{(n-1)}^* = \begin{bmatrix} 1 & 0 & \dots & 0 & x_{11} & x_{11} & \dots & x_{11} \\ 0 & 1 & \dots & 0 & x_{11} & x_{11} & \dots & x_{11} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & x_{11} & x_{11} & \dots & x_{11} \end{bmatrix};$$

- 3) вилучаємо з одержаної матриці її праву половину - вона й є оберненою матрицею.

Обчислення оберненої матриці є доцільним і раціональним у тих самих випадках, коли застосовується LU-розкладення матриці.

Дійсно, помножуючи зліва матричне рівняння (3) на матрицю A^{-1} , обернену до матриці A , одержимо:

$$A^{-1} \cdot A \cdot X = A^{-1} \cdot B \Rightarrow E \cdot X = A^{-1} \cdot B \Rightarrow X = A^{-1} \cdot B. \quad (4.28)$$

Отже, якщо необхідно розв'язувати СЛАР виду (3) неодноразово, з тією самою матрицею A коефіцієнтів, але кожного разу з іншим вектором B вільних членів, доцільно один раз обчислити зворотну матрицю коефіцієнтів A^{-1} , а потім лише кожного разу домножати її на новий вектор B .

4.5. Розв'язування СЛАР на ЕОМ. Матричні операції на ЕОМ

Розв'язування СЛАР на ЕОМ потребує наявності ряду досить складних процедур з оперування й перетворення матриць. У більшості універсальних мов програмування такі процедури відсутні. Точніше, вони створені математиками й програмістами і знаходяться у спеціальних бібліотеках процедур, розроблених на відповідних мовах. Але зазвичай такі бібліотеки є недосяжними для пересічного користувача.

Тому, якщо у вашому розпорядженні є такі мови як Pascal, Delphi або Basic, то для розв'язування СЛАР потрібно утворити цілу низку підпрограм, які б реалізували основні операції з матрицями, починаючи з уведення, формування й виведення матриць і закінчуючи процедурами обчислення визначника матри-

ці і зворотної матриці. Таку серію процедур зручніше за все розробляти мовою Pascal, яка дозволяє утворювати нові обчислювальні об'єкти на основі використання даних типу запису (*record*).

Приємним виключенням є система MatLAB. В ній уже передбачені усі необхідні процедури й функції матричної математики. Особливо зручним є те, що головні дії з матрицями здійснюються на мові MatLAB за допомогою відомих звичних знаків арифметичних дій

'+' - для підсумовування матриць;

'-' - для віднімання матриць;

'*' - для множення матриць;

'^' - для піднесення квадратної матриці до цілого (додатного або від'ємного) степеня.

Транспонування матриць (заміна рядків і стовпців місцями) здійснюється у MatLAB за допомогою знаку апострофа, розміщеного одразу після ймення тієї матриці, яку потрібно транспонувати.

Найбільш цікавими для нас є дві матричні операції, яких немає у матричній алгебрі - ділення матриць зліва направо (A/B) і справа наліво ($A \setminus B$). Перша з них є рівносильною до сукупності операцій $A \cdot B^{-1}$, друга - $A^{-1} \cdot B$. Саме ці операції дозволяють відшукати розв'язки СЛАР.

Якщо задана СЛАР має вигляд (3)

$$A \cdot X = B,$$

то його розв'язок у матричній формі матиме вигляд (28), що можна записати мовою MatLAB у такий спосіб

$$X = A \setminus B. \quad (4.29)$$

Якщо ж СЛАР записана у наступній матричній формі

$$X \cdot A = B,$$

то розв'язок її у MatLAB відшукується записом команди

$$X = A / B. \quad (4.30)$$

Приклад. Застосуємо засоби системи MatLAB для розв'язку СЛАР:

$$\begin{cases} x_1 + x_2 - x_3 = 6 \\ -2x_1 + 3x_2 = 4 \\ 4x_1 + 15x_2 + 2x_3 = 28 \end{cases} \quad (*)$$

Якщо, знаходячись у командному вікні MatLAB, ввести значення елементів матриці коефіцієнтів командою

$$A = [1 \ 1 \ -1; -2 \ 3 \ 0; 4 \ 15 \ 2]$$

то одержимо у тому ж вікні

$$A = \begin{matrix} 1 & 1 & -1 \\ -2 & 3 & 0 \\ 4 & 15 & 2 \end{matrix}$$

Аналогічно введемо вектор-стовпець вільних членів

$$B = [6 \ 4 \ 28]'$$

Одержимо

$$B =$$

```
6
4
28
```

Тепер можна застосувати команду (29) для відшукування розв'язків системи (*)

```
X=A\B
```

```
У вікні виникне відповідь
```

```
X =
 1.0000
 2.0000
-3.0000
```

Для перевірки, помножимо матрицю A на одержаний вектор X розв'язків

```
A*X
```

```
ans =
 6.0000
 4.0000
28.0000
```

Як бачимо, ми одержали вектор вільних членів, що підтверджує слушність проведених обчислень.

Функція *обернення матриці* - $\text{inv}(A)$ – обчислює матрицю, обернену до заданої матриці A . Початкова матриця A має бути квадратною, а її визначник не повинен дорівнювати нулеві.

Для прикладу знайдемо розв'язки тієї ж СЛАР іншим способом:

```
C = inv(A)
```

```
C =
 0.1154 -0.3269 0.0577
 0.0769 0.1154 0.0385
-0.8077 -0.2115 0.0962
```

Перевіримо слушність виконання операції обернення, застосовуючи її ще раз до отриманого результату:

```
inv(C)
```

```
ans =
 1.0000 1.0000 -1.0000
-2.0000 3.0000 0.0000
 4.0000 15.0000 2.0000
```

Знайдемо розв'язки СЛАР за допомогою співвідношення (28)

```
C*B
```

```
ans =
 1.0000
 2.0000
-3.0000
```

Розглянуті методи є точними. Точні (або прямі) методи працюють достатньо швидко і широко застосовуються на практиці, якщо є достатніми обсяги пам'яті для їхньої реалізації.

Але існують й інші шляхи відшукування коренів СЛАР, які відносяться до наближених. Наприклад, різні модифікації методу ітерацій. Вони знаходять за скінченну кількість кроків (ітерацій) лише наближені розв'язки із заданою припустимою відносною похибкою.

Метод ітерацій взагалі полягає у тому, що попередньо первісне рівняння $f(x) = 0$ перетворюється до виду

$$x = \varphi(x). \quad (4.30)$$

З області ізоляції $[a, b]$ шуканого кореня обирається деяке певне значення x_0 аргументу, яке приймається за початкове наближення кореня. Наближені значення кореня у наступних наближеннях визначаються з співвідношення

$$x_k = \varphi(x_{k-1}); \quad (k = 1, 2, \dots, n), \quad (4.31)$$

при цьому k має зміст номера ітерації (наближення).

Очевидно, ітераційний процес може приводити до послідовності значень, які наближаються до деякого значення аргумента (шуканого кореня), і тоді цей процес є стійким. Але зазначений процес може приводити до послідовності значень аргумента, які все далі віддаляються від первісного значення, тобто бути нестійким. Стійкість або нестійкість ітераційного процесу суттєво залежить від виду залежності $\varphi(x)$, яка, як неважко впевнитися визначається неоднозначно, і тому можна підібрати з можливих її варіантів і такі, які забезпечують стійкість ітерацій. Саме це й є найбільш складним у побудові ітераційного процесу.

Методи ітерацій можуть бути застосовані і для відшукування коренів СЛАР, якщо попередньо систему (1) подати у виді

$$\begin{cases} x_1 = \varphi_1(x_1, x_2, \dots, x_n) \\ x_2 = \varphi_2(x_1, x_2, \dots, x_n) \\ \dots\dots\dots \\ x_n = \varphi_n(x_1, x_2, \dots, x_n) \end{cases} \quad (4.32)$$

MatLAB має кілька вбудованих процедур, які дозволяють розв'язувати систему лінійних алгебричних рівнянь виду (3) наближеними ітераційними методами. Їх застосовують при обчисленнях тоді, коли матриці коефіцієнтів СЛАР є розрідженими і великими за розмірами. До них відносяться:

bicg – метод біспряжених градієнтів;

bicgstab – стабілізований метод біспряжених градієнтів;

cgs – квадратичний метод спряжених градієнтів;

gmres – узагальнений метод мінімального відхилення;

qmr – метод квазімінімального відхилення;

pcg – передобумовлений метод спряжених градієнтів (застосовується лише для симетричних матриць A).

Загальне звернення до цих процедур має вигляд

$$x = \mathbf{bicg}(A, B, \text{tol}, \text{maxit}).$$

Тут A – квадратна матриця розміром $(n \times n)$ коефіцієнтів при аргументах системи рівнянь (1), B – матриця-стовпець розміром $(n \times 1)$ вільних членів, tol – припустима межа відносна похибка визначення коренів, maxit – межа припустима кількість ітерацій, x – вектор одержаних наближених значень коренів рівняння (1).

За початкове наближення обирається вектор x_0 з нульових елементів.

Наведемо приклад. Введемо наступну послідовність операторів:

```
A=[1 1 2 3;3 -1 -1 -2;2 3 -1 -1; 1 2 3 -1]
```

```
B=[1; -4; -6; -4]
```

```
X=A\B;
```

```
X1=bicg(A,B);
```

```
X2=bicgstab(A,B);
```

```
X3=cgs(A,B);
```

```
X4=gmres(A,B);
```

```
X5=qmr(A,B);
```

```
disp(' ')
```

```
disp(' ')
```

```
disp([' Точно ';' BICG ';' BICGSTAB ';' CGs ';' GMRES ';' QMR '])
```

```
Y=[X X1 X2 X3 X4 X5]
```

```
disp(' ')
```

```
disp(' ПОХИБКИ')
```

```
disp([' BICG ';' BICGSTAB ';' CGs ';' GMRES ';' QMR '])
```

```
dY=[X1-X X2-X X3-X X4-X X5-X]
```

Виконавши їх, система MATLAB виведе результат у наступному вигляді:

```
A =
```

```
 1  1  2  3
 3 -1 -1 -2
 2  3 -1 -1
 1  2  3 -1
```

```
B =
```

```
 1
 -4
 -6
 -4
```

```
 Точно      BICG      BICGSTAB      CGS      GMRES      QMR
Y =
```

```
-1.0000e+000 -1.0000e+000 -1.0000e+000 -1.0000e+000 -1.0000e+000 -1.0000e+000
-1.0000e+000 -1.0000e+000 -1.0000e+000 -1.0000e+000 -1.0000e+000 -1.0000e+000
 0          3.5638e-014  3.8956e-011 -5.8953e-014 -1.6653e-016 -4.1078e-015
1.0000e+000 1.0000e+000 1.0000e+000 1.0000e+000 1.0000e+000 1.0000e+000
```

```
 ПОХИБКИ
 BICG      BICGSTAB      CGs      GMRES      QMR
dY =
 1.0436e-014 -4.6485e-012 -7.4163e-014 2.2204e-016 -3.1530e-014
-1.7319e-014 2.6083e-011 2.5535e-015 4.4409e-016 8.5487e-015
 3.5638e-014 3.8956e-011 -5.8953e-014 -1.6653e-016 -4.1078e-015
 4.8406e-014 2.0103e-011 -2.8422e-014 0 -2.0428e-014
```

Окрім вищезазначених, у MatLAB передбачені ще такі матричні функції й операції.

Операція *транспонування матриці* здійснюється за допомогою знака апострофа, записаного після ймення матриці, наприклад:

```
X=[1 2 3; 4 5 6]
```

```
X =
```

```
 1  2  3
 4  5  6
```

```
X'
```

```
ans =
```

```

1 4
2 5
3 6

```

Піднесення матриці до цілого степеня здійснюється у MatLAB за допомогою знаку " \wedge ": A^n . При цьому матриця має бути квадратною, а n цілим числом (додатним або від'ємним). Ця матрична дія є еквівалентною до множення матриці A на себе n разів (якщо n - додатне) або множенню оберненої матриці на себе (при n від'ємному). Наприклад:

```

D=A^2
D =
-5 -11 -3
-8 7 2
-18 79 0

```

Функція $\text{sqrtm}(A)$ обчислює таку матрицю Y , що $Y*Y = A$:

```

sqrtm(D)
ans =
1.0000 1.0000 -1.0000
-2.0000 3.0000 -0.0000
4.0000 15.0000 2.0000

```

Обчислення **матричної експоненти** (e^A) здійснюється за допомогою функцій expm , expm1 , expm2 expm3 . Ці функції слід відрізнити від раніше згадуваної функції $\text{exp}(A)$, яка формує матрицю, кожний елемент якої дорівнює e у степені, що дорівнює відповідному елементу матриці A .

Поняття матричної експоненти можна ввести наступним чином. Відомо, що розклад експоненти від деякого аргументу x у ряд Маклорена має вигляд:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} + \dots$$

За аналогією матричною експонентою називають функцію, яка визначається відповідним матричним розкладом

$$e^A = E + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \dots + \frac{A^n}{n!} + \dots,$$

де A - квадратна матриця, а E - одинична матриця тих самих розмірів.

Функція $V = \text{expm}(A)$ є вбудованою функцією MatLAB. Функція $V = \text{expm1}(A)$ є М-файлом, який обчислює матричну експоненту шляхом використання розкладення Паде матриці A . Функція $V = \text{expm2}(A)$ обчислює матричну експоненту, використовуючи розкладення Тейлора матриці A . Нарешті, функція $V = \text{expm3}(A)$ здійснює ту саму операцію на основі використання спектрального розкладу матриці A .

Функція $V = \text{logm}(A)$ робить зворотну операцію – відшукує таку матрицю V , матрична експонента від якої дорівнює заданій матриці A . Наведемо приклади:

```

A = [1 1 -1; 2 3 0; 4 15 2]
A =
1 1 -1
2 3 0
4 15 2
K=expm(A)
K =

```

```
-18.3503 -30.1580  2.3384
-0.7025 -11.1043 -3.9742
50.2602 32.4867 -24.6629
```

F=logm(K)

F =

```
1.0000  1.0000 -1.0000
2.0000  3.0000  0.0000
4.0000 15.0000  2.0000
```

Ми одержали значення початкової матриці, що є свідченням слушності обчислень за обома функціями.

Розглянемо також деякі процедури лінійної алгебри, передбачені у системі MatLAB.

За традицією до лінійної алгебри відносять такі задачі, як обернення та псевдообернення матриць, спектральне та сингулярне розкладення матриці, обчислення власних значень і векторів, сингулярних чисел матриць, обчислення функцій від матриць.

Визначення характеристичного полінома квадратної матриці A розміром $n \times n$ можна здійснити за допомогою процедури *poly(A)*. Звернення до неї вигляду

$$p = \text{poly}(A)$$

дає змогу побудувати вектор-рядок $p = [p_1 \ p_2 \ \dots \ p_{n+1}]$ коефіцієнтів характеристичного полінома

$$p(x) = \det(s \cdot E - A) = p_1 s^n + p_2 s^{n-1} + \dots + p_n s + p_{n+1},$$

де E - позначення для одиничної матриці розміром $n \times n$.

Обчислення власних значень і власних векторів матриці здійснює процедура *eig(A)*. Звичне звернення до неї дає змогу одержати вектор *власних значень матриці* A , тобто *коренів характеристичного полінома матриці*. Якщо ж звернення має вигляд

$$[R,D] = \text{eig}(A),$$

то в результаті одержують діагональну матрицю D власних значень і матрицю R правих власних векторів, які задовольняють умову

$$A \cdot R = R \cdot D.$$

Ці вектори знормовані у такий спосіб, що норма кожного з них дорівнює одиниці.

Сингулярне розкладення матриці здійснює процедура *svd(A)*. Спрощене звернення до неї дає змогу одержати *сингулярні числа матриці* A , під якими розуміють *невід'ємні значення квадратних коренів з власних значень матриці* $A^T \cdot A$. Більш складне звернення вигляду

$$[U,S,V] = \text{svd}(A)$$

дозволяє одержати три матриці – U , яку сформовано з ортонормованих власних векторів, які відповідають найбільшим власним значенням матриці $A \cdot A^T$, V – з ортонормованих власних векторів матриці $A^T \cdot A$ і S – діагональну матрицю з сингулярних чисел матриці A . Ці матриці задовольняють співвідношення

$$A = U * S * V^T.$$

Обчислення **власних значень матричного полінома** здійснює процедура ***polyeig***(A0,A1,...,Ap). Звернення до неї

$$[R,d] = \mathit{polyeig}(A_0,A_1,\dots,A_p)$$

дозволяє вирішити проблему власних значень для матричного полінома степеня p вигляду $A_0 + A_1\lambda + A_2\lambda^2 + \dots + A_p\lambda^p$. Вхідними величинами цієї процедури є квадратні матриці A_0, A_1, \dots, A_p порядку n . Вихідними величинами – матриця R власних векторів розміром $n \times (n \cdot p)$ і вектор d власних значень довжиною $n \cdot p$.

Функція ***polyvalm***(p,X) призначена для **обчислення значення матричного полінома** вигляду

$$Y(X) = p_n X^n + p_{n-1} X^{n-1} + \dots + p_2 X^2 + p_1 X + p_0 E$$

за заданого значення матриці X і вектора $p = [p_n, p_{n-1}, \dots, p_0]$ коефіцієнтів полінома.

Слід розрізняти процедури ***polyval*** і ***polyvalm***. Перша обчислює значення полінома для кожного з елементів матриці аргументу, а друга при обчисленні полінома підносить до відповідного степеня усю матрицю-аргумент.

Система MatLAB містить ще багато корисних процедур, що дозволяють здійснювати математичну обробку її елементів як масиву даних.

Процедура ***size***(A) призначена для визначення кількості рядків і стовпців матриці A. Вона формує вектор [n,p], де n - кількість рядків матриці A, а p - кількість її стовпців.

В результаті застосування процедур ***max***(A), ***min***(A), ***mean***(A) та ***std***(A) одержуються вектори-рядки, кількість елементів яких дорівнює кількості стовпців матриці A. Кожний елемент цього рядка містить, відповідно, максимальний, мінімальний елемент відповідного стовпця матриці A, середньоарифметичне значення елементів цього стовпця або середньоквадратичне відхилення елементів стовпця від нього.

Функції ***sum***(A) і ***prod***(A) формують вектор-рядок, кожний елемент якого є сумою або добутком елементів відповідного стовпця матриці A.

Процедури ***cumsum***(A) і ***cumprod***(A) утворюють матриці того самого розміру, що й матриця A, елементи кожного стовпця яких складають суми або добутки елементів відповідного стовпця матриці A, які розташовані вище за відповідний елемент, включаючи і сам цей елемент.

Функція ***diff***(A) утворює з матриці A розміром $m \times n$ матрицю розміром $(m-1) \times n$, елементи якої дорівнюють різниці між суміжними рядками матриці A.

Процедура ***diag***(x) формує або "витагує" діагональ матриці. Якщо x - вектор, то утворюється квадратна матриця з вектором x на головній діагоналі. Якщо ж x є матрицею, утворюється вектор-стовпець, що складається з елементів головної діагоналі заданої матриці x .

Функція $k=norm(A,p)$ обчислює p -норму матриці A за формулою

$$k=\max(\sum(abs(A).^p))^{(1/p)},$$

де p – ціле додатне число або inf або $'fro'$. Якщо аргумент p не вказаний, обчислюється 2-норма. При цьому є слушними співвідношення:

$$norm(A,1)=\max(\sum(abs(A)))$$

$$norm(A,inf)=\max(\sum(abs(A')))$$

$$norm(A,'fro')=\sqrt{\sum(diag(A'*A))}$$

$$norm(A)=norm(A,2)=\sigma_{\max}(A).$$

Функція $r=rank(A)$ обчислює ранг матриці, який визначається як кількість сингулярних чисел матриці, що перевищують поріг $\max(size(A))*norm(A)*eps$. Тут eps ($2,2204 \cdot 10^{-16}$) - похибка операцій над числами з плаваючою комою.

Процедура $k=cond(A)$ обчислює число обумовленості матриці A по відношенню до операції обернення. Воно дорівнює відношенню максимального сингулярного числа матриці до мінімального.

$k=rcond(A)$ обчислює величину, зворотну до значення числа обумовленості матриці A відносно її 1-норми. Якщо матриця A є добре обумовленою, значення k близько до одиниці, якщо ж вона є погано обумовленою, k наближається до нуля.

Наостанок розглянемо вельми корисні вбудовані функції Matlab, призначені для формування матриць певного виду, а також деякі функції, що дозволяють здійснювати перетворення матриць, не передбачені матричною алгеброю.

Наступні функції забезпечують формування матриць з певних елементів:

$zeros(m,n)$ утворює матрицю розміром $(m \times n)$ з усіма елементами, рівними нулю;

$ones(m,n)$ утворює матрицю розміром $(m \times n)$ з усіма елементами, рівними одиниці;

$eye(m,n)$ утворює одиничну матрицю розміром $(m \times n)$, тобто таку, що елементи вдовж головної діагоналі дорівнюють одиниці, а решта елементів дорівнюють нулю;

$rand(m,n)$ утворює матрицю розміром $(m \times n)$ з елементами, що приймають випадкові значення, які рівномірно розподілені у діапазоні від 0 до 1;

$randn(m,n)$ утворює матрицю розміром $(m \times n)$ з елементами, що приймають випадкові значення, розподілені за нормальним (гауссовим) законом з математичним сподіванням, рівним нулю і з середньоквадратичним відхиленням, рівним одиниці.

Для поелементного перетворення будь-якої заданої матриці придатні усі вказані раніше елементарні математичні функції, передбачені в Matlab. Кожна така функція, в якій як аргумент використана задана матриця, утворює матрицю тих самих розмірів, кожен елемент якої обчислений як вказана функція від відповідного елемента матриці-аргумента.

Окрім того, з матрицями однакових розмірів можна здійснювати поелементне множення, ділення і піднесення до степеня, використовуючи ті самі знаки арифметичних дій. Для цього до звичайних знаків множення, ділення і піднесення до степеня слід додати зліва знак крапки.

Так, запис $A.*B$ забезпечує утворення матриці тих самих розмірів, що й кожна з матриць A і B , кожний елемент якої дорівнює добуткові відповідних елементів матриць A і B . Аналогічно, $A./B$ утворює матрицю, кожний елемент якої дорівнює відповідному елементу матриці A , поділеному на відповідний елемент матриці B . Якщо ж у командному вікні Matlab записано $A.\backslash B$, то виходить матриця, кожний елемент якої дорівнює відповідному елементу матриці B , поділеному на відповідний елемент матриці A . Нарешті запис $A.^B$ приведе до утворення матриці, в якій кожний елемент дорівнює відповідному елементу матриці A , піднесеного до степеня, що дорівнює відповідному елементу матриці B .

4.6. Завдання

Завдання 8.

1. Уведіть довільну матрицю розміром $(4*6)$. Відшукайте суму найбільших елементів її рядків.
2. Уведіть квадратну матрицю $(5*5)$ із єдиним найменшим елементом. Відшукайте суму елементів рядка, у якому розміщений елемент із найменшим значенням.
3. Уведіть матрицю $(6*9)$, у якій є єдині найбільший і найменший елементи і вони розташовані у різних рядках. Поміняйте місцями рядки з найбільшим і найменшим елементами.
4. Уведіть матрицю $(5*6)$ із різними значеннями елементів. У кожному рядку виберіть найменший елемент, з отриманих чисел виберіть найбільше. Знайдіть індекси отриманого елемента.
5. Уведіть матрицю $(5*6)$. Побудуйте вектор, елементами якого є найбільші елементи відповідних рядків матриці.
6. Уведіть матрицю $(5*6)$. Побудуйте вектор, елементами якого є суми найбільшого й найменшого елементів відповідних рядків матриці.
7. Уведіть матрицю $(5*6)$. Побудуйте вектор, елементами якого є середні значення елементів відповідних рядків матриці.
8. Уведіть матрицю $(5*6)$. Побудуйте вектор, елементами якого є середньоквадратичні відхилення елементів відповідних рядків матриці від їхнього середнього значення.
9. Уведіть матрицю $(5*6)$. Побудуйте вектор, елементами якого є середні арифметичні значення найбільшого й найменшого елементів відповідних рядків матриці.

10. Уведіть матрицю (6*5). Побудуйте вектор, елементами якого є суми квадратів елементів відповідних стовпців матриці.

11. Уведіть матрицю (5*5). Побудуйте вектор, елементами якого є суми квадратів елементів відповідних стовпців матриці.

12. Уведіть матрицю (5*6). Відшукajte середнє арифметичне найбільшого й найменшого її елементів.

13. Уведіть матрицю (5*5). Побудуйте вектор, елементами якого є елементи головної діагоналі матриці. Відшукajte слід матриці.

14. Уведіть дві матриці (4*4). Побудуйте нову матрицю розміром (4*8), включаючи у перші 4 стовпчика рядки першої матриці, а в інші – рядки другої матриці.

15. Відшукajte суму усіх елементів матриці розміром (4*3).

Завдання 9. Уведіть довільну матрицю розміром (5*5). Відшукajte:

- 1) визначник матриці; якщо визначник дорівнює нулеві, або занадто малий, змініть деякі елементи матриці і повторіть обчислення;
- 2) обернену матрицю; перевірте слушність шляхом обернення отриманої матриці;
- 3) характеристичний поліном матриці;
- 4) корені характеристичного рівняння матриці; упорядкуйте корені по комплексно-спряжених парах за величиною;
- 5) власні значення матриці; порівняйте з раніше відшуканими коренями характеристичного рівняння;
- 6) LU-розкладення матриці; перевірте його слушність;
- 7) QR-розкладення матриці; перевірте його слушність;
- 8) сингулярні числа матриці;
- 9) слід матриці;
- 10) число обумовленості матриці;
- 11) експоненту від матриці;
- 12) логарифм від експоненти матриці; порівняйте з первісною матрицею.

Завдання 10. Відшукajte розв'язки системи лінійних рівнянь, заданої у таблицю 4.1:

- 1) одним з методів точного розв'язування;
 - 2) методами ітерації.
- Порівняйте результати.

Таблиця 4.1

Вар.	А			В	Вар.	А			В	Вар.	А			В
1	4	0.86	0.3	3	10	5	-1.2	1	2	19	4	0.12	1.8	-1
	0.05	2	0.14	4		0.23	3	-0.08	-3		0.31	4	0.19	2
	0.07	0.5	-3	-2		0.92	-0.33	2	1		0.4	-1	3	5

2	2 0.3 0.04	-0.1 3 0.1	0.3 -0.01 -4	-7 4 1	11	7 1.3 0.1	0.52 -4 0.02	-0.08 0.2 -2	2 -1 5	20	2.1 1.8 0.2	0.3 -7 1.2	0.2 0.1 -3	2 3 -1
3	1 0.02 0.4	0.02 -4 1	-0.03 -0.01 3	2 4 1	12	4 0.01 0.08	-0.3 2 -0.4	-0.8 -0.5 -3	3 -2 4	21	8 0.1 0.6	0.2 -2 1.3	-1 0.5 3	-2 3 -4
4	3 0.28 0.06	-0.01 4 -0.8	0.5 0.03 -2	-2 3 0.1	13	2 0.03 0.02	0.3 -4 0.61	-0.1 0.01 -7	7 -4 1	22	8 0.83 0.08	-0.02 3 -0.21	1.1 -1.65 -4	8 2 7
5	5 0.04 0.12	2 -3 0.28	0.08 0.02 -1	4 -3 7	14	5 0.4 0.15	0.35 5 0.24	-0.18 -0.1 1	-1 3 -4	23	8 0.15 0.21	0.27 -2 0.07	-0.32 0.5 -1	0.2 2 -3
6	7 0.03 0.12	-0.04 -2 0.28	-0.8 0.02 -1	3 -2 7	15	8 0.84 0.03	-0.35 3 -0.21	1 -1.21 -4	8 2 7	24	9 0.4 0.7	1.1 -1 -0.2	-0.2 0.1 3	8.5 -2 -1
7	3 0.43 0.17	0.1 -5 0.35	0.18 2 -2	-1 4 3	16	2 0.03 0.06	0.03 -2 0.17	-0.15 0.07 -1	2 -4 8	25	1 0.9 0.2	0.4 9 0.7	0.1 -0.1 2	0.5 -2 3
8	2 1.2 0.17	-0.17 6 0.41	0.41 -0.1 -3	1 -2 4	17	4 0.81 1.05	1.8 -3 0.43	-0.2 0.2 -4	1 -3 2					
9	3 0.01 0.7	-0.18 2 0.02	0.55 -0.5 -3	7 -3 -2	18	1.27 0.17 0.11	-0.1 -6 0.2	-0.2 0.2 7	2 -3 4					

4.7. Контрольні запитання

1. Що називають СЛАР? Що означає "розв'язати СЛАР"?
2. У чому суть метода Крамера? Які його недоліки й переваги у порівнянні з іншими методами розв'язування СЛАР? Коли вони виявляються?
3. У чому сутність методу Гаусса розв'язування СЛАР? Які його модифікації вам відомі? Чим вони відрізняються? Які їхні переваги й недоліки відносно інших методів?
4. Як найраціональніше побудувати процес обчислення визначника матриці високого порядку?
5. Як найраціональніше побудувати процес відшукування оберненої матриці високого порядку?
6. Яким чином найпростіше розв'язати СЛАР у системі MatLAB?

7. Коли раціонально для розв'язування СЛАР використовувати LU-розкладання матриці? обернену матрицю?
8. Як вводяться вектори у мові MatLAB? Якими функціями можна формувати їх у мові MatLAB?
9. Які функції MatLAB дозволяють перетворювати вектор поелементно?
10. За допомогою яких засобів у MatLAB здійснюються основні операції з векторами?
11. Як вводяться матриці у системі MatLAB ?
12. Які функції є у MatLAB задля формування матриць визначеного виду?
13. Як сформувати матрицю: а) по заданих векторах її рядків? б) по заданих векторах її стовпців? в) по заданих векторах її діагоналей?
14. Які функції поелементного перетворення матриці є у MatLAB?
15. Як здійснюються у MatLAB звичайні матричні операції?
16. Як розв'язати у MatLAB систему лінійних алгебричних рівнянь?
17. Як відшукати характеристичний поліном матриці ?
18. Як обчислити у MatLAB визначник матриці?
19. Як обчислити у MatLAB обернену матрицю?
20. Що таке "псевдообернення" матриці і як воно здійснюється у MatLAB?
21. Що таке "власні числа" і "власні вектори" матриці і як вони визначаються у MatLAB?
22. Що таке "сингулярні числа" матриці і як вони визначаються у MatLAB?
23. Які функції від матриці передбачені у MatLAB?
24. Що таке "матрична експонента"? "корінь квадратний з матриці"? "логарифм від матриці"? Як їх відшукати у системі MatLAB?

5. Розв'язування нелінійних алгебричних рівнянь

Нелінійним рівнянням називають алгебричне рівняння виду

$$f(x) = 0, \quad (5.1)$$

де x - аргумент, а функція

$$y = f(x) \quad (5.2)$$

є деякою довільною неперервною алгебричною функцією цього аргументу.

Під розв'язком нелінійного рівняння (нулем відповідної нелінійної функції $f(x)$) розуміють таке значення аргументу x , підставлення якого в рівняння (1) перетворює його у тотожність.

У цьому поділі мова буде йти про відшукування дійсних нулів нелінійних рівнянь.

Загальних способів відшукування дійсних нулів довільних нелінійних рівнянь не існує. Нелінійна функція може не мати взагалі дійсних нулів, може мати один, два і навіть нескінченну кількість їх.

У загальному випадку процес відшукування нулів розпадається на два етапи. Перший етап, дії за яким не можна формалізувати, - пошук інтервалу змінювання аргументу, всередині якого існує єдиний нуль. Немає загальних правил, за якими можна було б установлювати такий (такі) інтервал. Перш за все це має бути аналітичне дослідження поведінки заданої функції у різних інтервалах змінювання аргументу. Наприклад, якщо задана нелінійна функція є аналітичною, тобто необмежено диференційованою, то умови, за яких функція $y = f(x)$ має єдиний нуль всередині інтервалу $[a, b]$ змінювання аргументу x , мають бути наступними:

- значення функції на кінцях цього інтервалу повинні бути протилежного знаку

$$f(a) \cdot f(b) < 0; \quad (5.3)$$

- перша похідна функції $f'(x) = \frac{df(x)}{dx}$ не змінює знаку на цьому інтервалі;

- друга похідна функції $f''(x) = \frac{d^2 f(x)}{dx^2}$ також не змінює знака на цьому інтервалі.

Другий етап - процес уточнення значення нуля. Відомі чисельні алгоритми відшукування нулів нелінійного рівняння, розглядувані у подальшому, насправді є лише алгоритмами уточнення значення нуля, бо зводяться до послідовного звуження інтервалу $[a, b]$, всередині якого знаходиться нуль. Слід узяти до уваги, що встановлення інтервалу, всередині якого знаходиться єдиний нуль функції, фактично є визначенням наближеного значення цього нуля

$$x^* = \frac{a+b}{2}. \quad (5.4)$$

При цьому похибкою такого значення може вважатися величина

$$\Delta x = \frac{|a-b|}{2}. \quad (5.5)$$

5.1. Метод дихотомії (ділення навпіл)

Алгоритм методу легко зрозуміти з рис. 5.1.

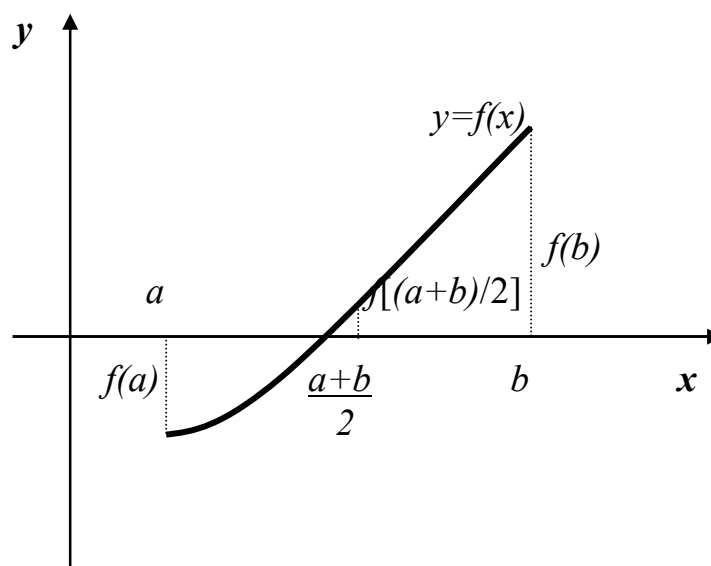


Рис. 5.1. Схема методу дихотомії

Заданий інтервал $[a, b]$ ділиться навпіл. Цим знаходиться наближене значення $x^* = \frac{a+b}{2}$ нуля функції. Обчислюється значення $y^* = f(x^*)$ функції при цьому значенні аргументу. Якщо воно дорівнює нулю, x^* є точним значенням нуля й процес закінчується. Якщо ні, то визначається знак значення y^* . Обирається той інтервал, на межах якого задана функція набуває значень протилежного знаку. Наприклад, якщо виявиться, що $f(x^*) \cdot f(a) < 0$, то як нове значення верхньої межі інтервалу приймається x^* : $b = x^*$. У протилежному випадку змінюється нижня межа інтервалу $a = x^*$. Далі процес повторюється для нового звуженого удвічі інтервалу $[a, b]$ доти, поки значення похибки (5) не стане меншою за задане припустиме її значення

$$\Delta x < \Delta_{\text{дон}}. \quad (5.6)$$

За остаточне значення нуля при цьому слід узяти значення (4).

Якщо обчислення потрібно проводити з максимальною точністю, процес звуження інтервалу слід продовжувати доти, поки нижня й верхня межі інтервалу $[a, b]$ не збіжаться у машинному поданні.

Схема алгоритму метода дихотомії для останнього випадку наведена на рис.5.2.

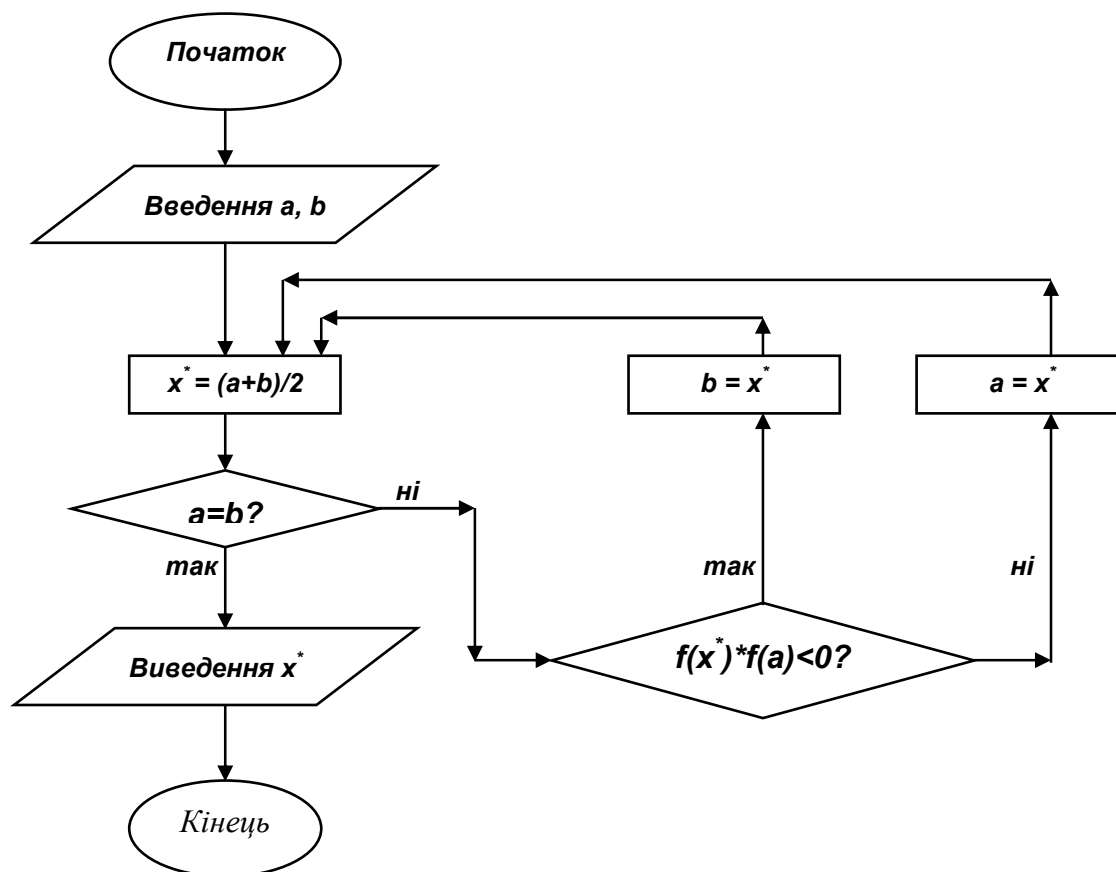


Рис. 5.2. Схема алгоритму метода дихотомії

До переваг метода дихотомії слід віднести те, що він може бути застосований навіть до тих неперервних функцій, що є недиференційованими у деяких точках усередині заданого інтервалу визначення кореня.

5.2. Метод хорд

Графічне уявлення про ідею метода хорд дає рис. 5.3.

З'єднуємо точки графіка функції, що відповідають кінцям заданого інтервалу $[a, b]$ прямою лінією. Відшукуємо точку перетинання цієї прямої з віссю аргументу - наближене значення x^* нуля функції. Обчислюємо значення функції у цій точці $y^* = f(x^*)$. Далі, як і у методі дихотомії, визначаємо, у якому з двох інтервалів $[a, x^*]$ або $[x^*, b]$ міститься нуль і відповідно до того змінюємо межі інтервалу, в першому випадку $b = x^*$, у другому - $a = x^*$.

Як впливає з рис. 5.3., на відміну від методу дихотомії, у методі хорд звуження інтервалу може бути не необмеженим (не збіжним до нуля). Одна з меж інтервалу може не змінюватися. Тому умова закінчення процесу тут має бути дещо іншою, а саме: на першому кроці методу слід запам'ятати перше наближене значення нуля x_1 ; здійснити другий крок і знайти друге наближене значення нуля x_2 ; відшукати різницю між цими нулями

$$\Delta x = |x_1 - x_2|. \quad (5.7)$$

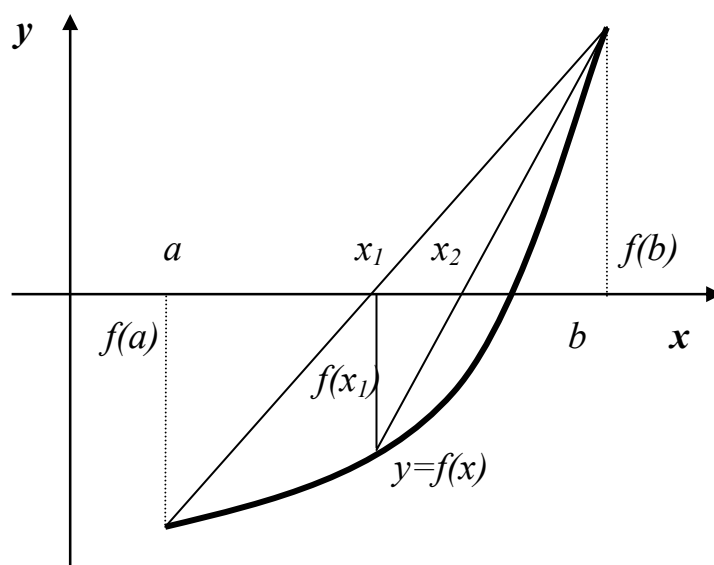


Рис. 5.3. Схема метода хорд

Перевірити умову (6). Якщо її виконано, процес завершується. За значення нуля слід узяти останнє наближене значення x_2^* . Якщо ні, слід замінити межі інтервалу, як вказано раніше, і замінити $x_1^* = x_2^*$. На наступному кроці наближення відшукується лише нове значення x_2^* .

Отже, внаслідок однобічного наближення до кореня, критерієм близькості до кореня є не довжина поточного інтервалу, а близькість двох послідовних наближених значень кореня.

Перейдемо до математичного оформлення цього алгоритму.

Перш за все потрібно обчислити значення функції $f(a)$ і $f(b)$ по краях інтервалу. Відшукаємо рівняння прямої, що проходить крізь точки $f(a), a$ і $f(b), b$:

$$\frac{y - f(b)}{f(a) - f(b)} = \frac{x - b}{a - b}.$$

Припускаючи у цьому рівнянні прямої $y = 0$, відшукаємо значення аргументу x_1 точки перетинання цієї прямої з віссю аргумента

$$x_1 = \frac{a \cdot f(b) - b \cdot f(a)}{f(b) - f(a)}. \quad (5.8)$$

Тепер алгоритм метода можна подати у вигляді схеми рис. 5.4.

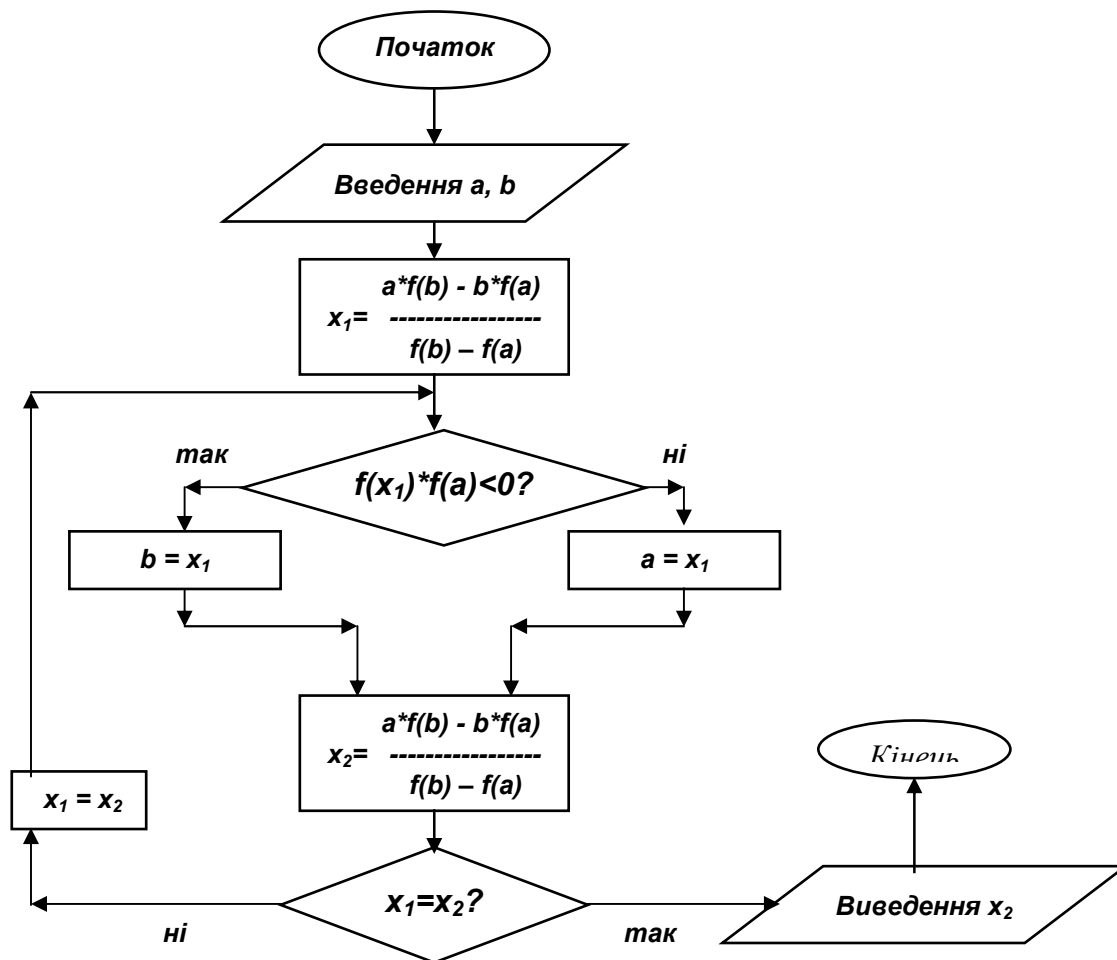


Рис. 5.4. Схема алгоритма метода хорд

Метод хорд, як і метод дихотомії, пристосований і для недиференційованих функцій. Його недоліком є однобічність наближення до кореня.

5.3. Метод дотичних (Ньютона)

Наступний метод може бути застосований лише для диференційованих функцій і потребує не лише задання функції $f(x)$, а й функції $f'(x)$ її похідної.

Сутність метода графічно подано на рис. 5.5.

Проводиться пряма, що є дотичною до графіка заданої функції у точці одній з меж заданого інтервалу $[a, b]$. Відшукується значення x_1 аргументу точки перетинання цієї прямою осі абсцис (аргументу). Обчислюється

значення функції у цій точці $y_1 = f(x_1)$. Далі будується нова пряма, дотична до графіка функції у точці x_1, x_1 . Відшукується точка перетинання x_2 , і процес повторюється.

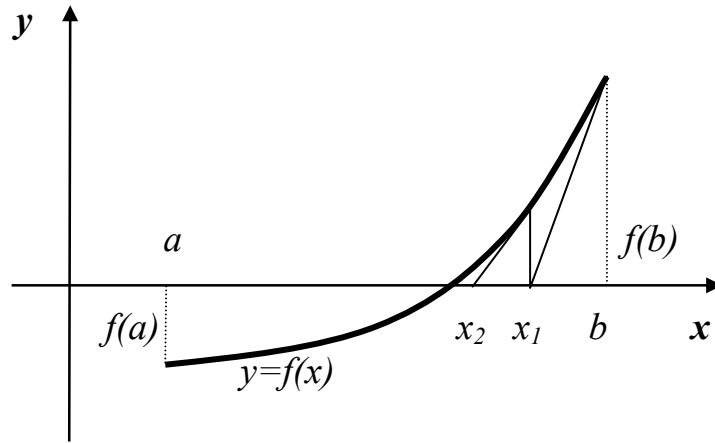


Рис. 5.5. Схема метода дотичних

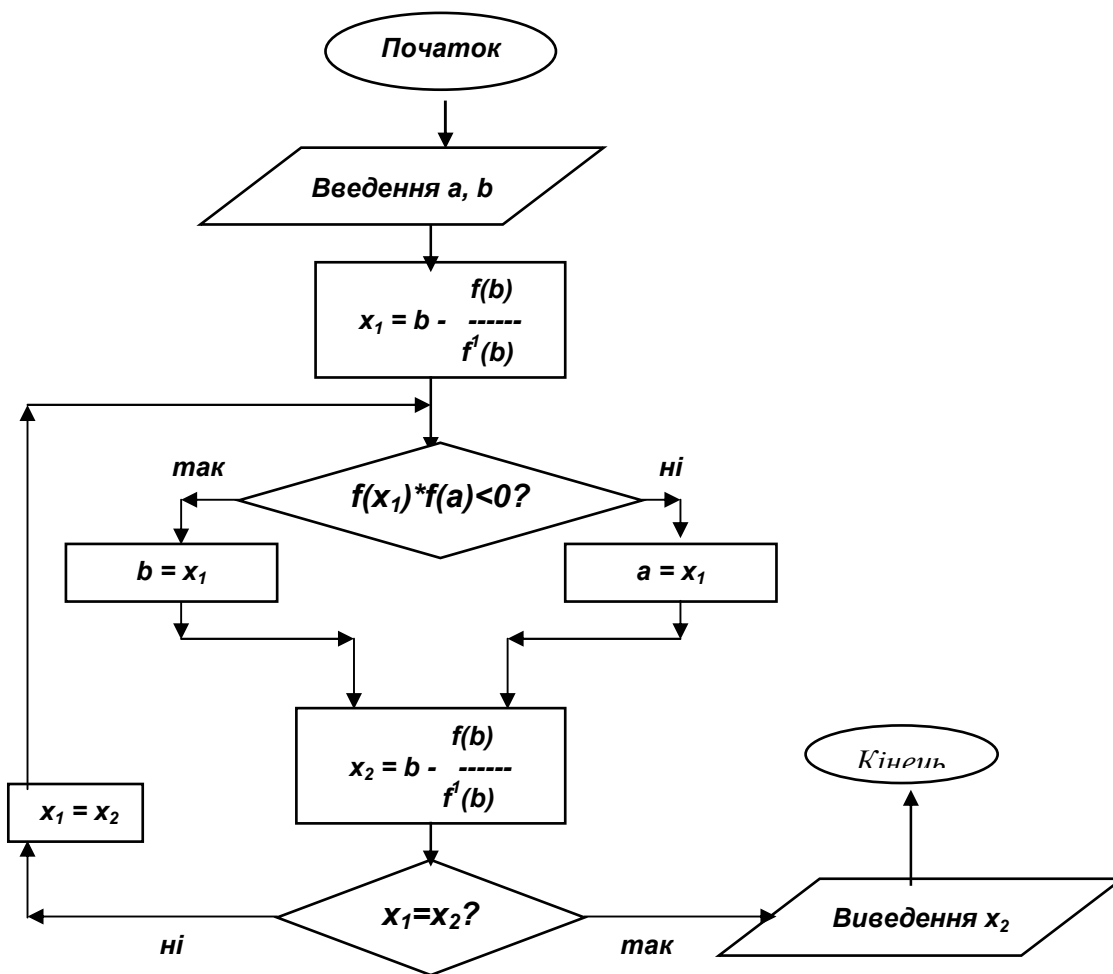


Рис. 5.6. Схема алгоритма метода Ньютона

Як і в методі хорд, процес наближення нулів переважно є однобічним. Тому, як і у попередньому випадку, наближення нулів слід оцінювати через близькість двох сусідніх послідовних наближених значень нулів.

Рівняння прямої, дотичної до кривої $f(x)$ у точці $x = b$, має вигляд:

$$y - f(b) = f'(b) \cdot (x - b).$$

Звідси точка перетинання цієї прямою осі абсцис визначиться формулою:

$$x_1 = b - \frac{f(b)}{f'(b)}. \quad (5.9)$$

Схему алгоритма реалізації метода дотичних подано на рис. 5.6.

Перевагою метода Ньютона є велика швидкість збіжності наближеного значення кореня до істинного. Недоліки - пристосовуваність лише для диференційовних функцій, необхідність задання додаткової функції - похідної від заданої, однобічність наближення коренів.

5.4. Комбінований метод (хорд і дотичних)

Як неважко впевнитися однобічні методи хорд і дотичних наближують до істинного значення нуля із протилежних боків. Тому можна суттєво поліпшити чисельне обчислення нулів, якщо поєднати ці два методи у єдиний метод.

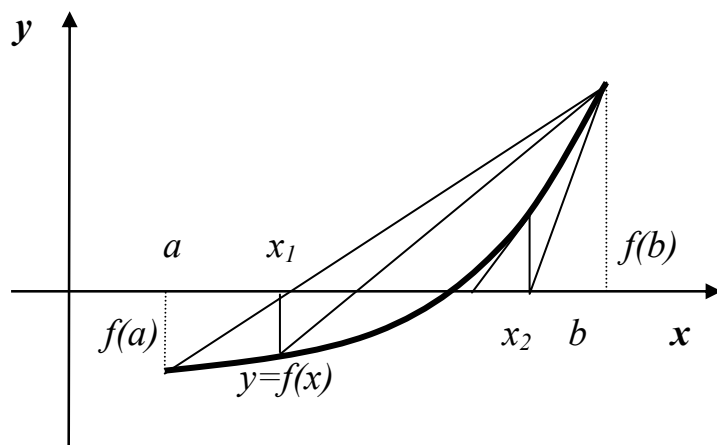


Рис. 5.7. Схеми методу хорд і дотичних

Дійсно, якщо на одному кроці застосувати обидва методи, одержимо вже звужений із двох боків новий інтервал змінювання аргументу, всередині якого міститься корінь рівняння.

Графічне уявлення про сутність метода можна одержати з рис. 5.7.

На кожному кроці метода здійснюються дві операції - обчислення наближеного значення x_1^* кореня за формулою (8) метода хорд і обчислення другого наближеного значення x_2^* того самого кореня за формулою (9) метода Ньютона. Установлюються нові межі інтервалу аргументу: за межу a

обирається менше з цих двох значень, за межу b - більше. Далі процес продовжується до виконання умови (6).

Алгоритм методу вказаний на рис 5.8.

До переваг метода відноситься велика швидкість збіжності значень кореня до істинного значення.

Недолік - пристосовуваність лише до диференційованих функцій, необхідність знання функції-похідної.

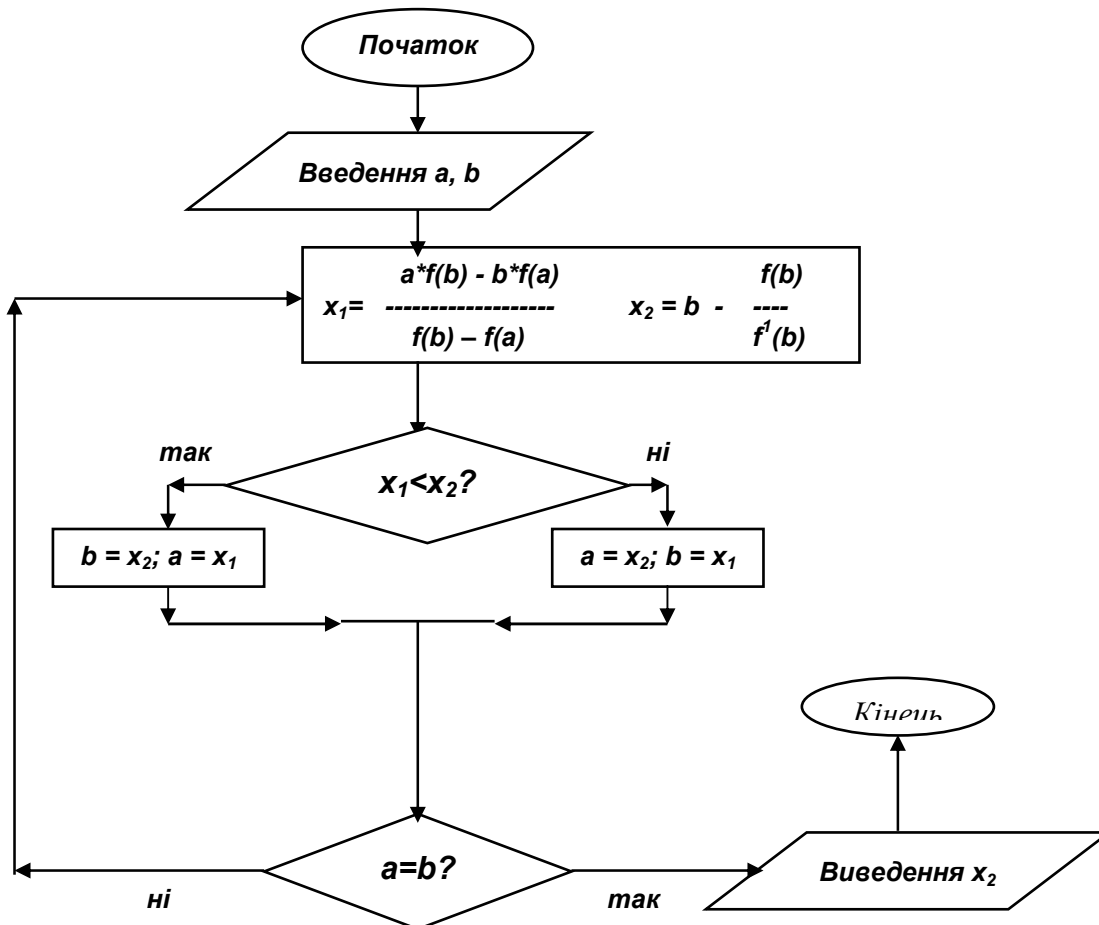


Рис. 5.8. Схема алгоритму комбінованого методу

5.5. Метод ітерацій

У методі ітерацій для уточнення значення нуля використовується ітераційна формула, яка пов'язує попереднє наближене значення нуля з його подальшим, більш точним значенням.

Найбільш просто одержати ітераційну формулу, якщо попередньо подати вихідне нелінійне рівняння (1) у вигляді

$$x = \varphi(x). \quad (5.10)$$

Перетворення рівняння (1) до форми (10) не є однозначним і визначається досвідченістю дослідника.

Щоб одержати графічне уявлення про сутність метода ітерацій, введемо позначення двох функцій

$$y_1(x) = x; \quad y_2(x) = \varphi(x). \quad (5.11)$$

Тоді розв'язок рівняння (10) можна інтерпретувати як аргумент точки перетину графіків цих двох функцій, а саме - прямої $y_1(x) = x$ (бісектриси координатного кута) і нелінійної функції $y_2(x) = \varphi(x)$ - див. рис. 5.9

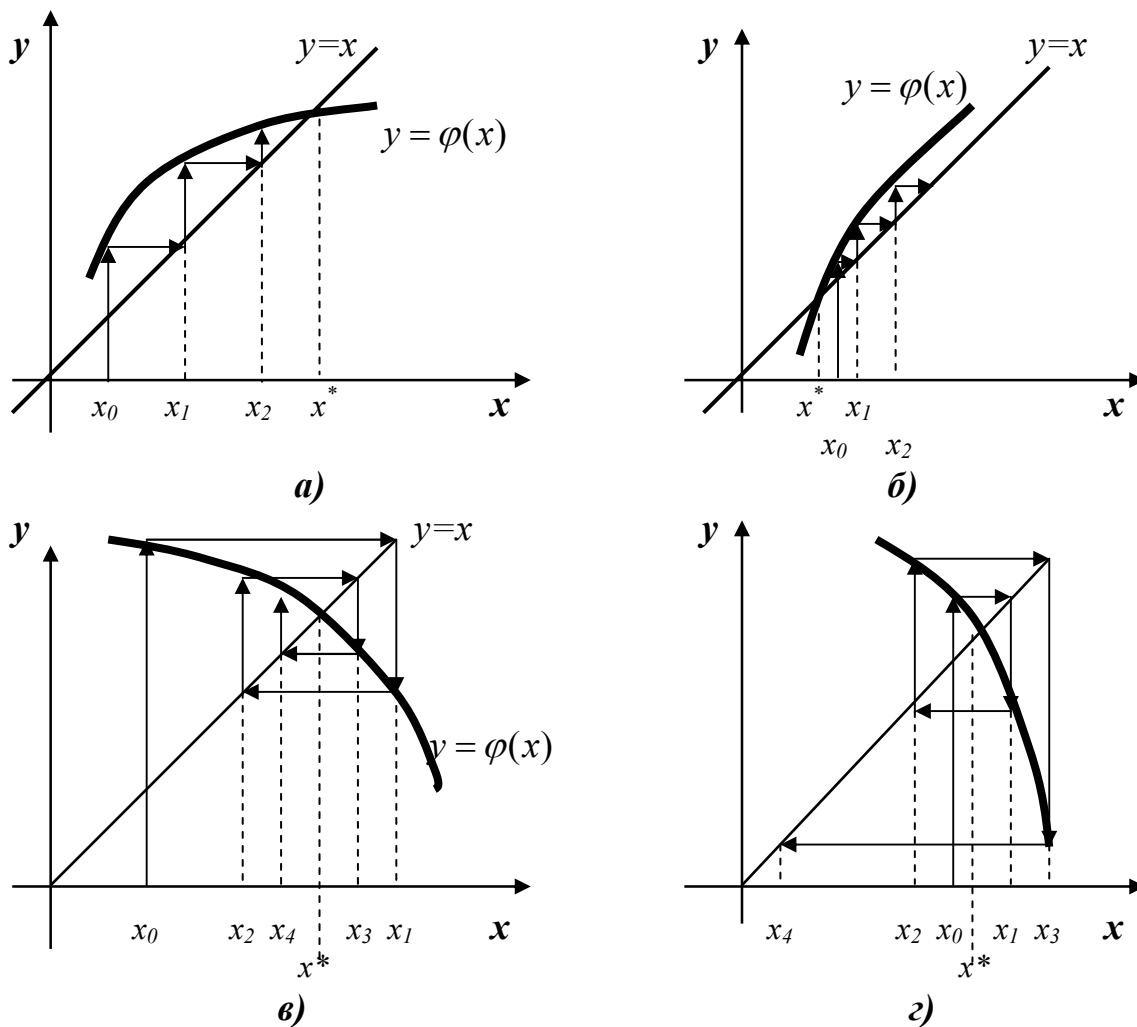


Рис. 5. 9. Графічні схеми методу ітерацій

Власне відшукування нуля методом ітерацій у цьому випадку полягає у застосуванні такої ітераційної формули, яка безпосередньо впливає з (10):

$$x_{i+1} = \varphi(x_i), \quad (5.12)$$

де i - номер ітерації (наближення).

Для її використання спочатку задаються певним значенням x_0 кореня усередині встановленого інтервалу існування єдиного кореня (точка 0 на рис. 5.9). Обчислюють значення нелінійної функції при цьому значенні аргументу (рух удовж вертикальної лінії до перетинання із графіком кривої $y_2(x) = \varphi(x)$). Одержуємо точку 1 на графіку. У відповідності до ітераційної формули (12) за

нове наближене значення кореня приймають одержане значення функції $x_1 = y_{20}$ (рух удовж горизонтальної прямої до перетинання із прямою $y_1(x) = x$). Одержуємо точку 2 графіка. Далі увесь процес повторюється доти, поки різниця між послідовними наближеними значеннями кореня не зменшиться до заданого рівня.

Як бачимо з рис. 5.9, успішність такого алгоритму забезпечується далеко не завжди. Лише за умови, що різниця між двома послідовними значеннями аргументу, одержаними при таких діях, зменшується за модулем, можна бути впевненим, що відбувається дійсно наближення до істинного кореня. Неважко впевнитися, що останнє можливе, якщо виконується співвідношення

$$\left| \frac{d\varphi(x)}{dx} \right| < 1 \quad (5.13)$$

у всіх точках інтервалу. Це й є умова стійкості (збіжності) ітераційного процесу.

Впливати на виконання цієї умови можна, по-різному формуючи нелінійну функцію $\varphi(x)$.

5.6. Реалізація процесу відшукування коренів на ЕОМ

Щоб реалізувати зазначені алгоритми на ЕОМ, потрібно мати у розпорядженні засоби, які б дозволяли використовувати довільну функцію (задану певною підпрограмою з відомим ім'ям), використовуючи ім'я цієї підпрограми як один із параметрів методів уточнення кореня. Такі засоби є майже в усіх мовах високого рівня. У Фортрані - це так звані зовнішні функції, що оголошуються за допомогою оператора EXTERNAL. У мові Паскаль для цього призначений так званий процедурний тип даних.

Особливо просто відшукування коренів нелінійного рівняння (1) здійснюється у середовищі MatLAB. Покажемо, як це робиться. Утворимо окремий М-файл, в якому обчислюється значення функції

$$f(x) = x^3 + 6x^2 + 3x - 10,$$

яка має три корені $x_1 = -5$, $x_2 = -2$ і $x_3 = +1$, і назвемо його "pol3fun":

```
function y=pol3fun(x)
% Обчислення функції y=x^3+6x^2+3x-10
v=[1 6 3 -10];
y=polyval(v,x)
```

Для відшукування коренів використовують процедуру *fzero*, яка обчислює нуль заданої функції. Звичайне звернення до неї є таким

$$x=fzero('pol3fun', x0, tol, trace),$$

де у дужках першим вказується (в апострофах) ім'я М-файлу, в якому обчислюється значення функції $f(x)$ за заданим значенням аргументу; другим задається початкове значення аргументу x_0 або вектор $[a \ b]$ із двох значень - нижньої й верхньої меж початкового інтервалу існування єдиного кореня; аргументи *tol* і *trace* можуть не вказуватися. Параметр *tol* задає значення

максимальної припустимої похибки при обчисленні значення x шуканого кореня. Нарешті, параметр *trace* вказує, що проміжні результати слід виводити на екран дисплея.

Наведемо приклади. Задамо початкове значення кореня $x_0 = -6$:

x0=-6

x0 = -6

Тепер звернемося до процедури *fzero* у найпростішому варіанті:

X=fzero('pol3fun',x0)

У командному вікні MatLAB одержимо результат

x = -5

Щоб одержати проміжні результати й інформацію про точність результатів, треба вказати замість параметра *trace* будь-яке ціле додатне число:

X=fzero('pol3fun',x0,[],1)

Тоді у командному вікні виникне таблиця

Func	evals	x	f(x)	Procedure
1		-6	-28	initial
2		-5.83029	-21.7222	search
3		-6.16971	-34.969	search
4		-5.76	-19.3174	search
5		-6.24	-38.065	search
6		-5.66059	-16.1063	search
7		-6.33941	-42.6585	search
8		-5.52	-11.9342	search
9		-6.48	-49.5954	search
10		-5.32118	-6.74272	search
11		-6.67882	-60.3165	search
12		-5.04	-0.734464	search
13		-6.96	-77.3839	search
14		-4.64235	5.33217	search
Looking for a zero in the interval [-4.6424, -6.96]				
15		-4.79176	3.3671	interpolation
16		-5.03708	-0.679922	interpolation
17		-4.99587	0.0742381	interpolation
18		-4.99992	0.00136181	interpolation
19		-5	-9.09834e-008	interpolation
20		-5	3.43725e-012	interpolation
21		-5	0	interpolation

x = -5

Як бачимо, загальна процедура відшукування кореня у випадку, коли задається не інтервал існування кореня, а його наближене значення, містить два етапи - пошук інтервалу, всередині якого є корінь, і потім уточнення значення кореня усередині знайденого інтервалу.

Число операцій значно скорочується, якщо замість одного наближеного значення кореня як другий аргумент функції вказати вектор із двох меж інтервалу, всередині якого міститься шуканий корінь. У цьому випадку необхідність у першому етапі - пошуку меж інтервалу - зникає:

X=fzero('pol3fun',[x0 -4],[],1)

Func	evals	x	f(x)	Procedure
1		-6	-28	initial
2		-4	10	initial
3		-4.52632	6.61321	interpolation
4		-5.35767	-7.63518	interpolation
5		-4.91218	1.51205	interpolation
6		-5.00765	-0.138173	interpolation
7		-4.99965	0.00623681	interpolation
8		-5	2.3783e-005	interpolation

```

      9          -5 -2.78817e-011      interpolation
     10          -5          0      interpolation
x =      -5

```

Неважко впевнитися, що у випадку, коли відносна припустима похибка *tol* у явному вигляді при зверненні не вказується, система MatLAB виконує обчислення нуля з максимально досяжною (машинною) точністю.

Значення відносної похибки обчислення нуля можна змінювати за власним бажанням, наприклад:

```

tol=1e-3
tol = 0.0010

```

Тоді, якщо звернення містить вказівку цієї похибки

```

X=fzero('pol3fun',[x0 -4],tol,1)

```

одержують із відповідною точністю:

Func evals	x	f(x)	Procedure
1	-6	-28	initial
2	-4	10	initial
3	-4.52632	6.61321	interpolation
4	-5.35767	-7.63518	interpolation
5	-4.91218	1.51205	interpolation
6	-5.00765	-0.138173	interpolation
7	-4.99763	0.0425789	interpolation

```

x = -4.9976

```

За допомогою функції *fzero* можна знайти корінь лише одного нелінійного рівняння.

Якщо ж потрібно відшукати корені системи з *n* алгебричних рівнянь, доцільно користуватися процедурами *solve* і *fsolve*. Наприклад, припустимо, що потрібно відшукати корені системи нелінійних рівнянь

$$\begin{cases} x_1 x_2 + x_3 = 6,5 \\ x_1 x_2^4 + x_3 = 167 \\ x_1 x_2^6 + x_3 = 1470 \end{cases}$$

Якщо тепер звернутися до процедури *solve* у такий спосіб:

```

[x1,x2,x3]= solve('x1*x2+x3=6.5', 'x1*x2^4+x3=167', 'x1*x2^6+x3=1470')

```

то одержимо

```

x1 =
[
    1.7554934310605148014354104265164]
[ 762.52961873540473151333419519262-621.90701499914181279559780253477*i]
[ 762.52961873540473151333419519262+621.90701499914181279559780253477*i]
[
    2.1511971917755071217289751079671]
x2 =
[
   -3.0659535021748359282404774061941]
[ -.45092048655690998589972532425192-.82999197912837056252692361812036*i]
[ -.45092048655690998589972532425192+.82999197912837056252692361812036*i]
[
    2.9677944752886559000399280546980]
x3 =
[
   11.882261233004904252923256860344]
[ 866.51806090717871601749484835904+352.46285360163186381239610138509*i]
[ 866.51806090717871601749484835904-352.46285360163186381239610138509*i]
[
   .11568885899217876225427050184419]

```

Щоб скористатися функцією *fsolve*, потрібно утворити М-файл процедури, яка обчислює вектор-стовпець $y = [y_1; y_2; y_3]$ з функцій, корені яких

потрібно відшукати, за заданим (вхідним) вектором стовпцем $x = [x_1; x_2; x_3]$.

Назвемо процедуру *fun33*:

```
function y=fun33(x)
y=[x(1)*x(2)+x(3)-6.5;
  x(1)*x(2)^4+x(3)-167;
  x(1)*x(2)^6+x(3)-1470];
```

Тепер можна звернутися до процедури *fsolve* у такий спосіб:

```
X=fsolve('fun33',x0)
```

попередньо ввівши вектор стовпець початкових значень шуканих коренів:

```
x0 = [1; 1; 1];
```

Результатом буде поява у командному вікні вектора коренів

```
X =
  2.1512
  2.9678
  0.1157
```

Як бачимо, процедура *fsolve* реалізує ітераційний процес і відшукує лише дійсні корені, а *solve* визначає усі корені, у тому числі й комплексні.

Набагато складніше досягти аналогічного результату мовою Паскаль.

Перш за все необхідно утворити спеціальний процедурний тип даних, яким би можна було описувати функції одного змінного. Назвемо цей тип **FunX**.

Потім треба визначити у вигляді підпрограми-функції **Pol3Fun** процедуру обчислення значення заданої функції за заданим значенням її аргументу. Причому, робити це потрібно у спеціальному режимі з використанням директиви `{SF+}` далекого виклику.

У звичайному режимі визначається процедура метода уточнення кореня. Як такий візьмемо метод дихотомії й утворимо процедуру **Dyhotomy**, що здійснює просування на один крок, звужуючи заданий інтервал існування єдиного кореня.

Нарешті, потрібно утворити основний текст програми, у якому були б подані усі зазначені процедури й оголошення, здійснювалося введення початкових параметрів (меж інтервалу існування кореня, заданої припустимої похибки, тощо), організація циклічного виклику процедури метода до досягнення заданої точності обчислення кореню й виведення результатів до екрана.

У цілому можливий текст такої програми наведено нижче.

```
program NelinUr;
USES crt;
TYPE
  {Введення процедурного типу - функції однієї змінної}
  FunX=Function(x:extended):extended;
VAR
  Fun1: FunX;   {Оголошення, що змінна Fun1 має процедурний тип FunX}
  ad,ap,an,bd,bp,bn,x,xs,eps,e:   extended;
  i: word;
{+++++}
{SF+}
{=== Підпрограма обчислення заданої функції ===}
```

```

Function Pol3Fun(x:extended):extended;
Begin
  Pol3Fun:=((x+6)*x+3)*x-10;
End;
{$F-}
{+++++}
PROCEDURE Dyhotomy(Fun:FunX;a0,b0:extended;var a,b,x:extended);
  {==== Процедура методу дихотомії ====}
Label d1;
Var
  Fa, Fb, Fx: extended;
Begin
  a:=a0; b:=b0;
  Fa:=Fun(a0); { }
  if Fa=0 then begin x:=a; b:=a; goto d1 end;
  Fb:=Fun(b0);
  if Fb=0 then begin x:=b; a:=b; goto d1 end;
  x:=(a+b)/2;
  Fx:=Fun(x);
  if Fx=0 then begin a:=x; b:=x; goto d1 end;
  if Fa*Fx<0 then b:=x else a:=x;
  d1:
End;
{=====}
BEGIN
  clrscr;
  Fun1:=Pol3Fun; {Присвоєння першому параметру процедури
                 метода дихотомії значення ймення
                 заданої функції}

  ad:=-6; bd:=-4.5;
  eps:=1e-6;
  e:=1;
  ap:=ad; bp:=bd;
  xs:=ad;
  i:=1;
  While e>eps do { Цикл застосування методу дихотомії }
  begin
    Dyhotomy(Fun1,ap,bp,an,bn,x);
    e:=abs(x-xs);
    writeln(i, ' a=',an:13,'; b=',bn:13,'; x=',x:16,'; e=',e:9);
    ap:=an; bp:=bn;
    xs:= x;
    i:=i+1;
  end;
  Writeln('Відповідь: x=',x:18,'; Похибка: deltaX=',e:9);
  readln
END.

```

5.7. Розв'язування поліноміальних рівнянь

Як було показано у п. 3.5, для інженерного дослідження лінійних стаціонарних систем важливо мати змогу відшукувати корені поліномів, що утворюють чисельники і, особливо, знаменники передатних функцій. Причому, необхідно визначати усі корені цих поліномів, включаючи *комплексно спряжені*, які *відповідають коливальним системам* (тобто таким, які при відсутності зовнішніх дій приходять у стан рівноваги, здійснюючи *навколо*

$$c_1 = c_0 = 0. \quad (5.20)$$

Тому для відшукування невідомих коефіцієнтів p, q квадратного полінома-множника, а також коефіцієнтів b_k ($k = 0, 1, \dots, n-3$) полінома - другого множника можна запропонувати такі дії:

■ задаються початкові довільні значення дійсних коефіцієнтів p і q (наприклад, рівні нулю);

■ з перших формул (19) визначаються невідомі коефіцієнти b_k :

$$b_{n-3} = a_{n-1} - p; \quad b_{n-4} = a_{n-2} - p \cdot b_{n-3} - q; \quad \dots \quad (5.21)$$

$$b_{n-k-2} = a_{n-k} - p \cdot b_{n-k-1} - q \cdot b_{n-k}; \quad \dots \quad b_0 = a_2 - p \cdot b_1 - q \cdot b_2;$$

■ з останніх двох формул (19) і із врахуванням умови (20) обчислюються наступні (нові) значення коефіцієнтів p і q :

$$q^* = \frac{b_0}{a_0}; \quad p^* = \frac{a_1 - q^* \cdot b_1}{b_0}. \quad (5.22)$$

Далі усі дії повторюються з новими значеннями p і q доти, поки різниці $p - p^*$ і $q - q^*$ не стануть достатньо малими за абсолютною величиною.

Реалізуємо цей метод у системі MatLAB.

Перш за все утворимо процедуру *PolySqMnoj*, яка здійснює обчислення одного кроку ітераційного процесу з виділення квадратного множника заданого полінома A:

```
function [Qout,B]=PolySqMnoj(A,Q)
% Уточнення значення квадратного множника Q полінома A
% і визначення результату ділення A на Q

% Лазарєв Ю.Ф., 24-08-2000р.
n=length(A);
B=zeros(1,n-2);
p=Q(1); q=Q(2);
if n==2
    Qout(1)=A(1);
    Qout(2)=A(2);
    B=0;
elseif n==3
    B(1)=A(1)-p;
    Qout(2)=A(3)/B(1);
    Qout(1)=(A(2)-q)/B(1);
elseif n==4
    B(1)=A(1)-p;
    B(2)=A(2)-p*B(1)-q;
    Qout(2)=A(4)/B(2);
    Qout(1)=(A(3)-q*B(1))/B(2);
elseif n>4
    B(1)=A(1)-p;
    B(2)=A(2)-p*B(1)-q;
    for k=1:(n-4)
        B(k+2)=A(k+2)-p*B(k+1)-q*B(k);
    end
    Qout(2)=A(n)/B(n-2);
    Qout(1)=(A(n-1)-q*B(n-3))/B(n-2);
```

end

Тепер створимо процедуру *Korny* обчислення коренів квадратного рівняння

```
function x=Korny(a,b,c)
% Процедура обчислення коренів квадратного рівняння
%  $a*x^2+b*x+c=0$ 

% Лазарєв Ю.Ф., 24-08-2000
p=-b/a/2; q=c/a;
sd=sqrt(p^2-q);
x(1)=p+sd; x(2)=p-sd;
```

Нарешті, утворимо головну, керуючу програму, яка утворює поліном із задалегідь відомими значеннями коренів, організує ітераційний процес із виділення множника-полінома другого степеня й відшукування його коренів:

```
%PolyKor
% Програма відшукування коренів полінома
% методом Ліна

% Лазарєв Ю.Ф., 24-08-2000
clc
%-----
% 1. Формування полінома за заданими його коренями
disp('Задані значення коренів:');
x1=-20;
x2=-0.1+5i; x3=-0.1-5i;
x4=-0.02+i; x5=-0.02-i;
xz1=[x4 x5]
xz2=[x2 x3]
xz3=x1
P1=[1 -x1]; P2=[1 -x2]; P3=[1 -x3];
P4=[1 -x4]; P5=[1 -x5];
P23=conv(P3,P2);
P45=conv(P4,P5);
P2_5=conv(P23,P45);
disp('Початковий поліном:');
P=conv(P1,P2_5)
disp('Знайдені його корені:');
%-----
% 2. Відшукування першої пари коренів
A=P(2:end);
Q=[0 0]; Q1=[1 1];Qout=Q1; k=0;
while Q(1)~=Q1(1)|Q(2)~=Q1(2) & k<100
k=k+1;
Q1=Qout;
[Qout,B]=PolySqMnoj(A,Q);
Q=Qout;
end
a=1;b=Qout(1); c=Qout(2);
x1=Korny(a,b,c)
%-----
% 3. Відшукування другої пари коренів
Q=[0 0]; Q1=[1 1];Qout=Q1; k=0;
while (Q(1)~=Q1(1)|Q(2)~=Q1(2))& k<100
```

```

k=k+1;
Q1=Qout;
[Qout,C]=PolySqMnoj(B,Q);
Q=Qout;
end
a=1;b=Qout(1); c=Qout(2);
x2=Korny(a,b,c)
%-----
% 4. Відшукування останнього кореня
x3=-C

```

Запускаючи цю програму, одержимо у командному вікні MatLAB:

```

Задані значення коренів:
xz1 =
    -0.02 +    1i    -0.02 -    1i
xz2 =
    -0.1 +    5i    -0.1 -    5i
xz3 =
    -20
Початковий поліном:
P =
     1    20.24    30.818    521.57    49.03    500.4
Знайдені його корені:
x1 =
    -0.02 +    1i    -0.02 -    1i
x2 =
    -0.1 +    5i    -0.1 -    5i
x3 =
    -20

```

Як бачимо, ми одержуємо ті самі значення коренів, які були задані.

Взагалі, при використанні системи MatLAB, для визначення коренів полінома немає потреби створювати якісь спеціальні програми. Досить використати вбудовану процедуру *roots* (див. п. 3.1). Наприклад, застосовуючи цю процедуру до раніше побудованого вектора P коефіцієнтів полінома, одразу одержуємо

```

» roots(P)
ans =
    -20
    -0.1 +    5i
    -0.1 -    5i
    -0.02 +    1i
    -0.02 -    1i

```

Результат при цьому одержують точніший і набагато швидше.

Наостанок розглянемо функції Matlab, які відшуковують мінімуми функцій.

До них відносяться:

fmin – відшукування мінімуму функції однієї змінної;

fmins – відшукування мінімуму функції кількох змінних.

Звернення до першої з них у загальному випадку має такий вигляд:

```
Xmin=fmin('ім'я М-файла функції',X1,X2)
```

Результатом цього звернення буде значення $Xmin$ аргументу функції, яке відповідає локальному мінімуму функції, поточні значення якої обчислюються у М-файлі з вказаним ім'ям, і яке міститься в інтервалі $X1 < X < X2$.

Звернення до другої процедури має форму:

$X_{\min} = \text{fmins}(\text{'<ім'я М-файла функції>'}, X_0)$

При цьому, на відміну від попереднього, X є не скаляром, а вектором аргументів, X_0 позначає початкове значення цього вектора, в околі якого відшукується найближчий локальний мінімум функції. В результаті одержується вектор X_{\min} аргументів, який відповідає цьому мінімуму.

5.8. Завдання

Завдання 11.

Відшукайте найменший за модулем дійсний корінь рівняння $f(x)=0$ за допомогою процедур *fzero*, *solve* і *fsolve*. Результати порівняйте.

1. $x^2 - 2x + \ln(x) = 0;$

2. $x^4 - 6x^2 + 12x - 8 = 0;$

3. $x^2 - 2 \ln(x + 2) = 0;$

4. $2^x + 2x^2 - 3 = 0;$

5. $x^3 + 2x - 13 = 0;$

6. $x^2 + \text{arctg}(x) - 0.5 = 0;$

7. $x \cdot e^{2x} - 4 = 0;$

8. $\text{ctg}(0,8x) - 2x^2 = 0;$

9. $x^5 + 5x + 1 = 0;$

10. $x^5 + 18x^3 - 34 = 0;$

11. $(x - 2)^2 - e^x = 0;$

12. $2x \cdot e^{x^2} - 5 = 0;$

13. $x^3 + 2x^2 - 11 = 0;$

14. $2e^{-x^2} - 3x + 4 = 0;$

15. $x^2 - 1 - \cos(1,2x) = 0;$

16. $2x - 3 \sin(2x) - 1 = 0;$

17. $(x - 0,5)^2 - \sin(\pi x) = 0;$

18. $x^3 + 3x^2 - 6x - 1 = 0;$

19. $x^3 - 2 \cos(\pi x) = 0;$

20. $\text{tg}(0,8x) - x - 2 = 0;$

21. $\text{tg}(1,2x) + 3x - 2 = 0;$

22. $x^4 + 3x - 3 = 0;$

23. $(x - 1)^2 - 0,5e^x = 0;$

24. $3 - x^3 + \sin\left(\frac{\pi}{2}x\right) = 0$

Завдання 12.

Відшукайте корені системи нелінійних рівнянь за допомогою процедур *solve* і *fsolve*. Результати порівняйте.

Варіант наближення	Система рівнянь	Початкові
1.	$\begin{cases} \sin(x_1 + x_2) - 1,24x_1 = 0,1 \\ x_1^2 + x_2^2 = 1 \end{cases}$	$x_1 = 0,74; \quad x_2 = 0,67$
2.	$\begin{cases} \sin(x_1 + 1) - x_1 = 0,1 \\ \cos x_1 + 2x_2 = 2 \end{cases}$	
3.	$\begin{cases} \text{tg}(x_1 x_2 + 0.2) = x_1^2 \\ 0.6x_1^2 + 2x_2^2 = 1 \end{cases}$	$x_1 = 0,88; \quad x_2 = 0,52$

4.
$$\begin{cases} \cos(x_2 - 1) + x_1 = 0,5 \\ x_2 - \cos x_1 = 3 \end{cases}$$
5.
$$\begin{cases} \sin x_1 + 2 \sin x_2 = 1 \\ 2 \sin 3x_1 + 3 \sin 3x_2 = 0,3 \end{cases} \quad x_1 = 1,08; \quad x_2 = 0,06$$
6.
$$\begin{cases} (x_1^2 - 1)x_2 = 9 \\ x_1 x_2 - x_1^2 + 10 = 0 \end{cases}$$
7.
$$\begin{cases} \operatorname{tg}(x_1 - x_2) - 4x_1 = 0 \\ x_1^2 + 2x_2^2 = 1 \end{cases} \quad x_1 = -0,5; \quad x_2 = 0,6$$
8.
$$\begin{cases} \sin(x_1 + x_2) - 2,4x_1 = -3,2 \\ x_1^2 + x_2^2 - x_1 x_2 = 0 \end{cases}$$
9.
$$\begin{cases} x_1^4 + x_2^2 = 3 \\ x_1^3 + x_2^3 = 4 \end{cases} \quad x_1 = 0,95; \quad x_2 = 1,4$$
10.
$$\begin{cases} 2x_1 x_2^2 - 4x_2 = 7,5 \\ x_1^2 - 3x_1 x_2 = -4,5 \end{cases}$$
11.
$$\begin{cases} \sin x_1 - x_2 = 1,3 \\ \cos x_2 - x_1 = -0,82 \end{cases} \quad x_1 = 0,8; \quad x_2 = -0,35$$
12.
$$\begin{cases} \sin x_1 + 2 \cos x_2 = 0,8 \\ x_1 x_2^2 + 3x_1 = 4,5 \end{cases}$$
13.
$$\begin{cases} x_1^3 + x_2^3 - 6x_1 + 3 = 0 \\ x_1^3 - x_2^3 - 6x_2 = 2 \end{cases} \quad x_1 = 0,52; \quad x_2 = -0,37$$
14.
$$\begin{cases} x_2 + e^{x_1 - x_2} = 0 \\ x_1 + e^{x_1 - x_2} = 0 \end{cases} \quad x_1 = -0,7; \quad x_2 = -0,35$$
15.
$$\begin{cases} (x_1^2 - 1)x_2^2 = 18,75 \\ x_1 + x_2^2 = 8,25 \end{cases}$$
16.
$$\begin{cases} \operatorname{tg}(x_1 - x_2) - 4x_1 = 0 \\ x_1^2 + 3x_2^2 = 4 \end{cases}$$
17.
$$\begin{cases} x_1^2 x_2 - 8x_1 = -5,5 \\ x_1 x_2 + 3x_2 = 10 \end{cases}$$
18.
$$\begin{cases} x_1^2 \sin x_2 + x_2^2 \sin x_1 + 1 = 0 \\ 2x_1 + e^{x_1 + x_2} = 4 \end{cases}$$
19.
$$\begin{cases} e^{x_1} + 2x_2 \ln x_1 = 3 \\ x_1^2 x_2 - 3x_1 + 5,4 = 0 \end{cases}$$

$$20. \quad \begin{cases} \sin x_1 + 3,5 \sin x_2 = 1 \\ 2 \sin 3x_1 + 3 \sin 2x_2 = 0,4 \end{cases}$$

5.9. Контрольні запитання

1. Що таке нелінійне алгебричне рівняння? Що означає відшукати розв'язок нелінійного рівняння?
2. На які етапи поділяється процес відшукування розв'язків нелінійного рівняння?
3. У чому логіка відшукування розв'язку нелінійного рівняння методом ділення навпіл? Які переваги і недоліки цього метода?
4. У чому сутність метода хорд? Які його переваги і недоліки?
5. Яка ідея методу дотичних? Які переваги і недоліки цього метода?
6. Які особливості комбінованого методу? У чому полягають його переваги і недоліки у порівнянні з методом хорд? з методом Ньютона?
7. У чому полягає сутність метода ітерацій розв'язування нелінійного рівняння? Які в нього переваги і недоліки у порівнянні з іншими методами?
8. Які засоби розв'язування нелінійних рівнянь є у сучасних мовах програмування?
9. Як відшукати корінь нелінійного алгебричного рівняння у системі MatLAB?
10. Як можна відшукати комплексні нулі полінома?
11. Які засоби відшукування комплексних нулів поліномів є у сучасних мовах програмування?
12. Як відшукати усі нулі заданого полінома у системі MatLAB?

6. Наближення функцій

6.1. Постановка задачі наближення функції

Стисло поняття наближення можна охарактеризувати як заміну деякої неперервної функції на деяку іншу, більш зручну у деякому відношенні і, в той же час, досить близьку до першої (теж в деякому певному відношенні).

Така задача виникає у кількох випадках. Наприклад, при теоретичних дослідженнях іноді виникає потреба замінити складну табульовану функцію, результати деяких потрібних дій над якою (наприклад, добуття похідної, або інтегралу) не можуть бути подані через елементарні функції, на таку, щоб вона була близькою до відомої, але з нею можливо було б досить просто робити необхідні операції. Зазвичай як така функція обирається поліном деякого степеня через простоту здійснювання над ним практично усіх математичних перетворень.

Другим важливим випадком, більш близьким до інженерної практики, є такий, коли значення деякої невідомої, але неперервної функціональної залежності надані лише при певних окремих значеннях аргумента і одержані шляхом вимірювання певної фізичної величини. Потрібно описати цю невідому залежність якомога простішою і зручною неперервною математичною залежністю, найближчою до невідомої у деякому розумінні, щоб можна було оперувати значеннями цієї фізичної величини не тільки у точках здійснених вимірів, але і у будь-яких точках між ними.

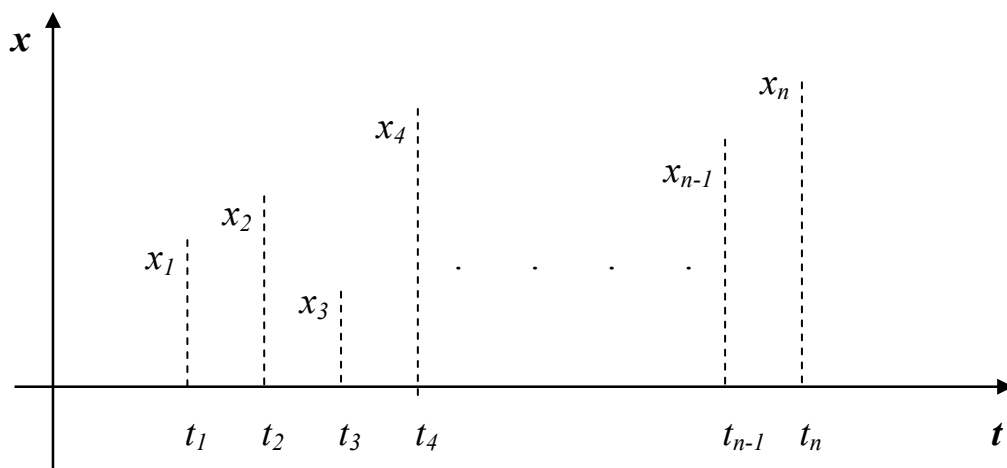


Рис. 6.1. Графік таблично заданої функції

У обох випадках можна вважати, що є відомим (заданим чи вимірним) масив t_1, t_2, \dots, t_n ($t_i < t_{i+1}$) дискретних значень аргумента t і відповідний масив x_1, x_2, \dots, x_n значень деякої функції $x(t)$, яка має бути наближеною, за вказаних значень аргумента (див. рис. 6.1). Тобто задана таблиця значень функції. Априорно відомо також, що ця функція насправді є неперервною функцією вла-

сного аргумента (наприклад, виміряна фізична величина є неперервною функцією часу).

Якщо, маючи таку таблицю, потрібно:

- визначити значення функції $x(t)$ у проміжку між будь-якими сусідніми значеннями t_i і t_{i+1} ;
- проінтегрувати функцію $x(t)$ у межах між t_1 і t_n ;
- відшукати значення похідної від функції $x(t)$ при будь-якому значенні аргумента між t_1 і t_n ,

це можна зробити тільки на основі наближення таблично заданої функції.

Задачу наближення можна розділити на два види, які значно відрізняються один від одного за постановкою і засобами розв'язування.

Перший вид задачі наближення виникає тоді, коли відомо, що задані значення визначені досить точно (тобто з похибкою, якою можна нехтувати). Тоді наближаючу неперервну функцію $X(t)$ можна (і слід) будувати так, щоб вона приймала задані значення x_i при заданих значеннях аргумента t_i , тобто графік цієї функції проходив крізь усі задані точки рис. 6.1. Таку постановку задачі прийнято називати інтерполюванням функції (див. рис. 6.2). Точки (x_i, t_i) у цьому випадку нізивають вузлами інтерполяції, бо в цих точках збігаються значення наближаючої (інтерполюючої) функції $X(t)$ і первісної функції $x(t)$.

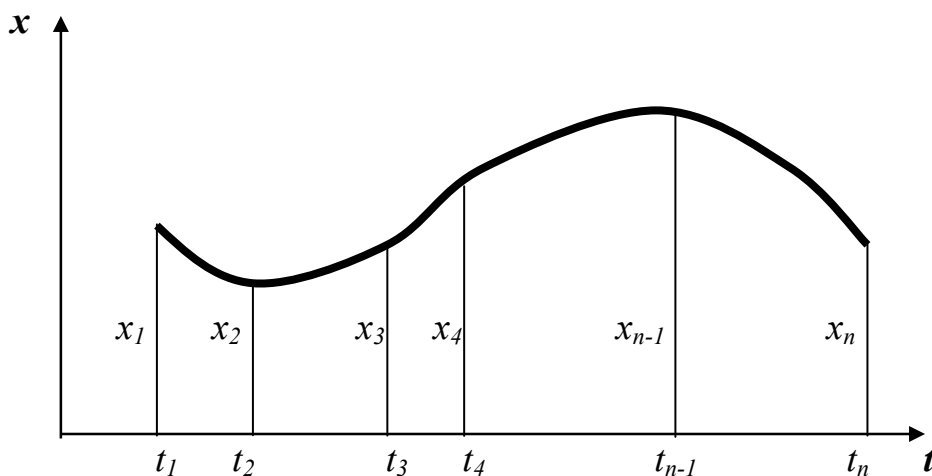


Рис. 6.2. Схема інтерполювання таблично заданої функції

По-іншому ставиться задача наближення у випадку, коли апріорно відомо, що задані значення x_i презентують первісну функцію зі значними, заздалегідь невідомими, похибками. У цьому випадку немає сенсу домогтися щоб наближаюча (у цьому випадку її називають апроксимуючою) функція обов'язково збігалася з заданою табличною при відповідних заданих значеннях аргумента. Достатньо лише, щоб вона у середньому у заданому діапазоні аргументу

наближалася найкращим чином у цих точках. Очевидно така задача є розв'язною лише за умови, що вид, характер апроксимуючої залежності є заздалегідь відомим (наприклад, на основі попередніх знань про можливий характер змінування відповідної фізичної величини), а невизначеними є тільки значення кількох коефіцієнтів у цій залежності. Таке наближення називатимемо апроксимуванням функції (див. рис. 6.3).

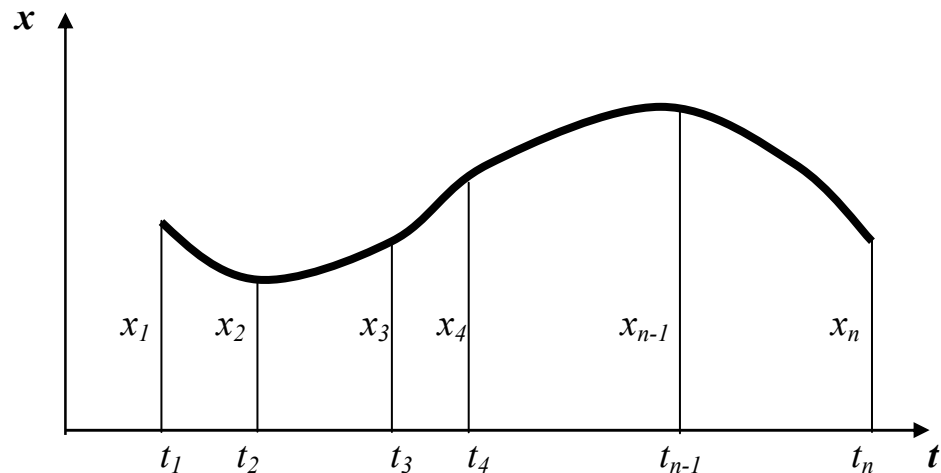


Рис. 6.3. Схема апроксимування таблично заданої функції

6.2. Інтерполювання поліноміальне

Отже, загальна постановка задачі інтерполювання така.

Задані значення x_1, x_2, \dots, x_n функції аргумента t при його значеннях t_1, t_2, \dots, t_n ($t_i < t_{i+1}$) відповідно. Побудувати неперервну функцію $X(t)$, що належить до заданого класу функцій, таку, що вона збігається з x_i при значеннях аргумента t_i ($i = 1, 2, \dots, n$).

У такому формулюванні розв'язок задачі є невизначеним, бо крізь задані точки можна провести безліч кривих. Тому загальну постановку дещо звужують, задаючи не тільки клас інтерполюючої функції, але й додаткову умову мінімальної її складності.

Наприклад, для найбільш поширеного *поліноміального інтерполювання* (при якому інтерполююча функція обирається серед поліномів аргументу t), додатковою умовою є мінімальний порядок інтерполюючого поліному. З цього випливає, що якщо первісну функцію задано лише двома точками, її треба інтерполювати поліномом першого порядку (крізь дві задані точки проходить єдина пряма), якщо трьома - параболою другого порядку і так далі. Взагалі функція, задана своїми n значеннями (у n точках) інтерполюється однозначно поліномом $(n - 1)$ -го порядку, тобто таким

$$X(t) = a_0 + a_1 \cdot t + a_2 \cdot t^2 + \dots + a_{n-1} \cdot t^{n-1}. \quad (6.1)$$

Тепер задача інтерполювання зветься до відшукування значень n невідомих коефіцієнтів полінома (1) з умови прийняття їм значень x_i при значеннях аргумента t_i ($i = 1, 2, \dots, n$).

Існують кілька способів визначення цих коефіцієнтів. Вони відрізняються методикою обчислень, зручною в одних і незручною в інших випадках. Але при ідеальних обчисленнях вони, природно, призводять до тих самих результатів, тобто до того самого поліному.

6.2.1. Інтерполювання за Лагранжем

За цією методикою попередньо визначають допоміжні поліноми $(n-1)$ -го порядку $L_{n-1}(t, t_i)$ такі, що

$$L_{n-1}(t_k, t_i) = \begin{cases} 0, & \text{якщо } k \neq i \\ 1, & \text{якщо } k = i \end{cases} \quad (6.2)$$

Тобто кожен з них приймає значення 1 тільки при $t = t_i$, а за рештою заданих значень аргумента він дорівнює нулю. Такі поліноми одержали назву лагранжєвих коефіцієнтів або множників впливу відповідних вузлів інтерполювання.

Щоб виконувалася перша умова (2), поліном $L_{n-1}(t, t_i)$ має мати вигляд:

$$L_{n-1}(t, t_i) = A(t-t_1)(t-t_2)\dots(t-t_{i-1})(t-t_{i+1})\dots(t-t_n), \quad (6.3)$$

тобто добутку $(n-1)$ різниць між поточним значенням аргументом і одним з заданих, окрім i -го, з деяким коефіцієнтом A . Друга умова (2) дозволяє визначити цей коефіцієнт A . Для цього у поліномі (3) слід покласти $t = t_i$ і прирівняти результат одиниці. З цього матимемо:

$$A = \frac{1}{(t_i - t_1)(t_i - t_2)\dots(t_i - t_{i-1})(t_i - t_{i+1})\dots(t_i - t_n)}.$$

Враховуючи це, одержимо остаточний вигляд поліному Лагранжа:

$$L_{n-1}(t, t_i) = \frac{(t-t_1)(t-t_2)\dots(t-t_{i-1})(t-t_{i+1})\dots(t-t_n)}{(t_i-t_1)(t_i-t_2)\dots(t_i-t_{i-1})(t_i-t_{i+1})\dots(t_i-t_n)}. \quad (6.4)$$

Тепер шуканий інтерполюючий поліном можна подати у вигляді

$$\begin{aligned} X(t) &= \sum_{i=1}^n L_{n-1}(t, t_i) \cdot x_i = \\ &= L_{n-1}(t, t_1) \cdot x_1 + L_{n-1}(t, t_2) \cdot x_2 + \dots + L_{n-1}(t, t_n) \cdot x_n = \\ &= \frac{(t-t_2)(t-t_3)\dots(t-t_n)}{(t_1-t_2)(t_1-t_3)\dots(t_1-t_n)} x_1 + \frac{(t-t_1)(t-t_3)\dots(t-t_n)}{(t_2-t_1)(t_2-t_3)\dots(t_2-t_n)} x_2 + \dots \\ &\dots + \frac{(t-t_1)(t-t_2)\dots(t-t_{n-1})}{(t_n-t_1)(t_n-t_2)\dots(t_n-t_{n-1})} x_n. \end{aligned} \quad (6.5)$$

Це й є інтерполяційний поліном Лагранжа степеня n .

Кількість арифметичних операцій для його обчислення дорівнює n^2 .

Інтерполювання за Лагранжем зручно використовувати тоді, коли ведеться багаторазове інтерполювання різних функцій за тих самих значень масиву аргументів. Тоді можна заздалегідь одноразово обчислити коефіцієнти Лагранжа, оскільки вони не залежать від функції, що інтерполюється.

Розглянемо часткові випадки.

Лінійна інтерполяція

У цьому разі маємо $n = 2$ два вузли інтерполяції. Інтерполяційний поліном Лагранжа має вигляд:

$$\begin{aligned} X(t) &= L_1(t, t_1) \cdot x_1 + L_1(t, t_2) \cdot x_2 = \frac{t - t_2}{t_1 - t_2} x_1 + \frac{t - t_1}{t_2 - t_1} x_2 = \\ &= \frac{1}{t_2 - t_1} [(x_1 t_2 - x_2 t_1) + (x_1 - x_2) t] \end{aligned} \quad (6.6)$$

Квадратична інтерполяція

У цьому випадку є три вузли інтерполяції ($n = 3$). Інтерполяційний поліном Лагранжа набуває виду:

$$\begin{aligned} X(t) &= L_2(t, t_1) \cdot x_1 + L_2(t, t_2) \cdot x_2 + L_2(t, t_3) \cdot x_3 = \\ &= \frac{(t - t_2)(t - t_3)}{(t_1 - t_2)(t_1 - t_3)} x_1 + \frac{(t - t_1)(t - t_3)}{(t_2 - t_1)(t_2 - t_3)} x_2 + \frac{(t - t_1)(t - t_2)}{(t_3 - t_1)(t_3 - t_2)} x_3 \end{aligned} \quad (6.7)$$

6.2.2. Інтерполювання за Ньютоном

Недоліком інтерполювання за Лагранжем є те, що якщо для поліпшення наближення додати ще один вузол інтерполювання, прийдеться усі обчислення проводити наново.

На практиці часто зустрічаються випадки, коли вузли інтерполяції стають відомими не одразу, а поступово, один за одним, у міру, наприклад, процесу вимірювання. Тоді зручно побудувати процес інтерполювання у такий спосіб, щоб поява даних про новий вузол інтерполювання, призводила б до необхідності мінімального перерахунку попередніх обчислень. Саме таку властивість має інтерполювання за Ньютоном.

Нехай вузли інтерполяції рівновідстоюють один від одного за аргументом, тобто виконується умова:

$$t_i - t_{i-1} = h; \quad (i = 2, 3, \dots, n) \quad (6.8)$$

Різниця

$$\Delta x_i = x_{i+1} - x_i \quad (6.9)$$

називають скінченними різницями першого порядку. Різниця сусідніх скінченних різниць першого порядку

$$\Delta^2 x_i = \Delta x_{i+1} - \Delta x_i = x_{i+2} - 2x_{i+1} + x_i \quad (6.10)$$

називаються скінченними різницями другого порядку. Аналогічно

$$\Delta^k x_i = \Delta^{k-1} x_{i+1} - \Delta^{k-1} x_i \quad (6.11)$$

є скінченними різницями k -го порядку.

Розглянемо поліном

$$X(t) = q_0 + q_1(t - t_1) + q_2(t - t_1)(t - t_2) + \dots + q_{n-1}(t - t_1)(t - t_2)\dots(t - t_{n-1}). \quad (6.12)$$

Визначимо його коефіцієнти.

Коефіцієнт q_0 визначимо з умови проходження графіка полінома через першу точку (x_1, t_1) :

$$q_0 = X(t_1) = x_1. \quad (6.13)$$

З умови проходження графіка полінома крізь точку (x_2, t_2) одержимо значення q_1 :

$$q_0 + q_1(t_2 - t_1) = X(t_2) = x_2; \Rightarrow q_1 = \frac{x_2 - x_1}{t_2 - t_1} = \frac{\Delta x_1}{h}. \quad (6.14)$$

Аналогічно визначається решта коефіцієнтів:

$$q_i = \frac{\Delta^i x_1}{i! \cdot h^i}. \quad (6.15)$$

Підставляючи у (12), одержуємо:

$$X(t) = x_1 + \sum_{k=1}^{n-1} \frac{\Delta^k x_1}{k! \cdot h^k} (t - t_1)\dots(t - t_k). \quad (6.16)$$

Це є перша інтерполяційна формула Ньютона (формула інтерполювання вперед).

Як бачимо, особливостями інтерполювання за Ньютоном є:

- при появі нового вузла додається лише новий член, решта не перераховується;
- коефіцієнти швидко зменшуються з зростанням k , бо у знаменнику міститься факторіал від k .

Іноді використовується формула для інтерполювання назад:

$$X(t) = x_n + \sum_{k=1}^{n-1} \frac{\Delta^k x_{n-k}}{k! \cdot h^k} (t - t_n)\dots(t - t_{n-k}).$$

Слід зазначити, що *при збільшенні кількості вузлів процес обчислення скінченних різниць стає все більш обчислювально нестійким* - похибка визначення скінченних різниць великого порядку різко зростає зі збільшенням порядку скінченної різниці. Тому метод Ньютона може бути застосований лише для невеликої кількості вузлів. Крім того, обмеженням у застосуванні цього методу може бути ще й необхідність, щоб *вузли інтерполяції були рівновіддаленими один від одного за аргументом*.

6.2.3. Інтерполювання за Ермітом

У більш загальному випадку потрібно, щоб в вузлах інтерполяції збігалися не лише значення інтерполюючої функції і функції, яку необхідно інтерполювати, але й значення їхніх похідних до деякого порядку. У цьому випадку застосовують інтерполювання за Ермітом.

ка збігалися. Одержані таким чином "склеєні" функції одержали назву сплайнів.

Найбільш поширеною є інтерполяція кубічними сплайнами, коли будь-які сусідні вузли інтерполяції з'єднуються поліномами третього порядку.

Визначення. Функція $X(t)$ називається кубічним сплайном, що інтерполює функцію $x(t)$, задану своїми значеннями x_i в узлах інтерполяції ($t = t_i$), якщо:

- вона є поліномом не вище за третього степеня на кожному з відрізків $[t_i, t_{i+1}]$ ($i = 1, \dots, n-1$);
- вона є неперервною у діапазоні $[t_1, t_n]$ разом з її похідними до другого порядку включно;
- у вузлах інтерполяції вона задовільняє умови інтерполяції

$$X(t_i) = x_i; \quad (i = 1, \dots, n). \quad (6.19)$$

Розглянемо докладніше процес аналітичної побудови такої інтерполюючої функції $X(t)$.

Будемо шукати її у вигляді

$$X(t) = a_{i0} + a_{i1} \cdot (t - t_i) + a_{i2} \cdot (t - t_i)^2 + a_{i3} \cdot (t - t_i)^3; \\ t_i < t < t_{i+1}; \quad (i = 1, \dots, n-1), \quad (6.20)$$

тобто кусково визначеної функції, що складається з відрізків кубічних парабол на вказаних інтервалах.

На кожному з $(n-1)$ відрізків кубічний сплайн (20) визначається чотирма параметрами $a_{i0}, a_{i1}, a_{i2}, a_{i3}$. На всьому діапазоні $t_i < t < t_{i+1}$ необхідно визначити $4(n-1)$ таких параметрів. Для цього потрібно використати n умов (19) таким чином: для першого (лівого) і останнього (правого) вузлів відповідна умова використовується один раз, для решти вузлів - подвійно (для лівого і правого відрізків кривої, що стикаються у цьому вузлі). Усього умови (19) дають $2+2(n-2)=2(n-1)$ умов для визначення коефіцієнтів сплайну. Окрім цього слід скористатися умовами неперервності першої та другої похідної сплайна у внутрішніх $(n-2)$ вузлах інтерполяції, тобто рівності цих похідних, обчислених для лівого і правого відрізків сплайна, що стикаються у цих вузлах. Це дає ще $2(n-2)$ умови. Усього, таким чином одержуємо $2(n-1)+2(n-2) = 4(n-1)-2$ умови.

Недостатні дві умови задаються додатково у вигляді обмежень на значення похідних від сплайна на кінцях діапазону. Тут можливі такі варіанти цих обмежень:

- задаються значення перших похідних на кінцях інтервалу

$$\frac{dX}{dt}(t_1) = x'(t_1); \quad \frac{dX}{dt}(t_n) = x'(t_n);$$

- задаються значення других похідних на кінцях інтервалу

$$\frac{d^2 X}{dt^2}(t_1) = x''(t_1); \quad \frac{d^2 X}{dt^2}(t_n) = x''(t_n);$$

часто використовується частковий випадок цієї умови - умови рівності нулю других похідних на кінцях діапазону

$$\frac{d^2 X}{dt^2}(t_1) = 0; \quad \frac{d^2 X}{dt^2}(t_n) = 0; \quad (6.21)$$

■ умови рівності відповідних похідних на кінцях інтервалу (періодичності при періоді $T = t_n - t_1$)

$$\frac{dX}{dt}(t_1) = \frac{dX}{dt}(t_n); \quad \frac{d^2 X}{dt^2}(t_1) = \frac{d^2 X}{dt^2}(t_n).$$

Введемо позначення:

■ кроку інтерполювання

$$h_i = t_{i+1} - t_i; \quad (6.22)$$

■ нахилу сплайна у i -му вузлі

$$m_i = \frac{dX}{dt}(t_i); \quad (6.23)$$

■ момента сплайна у i -му вузлі

$$M_i = \frac{d^2 X}{dt^2}(t_i). \quad (6.24)$$

Через те, що $X(t)$ є кубічним поліномом, його друга похідна є лінійною функцією t на кожному відрізку. Тому для її визначення можна скористатися формулою лінійної інтерполяції

$$\frac{d^2 X}{dt^2}(t) = M_i + \frac{M_{i+1} - M_i}{h_i}(t - t_i);$$

$$t_i < t < t_{i+1}; \quad (i = 1, \dots, n-1). \quad (6.25)$$

Інтегруючи, одержуємо

$$\frac{dX}{dt}(t) = m_i + M_i(t - t_i) + \frac{M_{i+1} - M_i}{2h_i}(t - t_i)^2;$$

$$t_i < t < t_{i+1}; \quad (i = 1, \dots, n-1). \quad (6.26)$$

Проінтегруємо ще раз, враховуючи умови (19):

$$X(t) = x_i + m_i(t - t_i) + \frac{M_i}{2}(t - t_i)^2 + \frac{M_{i+1} - M_i}{6h_i}(t - t_i)^3;$$

$$t_i < t < t_{i+1}; \quad (i = 1, \dots, n-1). \quad (6.27)$$

Підставимо у (27) $t = t_{i+1}$. Тоді з врахуванням (17) маємо:

$$x_{i+1} = x_i + m_i(t_{i+1} - t_i) + \frac{M_i}{2}(t_{i+1} - t_i)^2 + \frac{M_{i+1} - M_i}{6h_i}(t_{i+1} - t_i)^3.$$

Звідси знайдемо вираз для m_i :

$$m_i = \frac{x_{i+1} - x_i}{h_i} - \frac{h_i}{6}(2M_i + M_{i+1}). \quad (6.28)$$

Підставимо (28) у (27) і (26):

$$X(t) = x_i + \left[\frac{x_{i+1} - x_i}{h_i} - \frac{h_i}{6} (2M_i + M_{i+1}) \right] (t - t_i) + \frac{M_i}{2} (t - t_i)^2 + \frac{M_{i+1} - M_i}{6h_i} (t - t_i)^3; \quad (6.29)$$

$$\frac{dX}{dt}(t) = \frac{x_{i+1} - x_i}{h_i} - \frac{h_i}{6} (2M_i + M_{i+1}) + M_i (t - t_i) + \frac{M_{i+1} - M_i}{2h_i} (t - t_i)^2. \quad (6.30)$$

Порівнюючи (29) з (20), одержимо такі вирази для визначення параметрів

$$a_{i0} = x_i; \quad a_{i1} = \frac{x_{i+1} - x_i}{h_i} - \frac{h_i}{6} (2M_i + M_{i+1});$$

$$a_{i2} = \frac{M_i}{2}; \quad a_{i3} = \frac{M_{i+1} - M_i}{6h_i}. \quad (6.31)$$

Ці коефіцієнти визначені з умов, що відповідний відрізок проходить через обидві точки вузлів інтерполяції відповідного піддіапазону і що другі похідні двох відрізків сплайну, що стикуються у певному вузлі інтерполяції, дорівнюють одна одній. Skorистуємося тепер тим, що і перші похідні у внутрішніх вузлах мають дорівнювати одна одній. Для цього у (30) покладемо $i = i + 1$:

$$\frac{dX}{dt}(t) = \frac{x_{i+2} - x_{i+1}}{h_{i+1}} - \frac{h_{i+1}}{6} (2M_{i+1} + M_{i+2}) + M_{i+1} (t - t_{i+1}) + \frac{M_{i+2} - M_{i+1}}{2h_{i+1}} (t - t_{i+1})^2. \quad (6.32)$$

У точці $t = t_{i+1}$ значення (30) і (32) мають збігтися:

$$\frac{x_{i+2} - x_{i+1}}{h_{i+1}} - \frac{h_{i+1}}{6} (2M_{i+1} + M_{i+2}) =$$

$$= \frac{x_{i+1} - x_i}{h_i} - \frac{h_i}{6} (2M_i + M_{i+1}) + M_i h_i + \frac{M_{i+1} - M_i}{2} h_i,$$

звідки випливає

$$M_i h_i + 2(h_i + h_{i+1}) M_{i+1} + h_{i+1} M_{i+2} = 6 \left(\frac{x_{i+2} - x_{i+1}}{h_{i+1}} - \frac{x_{i+1} - x_i}{h_i} \right); \quad (6.33)$$

$$(i = 1, \dots, n - 2).$$

Але загальна кількість невідомих моментів M_i дорівнює n . Тому потрібно використати додаткові умови. Наприклад, якщо скористатися умовами (21), одержимо з (25):

$$M_1 = 0; \quad \frac{d^2 X}{dt^2}(t_n) = M_{n-1} + \frac{M_n - M_{n-1}}{h_{n-1}} (t_n - t_{n-1}) = M_n = 0$$

Тоді матимемо таку систему лінійних алгебричних рівнянь

$$2(h_{1i} + h_2) M_2 + h_2 M_3 = 6 \left(\frac{x_3 - x_2}{h_2} - \frac{x_2 - x_1}{h_1} \right);$$

$$M_i h_i + 2(h_i + h_{i+1}) M_{i+1} + h_{i+1} M_{i+2} = 6 \left(\frac{x_{i+2} - x_{i+1}}{h_{i+1}} - \frac{x_{i+1} - x_i}{h_i} \right); \quad (i = 2, \dots, n-3)$$

$$M_{n-2} h_{n-2} + 2(h_{n-2} + h_{n-1}) M_{n-1} = 6 \left(\frac{x_n - x_{n-1}}{h_{n-1}} - \frac{x_{n-1} - x_{n-2}}{h_{n-2}} \right)$$

$(n-2)$ -го порядку для визначення $(n-2)$ невідомих моментів M_2, M_3, \dots, M_{n-1} .

6.4. Засоби інтерполювання на ЕОМ

На завершення охарактеризуємо засоби інтерполювання, що мають сучасні мови програмування.

Універсальні мови програмування не мають засобів інтерполювання взагалі. Це обумовлено саме їхньою універсальністю, бо включення у їх склад великої кількості математичних процедур зробило б їх надто складними.

З розглядуваних засобів програмування персональних ЕОМ тільки система MatLAB має процедури інтерполювання.

Найпростіша і найуживаніша з них - процедура *interp1*. У загальному випадку звернення до неї у середовищі MatLAB здійснюється за наступною формою

$$Y_i = \text{interp1}(X, Y, X_i, \text{'method'}),$$

де Y - заданий вектор значень функції, яка інтерполюється, при значеннях аргумента, заданих у векторі X ; X_i - вектор значень аргументів, при яких треба визначити значення інтерполюючої функції; 'method' - символічна (рядкова) змінна, яка визначає метод інтерполювання і яка може набувати лише наступних значень:

'nearest' - східчаста інтерполяція;

'linear' - лінійна інтерполяція;

'spline' - інтерполяція кубічним сплайном;

'cubic' - інтерполяція кубічним поліномом;

Y_i - вектор значень інтерполюючої функції, що відповідає заданому вектору X_i аргументів.

Якщо останній параметр (назву метода) не вказано, застосовується лінійна інтерполяція.

Нижче наведений текст програми, яка демонструє можливості функції *interp1*:

```
% InterpDemo.m
% Програма, що демонструє використання функції INTERP1
% задля здійснення інтерполювання
% Лазарєв Ю.Ф., 21-08-2000 р.
x=[2 3 4 5];
y=[0 -3 5 3];
xi=1.4:0.1:5.6;
y1=interp1(x,y,xi,'nearest');
y2=interp1(x,y,xi);
```

```

y3=interp1(x,y,xi,'spline');
y4=interp1(x,y,xi,'cubic');
plot(x,y,'kS',xi,y1,'k.',xi,y2,'k+',xi,y3,'k--',xi,y4,'ko')
grid
set(gca,'FontSize',10)
legend('Задані точки','nearest','linear','spline','cubic')
set(gca,'FontSize',12)
title('Приклади інтерполяції функцією INTERP1')
xlabel('Аргумент')
ylabel('Значення функції')

```

Результат роботи цієї програми поданий на рис. 6.4.

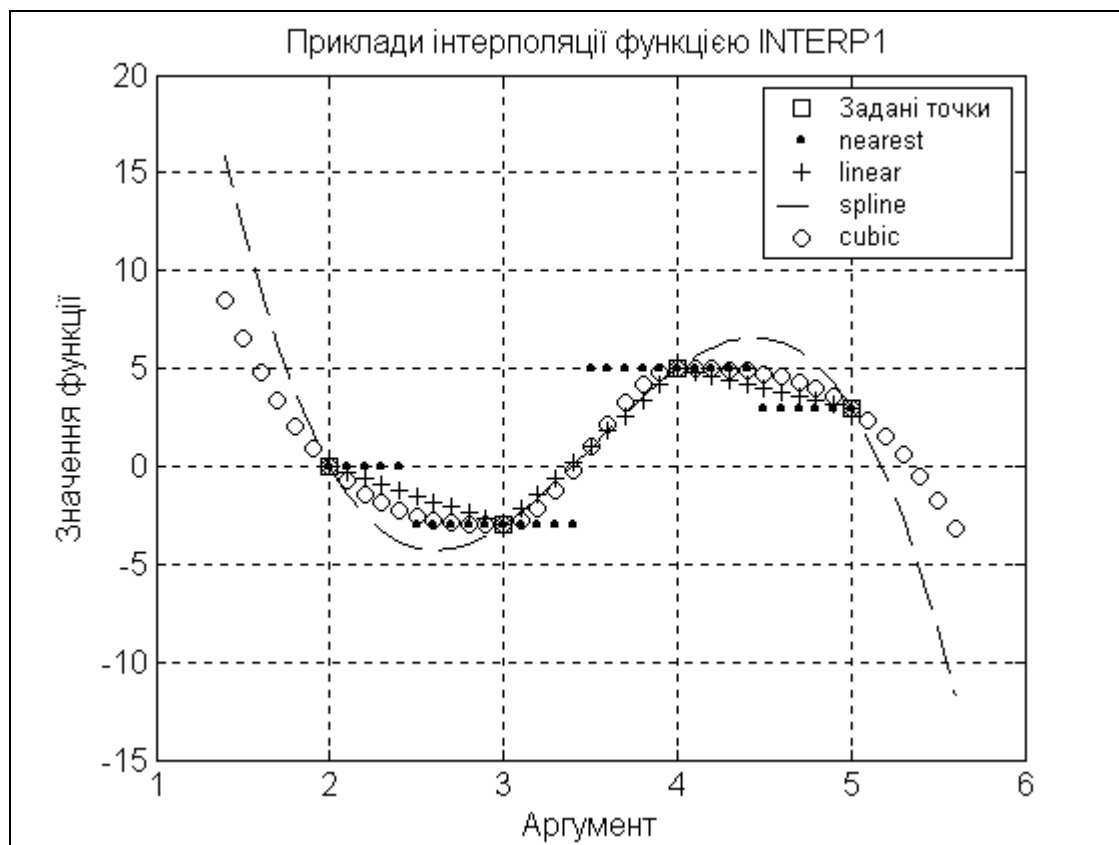


Рис. 6.4. Результати застосування процедури INTERP1

Фактично методи 'nearest', 'linear' і 'cubic' здійснюють інтерполювання сплайнами відповідно нульового, першого і третього порядків. У першому з них інтерполююча функція складається з відрізків прямих, паралельних осі аргументів, що проходять крізь вузли інтерполяції і займають приблизно половину інтервалів у обидва боки від вузла інтерполяції (див. рис. 6.4, крива з окремих точок). Другий метод - лінійне інтерполювання - з'єднує сусідні вузли інтерполяції відрізками прямих. Її зображено штриховою лінією на рис. 6.4. Метод 'cubic'

здійснює розглянуту раніше інтерполяцію кубічними сплайнами з додатковими умовами (21). Відповідну інтерполюючу криву зображено суцільною лінією.

Метод 'spline', всупереч назві, будує інтерполюючу криву, проводячи поліном третього порядку без додаткових умов. Замість цього, перші і останні кубічні поліноми з'єднують вже не два сусідні вузли інтерполяції, а три, іноді навіть чотири. Побудована цим методом інтерполююча крива подана штрих-пунктирною лінією.

6.5. Апроксимування. Метод найменших квадратів

Резюмуючи попереднє, можна зазначити, що інтерполювання може бути здійснено лише на невеликому інтервалі по кількох вузлах інтерполяції, процес обчислення скінченних різниць є нестійким.

Окрім того, якщо значення x_i подають значення функції, яка наближується, зі значними похибками, інтерполювати ці значення не доцільно.

За таких умов застосовують середньо-квадратичне наближення. Найбільш ефективним методом побудови середньо-квадратичного наближення функції є *метод найменших квадратів (МНК)*.

Нехай є відомими n значень x_i ($i = 1, 2, \dots, n$) деякої фізичної величини $x(t)$, вимірної у моменти часу t_i . Припустимо, що ці значення подають істинні значення функції $x(t)$ у відповідні моменти часу зі значними похибками, значення яких невідомі, але припускається, що ці похибки є випадковими з математичним сподіванням, що дорівнює нулю.

Будемо наближати невідому функцію $x(t)$ за допомогою лінійної комбінації деяких відомих m функцій $\varphi_k(t)$:

$$X(t) = \sum_{k=0}^{m-1} c_k \cdot \varphi_k(t), \quad (6.34)$$

де функції $\varphi_0(t), \varphi_1(t), \dots, \varphi_{m-1}(t)$ називатимемо *базовими функціями*.

Задачею є визначення m невідомих коефіцієнтів c_k ($k = 0, 1, \dots, m-1$) з умови, щоб сума квадратів відхилення апроксимуючої функції $X(t)$ від апроксимованої $x(t)$, обчислених за заданих значень t_i аргумента t

$$\sigma^2(c, \varphi) = \sum_{i=1}^n (x_i - X(t_i))^2 = \sum_{i=1}^n (x_i - \sum_{k=0}^{m-1} c_k \cdot \varphi_k(t_i))^2 \quad (6.35)$$

була мінімальною (саме тому відповідний метод називається МНК).

Сума (35) квадратів відхилень є функцією m невідомих коефіцієнтів c_k ($k = 0, 1, \dots, m-1$). Тому для відшукування його мінімуму слід знайти m часткових похідних від цього квадрата по окремих коефіцієнтах

$$\frac{d\sigma^2(c, \varphi)}{dc_k} = -2 \sum_{i=1}^n (x_i - \sum_{k=0}^{m-1} c_k \cdot \varphi_k(t_i)) \cdot \varphi_k(t_i); \quad (6.36)$$

$$(k = 0, 1, \dots, m-1)$$

і прирівняти їх нулю. В результаті одержується система з m лінійних алгебричних рівнянь з m невідомими c_0, c_1, \dots, c_{m-1} :

$$\left\{ \begin{array}{l} c_0 \sum_{i=1}^n \varphi_0^2(t_i) + c_1 \sum_{i=1}^n \varphi_0(t_i) \cdot \varphi_1(t_i) + \dots + c_{m-1} \sum_{i=1}^n \varphi_0(t_i) \cdot \varphi_{m-1}(t_i) = \sum_{i=1}^n x_i \cdot \varphi_0(t_i) \\ c_0 \sum_{i=1}^n \varphi_0(t_i) \cdot \varphi_1(t_i) + c_1 \sum_{i=1}^n \varphi_1^2(t_i) + \dots + c_{m-1} \sum_{i=1}^n \varphi_1(t_i) \cdot \varphi_{m-1}(t_i) = \sum_{i=1}^n x_i \cdot \varphi_1(t_i) \\ \dots \\ c_0 \sum_{i=1}^n \varphi_0(t_i) \cdot \varphi_{m-1}(t_i) + c_1 \sum_{i=1}^n \varphi_1(t_i) \cdot \varphi_{m-1}(t_i) + \dots + c_{m-1} \sum_{i=1}^n \varphi_{m-1}^2(t_i) = \sum_{i=1}^n x_i \cdot \varphi_{m-1}(t_i) \end{array} \right. \quad (6.37)$$

Визначником цієї системи є визначник Грама сукупності функцій $\varphi_k(t)$:

$$\Delta = \begin{bmatrix} \sum_{i=1}^n \varphi_0^2(t_i) & \sum_{i=1}^n \varphi_0(t_i) \cdot \varphi_1(t_i) & \dots & \sum_{i=1}^n \varphi_0(t_i) \cdot \varphi_{m-1}(t_i) \\ \sum_{i=1}^n \varphi_0(t_i) \cdot \varphi_1(t_i) & \sum_{i=1}^n \varphi_1^2(t_i) & \dots & \sum_{i=1}^n \varphi_1(t_i) \cdot \varphi_{m-1}(t_i) \\ \dots & \dots & \dots & \dots \\ \sum_{i=1}^n \varphi_0(t_i) \cdot \varphi_{m-1}(t_i) & \sum_{i=1}^n \varphi_1(t_i) \cdot \varphi_{m-1}(t_i) & \dots & \sum_{i=1}^n \varphi_{m-1}^2(t_i) \end{bmatrix} \quad (6.38)$$

Як відомо, якщо функції $\varphi_k(t)$ ($k = 0, 1, \dots, m-1$) складають сукупність взаємо-незалежних функцій (тобто ніяку з цих функцій неможливо подати як лінійну комбінацію решти з них), то визначник Грама (38) цих функцій не дорівнює нулю.

Це означає, що як базові функції при апроксимуванні слід обирати сукупності незалежних функцій. Тоді рівняння (37) можуть бути розв'язані, і знайдені значення коефіцієнтів c_k ($k = 0, 1, \dots, m-1$), що забезпечують мінімум квадрата середньо-квадратичного відхилення апроксимуючої і апроксимованої функцій. Після визначення значень коефіцієнтів, шукана апроксимуюча функція визначиться виразом (34).

Ортогональними на деякому інтервалі $[t_1, t_n]$ функціями називається сукупність таких функцій, що

$$\sum_{i=1}^n \varphi_k(t_i) \varphi_j(t_i) = \begin{cases} 1 & \text{при } k = j \\ 0 & \text{при } k \neq j \end{cases}$$

Матриця Грама для ортогональних функцій є одиничною.

У випадку, коли як базові при апроксимуванні обрані ортогональні функції, обчислення коефіцієнтів апроксимації значно скорочується. У цьому випадку значення їх можна визначити співвідношенням:

$$c_k = \sum_{i=1}^n x_i \cdot \varphi_k(t_i). \quad (6.39)$$

Тому при апроксимуванні бажано обирати як базові системи ортогональних функцій.

Класичними прикладами ортогональних функцій-поліномів є поліноми Якобі, Лежандра, Лагерра, Чебишева, Ерміта. Наприклад, поліноми Лежандра $P_n(t)$ є ортогональними на відрізку $[-1; 1]$ і мають вигляд:

$$P_0(t) = 1; \quad P_1(t) = t; \quad P_2(t) = \frac{1}{2}(3t^2 - 1); \quad P_3(t) = \frac{1}{2}(5t^3 - 3t);$$

$$P_4(t) = \frac{1}{8}(35t^4 - 30t^2 + 3); \quad P_5(t) = \frac{1}{8}(63t^5 - 70t^3 + 15t);$$

$$P_6(t) = \frac{1}{16}(231t^6 - 315t^4 + 105t^2 - 5);$$

$$P_7(t) = \frac{1}{16}(429t^7 - 693t^5 + 315t^3 - 35t).$$

Поліноми Чебишева першого роду $T_n(t)$ є ортогональними також на інтервалі $[-1; 1]$. Їхній явний вираз є таким

$$T_n(t) = \cos(n \cdot \arccos(t)); \quad (n = 0, 1, 2, \dots),$$

и рекурентна формула їх визначення має такий вид

$$T_{n+1}(t) = 2t \cdot T_n(t) \cdot T_{n-1}(t); \quad (n = 1, 2, \dots).$$

Наведемо явні вирази поліномів Чебишева першого роду:

$$T_0(t) = 1; \quad T_1(t) = t; \quad T_2(t) = 2t^2 - 1; \quad T_3(t) = 4t^3 - 3t;$$

$$T_4(t) = 8t^4 - 8t^2 + 1; \quad T_5(t) = 16t^5 - 20t^3 + 5t;$$

$$T_6(t) = 32t^6 - 48t^4 + 18t^2 - 1; \quad T_7(t) = 64t^7 - 112t^5 + 56t^3 - 7t \dots$$

Поліноми Чебишева другого роду $U_n(t)$ також ортогональні на тому ж інтервалі і мають наступний вигляд:

$$U_0(t) = 1; \quad U_1(t) = 2t; \quad U_2(t) = 4t^2 - 1; \quad U_3(t) = 8t^3 - 4t;$$

$$U_4(t) = 16t^4 - 12t^2 + 1; \quad U_5(t) = 32t^5 - 32t^3 + 6t.$$

Поліноми Ерміта $H_n(t)$ ортогональні на всій числовій осі $(-\infty, +\infty)$ і мають вигляд

$$H_0(t) = 1; \quad H_1(t) = 2t; \quad H_2(t) = 4t^2 - 2; \quad H_3(t) = 8t^3 - 12t;$$

$$H_4(t) = 16t^4 - 48t^2 + 12; \quad H_5(t) = 32t^5 - 160t^3 + 120t \dots$$

Наведені системи ортогональних поліномів стають у нагоді, коли як апроксимуюча функція обирається поліном певного степеня, тобто для здійснення так званого поліноміального апроксимування.

Прикладом системи неортогональних базових поліномів може бути така система:

$$P_0(t) = 1; \quad P_1(t) = t; \quad \dots \quad P_k = t^k; \quad \dots$$

При цьому, природньо, у результаті апроксимування певної сукупності даних в усіх випадках одержується той самий поліном. Різниця полягає лише у зручності, простоті одержання коефіцієнтів цього поліному.

Якщо при поліноміальній апроксимації кількість базових функцій-поліномів дорівнює 2, тобто $m = 2$, апроксимація називається лінійною. В результаті лінійного апроксимування одержують так звану лінію регресії (пряму). При $m = 3$ апроксимування називають квадратичним, а при $m = 4$ - кубічним.

Звичайно, апроксимування не обов'язково має бути поліноміальним. Наприклад, якщо відомо, що вимірювана функція є періодичною з відомим періодом $T = \frac{2\pi}{\omega}$, де ω - кругова частота, то як базові функції зручно використовувати таку сукупність:

$$\begin{aligned} \varphi_0(t) &= 1; \quad \varphi_1(t) = \sin(\omega t); \quad \varphi_2(t) = \cos(\omega t); \quad \dots \\ \dots \quad \varphi_{2k-1}(t) &= \sin(k\omega t); \quad \varphi_{2k}(t) = \cos(k\omega t); \quad \dots, \end{aligned}$$

тобто використовувати апроксимування у вигляді ряду Фур'є. Тут k є цілим додатним числом, яке дорівнює номеру гармоніки у розкладі Фур'є.

Наведена сукупність функцій є ортогональною на інтервалі, кратному періодові T . Тому застосування її є вельми ефективним (потребує мінімуму обчислень), якщо як інтервал вимірювання обрати кратним періодові.

6.6. Засоби апроксимування на ЕОМ

Повертаючись до характеристики програмних засобів апроксимування, слід зауважити, що в універсальних мовах програмування не передбачено спеціальних вбудованих процедур, що здійснювали б апроксимування. Лише у системі MatLAB присутні такі засоби. Найбільш вживаною тут є процедура ***polyfit***, яка виконує поліноміальне апроксимування.

Якщо звернутися до цієї процедури у такий спосіб

$$P = \text{polyfit}(X, Y, n),$$

то можна одержати вектор P коефіцієнтів полінома порядку n , який апроксимує вимірювану функцію за заданими векторами Y і X вимірених її значень і відповідних значень її аргументу.

Розглянемо приклад. Нехай масив значень аргументу є наступним:

$$x = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8],$$

а масив відповідних значень вимірної величини - таким:

$$y = [-1. \ 1 \ 0.2 \ 0.5 \ 0.8 \ 0.7 \ 0.6 \ 0.4 \ 0.1].$$

Тоді, застосовуючи зазначену процедуру при різних значеннях порядку апроксимуючого поліному, одержимо:

```

» x = [1 2 3 4 5 6 7 8];
» y = [-1.1 0.2 0.5 0.8 0.7 0.6 0.4 0.1];
» polyfit(x,y,1)
ans = 0.1143 -0.2393
» polyfit(x,y,2)
ans = -0.1024 1.0357 -1.7750
» polyfit(x,y,3)
ans = 0.0177 -0.3410 1.9461 -2.6500
» polyfit(x,y,4)
ans = -0.0044 0.0961 -0.8146 3.0326 -3.3893.

```

Це означає, що задану послідовність можна апроксимувати або прямою

$$y(x) = 0,1143x - 0,2393,$$

або квадратною параболою

$$y(x) = -0,1024x^2 + 1,0357x - 1,7750,$$

або кубічною параболою

$$y(x) = 0,0177x^3 - 0,341x^2 + 1,9461x - 2,65,$$

або параболою четвертого степеня

$$y(x) = -0,0044x^4 + 0,0961x^3 - 0,8146x^2 + 3,0326x - 3,3893.$$

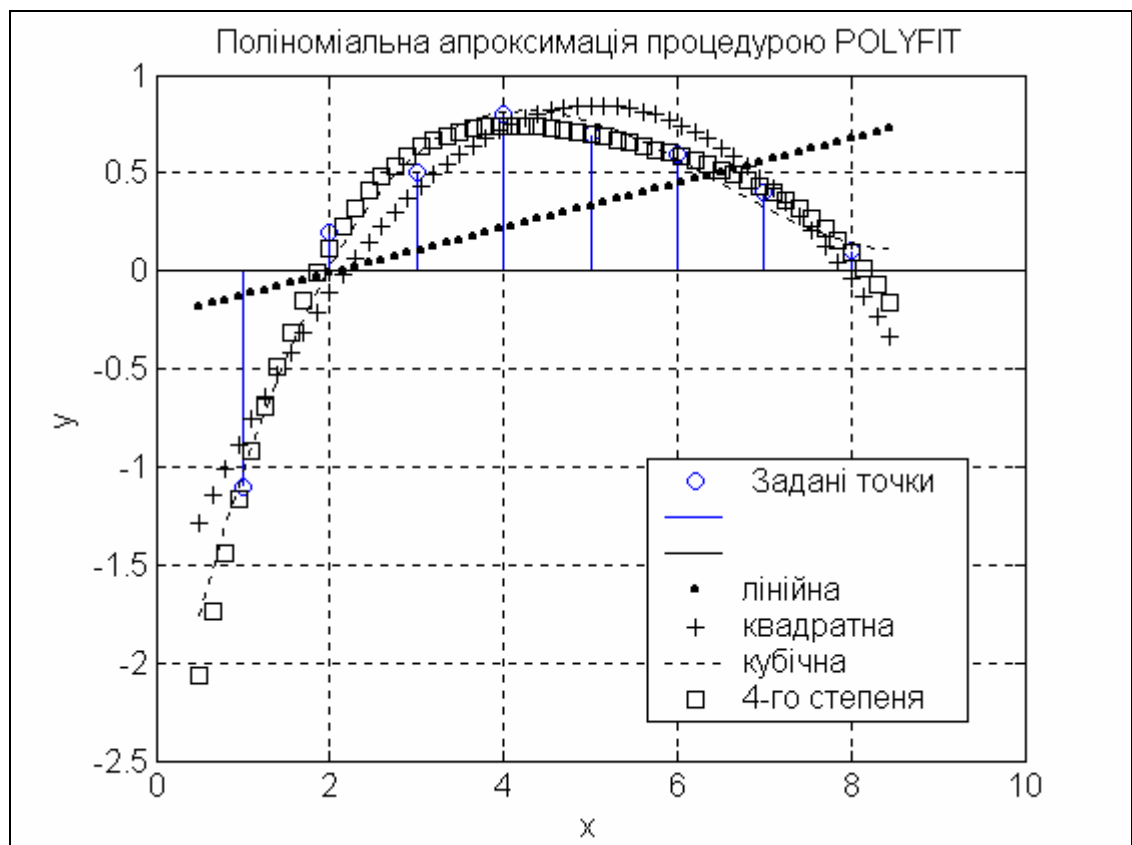


Рис. 6.5. Приклад застосування процедури POLYFIT

Побудуємо у одному графічному полі графіки заданої гратчастої функції і графіки усіх отриманих при апроксимації поліномів:

```
x = [1 2 3 4 5 6 7 8];
y = [-1.1 0.2 0.5 0.8 0.7 0.6 0.4 0.1];
P1=polyfit(x,y,1); P2=polyfit(x,y,2);
P3=polyfit(x,y,3); P4=polyfit(x,y,4);
stem(x,y);
x1 = 0.5 : 0.15 : 8.5;
y1=polyval(P1,x1); y2=polyval(P2,x1);
y3=polyval(P3,x1); y4=polyval(P4,x1);
hold on
plot(x1,y1,'k.',x1,y2,'k+',x1,y3,'k:',x1,y4,'ks'),
grid, set(gca, 'FontName', 'Arial Cyr', 'FontSize', 12),
title('Поліноміальна апроксимація процедурою POLYFIT');
```

```
xlabel('x'); ylabel('y')
legend('Задані точки',' ',' ','лінійна','квадратна','кубічна','4-го степеня',0)
```

Результат наведений на рис. 6.5.

6.7. Створення у середовищі MatLAB програми дослідження похибок апроксимації гармонічної функції

6.7.1. Мета створення програми і постановка задачі

Програма утворюється задля забезпечення можливості обчислення похибок апроксимування деякого вимірюваного процесу, який є гармонічно змінюваним процесом, спотвореним випадковими похибками вимірювання з нормальним законом розподілу, а також задля забезпечення можливості досліджування впливу на похибки цього апроксимування таких параметрів, як кількість вимірних точок, інтервалу змінювання аргументу і інтенсивності шуму.

Припустимо, що є деякий гармонічний процес

$$x(t) = a_0 + a_s \cdot \sin \omega t + a_c \cdot \cos \omega t = a_0 + a_m \cdot \sin(\omega t + \varphi), \quad (6.40)$$

де ω - колова частота цього процесу; a_m - амплітуда; φ - його початкова фаза a_0 - стала складова процесу, причому частота процесу відома апріорі точно.

Нехай цей процес вимірюється у окремі моменти часу, розділені однако-вим проміжком h , протягом проміжку часу t від 0 до T_k . При цьому при кожному з цих n вимірювань до істинного значення $x(t_i)$ вимірюваного сигналу додається значна похибка S_i (шум) вимірювання, яка є випадковою нормально розподіленою величиною із нульовим математичним сподіванням (середнім значенням) і стандартним відхиленням I_{nt} , яке визначає інтенсивність шуму вимірювання. У результаті одержується послідовність

$$X_i = x(t_i) + S_i, \quad (i = 1, 2, \dots, n), \quad (6.41)$$

яка є вхідною заданою послідовністю, яку потрібно апроксимувати.

При апроксимуванні вважатимемо, що нам відомі дві обставини:

- 1) що вимірний процес насправді є гармонічно змінюваним з часом із сталою складовою;
- 2) період T (частота ω) цього гармонічного процесу є відомим (точно, без похибок).

Виходячи з цього, як базові функції зручно використати такі

$$\varphi_1(t) = 1; \quad \varphi_2(t) = \sin \omega t; \quad \varphi_3(t) = \cos \omega t,$$

тобто апроксимуючу функцію обрати у вигляді

$$y(t) = A_0 + A_s \cdot \sin \omega t + A_c \cdot \cos \omega t, \quad (6.42)$$

де коефіцієнти A_0 , A_s і A_c підлягають визначенню за допомогою процесу апроксимування.

У відповідності до методу найменших квадратів сформуємо функціонал

$$\Phi(A_0, A_s, A_c) = \sum_{i=1}^n (y(t_i) - X_i)^2 = \sum_{i=1}^n (A_0 + A_s \cdot \sin \omega t_i + A_c \cdot \cos \omega t_i - X_i)^2.$$

Якщо відшукати похідні від цього функціоналу по шуканих коефіцієнтах і прирівняти їх нулеві, одержимо рівняння (див. 6.37)

$$\left\{ \begin{array}{l} A_0 \sum_{i=1}^n 1 + A_s \sum_{i=1}^n \sin \omega t_i + A_c \sum_{i=1}^n \cos \omega t_i = \sum_{i=1}^n X_i \\ A_0 \sum_{i=1}^n \sin \omega t_i + A_s \sum_{i=1}^n \sin^2 \omega t_i + A_c \sum_{i=1}^n \sin \omega t_i \cdot \cos \omega t_i = \sum_{i=1}^n X_i \cdot \sin \omega t_i \\ \dots \\ A_0 \sum_{i=1}^n \cos \omega t_i + A_s \sum_{i=1}^n \sin \omega t_i \cdot \cos \omega t_i + A_c \sum_{i=1}^n \cos^2 \omega t_i = \sum_{i=1}^n X_i \cdot \cos \omega t_i \end{array} \right. \quad (6.43)$$

Обчислення коефіцієнтів при невідомих A_0 , A_s і A_c значно спрощується, якщо проміжок, всередині якого вимірено процес, узяти кратним періоду шуканої гармонічної функції. У цьому випадку, як неважко впевнитися, усі не-діагональні коефіцієнти дорівнюватимуть нулеві, а діагональні коефіцієнти дорівнюють:

$$\sum_{i=1}^n 1 = n; \quad \sum_{i=1}^n \sin^2 \omega t_i = \frac{n}{2}; \quad \sum_{i=1}^n \cos^2 \omega t_i = \frac{n}{2}.$$

Враховуючи це у (6.43), дійдемо до шуканих формул:

$$\left\{ \begin{array}{l} A_0 = \frac{1}{n} \sum_{i=1}^n X_i \\ A_s = \frac{2}{n} \sum_{i=1}^n X_i \cdot \sin \omega t_i \\ \dots \\ A_c = \frac{2}{n} \sum_{i=1}^n X_i \cdot \cos \omega t_i \end{array} \right. \quad (6.44)$$

Як інтегральні показники похибки відновлення первісного сигналу (6.40) зручно обрати різниці між відшуканими коефіцієнтами (6.44) і відповідними точними їхніми значеннями a_0 , a_s і a_c :

$$\Delta_0 = A_0 - a_0; \quad \Delta_s = A_s - a_s; \quad \Delta_c = A_c - a_c. \quad (6.45)$$

Програма має дозволити у зручній інтерактивній формі провести дослідження похибок відновлення гармонічного сигналу із заданою частотою, який є відомим своєю вимірною послідовністю (6.41).

6.7.2. Створення програми

Найбільш зручно і наочно для розв'язування задачі, що поставлена, використати можливості інтерактивного пакету SimuLink, який входить у склад системи MatLab.

Пакет SimuLink забезпечує можливість так званого "візуального" програмування, коли математична програма утворюється не шляхом складання

тексту програми мовою високого рівня, а шляхом складання схеми з'єднань (блок-схеми) окремих візуальних блоків, кожний з яких фактично є певною математичною програмою. Набір готових блоків зосереджено у спеціальній бібліотеці блоків з аналогічною назвою SimuLink. Зображення блоків перетягуються з бібліотеки у спеціальне графічне вікно утвореної блок-схеми мишкою за технологією Drag-end-Drop (перетягни і залиш). Потім перетягнуті блоки з'єднуються між собою за тієї ж технологією так, щоб виходи одних блоків були з'єднані зі входами деяких інших, утворюючи логічно обгрунтовану блок-схему. При цьому виходи блоків фактично презентують собою вихідні (обчислені блоком (відповідною програмою)) величини, а входи – вхідні величини, потрібні для здійснення обчислень програмою блока. У цілому готова блок-схема презентує деяку (іноді досить складну) математичну програму, яку у цьому випадку називають S-моделлю. При запису блок-схеми на диск, вона набуває форму так званого MDL-файла.

Після закінчення складання блок-схеми можна здійснити її запуск до виконання. Виконання полягає у тому, що пакет SimuLink здійснює автоматичне змінювання деякого головного параметра, який інтерпретується як модельний час, за певним алгоритмом, параметри якого встановлюються при налаштуванні S-моделі. У відповідності з поточним значенням модельного часу усі величини, що входять у блоки, перераховуються, блоки здійснюють відповідні перетворення цих величин і у такий спосіб обчислюються поточні (відповідні до поточного моменту часу) значення шуканих величин.

Бібліотека SimuLink містить кілька розділів за типом задач, які розв'язуються блоками, що входять в них. Більшість розділів складаються з блоків, які здійснюють певні типові перетворення одних величин (входів блока) у інші величини (виходи блока). Але будь-яка блок-схема має містити хоча б один блок, в якому немає входу (є лише вихід), призначення якого – формувати на виході деяку послідовність чисел, значення членів якої змінюється у залежності від значення модельного часу. Такі блоки можна назвати блоками-джерелами. Вони генерують на виході певного виду сигнали як функцію часу. Блоки такого виду зосереджені у розділі Sources (Джерела) бібліотеки Simulink.

Окрім того у блок-схемі обов'язково має бути хоча б один блок, який здійснює зворотну операцію – робить досяжним для спостереження значення якоїсь величини (сигналу), яка обчислюється у блок-схемі і яка є потрібною для аналізу результатів моделювання. У таких блоків є тільки вхід і немає виходу. Їх називають блоками-приймачами. Вони містяться у розділі Sinks (Приймачи) бібліотеки Simulink. Найбільш уживаним з блоків-приймачів є блок Scope, завдяки якому сигнал, що надається до його входу, виводиться у вигляді графіка залежності його величини від модельного часу у окреме графічне вікно. У такий спосіб можна здійснити візуалізацію результатів моделювання за допомогою S-моделі.

Налаштування блоків у блок-схемі здійснюється шляхом попереднього виклику вікна налаштування блока. Для цього потрібно подвійно клацнути

мишкою на зображенні блока у блок-схемі. У вікні настроювання зазвичай є кілька вікон введення, у які можна ввести власні значення параметрів блока.

Головні відомості про особливості пакету Simulink і його використання наведені у [8, глава 7], [9, уроки 7 і 8], а також у розділі 9 цього посібника.

Використовуючи пакет Simulink, неважко скласти блок-схему формування первісного гармонічного сигналу зі сталою складовою, а також – зашумленого сигналу. Така блок-схема наведена на рис. 6.5. Запускаючи цю схему на моделювання, можна впевнитися, розкриваючи вікно блока *Scope*, що результатом моделювання є бажаний процес.

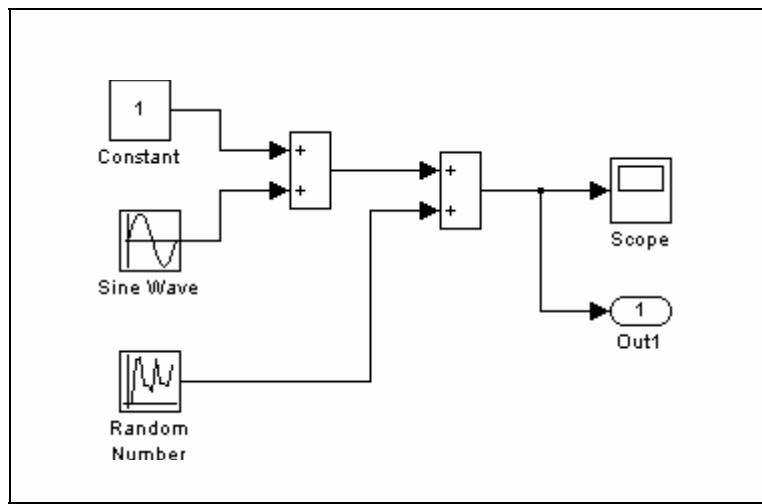


Рис. 6.5. SimuLink блок-схема формування зашумленого сигналу

Недоліком використання S-моделі є неможливість обробки сформованої послідовності згідно формул (6.44). Не зовсім зручно змінювати значення параметрів процесу і шуму, а також оформляти графіки процесів і виводити їх до друку. Тому є потреба використовувати засоби об'єднання S-моделі з робочим простором MatLab. Розглянемо їх.

Перш за все, значення усіх параметрів настроювання блоків S-моделі можна встановлювати не лише безпосередньо у вікнах настроювання, а й опосередковано у командному вікні MatLab, або у програмі системи MatLab. Для цього потрібно:

- 1) як значення параметрів у вікнах настроювання блоків використовувати не числа, а ідентифікатори;
- 2) у командному вікні, або програмі MatLab забезпечити присвоєння числових значень цим ідентифікаторам.

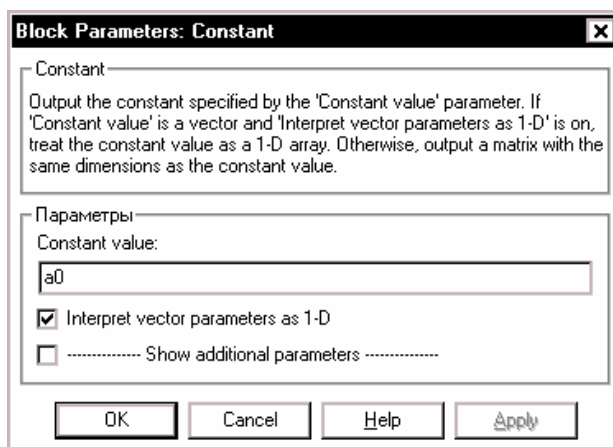


Рис. 6.6. Вікно налаштування блока *Constant*

Наприклад, у створеній блок-схемі у вікні налаштування блока *Constant* (рис. 6.6) у віконці *Constant value* слід встановити ідентифікатор a_0 , який відповідає значенню сталої складової a_0 первісного процесу $x(t)$.

Аналогічно, викликаючи вікно налаштування блока *Sine Wave*, потрібно встановити як амплітуду ідентифікатор am , як частоту - ωm і як фазу - e (рис. 6.7).

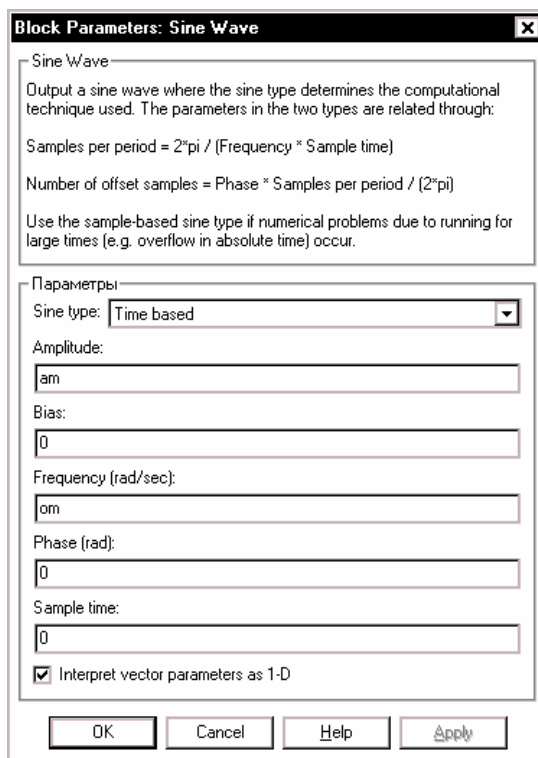


Рис. 6.7. Вікно налаштування блока *Sine Wave*

Нарешті у вікні налаштування блока *Random Number* (рис. 6.8) у віконці *Variance* встановимо ідентифікатор int .

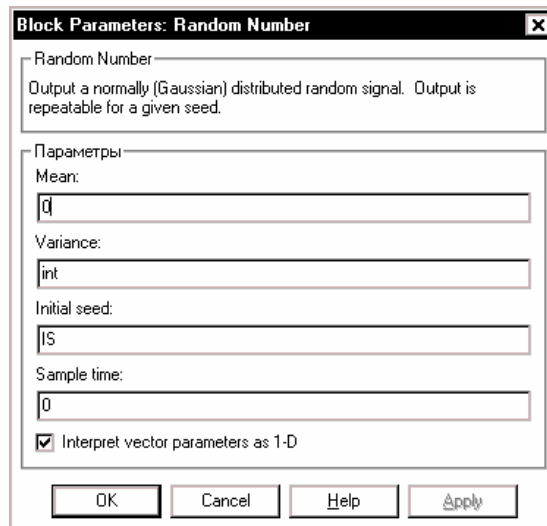


Рис. 6.8. Вікно налаштування блока *Random Number*

Якщо тепер спочатку запустити програму, а потім – запустити S-модель, то ми одержимо на виході процес з тими параметрами, які встановлені у MatLab.

Наступним кроком для забезпечення керованості S-моделі із програми MatLab (М-файла) є встановлення параметрів інтегрування усередині програми. Для цього потрібно:

- у вікні утвореної S-моделі шляхом *Simulation > Parameters > Solver* увійти у вкладинку Solver (рис. 6.9);

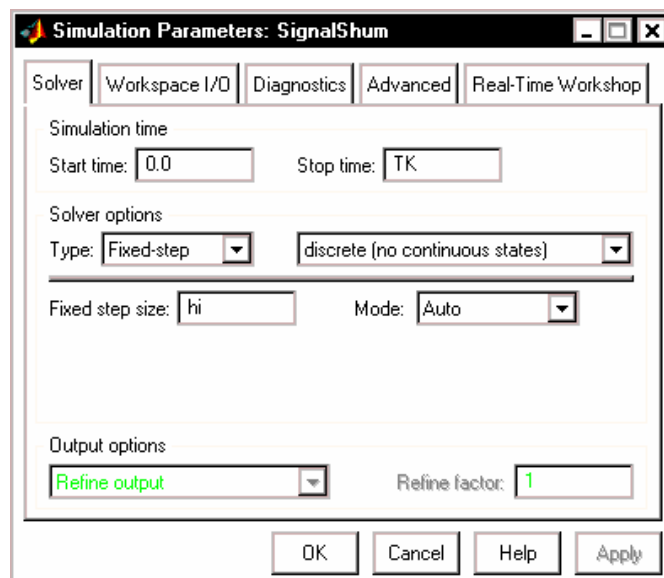


Рис. 6.9. Вікно *Simulation Parameters* (вкладка *Solver*)

- у віконці *Stop time* встановити ідентифікатор *TK*;
- у віконці *Type* поділу *Solver options* замість *Variable step* за допомогою випадаючого меню встановити *Fixed step*;

- у віконці Fixed step size встановити ідентифікатор hi ;

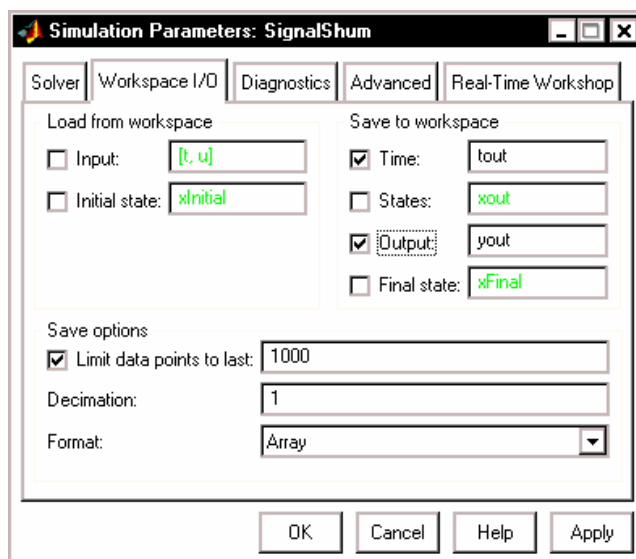


Рис. 6.10. Вікно Simulation Parameters (вкладка Workspace I/O)

- перейти до вкладки Workspace I/O (рис. 6.10);
- встановити у поділі Save to workspace галочки у віконцях ліворуч від написів Time та Output, а віконцях праворуч від них – встановити ідентифікатори tout і yout (рис. 6.10).

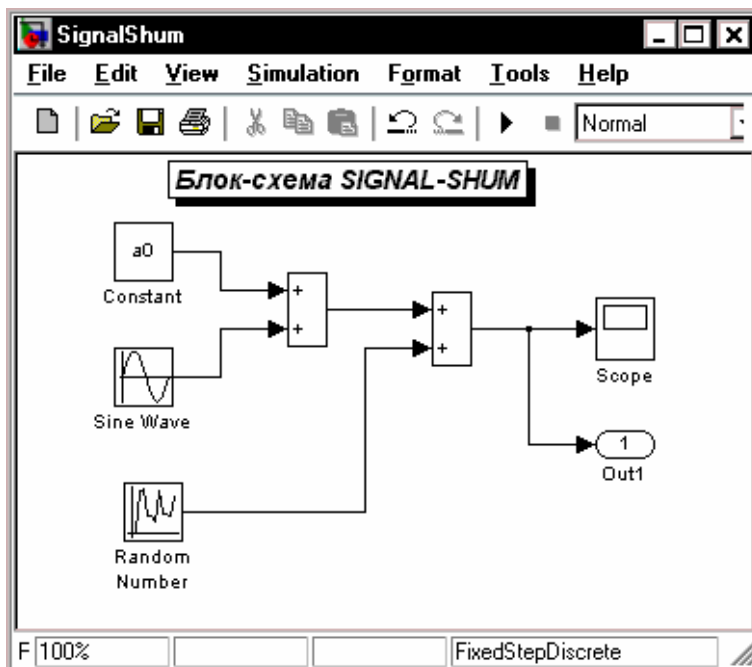


Рис. 6.11. Модифікована блок-схема формування зашумленого сигналу

Тепер, встановлюючи у програмі (М-файлі) числові значення параметрів розв'язувача, - Tk кінцевий термін подання процесів, hi - кроку розрахунку (вимірювання первісного процесу), - можна керувати процесом моделювання

стореної S-моделі. При цьому у робочому просторі MatLab буде автоматично утворюватися масив *tout* значень аргументу (часу), у які здійснюється формування сигналів (у розглядуваному випадку – вимірювання первісного сигналу).

Щоб здійснити наповнювання другої встановленої матриці *yout*, потрібно додатково ввести у блок схему блоки вихідних портів і подати на них сигнали, які потрібно записати у робочий простір. Саме сигнали, що подаються до цих портів, будуть записуватися у робочий простір під ім'ям *yout*.

Вводячи один вихідний порт і з'єднуючи його вхід з сформованим вектором, який складається із вимірюваного і зашумленого сигналів, одержимо модифіковану блок-схему S-моделі, подану на рис. 6.11. Запишемо її на диск під ім'ям *SignalShum.mdl*.

Тепер, щоб ця S-модель запрацювала, потрібно створити M-файл, в якому би:

- присвоювалися значення усім первісним даним;
- здійснювався запуск утвореної S-моделі на моделювання;
- використовувалися одержані при моделюванні результати для побудови графіків у графічному вікні MatLab;
- ті самі дані використовувалися для розрахунку коефіцієнтів апроксимації, розрахунку інтегральних похибок апроксимації, формування відновленого сигналу і побудови графіка поточної похибки.

Нижче наведений один з можливих варіантів тексту такої програми. Зазначимо, що виклик S-моделі із середини M-файла здійснюється за допомогою команди *sim*

```
sim('SignalShum')
```

Тут *SignalShum* – ймення файлу S-моделі, під яким вона записана на диск.

```
% Апрохимура.m
% Програма керування S-моделлю апрох.mdl
% Лазарєв Ю.Ф. 29-10-2001

% Задання первісних даних
om=2*pi;           % Частота гармонічної функції
T=2*pi/om;        % Розрахунок періода гармонічної функції
e=pi/6;           % Початкова фаза
a0=0.3;           % Стала складова
am=1;             % Амплітуда
as=am*cos(e);    ac= am*sin(e); % Розрахунок коефіцієнтів
clc
%%%% Введення змінюваних параметрів
ntk=input('Введіть кількість періодів: ')
TK=T*ntk;        % Розрахунок терміну вимірювання
ni=input('Введіть кількість точок на одному періоді: ')
hi=T/ni;         % Розрахунок інтервалу часу між вимірюваннями
int=input('Введіть інтенсивність шуму: ')

%==== Моделювання процесу вимірювання на S-моделі =====
n=ni*ntk         % Розрахунок кількості точок вимірювання
disp('ПОХИБКИ АПРОКСИМАЦІЇ')
for s=1:16
```

```

IS=s-1;
sim('SignalShum') % Запуск S-моделі SignalShum
%===== Обчислення коефіцієнтів АПРОКСИМАЦІЇ =====
t=tout; % Формування масиву відлікових значень часу
x=yout(:,1); % Формування масиву значень вимірного процесу
X=yout(:,2); % Формування масиву значень результатів вимірювань
% Розрахунок коефіцієнтів апроксимації
A0=sum(X)/n;
As=2*sum(X.*sin(om*t))/n;
Ac=2*sum(X.*cos(om*t))/n;
Y=A0+As*sin(om*t)+Ac*cos(om*t);
% Розрахунок ПОХИБОК визначення коефіцієнтів апроксимації
da0(s)=A0-a0; das(s)=As-as; dac(s)=Ac-ac; dam(s)=sqrt(As^2+Ac^2)-am;
disp([s,da0(s),das(s),dac(s),dam(s)])
end
%=== Розрахунок статистичних характеристик похибок ===
disp('СЕРЕДНІ значення ПОХИБОК')
SR0=mean(da0); SRs=mean(das); SRC=mean(dac); SRm=mean(dam);
disp([SR0,SRs,SRc,SRm])
disp('СЕРЕДНІ квадратичні ВІДХИЛЕННЯ ПОХИБОК')
SKO0=std(da0); SKOs=std(das); SKOc=std(dac); SKOm=std(dam);
disp([SKO0,SKOs,SKOc,SKOm])
%===== Виведення графіків =====
subplot(3,1,1)
plot(t,x),grid %t,X,','
hold on
stem(t,X,',' )
title('Вимірний процес і результати вимірювання')
hold off
subplot(3,1,2)
plot(t,x,t,Y),grid %t,X,','
title('Вимірний і відновлений процеси')
subplot(3,1,3)
plot(t,Y-x),grid %t,X,','
title('ПОХИБКА АПРОКСИМАЦІЇ')
xlabel('Час (с)')

```

Зверніть увагу на те, що

- 1) у S-моделі у блоці *Random Number* у вікно Initial Seed встановлений ідентифікатор IS;
- 2) моделювання за тих самих значень параметрів проводиться не один, а 16 разів при різних значеннях бази IS; похибки для кожного разу запам'ятовуються у масивах
 - da0 – вектора похибок визначення сталої складової;
 - das – вектора похибок визначення "амплітуди" при синусі;
 - dac – вектора похибок визначення "амплітуди" при косинусі;
 - dam – вектора похибок визначення амплітуди;
- 3) по завершенні циклу відшукуються статистичні характеристики кожного з цих векторів похибок, а саме:
 - середнє значення відповідної похибки (її математичне сподівання);
 - середнє квадратичне відхилення значень похибки від її середнього значення.

Як приклад розглянемо результати роботи програми при наступних значеннях параметрів: $ntk=4$; $ni=100$; $int=1$.

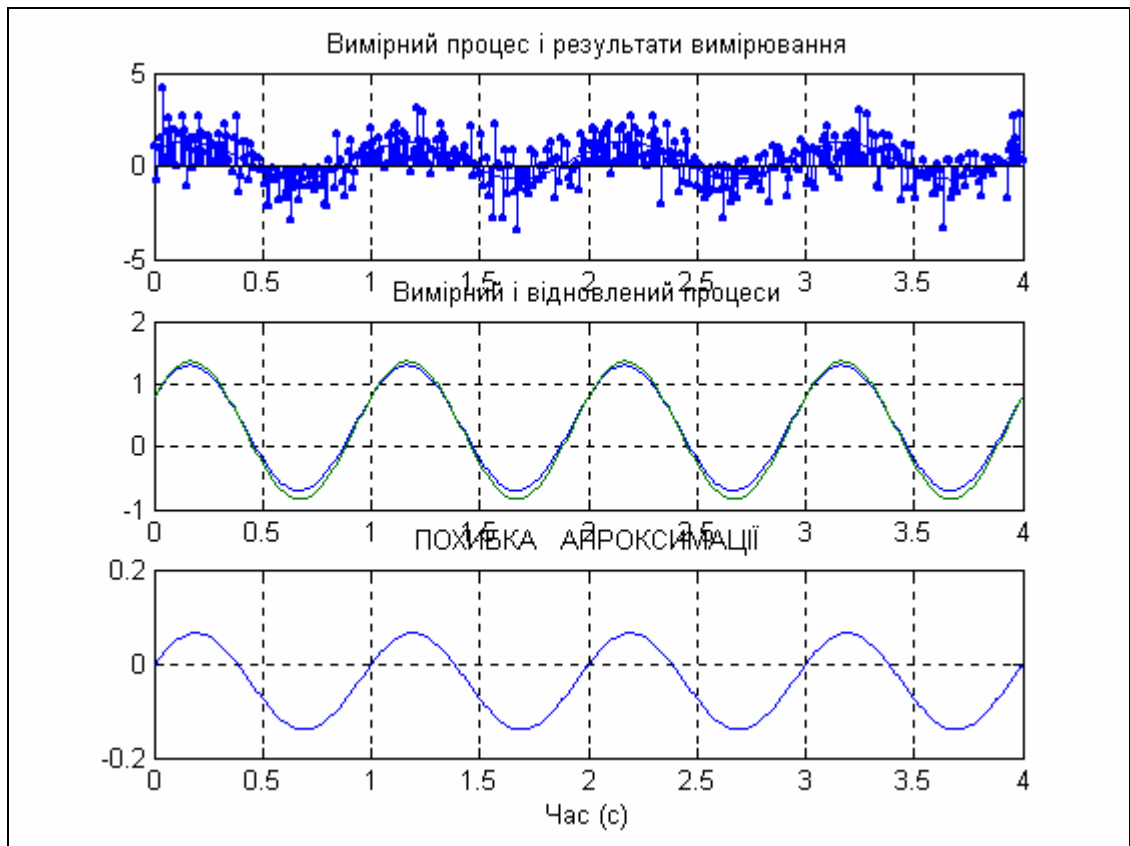


Рис. 6.12. Графічне подання результатів апроксимування

Нижче наведено текст, що виводиться у командне вікно MatLab:

Введіть кількість періодів: 4

$ntk = 4$

Введіть кількість точок на одному періоді: 100

$ni = 100$

Введіть інтенсивність шуму: 1

$int = 1$

$n = 400$

ПОХИБКИ АПРОКСИМАЦІЇ

1.0000	0.0465	-0.0630	0.0062	-0.0507
2.0000	-0.0225	-0.0352	0.1627	0.0628
3.0000	-0.0775	0.1110	-0.0725	0.0664
4.0000	-0.0307	0.0375	0.0188	0.0418
5.0000	-0.0605	-0.0737	-0.0000	-0.0631
6.0000	0.0589	-0.0011	0.0335	0.0162
7.0000	-0.0179	0.0982	-0.0380	0.0692
8.0000	0.0017	0.0437	0.0404	0.0581
9.0000	-0.0049	-0.0346	-0.0048	-0.0323
10.0000	0.0145	0.1332	0.0241	0.1283
11.0000	-0.0698	0.0152	0.0284	0.0275
12.0000	0.0185	0.0568	0.0241	0.0613
13.0000	-0.0661	-0.0481	0.0048	-0.0388
14.0000	0.0139	0.0715	-0.0356	0.0462
15.0000	-0.0188	-0.0243	0.1216	0.0463
16.0000	-0.0377	0.0957	0.0367	0.1014

СЕРЕДНІ значення ПОХИБОК

-0.0158 0.0239 0.0219 0.0338
СЕРЕДНІ квадратичні ВІДХИЛЕННЯ ПОХИБОК
0.0407 0.0665 0.0566 0.0547

По завершенні розрахунків у графічному вікні MatLab виникає зображення графіків, наведених на рис. 6.12.

6.8. Завдання

1. Створити програму дослідження похибок апроксимації гармонічного сигналу із заданою частотою у відповідності до вказівок, поданих у п. 6.7.
2. Провести дослідження статистичних характеристик похибок апроксимації. Побудувати графіки залежностей цих характеристик від:
 - часового інтервалу вимірювання;
 - кількості точок вимірювання;
 - інтенсивності шуму.

6.9. Контрольні запитання

1. Що зазвичай розуміють під "наближенням" функції? Які види наближень існують?
2. Що розуміють під "інтерполюванням" функції? Коли його застосовують?
3. Що розуміють під "апроксимуванням" функції? Коли його застосовують?
4. Що таке "поліноміальне" інтерполювання? За допомогою яких методів можна здійснювати поліноміальне інтерполювання?
5. Що таке метод найменших квадратів? Коли його застосовують?
6. Що розуміють під ортогональними функціями? Який чинник суттєво впливає на властивість ортогональності?
7. Чому ортогональні функції раціонально застосовувати при апроксимації функції?
8. Які засоби інтерполювання і апроксимування функцій є у сучасних мовах програмування?
9. Що таке "сплайн"? коли його застосовують?
10. Як здійснюється інтерполювання таблично заданої функції у системі MatLAB?
11. Як здійснюється поліноміальне апроксимування таблично заданої функції у системі MatLAB?
12. Від яких чинників залежить точність апроксимування заданої (таблично) функції?

7. Чисельне диференціювання і інтегрування функцій

Задача чисельного диференціювання і інтегрування ставиться так. Заданий масив (вектор) значень x_i функції $x(t)$, виміряних (можливо, з похибками) за значень t_i аргумента, заданих іншим вектором. Треба обчислити (оцінити якомога точніше) чисельне значення похідної від $x(t)$ у деякій точці усередині заданого діапазону змінювання аргумента, або обчислити інтеграл від $x(t)$ у цьому діапазоні.

Очевидно, ці задачі можна розв'язати лише за умови проведення через задані точки неперервної (або диференційовної) кривої. Тільки у цьому випадку задача відшукування похідної або інтегралу стає визначеною. Тому чисельне диференціювання і інтегрування тісно пов'язане з наближенням (інтерполюванням).

7.1. Чисельне диференціювання

Найпростіший (і найменш точний) варіант відшукування похідної усередині інтервала $t_i \leq t \leq t_{i+1}$ - використання поліноміальної інтерполяції за двома точками.

Для цього спочатку здійснюється лінійна інтерполяція кривої за двома точками - (x_i, t_i) і (x_{i+1}, t_{i+1}) . В результаті одержується рівняння прямої, що проходить через ці точки:

$$X(t) = x_i + \frac{x_{i+1} - x_i}{t_{i+1} - t_i} (t - t_i).$$

Якщо узяти похідну від цієї функції, одержимо:

$$\frac{dX}{dt}(t) = \frac{x_{i+1} - x_i}{t_{i+1} - t_i}.$$

Як бачимо, у цьому випадку похідна в усьому діапазоні $t_i \leq t \leq t_{i+1}$ є однією і тією самою величиною

$$\frac{dX}{dt}(t_i) = \frac{dX}{dt}(t_{i+1}) = \frac{dX}{dt}(t_i \leq t \leq t_{i+1}) = \frac{x_{i+1} - x_i}{h} + \frac{h}{2} X''(\xi). \quad (7.1)$$

Тут h - "крок" змінювання аргумента.

Перейдемо до побудови формул квадратичної інтерполяції "по трьох точках" (див. рис. 7.1). У цьому випадку через три точки проводиться інтерполююча квадратна парабола

$$X(t) = x_i + a_1(t - t_i) + a_2(t - t_i)^2.$$

З умови проходження цієї параболи через другу і третю точки одержуються значення коефіцієнтів a_1 і a_2 . В результаті рівняння параболи набуде, за умови рівновіддаленості сусідніх точок за аргументом, вигляду

$$X(t) = x_i + \frac{4x_{i+1} - 3x_i - x_{i+2}}{2h}(t - t_i) + \frac{x_{i+2} - 2x_{i+1} + x_i}{2h^2}(t - t_i)^2.$$

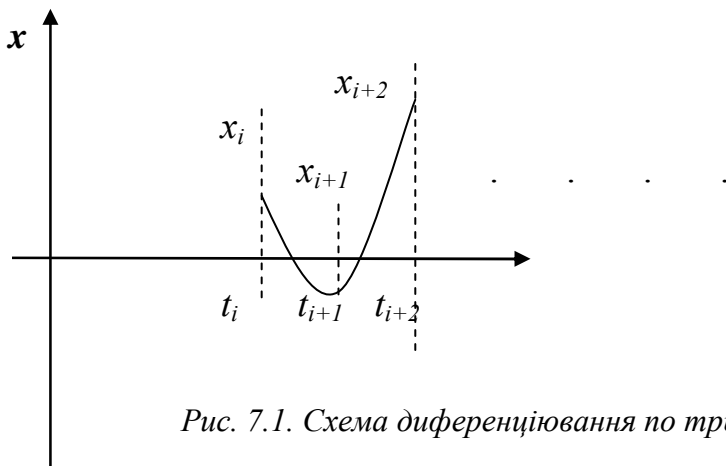


Рис. 7.1. Схема диференціювання по трьох точках

Похідна від цієї параболи є наступною функцією аргумента

$$\frac{dX}{dt}(t) = \frac{4x_{i+1} - 3x_i - x_{i+2}}{2h} + \frac{x_{i+2} - 2x_{i+1} + x_i}{h^2}(t - t_i).$$

Тепер можна записати формули для визначення похідної:

■ у першій точці

$$\frac{dX}{dt}(t_i) = \frac{4x_{i+1} - 3x_i - x_{i+2}}{2h} + \frac{h^2}{3} X'''(\xi); \quad (7.2)$$

■ у другій точці

$$\frac{dX}{dt}(t_{i+1}) = \frac{x_{i+2} - x_i}{2h} - \frac{h^2}{6} X'''(\xi); \quad (7.3)$$

■ у третій точці

$$\frac{dX}{dt}(t_{i+2}) = \frac{x_i - 4x_{i+1} + 3x_{i+2}}{2h} + \frac{h^2}{3} X'''(\xi). \quad (7.4)$$

Тут другі складові формул дають змогу оцінити величини похибок, що вносяться у визначення відповідної похідної за відповідною формулою. При цьому ξ - деяке значення аргументу усередині інтервала.

Інтерполювання по трьох точках дозволяє також обчислити і другу похідну всередині даного інтегралу. Для цього досить узяти похідну від першої похідної

$$\frac{d^2 X}{dt^2}(t) = \frac{x_{i+2} - 2x_{i+1} + x_i}{h^2}.$$

Друга похідна у цьому випадку є однаковою в усіх точках інтервалу, але похибка визначення в усіх точках є різною:

$$\frac{d^2 X}{dt^2}(t_i) = \frac{x_{i+2} - 2x_{i+1} + x_i}{h^2} - hX'''(\xi);$$

$$\frac{d^2 X}{dt^2}(t_{i+1}) = \frac{x_{i+2} - 2x_{i+1} + x_i}{h^2} - \frac{h^2}{12} X^{IV}(\xi); \quad (7.5)$$

$$\frac{d^2 X}{dt^2}(t_i) = \frac{x_{i+2} - 2x_{i+1} + x_i}{h^2} + hX'''(\xi).$$

Аналогічно виводяться формули чисельного диференціювання по чотирьох точках (кубічна інтерполяція):

■ у першій точці

$$\frac{dX}{dt}(t_i) = \frac{-11x_i + 18x_{i+1} - 9x_{i+2} + 2x_{i+3}}{6h} + \frac{h^3}{4} X^{IV}(\xi); \quad (7.6)$$

■ у другій точці

$$\frac{dX}{dt}(t_{i+1}) = \frac{-2x_i - 3x_{i+1} + 6x_{i+2} - x_{i+3}}{6h} + \frac{h^3}{12} X^{IV}(\xi); \quad (7.7)$$

■ у третій точці

$$\frac{dX}{dt}(t_{i+2}) = \frac{x_i - 6x_{i+1} + 3x_{i+2} + 2x_{i+3}}{6h} - \frac{h^3}{12} X^{IV}(\xi); \quad (7.8)$$

■ у четвертій точці

$$\frac{dX}{dt}(t_{i+3}) = \frac{-2x_i + 9x_{i+1} - 18x_{i+2} + 11x_{i+3}}{6h} + \frac{h^3}{4} X^{IV}(\xi). \quad (7.9)$$

Другі похідні у цьому випадку визначаються співвідношеннями:

$$\frac{d^2 X}{dt^2}(t_i) = \frac{2x_i - 5x_{i+1} + 4x_{i+2} - x_{i+3}}{h^2} + \frac{11h^2}{12} X^{IV}(\xi);$$

$$\frac{d^2 X}{dt^2}(t_{i+1}) = \frac{x_i - 2x_{i+1} + x_{i+2}}{h^2} - \frac{h^2}{12} X^{IV}(\xi); \quad (7.10)$$

$$\frac{d^2 X}{dt^2}(t_{i+2}) = \frac{x_{i+1} - 2x_{i+2} - x_{i+3}}{h^2} - \frac{h^2}{12} X^{IV}(\xi);$$

$$\frac{d^2 X}{dt^2}(t_{i+3}) = \frac{-x_i + 4x_{i+1} - 5x_{i+2} + 2x_{i+3}}{h^2} + \frac{11h^2}{12} X^{IV}(\xi).$$

Як впливає з наведених формул, зі зменшенням величини кроку h для досить гладких функцій величина залишкового члену, тобто похибка метода, зменшується. При цьому значення похідних є більш точними у вузлах, що розташовані усередині рівномірної сітки.

Але якщо значення x_i відомі не точно (з похибками вимірювання), то обчислювальна похибка, як це впливає з одержаних формул, навпаки, збільшується зі зменшенням кроку і ростом порядку похідної. Отже, чисельне диференціювання є за своєю природою нестійкою процедурою.

Якщо абсолютна похибка вимірювання ε є відомою, можна для конкретної формули чисельного диференціювання вказати оптимальну величину кроку, за якої сумарна похибка визначення похідної є мінімальною.

Наприклад, для формули (3) похибку метода можна оцінити так

$$\frac{h^2}{6} M; \text{ де } M = \max_{t_i \leq t \leq t_{i+2}} |X'''(t)|.$$

Похибка, спричинена похибкою вимірювання, у відповідності з тією ж формулою, дорівнюватиме $\frac{2\varepsilon}{2h} = \frac{\varepsilon}{h}$. Оптимальний крок досягається при такому значенні кроку, коли ці два види похибок дорівнюють один одному:

$$\frac{h^2}{6} M = \frac{\varepsilon}{h},$$

тобто при

$$h_{opt} = \sqrt[3]{\frac{6\varepsilon}{M}}.$$

Сумарна похибка при цьому оптимальному кроці дорівнюватиме

$$(\Delta_{X'})_{\min} = 2 \frac{\varepsilon}{h_{opt}} = 2\varepsilon \sqrt[3]{\frac{M}{6\varepsilon}}.$$

У випадку, коли потрібно знайти похідну не в одній точці, а у всіх багатьох вимірних точках, тобто коли потрібно одержати криву залежності похідної від аргументу у заданому діапазоні, раціональніше користуватися сплайновою кубічною інтерполяцією. Бажана функція похідної одержуватиметься як похідна від сплайна.

7.2. Чисельне інтегрування

При розв'язуванні багатьох інженерних задач доводиться обчислювати визначені інтеграли від функцій, у тому числі і від заданих при окремих заданих значеннях аргументу (таблично).

Більшість чисельних методів інтегрування функцій спираються на ідею заміни підінтегральної функції $x(t)$ деякою наближаючою її функцією $X(t)$, інтеграл від якої обчислюється досить просто. Наближення зазвичай здійснюється інтерполюванням у межах заданого діапазону змінювання аргументу. У подальшому вважатимемо, що функція $x(t)$ задана масивом (вектором) своїх значень x_i у n рівновіддалених точках діапазону змінювання аргументу t від a до b так, що $h = (b - a) / (n - 1)$ є кроком задання функції (одночасно це є крок інтегрування). На рис. 7.2. наведено графічне подання такої функції.

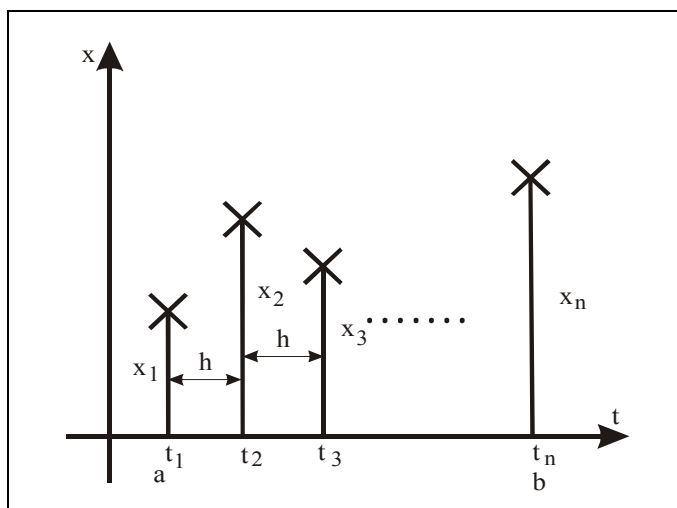


Рис. 7.2. Графічне подання табличної функції

Метод правих прямокутників. Цей найпростіший метод чисельного інтегрування полягає у тому, що на кожному кроці $t_i \leq t \leq t_{i+1}$ інтегрування функція $x(t)$ замінюється на сталу величину, рівну значенню x_i на лівому кінці відповідного інтервалу. Через те інтеграл від функції $x(t)$ на цьому інтервалі можна геометрично подати як площу прямокутника (рис. 7.3 а), що справа примикає до заданої кривої (звідси й назва методу) і обчислити за формулою

$$\int_{t_i}^{t_{i+1}} x(t) dt = x_i \cdot h + \frac{h^2}{2} X'(\xi). \quad (7.11)$$

Другий член формули подає оцінку похибки методу інтегрування (ξ - деяке значення аргументу усередині інтервалу). Сумуючи результати інтегрування по окремих кроках, одержимо загальну формулу

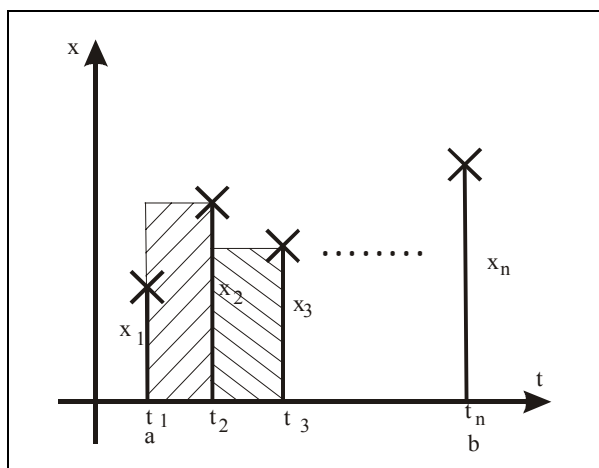
$$\int_a^b x(t) dt = h \sum_{i=1}^{n-1} x_i + \frac{h \cdot (b-a)}{2} X'(\xi). \quad (7.12)$$

Отже, для визначення повного інтегралу за методом правих прямокутників достатньо просумувати усі задані значення функції, окрім останнього, і результат домножити на крок інтегрування. при цьому похибка методу визначається середнім значенням похідної на інтервалі визначення інтегралу і пропорційна другому степеню кроку. Зменшуючі величину кроку інтегрування, можна суттєво зменшити похибку визначення інтегралу.

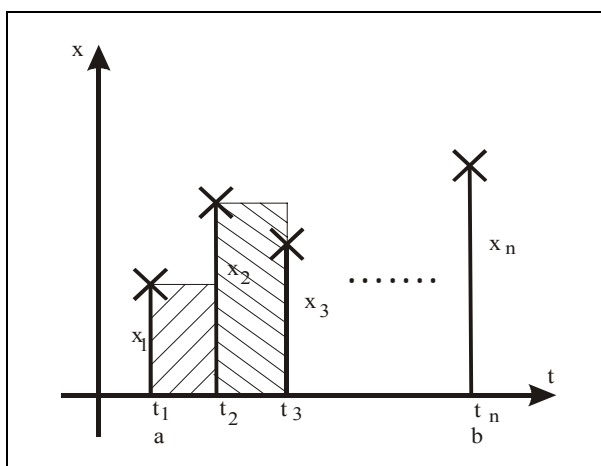
Метод лівих прямокутників. Сутність цього методу аналогічна, за винятком того, що апроксимуючі прямокутники тепер примикають зліва (див. рис. 7.3 б) до заданих точок. Відповідні формули чисельного інтегрування набувають вигляду

$$\int_{t_i}^{t_{i+1}} x(t) dt = x_{i+1} \cdot h - \frac{h^2}{2} X'(\xi); \quad \int_a^b x(t) dt = h \sum_{i=2}^n x_i - \frac{h \cdot (b-a)}{2} X'(\xi). \quad (7.13)$$

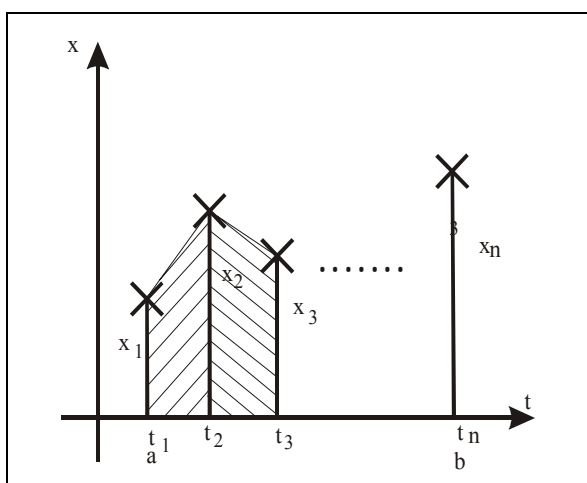
Властивості цього методу майже не відрізняються від попереднього. Слід звернути увагу лише на те, що похибка його має зворотний знак. З цього одразу можна зробити висновок, що комбінуючи ці два методи, можна суттєво збільшити точність чисельного інтегрування. У такий спосіб приходим до методу трапецій.



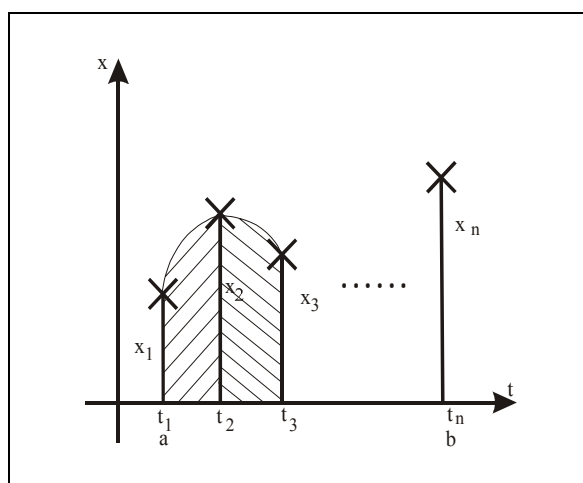
а) метод правих прямокутників



б) метод лівих прямокутників



в) метод трапецій



г) метод Симпсона

Рис. 7.3. Графічне подання методів інтегрування

Метод трапецій. Формули цього методу є такими:

$$\int_{t_i}^{t_{i+1}} x(t) dt = \frac{h}{2} (x_i + x_{i+1}) + \frac{h^3}{12} X'''(\xi);$$

$$\int_a^b x(t) dt = h \left(\frac{x_1 + x_n}{2} + \sum_{i=2}^{n-1} x_i \right) + \frac{h^2 \cdot (b-a)}{12} X'''(\xi). \quad (7.13)$$

Графічна інтерпретація: при інтегруванні цим методом обчислюються площі (рис. 7.2.в) трапецій, які утворюються, якщо з'єднати окремі задані точки відрізками прямих, тобто за лінійної інтерполяції.

Метод Симпсона. Поділимо інтервал інтегрування $[a, b]$ на парну кількість k рівних частин так, що загальна кількість точок $n = 2k - 1$ буде непарною, а крок інтегрування дорівнюватиме

$$h = (b - a) / (n - 1) = (b - a) / 2(k - 1).$$

На кожному з інтервалів $t_i \leq t \leq t_{i+2}$ ($i = 1, 3, 5, \dots, n - 2$) інтерполюємо задані точки квадратною параболою (див. п. 7.1, диференціювання за трьома точками):

$$X(t) = x_i + \frac{4x_{i+1} - 3x_i - x_{i+2}}{2h} (t - t_i) + \frac{x_{i+2} - 2x_{i+1} + x_i}{2h^2} (t - t_i)^2.$$

Інтегруючи цю функцію в інтервалі $t_i \leq t \leq t_{i+2}$, одержимо формулу Симпсона для окремої ділянки

$$\int_{t_i}^{t_{i+2}} x(t) dt = \frac{h}{3} (x_i + 4x_{i+1} + x_{i+2}) - \frac{h^5}{90} X^{IV}(\xi). \quad (7.14)$$

Підсумовуючи цей результат по усіх частинних відрізках ($i = 1, 3, 5, \dots, n - 2$), прийдемо до квадратурної формули Симпсона

$$\int_a^b x(t) dt = \frac{h}{3} (x_1 + 4x_2 + 2x_3 + 4x_4 + 2x_5 + \dots + 2x_{n-2} + 4x_{n-1} + x_n) - \frac{h^4 \cdot (b - a)}{180} X^{IV}(\xi)$$

На відміну від чисельного диференціювання, чисельне інтегрування є обчислювально стійкою операцією. Похибка будь-якого методу зменшується зі збільшенням кількості заданих точок початкової функції у заданому інтервалі, причому степінь зменшення залежить від методу.

Як впливає з розгляду залишкових членів, що описують похибку методу, остання є степеневою функцією від кроку інтегрування. Показник степеня у цій залежності похибки від кроку прийнято називати порядком метода інтегрування. Таким чином, методи прямокутників є методами першого порядку, метод трапеції - другого порядку, а метод Симпсона є методом четвертого порядку.

7.3. Засоби чисельного диференціювання і інтегрування на ЕОМ

Практично в усіх мовах програмування високого порядку відсутні вбудовані функції, які б дозволяли здійснювати чисельне диференціювання і інтегрування функцій. Деякі вбудовані функції (процедури) передбачені лише у новітніх комп'ютерних математичних системах.

У системі MatLab існують три вбудовані функції, які здійснюють чисельне інтегрування функцій – *trapz*, *quad* і *quad8*.

Процедура *trapz* здійснює обчислення площі під графіком функції $y(x)$ у випадку, коли функція подана у вигляді двох масивів (векторів) x - значень аргумента у зростаючому порядку і y - відповідних значень функції при цих значеннях аргумента. При цьому під графіком функції $y(x)$ розуміється ламана, що з'єднує сусідні точки функції. Звернення виду

$$I = \text{trapz}(x, y)$$

приводить до обчислення вказаної площі і присвоєння обчисленого значення ідентифікатору I.

Приклад. Обчислимо інтеграл від функції $y = \sin(x)$ у діапазоні від 0 до π . Його точне значення дорівнює 2. Візьмемо рівномірну сітку значень аргументу з 100 елементів. Тоді обчислення зведеться до сукупності операцій:

```
x = 0 : pi/100 : pi;
y = sin(x);
disp(trapz(x,y))
```

Результатом буде поява у командному вікні значення інтегралу 1.9998.

Обчислення інтегралів методом квадратур у випадку, коли обчислення функції здійснюється у m-файлі з деяким ім'ям, наприклад, FUN.m, здійснюється процедурою:

```
[I, cnt] = quad('FUN', a, b).
```

Тут a і b – нижня і верхня межі змінювання аргументу функції (межі інтегрування; I – одержане значення інтеграла; cnt – кількість звернень до обчислення підінтегральної функції (яку репрезентовано m-файлом FUN.m) при обчисленні інтегралу.

Функція **quad** використовує квадратурні формули Ньютона – Котеса.

Аналогічна функція **quad8** використовує більш точні формули 8-го порядку. Звернення до неї аналогічне.

Функції **quad** і **quad8** здійснюють розбиття заданого діапазону змінювання аргументу автоматично, забезпечуючи відносну похибку обчислення інтегралу, меншу за задану, яка за замовчуванням приймається рівною $1 \cdot 10^{-3}$. Якщо потрібно змінити точність обчислення інтегралу, слід задати межову припустиму відносну похибку, додаючи у перелік вхідних параметрів під ім'ям 'RelTol'. Наприклад, щоб встановити відносну похибку обчислення інтегралу меншу $1 \cdot 10^{-8}$, можна це зробити, звертаючись до цих функцій у такий спосіб:

```
[I, cnt] = quad('FUN', a, b, 'RelTol', 1e-8).
```

Як приклад обчислимо той самий інтеграл, використовуючи як файл, що обчислює підінтегральну функцію, вбудовану функцію сінуса 'sin':

```
[I, cnt] = quad('sin', 0, pi)
```

Результат буде наступним:

```
I = 2.0000
cnt = 17
```

7.4. Контрольні запитання

1. У чому полягає задача чисельного диференціювання? чисельного інтегрування функції?
2. Як вирішується задача відшукування значення похідної від функції, заданої таблично?
3. Які існують найпростіші методи чисельного інтегрування функції? Напишіть формули обчислення інтегралу для цих методів?
4. Які вбудовані функції, що здійснюють чисельне диференціювання і чисельне інтегрування функції, існують у мовах високого рівня, математичних комп'ютерних системах, у MatLab?
5. Чим відрізняється обчислення інтегралу у функціях *trapz*, *quad* і *quad8* системи MatLab? Як звертатися до цих функцій?

8. Чисельне інтегрування диференціальних рівнянь

Чисельне інтегрування диференціальних рівнянь іноді є єдиною можливістю дослідити поведінку деякої технічної системи, якщо вона може бути описана лише за допомогою таких складних рівнянь, що вони не можуть бути розв'язаними аналітично. Це стосується перш за все так званих нелінійних і нестационарних систем з суттєвими нелінійностями, що часто трапляються у інженерній практиці.

8.1. Однокрокові методи

8.1.1. Загальна постановка задачі чисельного інтегрування диференціальних рівнянь

Сутність методів чисельного інтегрування звичайних (тобто з одним аргументом t) диференціальних рівнянь (ЗДР) розглядатимемо на прикладі єдиного рівняння першого порядку:

$$\frac{dy}{dt} = Z(t, y). \quad (8.1)$$

Як буде показано далі, висновки і формули, одержані при дослідженні цього рівняння, досить легко узагальнюються на випадок системи ЗДР будь-якого порядку n , якщо попередньо цю систему приведено до так званої нормальної форми Коші, а саме - до вигляду:

$$\begin{aligned} \frac{dy_1}{dt} &= Z_1(t, y_1, y_2, \dots, y_n); \\ \frac{dy_2}{dt} &= Z_2(t, y_1, y_2, \dots, y_n); \\ &\dots \dots \dots \\ \frac{dy_n}{dt} &= Z_n(t, y_1, y_2, \dots, y_n) \end{aligned} \quad (8.2)$$

У принципі ніякий чисельний метод не в змозі забезпечити точне моделювання рівняння (1) через те, що останнє записано у нескінченно малих приростах, у той час як чисельні методи можуть оперувати лише скінченими приростами, хоча й, можливо, досить малими. Тому замість рівняння (1) чисельним методом відтворюється деякий його аналог у скінчених різницях. Задачею цього аналогу є обчислення значення y_{m+1} змінної y при наступному значенні t_{m+1} аргументу t , відділеному від попереднього значення t_m на величину h , по відомій інформації про стан системи у точці m (тобто про значення y_m змінної y , див. рис. 8.1).

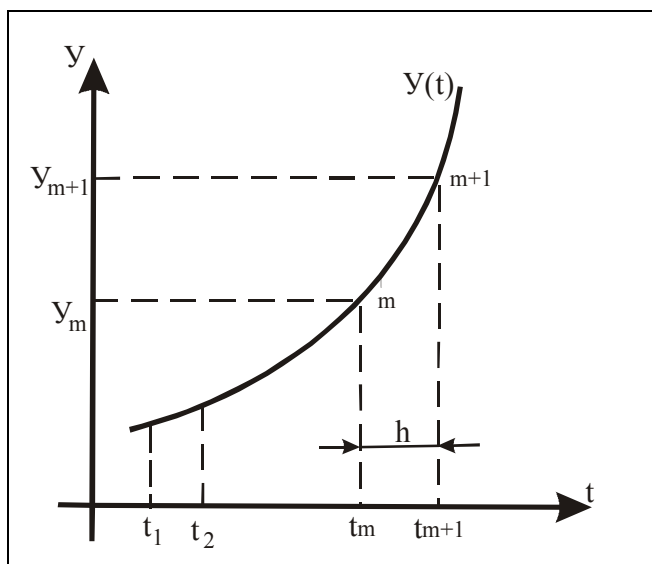


Рис. 8.1. Графічне подання чисельного інтегрування ДР

Величина h , що відділяє за аргументом наступну точку обчислень від попередньої, називається кроком інтегрування.

Методи, що дозволяють визначити наступну точку інтегральної кривої за інформацією тільки про єдину попередню точку, називають однокроковими. Усі однокрокові методи ще називають методами Рунге - Кутта (за прізвищами вчених, які побудували теорію цих методів).

Узагальнена формула відшукування наступного значення змінної за відомим значенням її на попередньому кроці у методах Рунге - Кутта виглядає наступним чином:

$$y_{m+1} = y_m + h \cdot F(t_m, y_m, h). \quad (8.3)$$

Залежність $F(t_m, y_m, h)$ може бути досить складною. Саме її виглядом і відрізняються модифікації методів Рунге-Кутта.

Щоб підійти до виведення конкретних формул методів Рунге-Кутта, зазначимо перш за все, що, якщо є відомим значення функції $y(t)$ у якійсь точці $y_m = y(t_m)$, то, у випадку, коли ця функція неперервна і необмежено диференційовна, значення її в околі точки (y_m, t_m) можуть бути подані через ряд Тейлора:

$$y(t) = y_m + y'_m(t - t_m) + \frac{y''_m}{2}(t - t_m)^2 + \frac{y'''_m}{6}(t - t_m)^3 + \dots \quad (8.4)$$

Нехай задача полягає у обчисленні значення y_{m+1} цієї функції у точці t_{m+1} , яка відстоїть від t_m на величину кроку h інтегрування

$$h = t_{m+1} - t_m. \quad (8.5)$$

Тоді для такого обчислення ми маємо точну формулу, яка впливає з (4):

$$y_{m+1} = y_m + h \cdot y'_m + h^2 \cdot \frac{y''_m}{2} + h^3 \cdot \frac{y'''_m}{6} + \dots, \quad (8.6)$$

що є еквівалентним до формули (3), якщо прийняти

$$F(t_m, y_m, h) = y'_m + h \cdot \frac{y''_m}{2} + h^2 \frac{y'''_m}{6} + \dots \quad (8.7)$$

Вираз (7) є точним скінчено-різницеvim аналогом похідної, яка стоїть у правій частині рівняння (1). Тому розрахунок $F(t_m, y_m, h)$ за цією формулою призвів би до точного значення y_{m+1} (звичайно, за умови, що усі обчислення здійснюються без похибок і відсутні похибки округлення).

Однак, як неважко бачити, реалізація обчислень за формулою (7) натикається на наступні перешкоди:

- 1) необхідність проводити нескінченну кількість операцій (обчислювати суму нескінченного ряду);
- 2) необхідність знати значення похідних усіх порядків у попередній точці.

Ці труднощі мають принциповий характер. Неможливо (якнайменше - нераціонально) організувати нескінченний процес для обчислення однієї точки. Доведеться відкидати усі члени нескінченного ряду за винятком деяких з них. У загальному випадку є неможливим і попереднє знання значень усіх похідних. Якщо першу похідну в силу рівняння (1) можна обчислити точно (якщо нехтувати похибками обчислень і округлень), то визначити похідні вищого порядку значно складніше. Скоріш за все для їхнього визначення потрібно також спиратися на скінчено-різницеві співвідношення.

Таким чином, принципові похибки чисельних методів інтегрування ДР мають подвійний характер:

- 1) похибки внаслідок того, що нескінченний ряд (7) замінюється на скінчений, який має декілька перших членів ряду (7); при цьому відкинуті члени утворюють похибку;
- 2) похибки внаслідок неточного обчислення похідних у попередній точці; у власну чергу, вони звичайно обумовлені також тим, що замість нескінченного ряду при їхньому обчисленні використовуються лише обмежена кількість його членів.

Як бачимо, обидві похибки - єдиного типу. Ці похибки чисельного методу не пов'язані з точністю здійснення математичних операцій, - вони викликані обмежуванням нескінченних рядів скінченою кількістю їх членів. Тому вони одержали назву похибок обмежування (на відміну від похибок округлення, які визначаються точністю проведення обчислень).

Характерна особливість похибок обмежування - при зменшенні кроку інтегрування вони також зменшуються (див. формулу (7)). Цим вони суттєво відрізняються від похибок округлення, які визначаються кількістю математичних операцій. Через те, що при зменшенні кроку кількість операцій, що припадає на відтворення одиниці часу процесу, що інтегрується, пропорційно збільшується, то похибки округлення теж збільшуються зі зменшенням кроку.

Якісна картина залежності похибок від кроку інтегрування наведена на рис. 8.2.

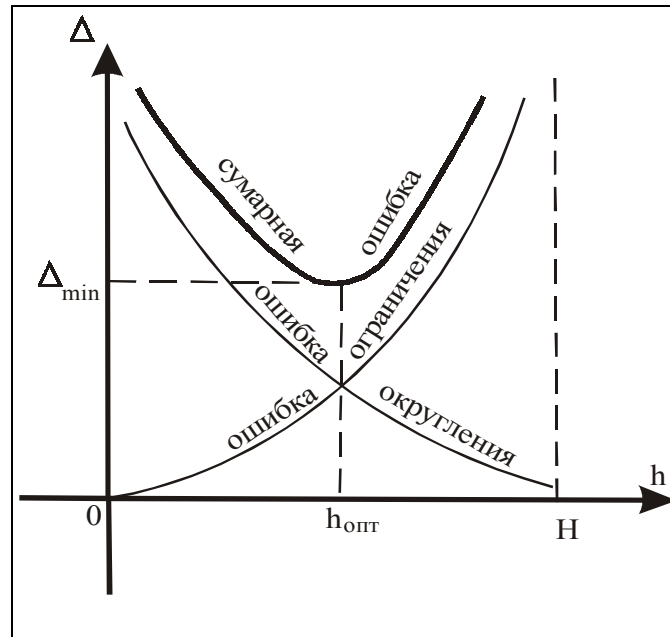


Рис. 8.2. Залежність похибок від кроку інтегрування

Як очевидно, для кожного методу (і кожного виду зберігання чисельних даних у оперативній пам'яті ЕОМ) існує деяке оптимальне значення кроку інтегрування, за якого сумарна похибка чисельного інтегрування даного диференційного рівняння сягає мінімального значення.

Методи Рунге-Кутта мають наступні відзначальні риси:

- ці методи є однокроковими: щоб визначити y_{m+1} потрібна інформація лише про попередню точку ;
- вони погоджуються з рядом Тейлора (6) впритул до членів порядку h^p , де степінь p є різною для різних методів і називається порядком метода;
- вони не потребують обчислення похідних від функцій $Z(t, y)$, а потребують лише обчислення самої функції $Z(t, y)$.

Саме завдяки останньої властивості методи Рунге-Кутта є більш зручними за ряд Тейлора (6). Однак, як і слід було чекати, для обчислення однієї наступної точки розв'язку доводиться обчислювати функцію $Z(t, y)$ кілька разів за різних значень t і y . Це є тою ціною, яку треба платити за можливість не обчислювати ніяких похідних.

8.1.2. Метод Ейлера

Це - найпростіший чисельний метод. У практиці обчислень він застосовується вельми рідко внаслідок великої похибки обмежування. Але на його прикладі зручно пояснити способи побудови і дослідження чисельних методів інтегрування.

Формула метода Ейлера відшукування наступної точки y_{m+1} за попередньою y_m випливає з (7), якщо у цьому ряду залишити лише перший член, тобто у виразі (3) як $F(t_m, y_m, h)$ взяти y'_m :

$$y_{m+1} = y_m + h \cdot y'_m.$$

У відповідності з початковим рівнянням (1) $y'_m = Z(t_m, y_m)$, тому матимемо остаточно

$$y_{m+1} = y_m + h \cdot Z(t_m, y_m). \quad (8.8)$$

Це й є формула, за якою здійснюється чисельне інтегрування рівняння (1) у методі Ейлера.

Геометрична інтерпретація формули подана на рис. 8. 3.

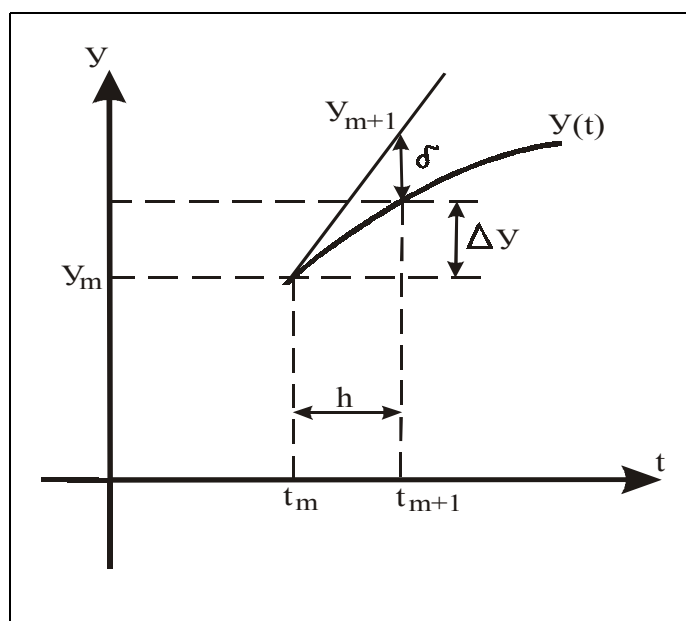


Рис. 8.3. Геометрична інтерпретація метода Ейлера

З нього випливає, що інтегрування за методом Ейлера графічно зводиться до проведення дотичної до інтегральної кривої у попередній точці до перетинання її з прямою, яка є паралельною осі ординат і має значення аргументу t_{m+1} . Похибка обмежування, яка виникає при такому русі від попередньої точки до наступної, показана у вигляді відрізка δ (на рисунку крива $y = y(t)$ відповідає точній інтегральній кривій, що проходить через точку (t_m, y_m)). Докладніше займемося досліджуванням похибки δ обмежування методу Ейлера.

Побудуємо таку схему міркувань. Якщо б ми у точності відтворили ряд (7), це було б еквівалентно відшуканню точного значення похідної у початковому рівнянні (1) для точки (t_m, y_m) . Перепишемо (6) у такий формі

$$y'_m = \frac{y_{m+1} - y_m}{h} - h \cdot \frac{y''_m}{2} - h^2 \frac{y'''_m}{6} - \dots \quad (8.9)$$

Тепер можна стверджувати, що обчислення за нескінченним рядом (7) є еквівалентним до обчислення похідної за формулою (9). Насправді метод Ейлера здійснює обчислення лише за першим членом ряду (7). Це означає, що похідна обчислюється не за точною формулою (9), а за наближеною, яка, з врахуванням (9), може бути поданою у вигляді

$$(y'_m)_u = \frac{(y_{m+1})_u - y_m}{h} = y'_m + \frac{h}{2} \cdot y''_m + \frac{h^2}{6} \cdot y'''_m + \dots, \quad (8.10)$$

де індекс u вказує на те, що у цьому випадку мова йде вже не про точну інтегральну криву $y(t)$, а про криву, яка виходить внаслідок чисельного інтегрування (що містить у собі похибки обмежування). Додаток до першого члена у правій частині (10) фактично і є похибкою обмежування:

$$D(y) = \frac{h}{2} \cdot y''_m + \frac{h^2}{6} \cdot y'''_m + \frac{h^3}{24} \cdot y^{IV}_m + \frac{h^4}{120} \cdot y^V_m + \dots \quad (8.11)$$

Позначимо похибку обмежування

$$\delta = y_u - y \quad (8.12)$$

Тоді матимемо

$$y'_u = \frac{dy_u}{dt} = \frac{dy}{dt} + \frac{d\delta}{dt} = y' + \delta'. \quad (8.13)$$

Якщо "ідеальне" рівняння має вид (1)

$$y' = Z(t, y), \quad (8.14)$$

то "реальне" рівняння, яке відтворюється чисельним методом, буде, у відповідності з (10), мати вигляд

$$y'_u - D(y) = Z(t, y_u). \quad (8.15)$$

Віднімемо з рівняння (15) рівняння (14) і врахуємо позначення (12):

$$\delta' = Z(t, y + \delta) - Z(t, y) + D(y). \quad (8.16)$$

Розкладаючи у ряд Тейлора функцію $Z(t, y + \delta)$ у околі значення (t, y) і обмежуючись утримування у цьому ряді лише перших двох членів

$$Z(t, y + \delta) = Z(t, y) + \delta \cdot \frac{\partial Z}{\partial y},$$

одержимо рівняння похибок

$$\delta' - \frac{\partial Z}{\partial y} \cdot \delta = D(y) = \frac{h}{2} y'' + \frac{h^2}{6} y''' + \frac{h^3}{24} y^{IV} + \dots \quad (8.17)$$

Якщо величина кроку є достатньо малою, то, нехтуючи впливом більш малих членів, одержуємо

$$\delta' - \frac{\partial Z}{\partial y} \cdot \delta = \frac{h}{2} y''. \quad (8.18)$$

Звідси випливає, що похибка обмежування метода Ейлера має перший порядок мализни відносно кроку h , тобто метод Ейлера відноситься до методів Рунге-Кутта першого порядку.

Знаючи характер змінювання з часом шуканої величини $y(t)$, можна за рівнянням (18) встановити й характер змінювання похибки з часом.

8.1.3. Модифікований метод Ейлера

Відшукаємо формули метода другого порядку. Для цього у ряді (7) потрібно залишити перші два члени:

$$F(t_m, y_m, h) = y'_m + h \cdot \frac{y''_m}{2} . \quad (8.19)$$

Щоб використати цей вираз у загальній формулі (3), потрібно якось оцінити (обчислити з відповідною точністю) другу похідну від інтегральної кривої у точці m . Це можна зробити, використовуючи скінченні різниці

$$y''_{um} = [Z(\tilde{t}, \tilde{y}) - Z(t_m, y_m)] / \Delta t + \Delta(\Delta t), \quad (8.20)$$

де через $\Delta(\Delta t)$ - позначено похибку такого визначення другої похідної і ту обставину, що вона має величину, пропорційну узятому проміжку часу $\Delta t = \tilde{t} - t_m$.

Обираючи різні \tilde{t} і способи визначення відповідного цьому значенню аргументу значення \tilde{y} шуканої змінної, можна одержати різні формули для методів другого порядку.

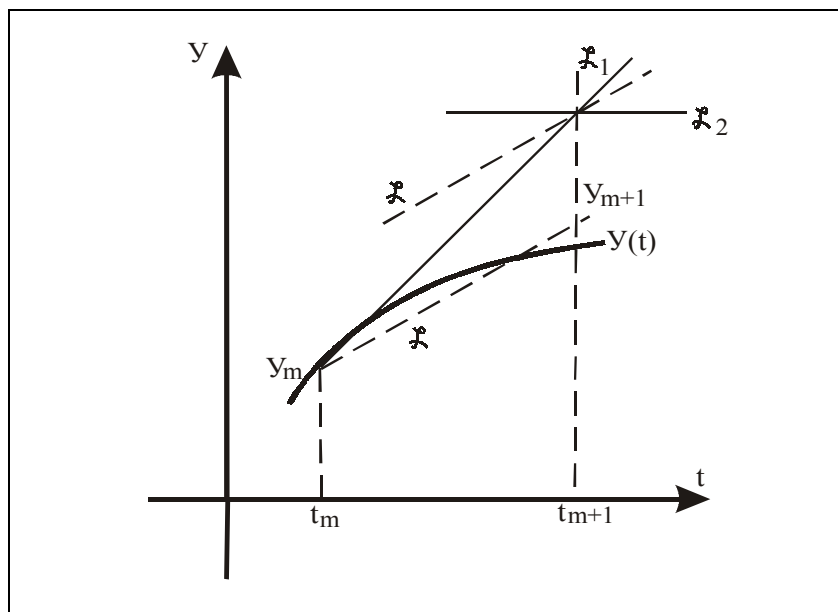


Рис. 8.4. Геометрична інтерпретація першої модифікації методу Ейлера

Так, якщо обрати $\tilde{t} = t_{m+1}$, а \tilde{y} визначити за методом Ейлера

$$\tilde{y} = y_m + h \cdot Z(t_m, y_m),$$

то, підставляючи цей результат у (20), а потім - у (19), одержимо формулу модифікованого методу Ейлера:

$$F(t_m, y_m, h) = Z(t_m, y_m) + \frac{h}{2} \cdot \frac{Z(t_m + h, y_m + h \cdot Z(t_m, y_m)) - Z(t_m, y_m)}{h} = \frac{1}{2}(k_1 + k_2) \quad (8.19)$$

де позначено

$$k_1 = Z(t_m, y_m); \quad k_2 = Z(t_m + h, y_m + h \cdot k_1). \quad (8.20)$$

Геометрична інтерпретація цього варіанту досить проста (рис. 8.4).

Спочатку обчислюється за методом Ейлера значення змінної стану y у точці з номером $m + 1$, потім – нахил дотичної L_2 до інтегральної кривої у цій точці. Далі знаходимо середній нахил дотичних у діапазоні між точками m і $m + 1$ (проводимо бісектрису L кута між дотичними L_1 і L_2 у цих точках) і по ньому уточнюється значення y_{m+1} .

Іншу формулу метода другого порядку можна одержати, якщо обрати $\tilde{t} = t_m + \frac{h}{2}$ і визначити \tilde{y} за тим же методом Ейлера $\tilde{y} = y_m + \frac{h}{2} \cdot k_1$. Тоді підставлення у (20) і (19) дає (k_1 має той самий зміст (20)):

$$F(t_m, y_m, h) = k_1 + \frac{h}{2} \cdot \frac{Z(t_m + h/2, y_m + h \cdot k_1 / 2) - k_1}{h/2} = Z(t_m + h/2, y_m + h \cdot k_1 / 2) \quad (8.21)$$

Геометричну інтерпретацію цієї формули наведено на рис. 8.5.

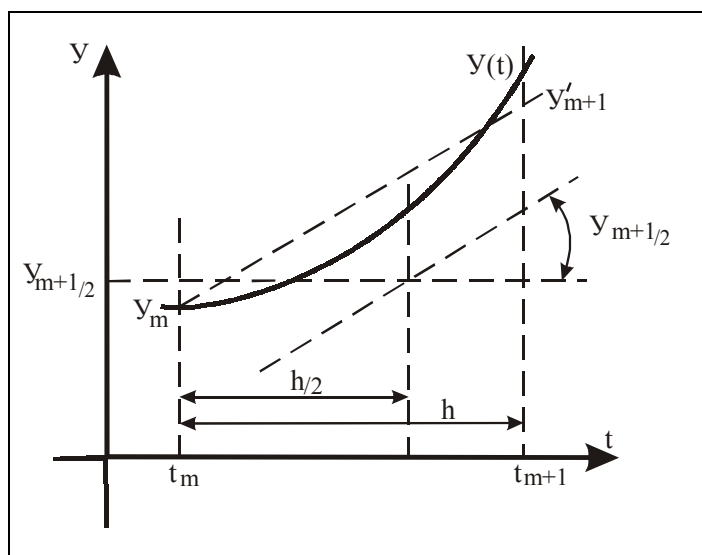


Рис. 8.5. Геометрична інтерпретація другої модифікації методу Ейлера

Вона зводиться до того, що спочатку методом Ейлера відшукують значення змінної y у середній точці (крок $h/2$), знаходять кут нахилу дотичній до інтегральної кривої у цій точці, потім проводять пряму через початкову точку y_m з цим нахилом до перетинання з вертикальною прямою з абсцисою t_{m+1} .

Обидві виведені формули характеризують методи Рунге-Кутта другого порядку, тобто такі, похибка обмежування яких є пропорційною до h^2 . Якщо при користуванні цими методами зменшити крок інтегрування у 10 разів, то похибка обмежування зменшиться у 100 разів.

8.1.4. Метод Хойне

Переходячи до методів третього порядку, залишимо у ряді (7) перші три члени

$$F(t_m, y_m, h) = y'_m + h \cdot \frac{y''_m}{2} + h^2 \frac{y'''_m}{6} . \quad (8.22)$$

Щоб скористуватися співвідношенням (22), треба обчислити додатково вже дві похідні - другого і третього порядку - у попередній точці.

Зазначимо, що для чисельного оцінювання похідних потрібно інтерполювати кілька точок інтегральної кривої. Для відшукування значення першої похідної потрібно мати, як мінімум, дві точки, щоб провести через них пряму, тангенс кута нахилу якого і буде шуканою похідною. Саме тому у попередніх методах необхідно було обчислювати (наближено) додаткову точку інтегральної кривої.

З огляду на це, у розглядуваному випадку задля обчислення двох додаткових (окрім першої, яка дається самим рівнянням (1)) похідних необхідно одержати значення інтегральної кривої хоча б ще у двох додаткових точках, причому ці значення потрібно визначати з точністю, яка забезпечується методами другого порядку.

Першою точкою, як завжди, є початкова точка (t_m, y_m) , в якій похідна дорівнює

$$k_1 = Z(t_m, y_m) . \quad (8.23)$$

Першу додаткову точку визначимо усередині інтервалу кроку інтегрування $t_2 = t_m + h/2$. Значення першої похідної від інтегральної кривої у цій точці визначимо за допомогою формули (21) другого порядку:

$$k_2 = Z(t_m + h/2, y_m + h \cdot k_1 / 2) . \quad (8.24)$$

Щоб визначити похідну k_3 у другій додатковій точці з аргументом $t_3 = t_m + h$, припустимо спочатку, що друга похідна на усьому інтервалі кроку інтегрування є сталою величиною, і тому різниці перших похідних на двох підінтервалах кроку дорівнюють один одному

$$k_3^* - k_2 = k_2 - k_1,$$

звідки

$$k_3^* = 2k_2 - k_1 .$$

Визначимо значення y_3 інтегральної кривої, використовуючи форму Ейлера для інтегрування назад і тому як похідну приймаючи відшукане значення

$$y_3 = y_m + h \cdot k_3^* = y_m + h \cdot (2k_2 - k_1) .$$

З врахуванням цього уточнене значення похідної у цій точці визначиться формулою

$$k_3 = Z(t_m + h, y_m + h \cdot (2k_2 - k_1)). \quad (8.25)$$

Тепер проведемо через визначені три точки (t_m, k_1) , $(t_m + h/2, k_2)$ і $(t_m + h, k_3)$ інтерполіюючу параболу і, використовуючи формули (7.2) і (7.5), знайдемо значення другої

$$y_m'' = \frac{4k_2 - 3k_1 - k_3}{2(h/2)} = \frac{4k_2 - 3k_1 - k_3}{h}$$

і третьої похідної

$$y_m''' = \frac{-2k_2 + k_1 + k_3}{(h/2)^2} = \frac{4(k_3 - 2k_2 + k_1)}{h^2}$$

у точці (t_m, y_m) .

Нарешті, враховуючи знайдені значення похідних у (22), одержимо остаточно формулу

$$\begin{aligned} F(t_m, y_m, h) &= y_m' + h \cdot \frac{y_m''}{2} + h^2 \frac{y_m'''}{6} = \\ &= k_1 + \frac{h}{2} \cdot \frac{-3k_1 + 4k_2 - k_3}{h} + \frac{h^2}{6} \cdot \frac{4(k_3 - 2k_2 + k_1)}{h^2} = \\ &= \frac{1}{6}(k_1 + 4k_2 + k_3). \end{aligned} \quad (8.26)$$

Це й є формула метода Хойне третього порядку.

Аналогічно можна побудувати формули чисельного інтегрування для методів вищого порядку.

8.1.5. Розповсюдження результатів на довільні системи диференціальних рівнянь

Одержані результати доволі просто узагальнюються на випадок довільної системи диференціальних рівнянь, якщо попередньо привести початкову систему ДР до нормальної форми Коші (2).

Одержаними раніше результатами можна користуватися практично без змін. Необхідно лише мати на увазі, що у загальному випадку системи n -го порядку:

- 1) кожна з функцій $Z_k(t, y_1, y_2, \dots, y_n)$ залежить від усіх n змінних стану; через те для обчислення їх значення у будь якій точці потрібно знати значення у ній усіх цих змінних;
- 2) розрахункові величини k_i є вже не числами, а векторами з кількістю елементів, що дорівнює кількості змінних стану (і порядку системи ДР);
- 3) головна розрахункова формула (3) набуває вигляду

$$y_{l,m+1} = y_{l,m} + h \cdot F_l(t_m, y_{1,m}, y_{2,m}, \dots, y_{n,m}, h); (l = 1, 2, \dots, n) \quad (8.27)$$

4) розрахункові формули для похідних набувають вигляду

$$F_l(t_m, y_{1,m}, y_{2,m}, \dots, y_{n,m}, h) = F_{i,k}(k_{1,l}, k_{2,l}, \dots, k_{i,l}), \quad (8.28)$$

де i - порядок метода; конкретний вид залежностей аналогічний тим, що виведений для однієї змінної.

Тому процес обчислення на кожному кроці слід оформлювати як цикл за усіма n змінними стану.

Нижче наведено таблицю 8.1, в якій зведені формули обчислень для методів Рунге-Кутта до четвертого порядку включно.

Таблиця 8.1. Методи Рунге-Кутта

Загальна формула $y_{m+1} = y_m + h F(t_m; y_m)$

№ МЕТОДА	Формула методу	Допоміжні величини	Назва методу
1	$F=k_1$	$k_1 = Z(t_m; y_m)$	Ейлера
2	$F=(k_1+k_2)/2$	$k_1=Z(t_m; y_m);$ $k_2=Z(t_m+h; y_m+hk_1)$	модифікований Ейлера
2	$F=Z(t_m+h/2; y_m+hk_1/2)$	$k_1=Z(t_m; y_m)$	
3	$F=(k_1+4k_2+k_3)/6$	$k_1=Z(t_m; y_m);$ $k_2=Z(t_m+h/2; y_m+hk_1/2);$ $k_3=Z(t_m+h; y_m+h(2k_2-k_1))$	Хойне
3	$F=(k_1+3k_3)/4$	$k_1=Z(t_m; y_m);$ $k_2=Z(t_m+h/3; y_m+hk_1/3);$ $k_3=Z(t_m+2h/3; y_m+2hk_2/3)$	
4	$F=(k_1+2k_2+2k_3+k_4)/6$	$k_1=Z(t_m; y_m);$ $k_2=Z(t_m+h/2; y_m+hk_1/2);$ $k_3=Z(t_m+h/2; y_m+hk_2/2);$ $k_4=Z(t_m+h; y_m+hk_3)$	Рунге-Кутта
4	$F=(k_1+3k_2+3k_3+k_4)/8$	$k_1=Z(t_m; y_m);$ $k_2=Z(t_m+h/3; y_m+hk_1/3);$ $k_3=Z(t_m+2h/3; y_m+h(k_2-k_1/3));$ $k_4=Z(t_m+h; y_m+h(k_1-k_2+k_3))$	

8.2. Багатокрокові методи

Визначальною рисою однокрокових методів є те, що для обчислення наступної точки використовується інформація лише про одну безпосередньо передуючу точку. Інформація про інші передуючі точки (за два, три кроки і т.і.) не використовується. При цьому у методах порядку вище за першій доводиться

обчислювати значення функцій правих частин рівнянь у кількох проміжних точках. Це здається нераціональним, оскільки, якщо процес інтегрування вже просунувся на декілька кроків, до послуг обчислювача є готова додаткова інформація, для використання якої не потрібні ніякі обчислення, а саме - інформація про декілька попередніх точок розв'язку. Через це доцільно розглянути методи, які могли б використовувати цю додаткову інформацію про попередні точки. Ці методи, внаслідок зазначеного, називають *багатокроковими*.

Загальною характерною їх рисою є те, що *за допомогою них неможливо розпочати процес чисельного інтегрування* (розв'язування задачі Коші), бо у початкову мить відомою є тільки одна точка, яка визначається початковими умовами системи ЗДР. Щоб розпочати розв'язування ЗДР за допомогою багатокрокових методів, потрібно спочатку застосувати один з однокрокових методів Рунге-Кутта.

Багатокрокові методи називають ще *методами прогнозу-корекції*. Це обумовлено тим, що розрахунки за цими методами побудовані на застосуванні двох послідовних етапів. На першому на основі інформації про попередні точки прогнозується значення шуканої змінної стану у наступній точці. На другому здійснюється уточнення ("корегування") знайденого значення за спеціальною неявною формулою.

Природньо, обчислення за формулою корекції здійснюються ітераційно кілька разів доти, поки у неявній формулі значення шуканої змінної справа і зліва формули корекції не збіжуться з потрібною точністю. Зазвичай за малих величини кроку інтегрування цей процес є завжди збіжним (і досить швидко – зазвичай - не більше кількох ітерацій). Але навіть повний збіг шуканої змінної у формулі корекції не означає, що знайдене значення є точним, через наближеність самої формули корекції.

Формули прогнозу і корекції багатокрокових методів до четвертого порядку наведені у таблиці 8.2.

Зазначимо, що порядок метода, як і раніше, характеризує показник степеня у залежності похибки метода від кроку інтегрування. Для багатокрокових методів порядок метода збігається з кількістю попередніх точок, інформація про які використовується в формулах цього методу. Так, методи другого порядку використовують дві точки (t_m, y_m) і (t_{m-1}, y_{m-1}) , а методи четвертого порядку - чотири - (t_m, y_m) , (t_{m-1}, y_{m-1}) , (t_{m-2}, y_{m-2}) і (t_{m-3}, y_{m-3}) .

Розглядаючи наведені у таблиці 8.2 формули, слід узяти до уваги, що більшість величин, що складають їх, не потрібно обчислювати, перш за все функції $Z(t_{m-k}, y_{m-k})$ ($k = 1, 2, \dots$). Це й робить методи прогнозу-корекції вельми ефективними у обчислювальному відношенні.

Головні особливості застосування багатокрокових методів пояснимо на прикладі метода другого порядку.

Таблиця 8.2. Багатокрокові методи

Поря док	Формула прогнозу	Формула корекції	Назва методу
2	$y_{m+1}=y_{m-1}+2hZ(t_m,y_m)$	$y_{m+1}=y_m+h[Z(t_{m+1};y_{m+1}^*)+Z(t_m;y_m)]/2$	
2	$y_{m+1}=y_m+h[Z(t_m;y_m)-Z(t_{m-1};y_{m-1})]/2$	$y_{m+1}=y_m+h[Z(t_{m+1};y_{m+1}^*)+Z(t_m;y_m)]/2$	
3	$y_{m+1}=y_m+h[23Z(t_m;y_m)-16Z(t_{m-1};y_{m-1})+5Z(t_{m-2};y_{m-2})]/12$	$y_{m+1}=y_m+h[5Z(t_{m+1};y_{m+1}^*)+8Z(t_m;y_m)-Z(t_{m-1};y_{m-1})]/12$	
4	$y_{m+1}=y_m+h[55Z(t_m;y_m)-59Z(t_{m-1};y_{m-1})+37Z(t_{m-2};y_{m-2})-9Z(t_{m-3};y_{m-3})]/24$	$y_{m+1}=y_m+h[9Z(t_{m+1};y_{m+1}^*)+19Z(t_m;y_m)-5Z(t_{m-1};y_{m-1})+Z(t_{m-2};y_{m-2})]/24$	Адамса-Башфорта
4	$y_{m+1}=y_{m-3}+4h[2Z(t_m;y_m)-Z(t_{m-1};y_{m-1})+2Z(t_{m-2};y_{m-2})]/3$	$y_{m+1}=y_{m-1}+h[Z(t_{m+1};y_{m+1}^*)+4Z(t_m;y_m)+Z(t_{m-1};y_{m-1})]/3$	Мілна
4	$y_{m+1}=y_{m-3}+4h[2Z(t_m;y_m)-Z(t_{m-1};y_{m-1})+2Z(t_{m-2};y_{m-2})]/3$	$y_{m+1}=\{9y_m - y_{m-2} + 3h[Z(t_{m+1};y_{m+1}^*)+2Z(t_m;y_m)-Z(t_{m-1};y_{m-1})]\}/8$	Хеммінга

Для прогнозу використаємо формулу (див. перший рядок таблиці 8.1):

$$y_{m+1}^{(0)} = y_{m-1} + 2h \cdot Z(t_m, y_m), \quad (8.29)$$

де горішній індекс (0) означає початкове наближення до y_{m+1} , тобто спрогнозоване значення. Безпосередньо з формули випливає, що її неможливо використати для відшукування першої точки y_1 , через те, що для цього потрібна інформація про точку, яка передуює точці y_0 (яка відповідає початковим умовам). Щоб розпочати чисельне розв'язування ДР за допомогою метода прогнозу-корекції, потрібно спочатку скоритися методом Рунге-Кутта.

Геометрично прогнозування зводиться до того, що відшукується кут нахилу дотичної до інтегральної кривої у точці (t_m, y_m) (пряма L_1 на рис. 8.6).

Після цього через точку (t_{m-1}, y_{m-1}) проводиться пряма L , паралельна L_1 . Передбачуване значення $y_{m+1}^{(0)}$ буде відповідати ординаті точки перетину прямої L з прямою, паралельною осі ординат і з абсцисою $t = t_{m+1}$.

Тепер потрібен деякий метод корегування (уточнення) відшуканого значення. Через те, що нам наближено відома величина y_{m+1} , можна обчислити нахил дотичної у точці (t_{m+1}, y_{m+1}) . Цю дотичну зображено на рис. 8.7 і позначено L_2 .

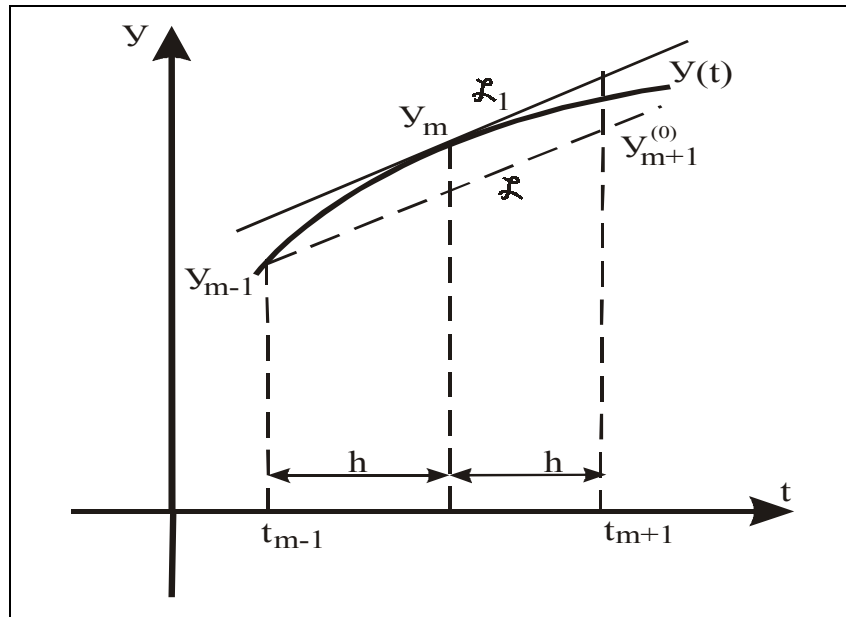


Рис. 8.6. Схема процесу прогнозу

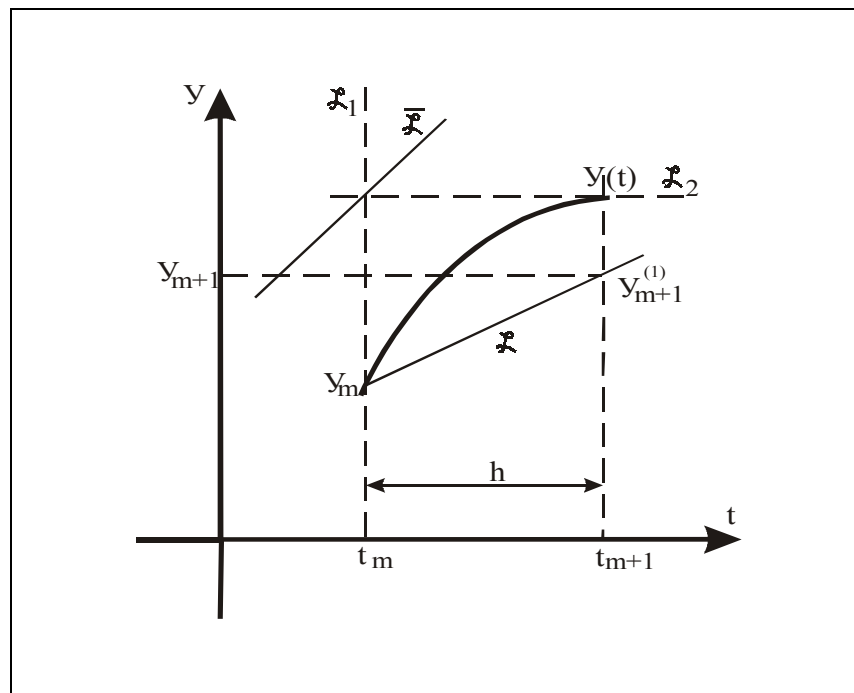


Рис. 8.7. Схема корегування

Пряма L_1 на цьому рисунку подає собою дотичну до інтегральної кривої у точці (t_m, y_m) . Проведемо бісектрису L кута між цими двома дотичними, і через ту ж точку (t_m, y_m) проводимо пряму, паралельну L до перетинання з прямою $t = t_{m+1}$. Ордината цієї точки перетинання дає нове (скореговане) наближення $y_{m+1}^{(1)}$.

Аналітично описана процедура відображується формулою (див. таблицю 8.2):

$$y_{m+1}^{(1)} = y_m + h \cdot \frac{Z(t_m, y_m) + Z(t_{m+1}, y_{m+1}^{(0)})}{2}.$$

Можна зробити нове, вочевидь ще краще, наближення до y_{m+1} , використовуючи значення $y_{m+1}^{(1)}$ і корегуючи знову. У загальному випадку k -те наближення обчислюється за аналогічною формулою

$$y_{m+1}^{(k)} = y_m + h \cdot \frac{Z(t_m, y_m) + Z(t_{m+1}, y_{m+1}^{(k-1)})}{2}. \quad (8.30)$$

Ітераційний процес припиняється, коли

$$|y_{m+1}^{(k+1)} - y_{m+1}^{(k)}| < \varepsilon \quad (8.31)$$

для деякої додатної заданої величини ε припустимої похибки.

Можна довести, що за досить малих кроках інтегрування

$$h < \frac{2}{\left| \frac{\partial Z}{\partial y} \right|_{\max}}$$

ітераційний процес є збіжним. Це означає, що послідовні значення $y_{m+1}^{(k)}$ прагнуть до деякого певного значення, але зовсім не обов'язково до точного розв'язку. Різниця між першим і другим є похибкою обмежування.

Перейдемо до аналізу похибок метода. Щоб визначити похибки обмежування при прогнозуванні (29), нагадаємо, що ряд Тейлора для $y(t)$ має вид (4).

Покладаючи у (4) $t = t_{m+1}$, одержимо ряд (6), а покладаючи $t = t_{m-1}$, одержимо

$$y_{m-1} = y_m - h \cdot y'_m + h^2 \cdot \frac{y''_m}{2} - h^3 \frac{y'''_m}{6} + \dots, \quad (8.32)$$

Якщо відняти з ряду (6) ряд (32), тоді матимемо

$$y_{m+1} = y_{m-1} + 2h \cdot y'_m + h^3 \frac{y'''_m}{3} + \dots. \quad (8.33)$$

Локальна похибка формули прогноза (29) визначається з співставлення точної (33) і наближеної (29) формул:

$$e_{np} = \frac{h^3}{3} y'''_m. \quad (8.34)$$

Перейдемо до похибок обмежування, що виникають в процесі корегування. Для цього, використовуючи (1), запишемо ряд (6) у дещо іншій формі:

$$y_{m+1} = y_m + h \cdot Z_m + h^2 \cdot \frac{Z'_m}{2} + h^3 \frac{Z''_m}{6} + \dots. \quad (8.35)$$

Аналогічний ряд, очевидно, можна записати і для функції $Z(t)$:

$$Z_{m+1} = Z_m + h \cdot Z'_m + h^2 \cdot \frac{Z''_m}{2} + h^3 \frac{Z'''_m}{6} + \dots,$$

звідки визначимо

$$Z'_m = \frac{Z_{m+1} - Z_m}{h} - \frac{h}{2} Z''_m - \dots \quad (8.36)$$

Підставляючи (36) у (35), одержимо

$$y_{m+1} = y_m + \frac{h}{2} \cdot (Z_m + Z_{m+1}) - \frac{h^3}{12} Z''_m + \dots \quad (8.37)$$

Формула (37) є точною. Тому локальна похибка корегування, яка здійснюється за наближеною формулою (30), визначатиметься додатковим доданком:

$$e_{\text{кор}} = -\frac{h^3}{12} y'''_m \quad (8.38)$$

Отже, у відповідності з (33)

$$y_{m+1} = y_{m+1}^{(0)} + e_{\text{нр}} \quad (8.39)$$

а у відповідності до (37)

$$y_{m+1} = y_{m+1}^{(k)} + e_{\text{кор}} \quad (8.40)$$

Віднімаючи ці дві рівності, одержимо з врахуванням (38) і (34)

$$y_{m+1}^{(0)} - y_{m+1}^{(k)} = e_{\text{кор}} - e_{\text{нр}} = -\frac{5}{12} y'''_m \quad (8.41)$$

У кінцевому підсумку після завершення процесу корегування остаточною похибкою усього методу прогнозу-корекції стає похибка обмежування корегування. З порівняння (38) і (41) впливає вельми просте правило оцінки похибки методу

$$e = e_{\text{кор}} = \frac{y_{m+1}^{(0)} - y_{m+1}^{(k)}}{12} \quad (8.42)$$

Величини, що потрібні для такого оцінювання, є просто результатами обчислень по формулах (29) і (30). Отже, на противагу однокроковим методам, метод прогнозу-корекції дає змогу визначити як побічний продукт обчислень оцінку похибки обмежування методу.

Примітка. Тут намічається деякий парадокс. Методи прогнозу-корекції дозволяють оцінити похибку обмежування, і, отже, визначити, як потрібно змінити величину кроку інтегрування, щоб похибка методу розміщувалася у заданих межах. Але самі методи прогнозу-корекції побудовані на припущенні, що величина кроку між відомими точками є незмінною, і тому не дозволяють змінити крок у процесі інтегрування. Для змінювання кроку слід спочатку перейти до відповідного однокрокового методу, а вже згодом, після накопичення потрібних даних з новим кроком, знову перейти до багатокрокового методу. У той же час, навпаки, однокрокові методи, як такі, що можуть самі стартувати, пристосовані для змінювання кроку хоч на кожному кроці. Але їхні алгоритми не дають змоги оцінити локальну похибку методу, щоб на цій основі обчислити потрібну величину кроку.

Перейдемо до оцінювання глобальної похибки $\delta = y_u - y$.

Для цього можна використати раніше доведене рівняння похибок (17). Потрібно лише для розглядуваного випадку конкретизувати вираз $D(y)$ похибки обчислення похідної. Міркуватимемо наступним чином.

Формула прогноза (29) є еквівалентною обчисленню похідної за формулою

$$(y')_u = \frac{y_{m+1} - y_{m-1}}{2h} = \frac{\Delta y_u}{\Delta t},$$

яка відрізняється від точної (33) на величину похибки

$$D(y) = (y')_u - y' = -\frac{h^2}{6} y''' = -\frac{e_{np}}{2h}. \quad (8.43)$$

Ми одержали співвідношення між похибкою обчислення похідної і локальною похибкою. Внаслідок корегування похибка прогнозу e_{np} заміниться на похибку корекції $e_{кор}$, і похибка подання похідної стане рівною

$$D(y_u) = -\frac{e_{кор}}{2h} = \frac{h^2}{24} y'''. \quad (8.43)$$

Тепер рівняння похибок (17) для метода прогнозу-корекції другого порядку набуде вигляду

$$\delta' - \frac{\partial Z}{\partial y} \cdot \delta = D(y) = \frac{h^2}{24} y'''. \quad (8.44)$$

Що й підтверджує, що метод має саме другий порядок.

8.3. Загальна характеристика явних методів

Однокрокові і багатокрокові методи разом складають так звані *явні методи* чисельного інтегрування диференціальних рівнянь. Назву їх обумовлено тим, що усі вони задля відшукування значень фазових змінних (змінних стану) на наступному кроці використовують інформацію лише про попередні точки.

Нагадаємо головні властивості кожної з двох груп явних методів.

Однокрокові методи

1. Через те, що у методах Рунге-Кутта використовується інформація лише про останню обчислену точку інтегральної кривої, за допомогою цих методів можна починати інтегрування ДР у випадку задачі Коші.
2. З тієї ж причини, ці методи дозволяють у процесі інтегрування легко змінювати його крок.
3. Однак, з тієї ж причини при використанні цих методів доводиться багаторазово обчислювати значення функцій $Z_k(t, y_1, y_2, \dots, y_n)$ ($k = 1, \dots, n$) і витрачувати на це багато машинного часу.
4. При застосуванні цих методів важко одержати оцінку виникаючої при інтегруванні похибки обмежування (для цього потрібні додаткові обчислення приблизно того ж обсягу, що й саме інтегрування).

Методи прогнозу-корекції

1. Через те, що у цих методах використовується інформація про декілька попередніх точки інтегральної кривої, розділені за аргументом однаковими інтервалами, за їх допомогою неможливо розпочати процес інтегрування у випадку задачі Коші.
2. Оскільки у цих методах замість обчислення $Z_k(t, y_1, y_2, \dots, y_n)$ ($k = 1, \dots, n$) у попередніх точках використовується значення, одержані при попередніх обчисленнях і збережені у пам'яті ЕОМ, вони є більш економічними у сенсі витрат машинного часу.
3. При будь-якому змінюванні величини кроку інтегрування, доводиться тимчасово звертатися до методів Рунге-Кутта.
4. У цих методах легко одержати добру оцінку похибки обмежування.

Загальним для явних методів є така, ще не обговорювана обставина: явні методи, при кроках інтегрування, більших за деякий крок H_{\max} , стають нестійкими у обчислювальному відношенні. Тобто якщо

$$h \geq H_{\max}, \quad (8.45)$$

похибка обмежування метода починає катастрофічно збільшуватися з кожним кроком, внаслідок чого одержувані результати інтегрування перестають хоч в якійсь мірі відображувати якісно і кількісно дійсну інтегральну криву.

Величина H_{\max} залежить від властивостей самої системи диференціальних рівнянь, яку потрібно чисельно інтегрувати і тому не може бути визначена заздалегідь, якщо невідома ця система ДР. Для систем лінійних ДР з постійними коефіцієнтами цю величину можна наближено обчислити за формулою

$$H_{\max} = \frac{T_{\min}}{N}, \quad (8.46)$$

де T_{\min} означає мінімальну за величиною сталу часу системи ДР, що інтегрується, N - деяке число, більше за одиницю, різне для різних методів інтегрування (наприклад, для метода Ейлера $N=20$, а для метода Рунге-Кутта 4-го порядку $N=5$). У власну чергу сталі часу T_i ($i = 1, 2, \dots, n$) можна визначити як величині, зворотні до значень дійсних та уявних частин (точніше – їхніх модулів) коренів характеристичного рівняння системи ДР.

Таким чином, щоб організувати процес чисельного інтегрування якоїсь системи диференціальних рівнянь одним із явних методів, потрібно здійснити такі дії:

- 1) визначити (обчислити за заданими значеннями параметрів диференціальних рівнянь) сталі часу заданої системи ДР;
- 2) відкинувши сталі часу, що дорівнюють нескінченності (які відповідають нульовим кореням характеристичного рівняння), знайти серед них максимальну T_{\max} за величиною і мінімальну T_{\min} ;
- 3) задатися кількістю n_{int} (>1) кроків інтегрування на інтервалі часу, що дорівнює мінімальній сталій часу; при цьому необхідно, щоб n_{int} було

більше за сталу N для заданого методу; слід при цьому пам'ятати також, що чим більшу за величиною n_{int} ми задаємо, тим з більшою точністю здійснюватиметься процес чисельного інтегрування, але й тим більший час він займе;

4) розрахувати величину кроку інтегрування за формулою

$$h = \frac{T_{\text{min}}}{n_{\text{int}}}; \quad (8.47)$$

5) задатися бажаною тривалістю процесу, що виходить внаслідок інтегрування, у вигляді кількості n_T максимальних сталих часу системи; розрахувати бажану тривалість процесу за формулою

$$T = n_T \cdot T_{\text{max}}; \quad (8.48)$$

6) здійснити присвоювання змінним стану y_1, y_2, \dots, y_n початкових значень;

7) організувати цикл за часом t , який би змінювався від 0 до T з кроком h , усередині якого за заданими значеннями вектора $y = [y_1, y_2, \dots, y_n]$ у попередній точці обчислюється значення того ж вектора у наступній точці, а також на основі цих розрахунків формуються масиви, які у подальшому будуть використовуватися як результати інтегрування.

Приклади організації програм чисельного інтегрування наведені у [5] і [6], п. 2.6.2, п. 2.7. Як можна організувати чисельне інтегрування засобами SimuLink показано у [6], п. 7.3.

8.4. Жорсткі системи. Неявні методи

В інженерній практиці трапляються випадки, коли відношення максимальної сталої часу T_{max} системи до мінімальної T_{min} є вельми великою величиною (наприклад, більше, ніж 10^5). Такі системи диференціальних рівнянь називаються *жорсткими*.

Чисельне інтегрування жорстких систем за допомогою явних методів натикається на суттєві труднощі. У цьому випадку загальна кількість кроків інтегрування, потрібна для повного інтегрування

$$k_{\text{int}} = \frac{T_{\text{max}}}{T_{\text{min}}} \cdot n_{\text{int}} \quad (8.49)$$

буде теж вельми великою. А через те, що похибки округлення накопичуються зі збільшенням кількості обчислень приблизно пропорційно цій кількості, на практиці вже після мільона операцій підсумовування-множення похибки округлення можуть несподівано спотворити інтегральну криву. Якщо врахувати, що кожний крок інтегрування може вміщувати тисячі операцій підсумовування-множення, то вже через тисячі кроків інтегрування відносна похибка внаслідок округлення збільшується до 100%. Подальше інтегрування стає недоцільним.

Перебороти цю перешкоду можливо тільки у разі, якщо збільшити крок інтегрування до величини, у десятки разів більшої за мінімальну сталу часу си-

стеми. Для розглянутих явних методів це не є можливим, бо за цих умов ці методи є нестійкими.

Тому для інтегрування жорстких систем використовують так звані неявні методи.

Найпростішим шляхом розв'язування "жорстких" задач є так званий неявний метод Ейлера, за яким інтегрування на одному кроці здійснюється за неявною формулою

$$y_{m+1} = y_m + h \cdot Z(t_m, y_{m+1}). \quad (8.50)$$

На відміну від явних методів, цей метод є стійким при величинах кроку, набагато більших за мінімальну сталу часу, хоча точність його є помірною (порядка 10%).

8.5. Засоби чисельного інтегрування ЗДР системи MatLAB

Як можна було впевнитися, організація чисельного інтегрування системи звичайних диференціальних рівнянь (ЗДР) з фіксованим кроком інтегрування є досить громіздкою і включає низку дій з обчислення параметрів самої системи диференціальних рівнянь. Більш зручним у застосуванні є методи з автоматичним добиранням величини кроку. Саме такі вбудовані методи передбачені у ядрі середовища MatLAB. Це процедури *ode23* і *ode45*. Назва процедур - це абрєвіатура словосполучення Ordinary Differential Equations (звичайні диференціальні рівняння). Обидва методи здійснюють чисельне інтегрування з автоматичним добиранням кроку. У процедурі *ode23* головний метод інтегрування - метод Рунге-Кутта 2-го порядку. На кожному кроці інтегрування здійснюється також методом Рунге-Кутта 3-го порядку задля одержання оцінки похибки головного методу. Якщо оцінка похибки не вміщується у задані межі, здійснюється розрахунок нового значення кроку і перерахунок інтегральної кривої за нового значення кроку. Це повторюється доти, поки похибка не увійде у задані межі. Аналогічно працює і процедура *ode45*. Але у цьому випадку головним методом чисельного інтегрування є метод Рунге-Кутта 4-го порядку, а контроль величини кроку здійснюється методом 5-го порядку.

Звернення до процедур чисельного інтегрування ЗДР має вид:

[t, y] = ode23('ім'я функції', tspan, y0, options)

[t, y] = ode45('ім'я функції', tspan, y0, options),

де використані параметри мають такий зміст: <ім'я функції> - рядок символів, що є ім'ям М-файлу, у якому обчислюється вектор-функція $Z(t, y)$, тобто *праві частини системи ЗДР*; **y0** - вектор початкових значень змінних стану; **t** - масив значень аргументу, що відповідають крокам інтегрування; **y** - матриця проінтегрованих значень фазових змінних, в якій кожний стовець відповідає одній з змінних стану, а рядок містить значення змінних стану, що відповідають певному кроку інтегрування; **tspan** - вектор-рядок [t0 tfinal], що містить два значення:

t_0 - початкове значення аргументу t ; t_{final} - кінцеве значення аргументу; $options$ - рядок із параметрів, що визначають значення припустимої відносної й абсолютної похибок інтегрування.

Параметр $options$ можна не вказувати. Тоді за замовчуванням припустима відносна похибка інтегрування приймається рівною $1 \cdot 10^{-3}$, а абсолютна (по кожній із змінних стану) - $1 \cdot 10^{-6}$. Якщо ж по якихось параметрах ці значення не влаштовують користувача, треба перед зверненням до процедури чисельного інтегрування встановити нові значення припустимих похибок за допомогою процедури *odeset* у такий спосіб:

options = odeset('RelTol',1e-4,'AbsTol',[1e-4 1e-4 1e-5]).

Параметр *RelTol* визначає відносну похибку чисельного інтегрування по усіх фазових змінних одночасно, а *AbsTol* є вектором-рядком, що складається з абсолютних припустимих похибок чисельного інтегрування по кожній із фазових змінних.

8.6. Програма досліджування похибок чисельних методів інтегрування ЗДР

8.6.1. Постановка задачі створення програми

Метою утворення програми є дослідження за її допомогою похибок чисельного інтегрування деякого, не надто складного диференційного рівняння (але й не зовсім примітивного). Щоб одержати можливість обчислення похибки інтегрування, потрібно з чимось порівнювати результати чисельного інтегрування. Таким еталоном може бути точний розв'язок диференційного рівняння. Тоді, обчислюючи значення цього розв'язку при відповідному значенні аргументу і відшукуючи різницю між ним, і результатом чисельного інтегрування, можна одержати значення похибки у цей момент часу, а по тому, - побудувати графік залежності похибки інтегрування від аргументу.

Для цього годиться лінійне диференційне рівняння зі сталими коефіцієнтами, яким описується поведінка довільної коливальної ланки, наприклад, рух твердого тіла, зв'язаного з основою пружиною і демпфером, при дії на нього сили, яка змінюється у часі за гармонічним законом зі сталою складовою:

$$m \cdot \ddot{x} + f \cdot \dot{x} + c \cdot x = F_o + F_m \cdot \sin \omega t. \quad (8.6.1)$$

Тут позначено: x - переміщення тіла відносно основи у напрямку дії сили; m - маса тіла; c - жорсткість пружини; f - коефіцієнт в'язкого тертя демпфера; F_o - величина сталої складової діючої сили; F_m - амплітуда змінювання сили; ω - колова частота змінювання сили.

Приведення рівняння до безрозмірної форми

Інтегрування рівняння (1) можна значно спростити, не зменшуючи (навіть збільшуючи) узагальненості одержуваних результатів, якщо замість нього

розглядати його безрозмірний аналог, тобто таке рівняння, в якому усі величини не мають фізичної розмірності.

Для цього спочатку поділимо усі члени рівняння (1) поділимо на масу:

$$\ddot{x} + \frac{f}{m} \cdot \dot{x} + \frac{c}{m} \cdot x = \frac{F_o}{m} + \frac{F_m}{m} \cdot \sin \omega t. \quad (8.6.2)$$

Через те, що усі складові (члени) диференціального рівняння мають однакову фізичну розмірність, а розмірність другої похідної \ddot{x} величини x дорівнює розмірності останньої, поділеній на квадрат розмірності часу, розмірність коефіцієнта c/m у третьому члені лівої частини (2) дорівнюватиме T^{-2} (де T - позначення одиниці часу). З іншого боку значення цього коефіцієнту для стійких систем (а тільки такі системи існують у реальності, бо нестійкі системи є саморуйнівними) є завжди додатним. Тому цей коефіцієнт можна позначити як квадрат деякої величини:

$$\frac{c}{m} = \omega_0^2. \quad (8.6.3)$$

Величина

$$\omega_0 = \sqrt{\frac{c}{m}} \quad (8.6.4)$$

має розмірність колової частоти і вимірюється у радіанах у секунду. Як у подальшому стане ясным, вона є *частотою власних коливань системи*, описуваної рівнянням (1), за умови, що тертя відсутнє ($f = 0$).

Фізична розмірність коефіцієнта f/m при другому члені лівої частини (2) дорівнює, як неважко впевнитися, T^{-1} . З цього випливає, що його можна подати як добуток деякої безрозмірної величини (зазвичай її позначають $2 \cdot \zeta$) на величину введеної частоти власних коливань ω_0 :

$$\frac{f}{m} = 2 \cdot \zeta \cdot \omega_0.$$

Введений у такий спосіб безрозмірний коефіцієнт

$$\zeta = \frac{f}{2m \cdot \omega_0} = \frac{f}{2\sqrt{c \cdot m}} \quad (8.6.5)$$

зазвичай називають *відносним коефіцієнтом загасання*. Він характеризує демпфіруючі властивості коливальної ланки.

З врахуванням цих нових позначень рівняння (2) можна подати у вигляді:

$$\ddot{x} + 2 \cdot \zeta \cdot \omega_0 \cdot \dot{x} + \omega_0^2 \cdot x = \frac{F_o}{m} + \frac{F_m}{m} \cdot \sin \omega t. \quad (8.6.6)$$

Тепер введемо поняття безрозмірного часу τ , який пов'язаний зі звичайним часом t співвідношенням:

$$\tau = \omega_0 \cdot t. \quad (8.6.7)$$

Зміст безрозмірного часу досить простий, - це час, який вимірюється в одиницях періоду $T_0 = 2 \cdot \pi / \omega_0$ власних коливань системи.

Замінімо розмірний час у рівнянні (6) на безрозмірний. Для цього подамо похідну за часом у вигляді:

$$\dot{x} = \frac{dx}{dt} = \frac{dx}{d\tau} \frac{d\tau}{dt} = x' \cdot \omega_0.$$

Аналогічно одержимо:

$$\ddot{x} = \frac{d}{dt} \frac{dx}{dt} = \frac{d}{dt} (x' \cdot \omega_0) = \omega_0 \frac{d}{d\tau} (x') \frac{d\tau}{dt} = x'' \cdot \omega_0^2.$$

Тут позначкою "штрих" позначена похідна по безрозмірному часу. Підставляючи це у (6), одержимо остаточно:

$$x'' + 2 \cdot \zeta \cdot x' + x = a_0 + a_m \cdot \sin \nu \tau, \quad (8.6.8)$$

де використані такі додаткові позначення: $a_0 = \frac{F_0}{c}$; $a_m = \frac{F_m}{c}$; $\nu = \frac{\omega}{\omega_0}$ - від-

носна частота зовнішньої дії (відношення частоти зовнішньої дії до частоти власних незгасаючих коливань).

Рівняння (8) має такі переваги по відношенню до первісного рівняння (1):

- у ньому всього чотири коефіцієнти (ζ , a_0 , a_m і ν), що характеризують поведження системи, замість шести (m , f , c , F_0 , F_m і ω) у рівнянні (1); через те значно спрощується як подання рівняння при чисельному інтегруванні, так і вираз точного розв'язку;
- до форми (8) зводиться рівняння не тільки тіла під дією сили, а й рівняння маятника, електричного контуру, поплавка на воді, гіротахометру і безлічі інших систем, поведження яких с достатньою точністю можна описати диференціальним рівнянням другого порядку; тому рівняння у безрозмірній формі (8) описує поведження цілого класу реальних систем, і його вивчення дозволяє поширити одержані результати не тільки на систему, що вивчається, а на весь клас аналогічних систем.

Через ці обставини попереднє приведення початкових рівнянь до безрозмірної форми завжди є доцільним.

Відшукування точного розв'язку рівняння

Точний розв'язок лінійного диференціального рівняння зі сталими коефіцієнтами, згідно теорії диференціальних рівнянь може бути поданий як сума двох розв'язків:

$$x(\tau) = x_{oo}(\tau) + x_{ch}(\tau), \quad (8.6.9)$$

де $x_{oo}(\tau)$ - це загальний розв'язок відповідної однорідної системи диференціальних рівнянь (тобто такої, у якій відкинуті усі члени, що залежать тільки від часу; $x_{ch}(\tau)$ - деякий частковий розв'язок початкової системи.

У випадку рівняння (8) відповідне однорідне рівняння має вигляд:

$$x'' + 2 \cdot \zeta \cdot x' + x = 0, \quad (8.6.10)$$

Для відшукування його загального розв'язку потрібно знайти його характеристичне рівняння:

$$p^2 + 2\zeta \cdot p + 1 = 0,$$

відшукати його корені p_1 і p_2 :

$$p_1 = -\zeta + j \cdot \sqrt{1 - \zeta^2}; \quad p_2 = -\zeta - j \cdot \sqrt{1 - \zeta^2}$$

і подати загальний розв'язок у виді:

$$\begin{aligned} x_{oo}(\tau) &= C_1 \cdot e^{p_1 \tau} + C_2 \cdot e^{p_2 \tau} = \\ &= e^{-\zeta \cdot \tau} \cdot [A \cdot \sin(\sqrt{1 - \zeta^2} \cdot \tau) + B \cdot \cos(\sqrt{1 - \zeta^2} \cdot \tau)] \end{aligned} \quad (8.6.11)$$

Тут припущено, що розглядувана ланка є коливною ($\zeta < 1$). Сталі A і B є сталими інтегрування. Вони визначаються початковими умовами рівняння (8).

Частковий розв'язок рівняння (8) шукатимемо у вигляді:

$$x_{ch}(\tau) = A_0 + A_S \cdot \sin v\tau + A_C \cdot \cos v\tau. \quad (8.6.12)$$

Коефіцієнти A_0 , A_S , A_C визначаються шляхом підставлення розв'язку у вигляді (12) у початкове рівняння (8) і подальшого прирівнювання коефіцієнтів у правій і лівій частинах при

- сталих членах;
- синусах від $v\tau$;
- косинусах від $v\tau$.

В результаті виходять три алгебричних рівняння для визначення трьох невідомих коефіцієнтів, розв'язуючи які, одержимо:

$$A_0 = a_0; \quad (8.6.13)$$

$$A_S = \frac{1 - v^2}{(1 - v^2)^2 + 4\zeta^2 v^2} a_m; \quad A_C = -\frac{2\zeta v}{(1 - v^2)^2 + 4\zeta^2 v^2} a_m$$

Отже загальний розв'язок рівняння (8), враховуючи (9), має вигляд:

$$\begin{aligned} x(\tau) &= e^{-\zeta \cdot \tau} \cdot [A \cdot \sin(\sqrt{1 - \zeta^2} \cdot \tau) + B \cdot \cos(\sqrt{1 - \zeta^2} \cdot \tau)] + \\ &+ A_0 + A_S \sin v\tau + A_C \cos v\tau \end{aligned} \quad (8.6.14)$$

Сталі інтегрування A і B визначаються з початкових умов, за які приймемо

$$x(0) = x_0; \quad x'(0) = x'_0. \quad (8.6.15)$$

Підставляючи у вираз (14) і впливаючий з нього вираз для похідної $x'(\tau)$ значення $\tau = 0$ і умови (15), одержимо:

$$x_0 = B + A_0 + A_C; \quad x'_0 = A\sqrt{1-\zeta^2} - \zeta B + A_S \cdot \nu, \quad (8.6.16)$$

звідки випливає

$$B = x_0 - A_0 - A_C; \quad A = \frac{x'_0 + \zeta(x_0 - A_0 - A_C) - A_S \cdot \nu}{\sqrt{1-\zeta^2}}.$$

З врахуванням цього розв'язок рівняння (8) остаточно має вид:

$$x(\tau) = e^{-\zeta\tau} \cdot \left[\frac{x'_0 + \zeta(x_0 - A_0 - A_C) - A_S \cdot \nu}{\sqrt{1-\zeta^2}} \cdot \sin(\sqrt{1-\zeta^2} \cdot \tau) + \right. \\ \left. + (x_0 - A_0 - A_C) \cdot \cos(\sqrt{1-\zeta^2} \cdot \tau) \right] + A_0 + A_S \sin \nu\tau + A_C \cos \nu\tau \quad (8.6.17)$$

Для утворюваної програми важливо, з метою зменшення очікуваного терміну інтегрування диференційного рівняння, підібрати початкові умови з умови відсутності у вихідному (проінтегрованому) процесі власних коливань. У цьому випадку розв'язок (14) стає особливо простим і збігається з частковим розв'язком (12):

$$x(\tau) = A_0 + A_S \cdot \sin \nu\tau + A_C \cdot \cos \nu\tau. \quad (8.6.18)$$

Для цього початкові умови мають бути наступними:

$$x_0 = A_0 + A_C; \quad x'_0 = A_S \cdot \nu. \quad (8.6.19)$$

Приведення рівняння до форми Коші

Методи чисельного інтегрування орієнтовані на системи диференційних рівнянь, приведених до нормальної форми Коші. Тому приведемо рівняння (8) до цієї форми. Для цього позначимо:

$$x_1 = x; \quad x_2 = x'. \quad (8.6.20)$$

У цих нових змінних (змінних стану) рівняння (8) може бути подано у вигляді наступної системи диференціальних рівнянь першого порядку:

$$\frac{dx_1}{d\tau} = x_2; \\ \frac{dx_2}{d\tau} = -2\zeta \cdot x_2 - x_1 + a_0 + a_m \sin \nu\tau. \quad (8.6.21)$$

Якщо ввести позначення:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}; \quad A = \begin{bmatrix} 0 & 1 \\ -1 & -2\zeta \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \quad u = a_0 + a_m \sin \nu\tau, \quad (8.6.22)$$

систему (21) можна подати у такій матричній формі:

$$\frac{dx}{d\tau} = A \cdot x + B \cdot u. \quad (8.6.23)$$

Саме ця остання форма стане нам у пригоді при складанні програми чисельного інтегрування.

8.6.2. Розроблення програми

Кінцевою метою утворення програми є дослідження за її допомогою похибок інтегрування диференційного рівняння одним (за вибором) з однокрокових методів чисельного інтегрування.

Це можна зробити суто програмно. Для цього потрібно, по-перше, створити М-файл обчислення поточних значень правих частин рівняння (21), по-друге, - розробити М-файл відповідного однокрокового методу (бо в ядрі системи MatLab немає вбудованих процедур однокрокових методів), а, по-третє, - розробити головну програму, в якій би організовувався цикл покрокового звертання до файлу методу інтегрування і формувалися масиви даних про проінтегрований процес і його похибки. Приклад такої організації програми наведений у [5] і [6], п. п. 2.6.2 і 2.7.

Набагато простіший шлях полягає у використанні можливостей пакета SimuLink. По-перше, у цьому пакеті вже передбачені (існують) вбудовані процедури усіх однокрокових методів чисельного інтегрування впритул до методу шостого порядку. По-друге, - графічними засобами SimuLink значно простіше створити програму (блок) розрахунку правих частин будь-якої системи диференційних рівнянь за умови, що вони є лінійними і зі сталими коефіцієнтами. Для цього призначений блок *State Space* з розділу *Continuous* бібліотеки блоків SimuLink. По-третє, - при використанні пакета SimuLink немає потреби організовувати циклічний процес інтегрування і формування вихідних масивів даних, - все це робить сам пакет власними внутрішніми засобами.

Отже, почнемо з утворення блок-схеми S-моделі, яка здійснює чисельне інтегрування диференційного рівняння (8). Вона є дуже простою і наведена на рис. 8.8.

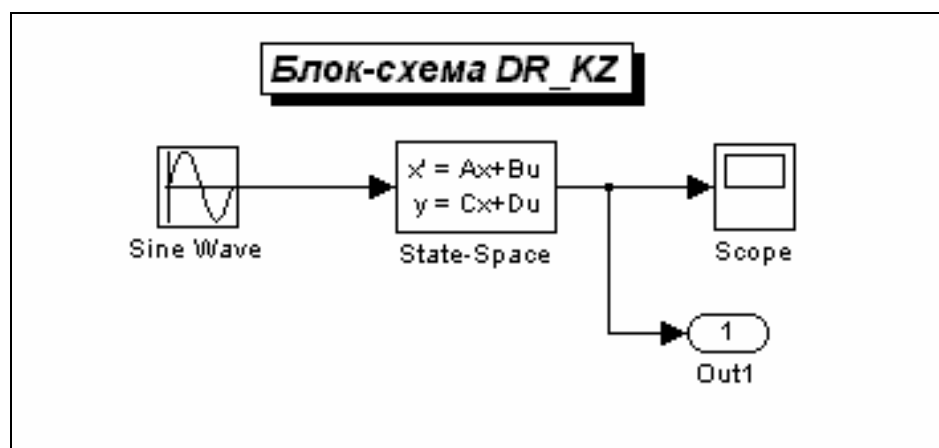


Рис. 8.8. Блок-схема S-моделі чисельного інтегрування ДР

Основу її складає блок *State-Space* (Простір Станів), який подає систему диференційних рівнянь, що підлягає чисельному інтегруванню. Як позначено

на зображенні блоку, диференційні рівняння системи мають бути приведені до матричної форми виду:

$$\frac{dx}{dt} = A \cdot x + B \cdot u;$$

$$y = C \cdot x + D \cdot u.$$

Тут використовуються наступні позначення:

u - вектор величин (процесів), що подаються на вхід блоку;

y - вектор величин, що "знімаються" з виходу блоку;

x - вектор змінних стану;

A, B, C, D - матриці з відомих членів, які визначають взаємозв'язок змінних стану, вхідних дій (u) і вихідних величин блоку.

Значення останніх матриць, а також початкові умови для процесу інтегрування, задаються у вікні настроювання блоку (рис. 8.9).

У розглядуваному випадку матриці A, B визначені у (22). Запишемо їх у відповідні віконця вікна настроювання блоку *State-Space*.

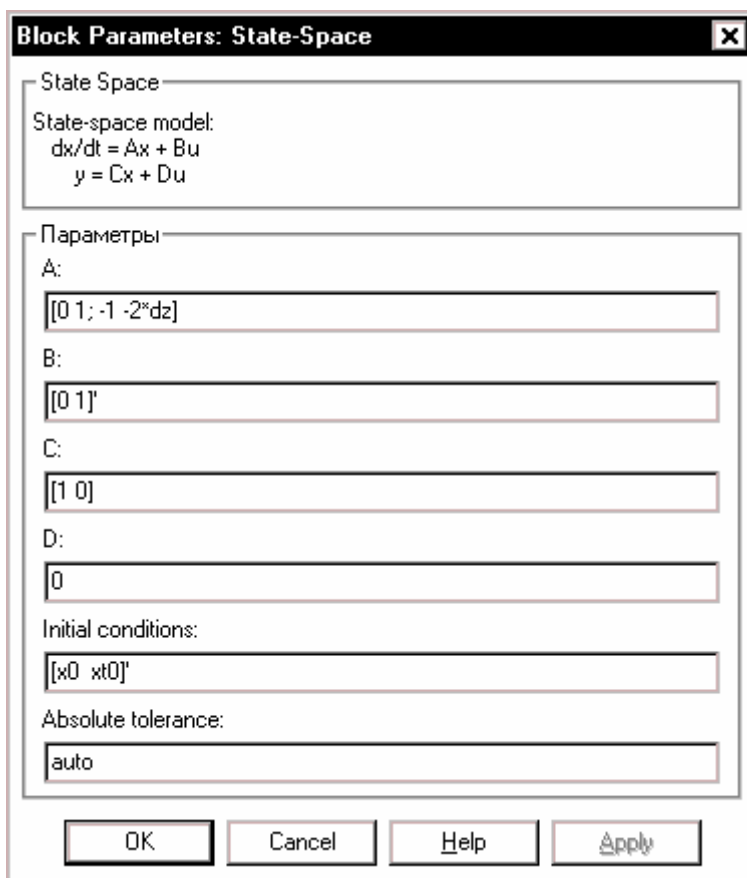


Рис. 8.9. Вікно настроювання блоку *State-Space*

Виход блоку сформуємо так, щоб його утворювала тільки величина переміщення x тіла, тобто

$$y = x_1 = [1 \quad 0] \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = [1 \quad 0] \cdot x.$$

Звідси випливає, що матриці C , D мають бути введені наступні
 $C = [1 \quad 0]$; $D = 0$.

На рис. 8.9. показане вікно настроювання з введеними вказаними значеннями матриць і початковими умовами (15). Формування величин усіх параметрів, що вказані у виді ідентифікаторів, має відбутися попередньо, перед запуском моделі на моделювання.

Після оформлення блок-схеми моделі в її вікні потрібно встановити параметри інтегрування. Для цього слід викликати меню *Simulation*, поділ *Parameters*, і у вкладинці *Solver* (Розв'язувач) встановити:

- у віконці *Type*, - режим *Fixed step* (фіксований крок) і у наступному віконці – заданий метод інтегрування (наприклад, *ode5*);
- у віконці *Stop time* (час припинення), - значення TK ;
- у віконці *Fixed step size* (розмір фіксованого кроку), - значення hi .

Ці установки відображені на рис. 8.10.

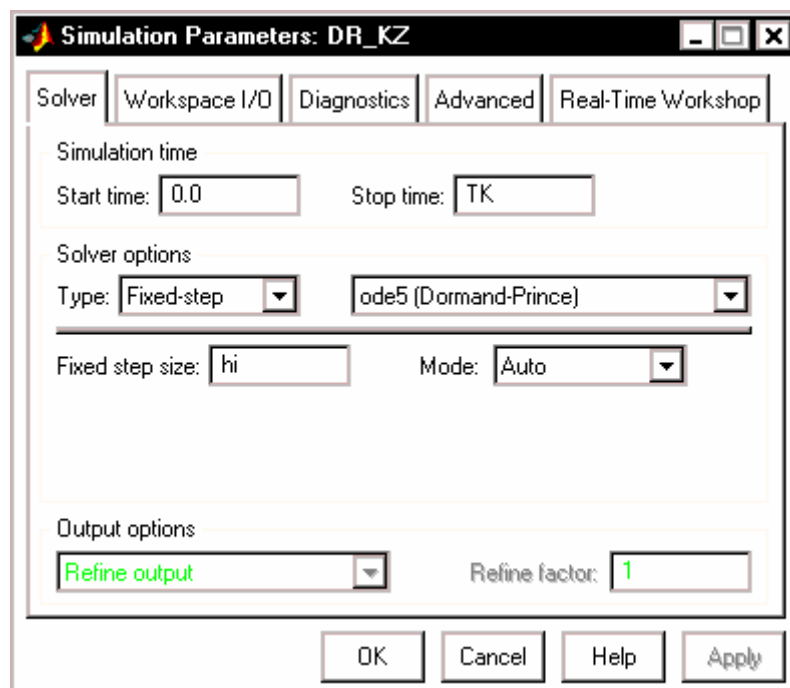


Рис. 8.10. Вкладка *Solver*

Наразі потрібно створити М-файл, в якому б, по-перше, присвоювалися значення усім параметрам рівняння і параметрам процесу чисельного інтегрування, по-друге, здійснювалося б звернення до утвореної S-моделі і, по-третє, забезпечувалося б виведення у графічне вікно графіків точного розв'язку, проінтегрованого процесу і похибки чисельного інтегрування з необхідними пояснювальними написами. Текст такої керуючої програми може бути наступним:

Файл DR_KZ_upr.m

```

% DR_KZ_upr.m
% Керуюча програма для дослідження похибок чисельного інтегрування
% диференційного рівняння  $x''+2*dz*x'+x=a0+am*\sin(nu*t)$ 

% Лазарєв Ю.Ф. 23-01-2003
%
clc
clear all
disp('Програма дослідження похибок чисельного інтегрування')
disp('диференційного рівняння  $x''+2*dz*x'+x=a0+am*\sin(nu*t)$ ')
disp(' при наступних значеннях параметрів')
% ===== Присвоювання значень параметрам рівняння =====
dz=0.04
a0=0
am=1
nu=0.2
% === Обчислення коефіцієнтів точного розв'язку ===
A0=a0;
znam=(1-nu^2)^2+4*dz^2*nu^2;
As=(1-nu^2)*am/znam;
Ac=-2*dz*nu*am/znam;
% === Обчислення початкових умов ===
x0=A0+Ac;
xt0=As*nu;
% == Встановлення параметрів процесу чисельного інтегрування ==
T1=2*pi; T2=T1/nu;
Tmin=min([T1,T2]); Tmax=max([T1,T2]);
Nhi=input('Уведіть кількість кроків на мінімальному періоді');
if isempty(Nhi)
    Nhi=20;
end
Nhi
hi=Tmin/Nhi;
NTK=input('Уведіть кількість максимальних періодів');
if isempty(NTK)
    NTK=4;
end
NTK
TK=Tmax*NTK;
%-----
% Звернення до S-моделі
sim('DR_KZ')
%-----
% ===== Обчислення значень точного розв'язку =====
t=tout;
x=yout(:,1);
xt=A0+As*sin(nu*t)+Ac*cos(nu*t);
% ===== Виведення графіків у графічне вікно =====
subplot(2,1,1)
Str1=' $x''+2*dz*x'+x=a0+am*\sin(nu*t)$  Чисельне інтегрування методом ode5';
plot(t,x,'.',t,xt),grid
title (Str1);
ylabel('Проінтегрований процес');
legend('чисельно','точний',0)
subplot(2,1,2)
plot(t,x-xt),grid
Str2=sprintf('Параметри: dz=%g; ',dz);

```

```

Str3=sprintf('a0=%g; ',a0);    Str4=sprintf('am=%g; ',am);
Str5=sprintf('nu=%g; ',nu);    Str=[Str2,Str3,Str4,Str5];
title(Str)
ylabel('Похибка чисельного інтегрування');
Str6=sprintf('Nhi=%g; ',Nhi);  Str7=sprintf('hi=%g',hi);
StrX=[Str6,Str7];
xlabel(['Безрозмірний час (Параметри інтегрування: ',StrX,')']);

```

8.6.3. Деякі попередні результати роботи з програмою

Далі (рис. 8.11...8.15) наведені результати роботи утвореного комплексу програм для наступних значень параметрів:

- метод чисельного інтегрування – *ode5*;
- параметри рівняння: $\zeta = 0,04$; $a_0 = 0$; $a_m = 1$; $\nu = 0,2$.

Рис. 8.11 відображує результати при кількості кроків інтегрування на мінімальному періоді $N_h = 20$, рис. 8.12, - при $N_h = 200$, рис. 8.13, - при $N_h = 500$, рис. 8.14, - при $N_h = 2,5$, рис.8. 15, - при $N_h = 2,2$.

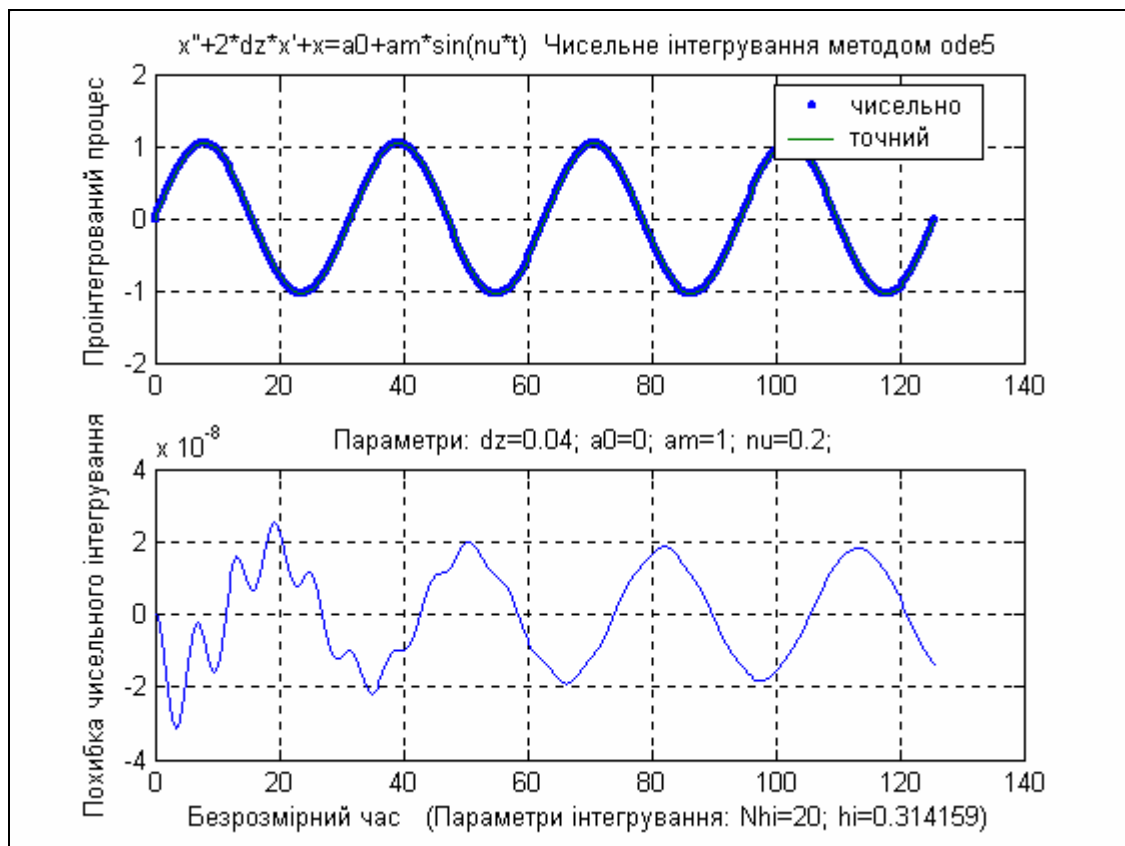


Рис. 8.11. Результати інтегрування при $N_k = 20$

Аналізуючи одержані результати, можна зробити висновки:

- 1) характер змінювання у часі похибки чисельного інтегрування при кількості кроків на мінімальному періоді від $N_h = 20$ до $N_h = 400$ нагадує характер змінювання у часі самої інтегрованої величини; вона складається з загасаючої складової, яка відповідає власним

коливанням з періодом у 6,28 і сталих вимушених коливань з утрічі більшим періодом 31,4; це є ознакою того, що у цьому діапазоні змінювання кроку інтегрування на першій план висуваються похибки методу інтегрування, які визначаються розв'язком рівняння похибок, аналогічному (18);

- 2) похибки чисельного інтегрування у цьому діапазоні змінювання кроку інтегрування зменшуються з його зменшенням пропорційно до величини кроку інтегрування у степені, який збігається з порядком методу інтегрування;

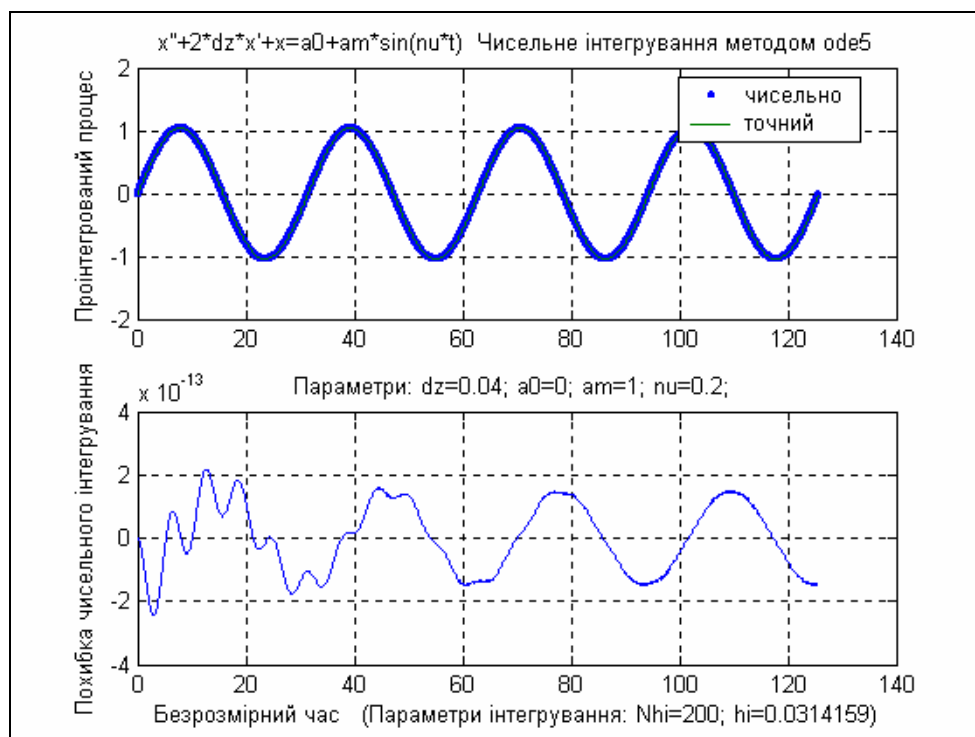


Рис. 8.12. Результати інтегрування при $N_k = 200$

- 3) при зменшенні кроку інтегрування менше за той, що відповідає кількості кроків $N_h = 500$ на мінімальному періоді, похибка інтегрування вже не зменшується, а, навпаки, збільшується; при цьому і характер змінювання похибки у часі теж різко змінюється, - в ньому вже важко виділити якісь періодичні складові, змінювання похибки набуває характеру випадкового процесу; це є ознакою того, що у цьому діапазоні змінювання кроків переважає друга складова похибки інтегрування – похибки внаслідок округлень при здійсненні операцій чисельного інтегрування;
- 4) якщо кількість кроків на мінімальному періоді збільшувати, процес чисельного інтегрування наближається до межі своєї стійкості; починаючи з $N_h = 2,5$ (рис. 8.14) похибка інтегрування (методу) починає збільшуватися у часі необмежено; при $N_h = 2,2$ (рис. 8.15) процес,

що одержується через чисельне інтегрування, вже не має нічого спільного з точним розв'язком диференціального рівняння, як говорять, метод "вибухає".

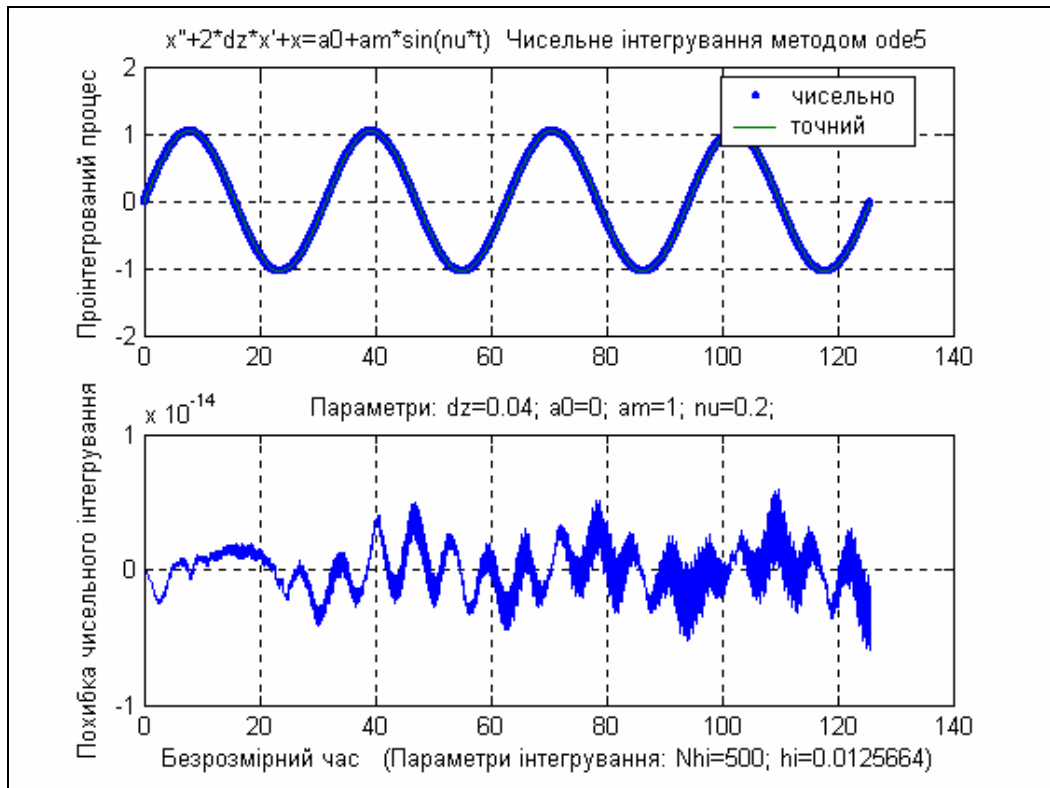


Рис. 8.13. Результати інтегрування при $N_k = 500$

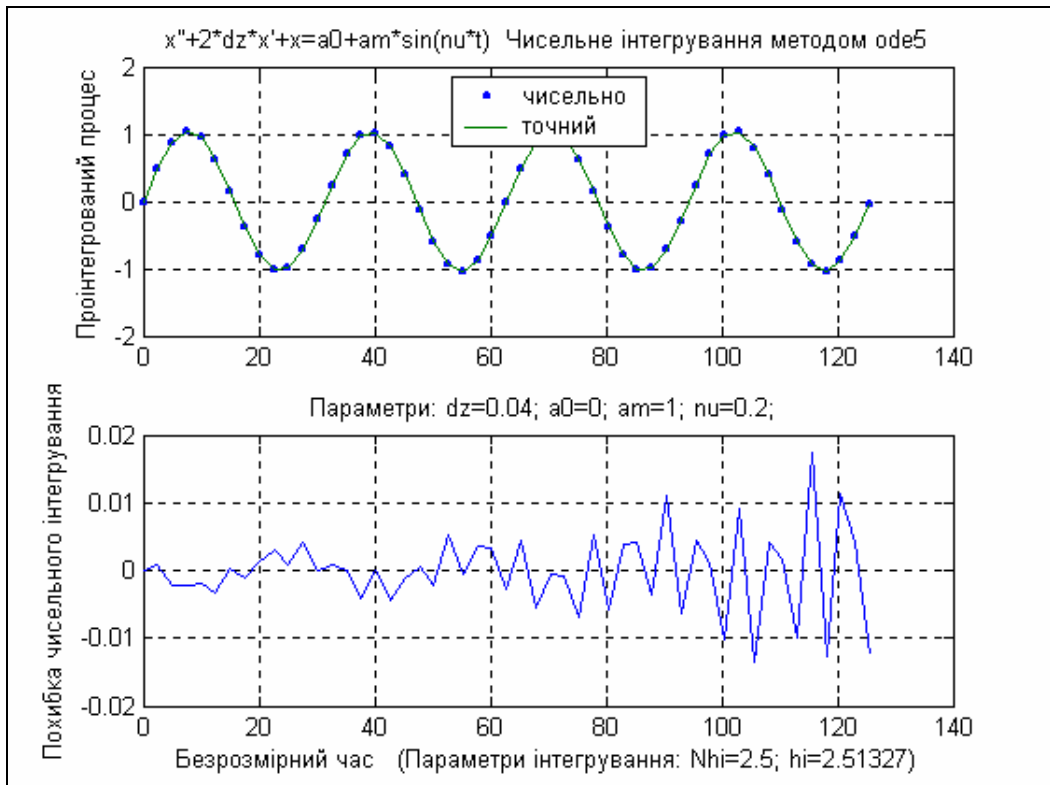
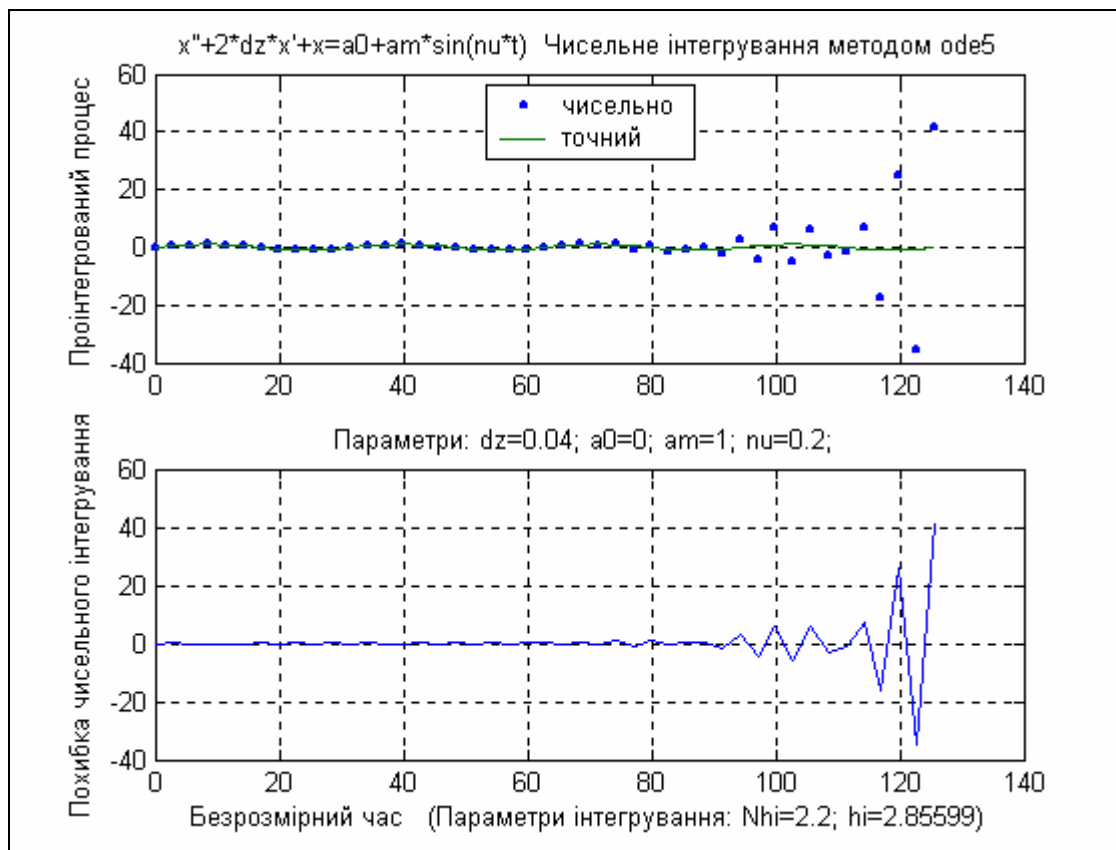


Рис. 8.14. Результати інтегрування при $N_k = 2,5$ Рис. 8.15. Результати інтегрування при $N_k = 2,2$

8.7. Завдання

По заданих викладачем параметрах рівняння (8) і заданому методу чисельного інтегрування на основі створеної програми дослідження похибок чисельного інтегрування:

- провести серію експериментів по визначенню амплітуди сталої складової похибки інтегрування при різних значеннях кроку інтегрування (або, що те ж саме, при різних кількостях кроків на мінімальному періоді); кількість кроків інтегрування на мінімальному періоді збільшувати доти, поки на першій план не вийдуть похибки округлень, а збільшувати до досягнення "вибуху" методу чисельного інтегрування;
- побудувати у графічному вікні у логарифмічному масштабі графік залежності амплітуди похибки інтегрування від кроку; вивести графік до друку;
- на основі одержаного графіка знайти формулу аналітичної залежності похибки від кроку;
- зробити висновки з проведених досліджень.

8.8. Контрольні запитання

1. Що таке "нормальна форма Коші" диференціальних рівнянь?
2. Як привести систему диференціальних рівнянь до форми Коші?
3. Що таке змінні стану? – фазові змінні?
4. Чим визначається кількість змінних стану?
5. У чому полягає задача чисельного інтегрування системи диференціальних рівнянь?
6. Чим вирізняються однокрокові методи різних порядків?
7. Що визначає величина порядку методу чисельного інтегрування?
8. Які основні особливості однокрокових методів чисельного інтегрування ДР?
9. Що таке багатокрокові методи чисельного інтегрування ДР? Чим вони вирізняються від однокрокових?
10. Які основні особливості багатокрокових методів чисельного інтегрування ДР?
11. Перелікуйте головні переваги і недоліки однокрокових і багатокрокових методів у порівнювальному плані?
12. Які головні джерела похибок чисельного інтегрування ДР?
13. Що таке похибка обмежування? Чим вона обумовлена?
14. Як залежить величина похибки методу чисельного інтегрування ДР від величини кроку інтегрування? – від модельного часу?
15. Що таке похибка округлень? Чим вона обумовлена?
16. Як залежить величина похибки округлень при чисельному інтегруванні ДР від величини кроку інтегрування? – від модельного часу?
17. Що таке оптимальний крок чисельного інтегрування? За яких умов він має місце?
18. Що таке "вибух" методу чисельного інтегрування? За яких умов він має місце?
19. Що таке "жорстка" система диференціальних рівнянь? За яких умов систему ДР можна вважати жорсткою?
20. У чому полягають труднощі чисельного інтегрування жорстких систем ДР? Чому у цьому випадку неможливо використовувати однокрокові і багатокрокові методи?
21. Що таке неявні методи чисельного інтегрування ДР? У чому їхня перевага у порівнянні з явними методами?
22. Що таке сталі часу системи диференціальних рівнянь? Як їх можна визначити?
23. Як пов'язана похибка методу чисельного інтегрування з порядком цього методу?



9. Моделювання процесів і систем на ЕОМ

9.1. Особливості моделювання технічних систем

Інженер – це творець технічних об'єктів. Технічний об'єкт – це штучне створіння людини, призначене самостійно виконувати потрібні людині дії з перетворення руху, енергії чи інформації.

Для користувача важливо, щоб технічний об'єкт був стійким (функціонував при заданих збуреннях, змінюванні зовнішніх умов у заданих межах), довговічним і забезпечував виконання покладених на нього функцій протягом усього заданого терміну використання. Це, звичайно, далеко не усі вимоги, що накладаються на утворюваний технічний об'єкт. Але для нас однією з найхарактерніших рис технічного об'єкта є те, що він завжди працює у часі, в ньому завжди протікають різноманітні процеси. Деякі з цих процесів суттєво впливають на характеристики функціонування. Задачею інженера при проектуванні є передбачування впливу різних факторів зовнішнього оточення на функціонування утворюваного технічного об'єкта і передбачити в його конструкції засоби зменшення шкідливого впливу з тим, щоб забезпечити, щоб відхилення найважливіших характеристик від запланованих не виходило за задані межі.

Поводження фізичних і технічних систем описується диференційними або інтегродиференційними рівняннями, одним з аргументів якого є час. У переважній більшості випадків з задовільною для практики точністю поведження можна описати за допомогою диференційних рівнянь з одним аргументом (часом). Такі рівняння у математиці називають звичайними диференційними рівняннями. *Технічні системи, поведження яких описується звичайними диференційними рівняннями, називають динамічними системами.*

У подальшому буде йти мова лише про моделювання саме динамічних систем.

Основним засобом моделювання динамічної системи на ЕОМ є чисельне інтегрування диференційних рівнянь, що описують його поведження у часі. Таке моделювання застосовується по-більшості на етапі ескізного проектування, коли розробляються принципіальні можливості нового утворюваного технічного об'єкта і розробляється первинна документація на вироблення перших (макетних) його зразків. Але таке моделювання стає у нагоді і тоді, коли виникає необхідність розібратися у причинах незапланованих дій технічного об'єкта, виявлених протягом його випробувань чи експлуатації.

Моделюючи поведження технічного об'єкта в умовах, наближених до умов його експлуатації, слід мати на увазі, що технічний об'єкт працює під впливом трьох головних чинників:

- 1) джерела енергії, який спричиняє виникнення в об'єкті потрібних процесів;

- 2) зовнішніх дій з боку оточуючого середовища; серед цих дій можна розрізнити корисні дії, які також спричиняють потрібні (корисні) процеси у технічному об'єкті, і шкідливі дії, які є причиною відхилень функціонування технічного об'єкта від запланованого;
- 3) приймача (навантаження), який являє собою інший реальний (матеріальний) об'єкт, що сприймає рух, енергію чи інформаційний сигнал від розглядуваного технічного об'єкта; приймач теж може впливати на поведінку технічного об'єкта як у позитивному, так і у негативному сенсі.

З огляду на це, у моделі, яка здійснює моделювання технічного об'єкта за умов, наближених до умов його експлуатації, мають бути передбачені наступні складові:

- 1) моделі, що імітують дії джерел енергії на технічний об'єкт;
- 2) імітатори зовнішніх корисних дій;
- 3) імітатори зовнішніх шкідливих дій;
- 4) модель функціонування власно технічного об'єкта;
- 5) імітатор приймача, який дозволяв би, по-перше, сприймати вихідні фізичні величини технічного об'єкта, а по-друге, імітувати дію реального приймача на поведінку технічного об'єкта.

Тому і при створенні моделі на ЕОМ теж потрібно мати засоби, що дозволяють утворювати чисельні імітатори дій різноманітної форми, які можна розглядати як сигнали, що є функціями поточного моменту часу. Такі імітатори називатимемо генераторами сигналів.

9.2. Засоби моделювання системи Matlab. Ознайомлення з пакетом Simulink

Найбільш пристосованою для створення чисельних моделей технічних об'єктів є наразі комп'ютерна система MatLab завдяки присутності в ній таких потужних пакетів моделювання як Simulink і Signal Toolbox.

Перший пакет дозволяє значно спростити складання обчислювальних програм чисельного моделювання завдяки застосуванню так званого візуального програмування, коли програма утворюється шляхом складання спеціальних блок-схем з готових візуальних блоків, що є у бібліотеці блоків Simulink. Кожний з візуальних блоків Simulink є фактично математичною (чисельною) програмою з'єднаною з її візуальним поданням у вигляді блока. З'єднання між собою блоків з'єднувальними лініями у блок-схемі забезпечує передавання інформації, виробленою одним блоком, до іншого блока для подальшого її перетворення. Цим забезпечується можливість моделювання доволі складних систем, у тому числі нелінійних і нестационарних, під впливом дій досить складної форми.

Пакет Signal Toolbox призначений для застосування у звичайному програмному режимі, коли програму записують у вигляді тексту на М-мові. У ньому зосереджені засоби:

- чисельного генерування сигналів різних найбільш типових форм;
- засоби чисельної обробки сигналів;
- засоби утворення фільтрів різного рода для перетворення сигналів.

Ознаймимося більш докладно з можливостями, що надаються бібліотекою Simulink.

Бібліотека складається з таких розділів (рис. 9.1):

- **Sources** (Джерела), який містить блоки, що генерують сигнали певних типових форм як функції модельного часу; ці блоки не мають входів (їм не потрібна зовнішня інформація), а мають тільки вихід;
- **Sinks** (Приймачи), у якому зосереджені блоки, які тільки сприймають інформацію; у цих блоках немає виходу;
- **Continuous** (Неперервні елементи), в якому містяться блоки, які забезпечують перетворення інформаційного сигналу, що надходить до них, за лінійними диференціальними рівняннями з постійними коефіцієнтами;
- **Discrete** (Дискретні елементи); тут містяться блоки, імітуючі дискретну обробку інформації (сигнала), що надходить;

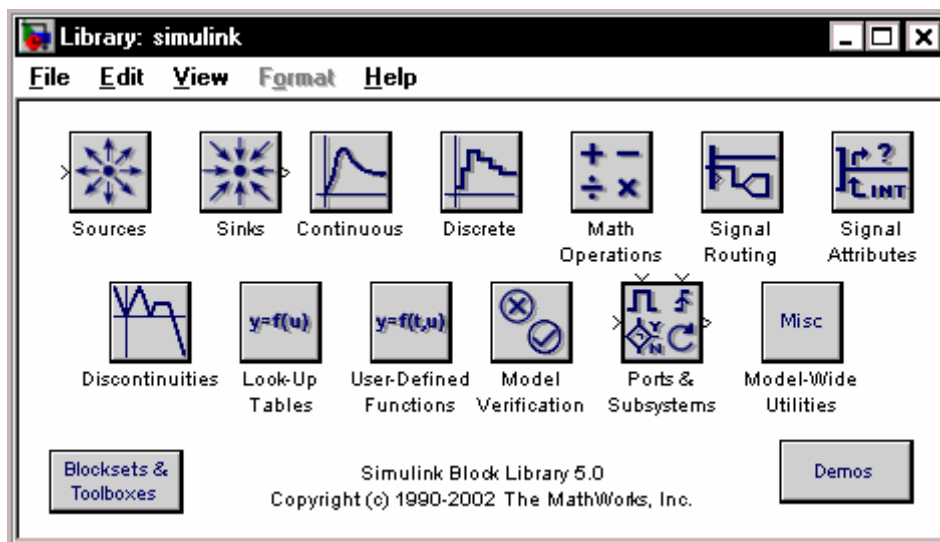


Рис. 9.1. Розділи бібліотеки Simulink

- **Math Operations** (Математичні операції), який містить блоки, що здійснюють найбільш поширені математичні перетворення величини, яка надходить до входу;
- **Signal Routing** (Впорядковування сигналів), що містить блоки, які здійснюють різноманітне переключення сигналів, об'єднання кількох сигналів у єдиний векторний сигнал (шину), зворотне роз'єднування векторного сигналу на окремі складові тощо;
- **Signal Attributes** (Атрибути сигналів), який містить блоки, що дозволяють встановити, виміряти або змінити деякі з атрибутів сигналів;

- Discontinuities (Розривні елементи), де зосереджені блоки, які перетворюють сигнали, що надходять до них, за деякими типовими кусково-лінійними залежностями;
- Look-Up Tables (Інтерполяція табличних залежностей); тут містяться блоки, що здійснюють поточне інтерполювання функцій, заданих таблично;
- User-Defined Functions (Функції, визначувані користувачем); в ньому містяться блоки, які дозволяють користувачеві утворювати власні потрібні йому блоки;

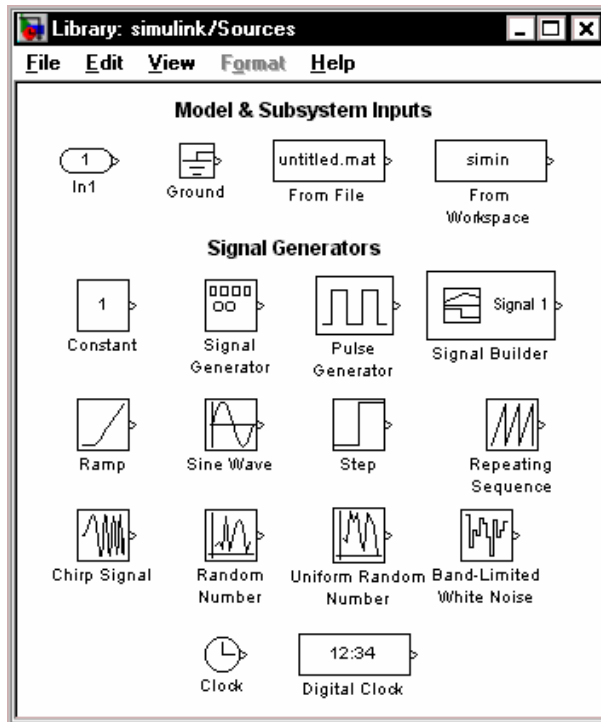


Рис. 9.2. Вміст розділу Sources

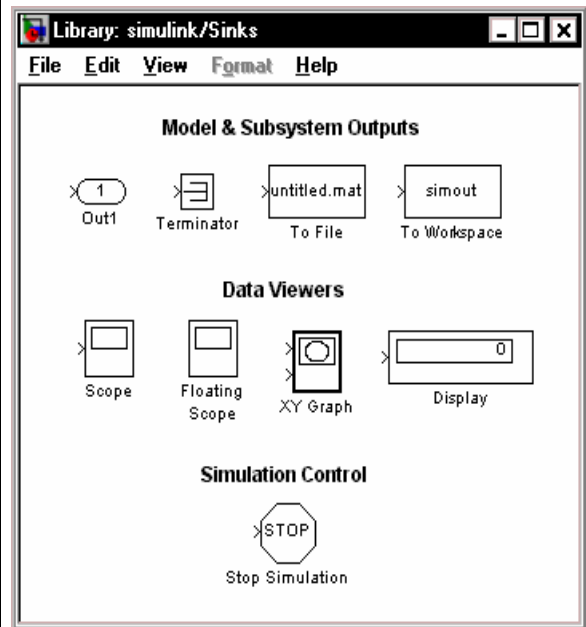


Рис. 9.3. Вміст розділу Sinks

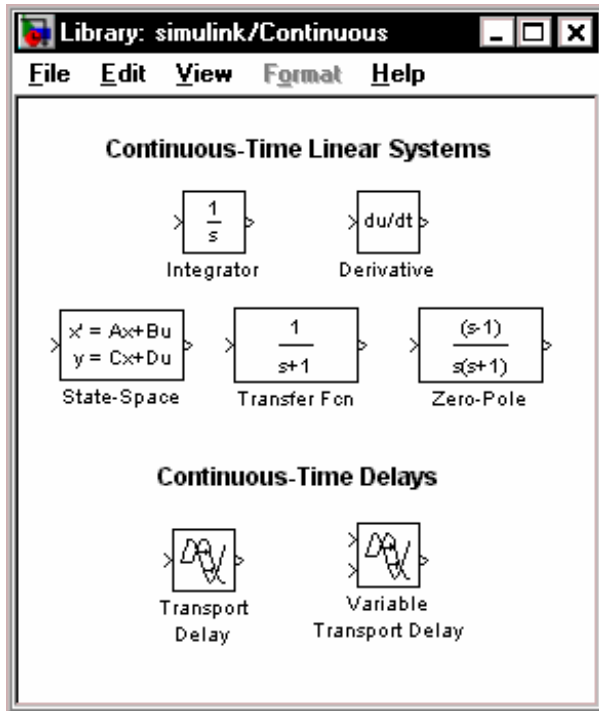


Рис. 9.4. Вміст розділу Continuous

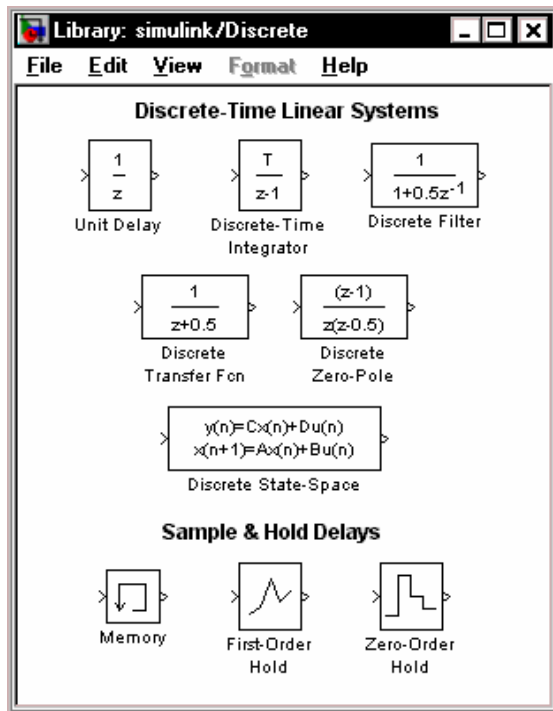


Рис. 9.5. Вміст розділу Discrete

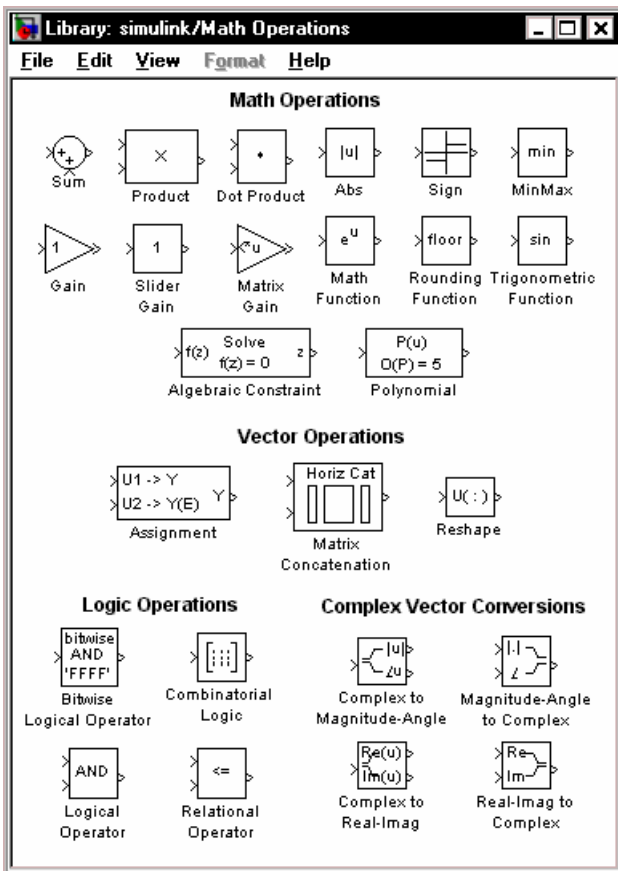


Рис. 9.6. Розділ Math Operations

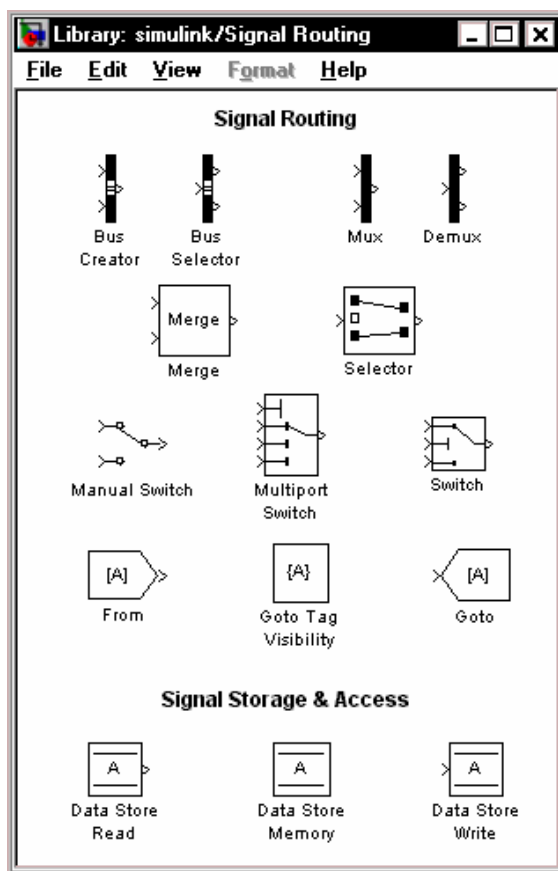


Рис. 9.7. Розділ Signal Routing

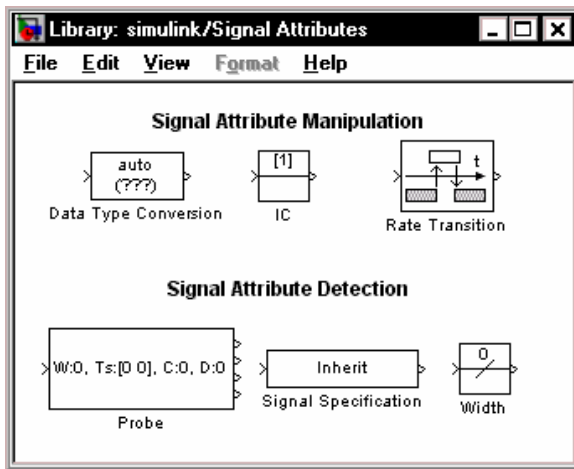


Рис. 9.8. Розділ Signal Attributes

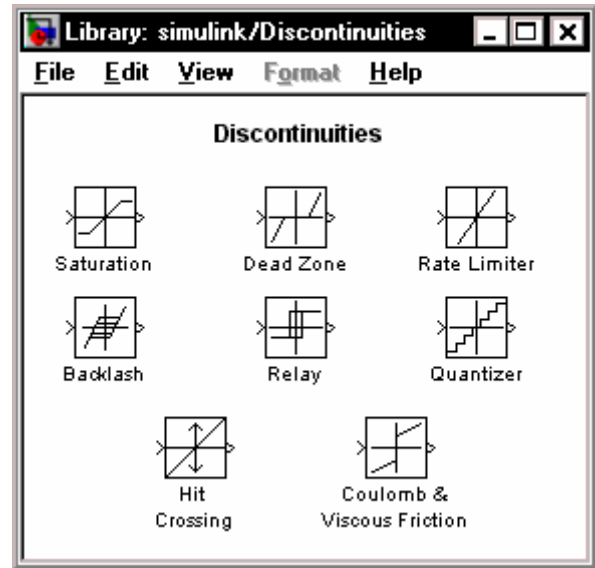


Рис. 9.9. Розділ Discontinuities

- Model Verification (Верифікація моделі), який містить блоки, що дозволяють проводити верифікацію моделі;
- Ports & Subsystems (Порти і підсистеми) складається з блоків, які дозволяють з'єднувати між собою моделі різного ступеня ієрархії;
- Model-Wide Utilities (Утіліти) містить блоки, що здійснюють деякі допоміжні перетворення.

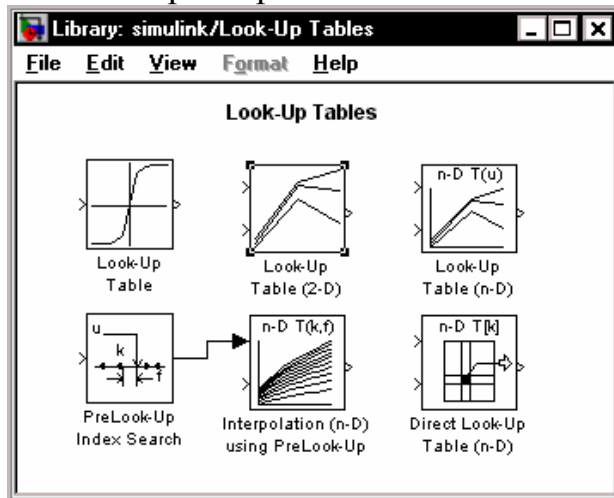


Рис. 9.10. Розділ Look-Up Tables

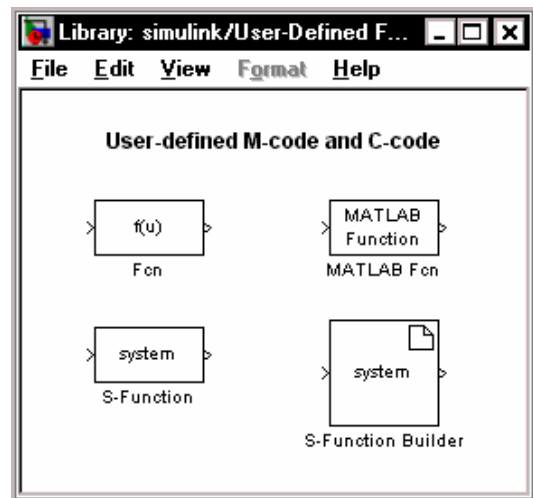


Рис. 9.11. Розділ User-Defined Functions

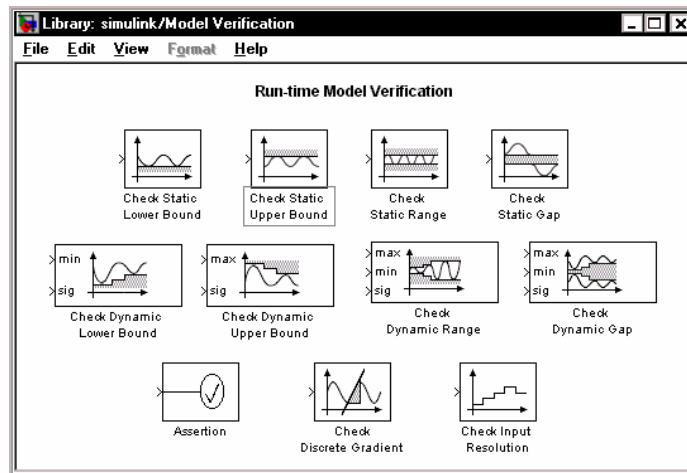


Рис. 9.12. Розділ Model Verification

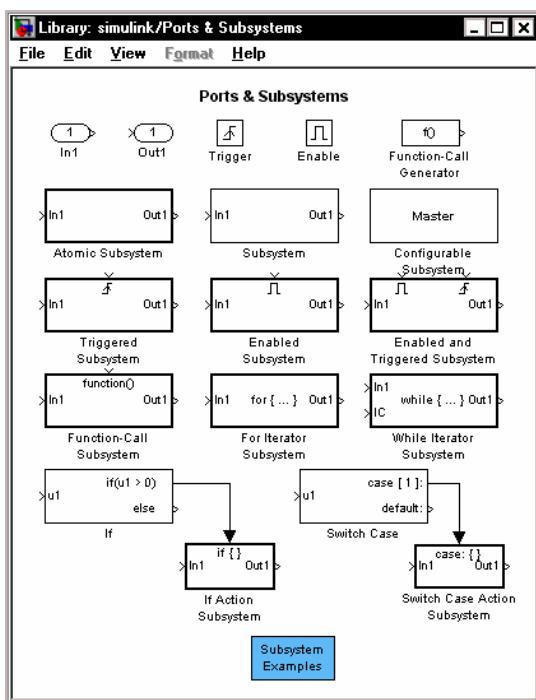


Рис. 9.13. Вміст розділу Ports & Subsystems

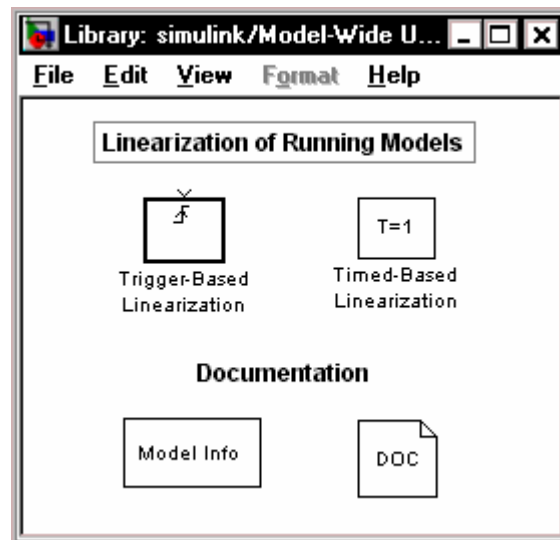


Рис. 9.14. Вміст розділу Model-Wide Utilities

Якщо подвійно клацнути мишкою на зображенні якогось з розділів (рис. 9.1.), на екрані виникне вікно з зображенням блоків, що входять у даний розділ.

На рис. 9.2... 9.14 наведені одержуваний при цьому зміст вказаних розділів.

9.3. Генерування типових процесів

Розглянемо детальніше можливості, що надають користувачеві пакети SIGNAL і SIMULINK у частині чисельного імітування (генерування) процесів заданої форми.

9.3.1. Формування імпульсних, неколивальних і нестационарних коливальних процесів

У пакеті SIGNAL передбачено декілька процедур, що утворюють послідовності даних, які подають деякі одиночні імпульсні процеси типових форм.

Процедура *rectpuls* забезпечує формування одиночного імпульсу прямокутної форми. Звернення до неї виду

$$y = \text{rectpuls}(t, w)$$

дозволяє утворити вектор *y* значень сигналу такого імпульсу одиничної амплітуди, шириною *w*, центрованого щодо $t=0$ за заданим вектором *t* моментів часу. Якщо ширина імпульсу *w* не зазначена, її значення за замовчуванням приймається рівним одиниці.

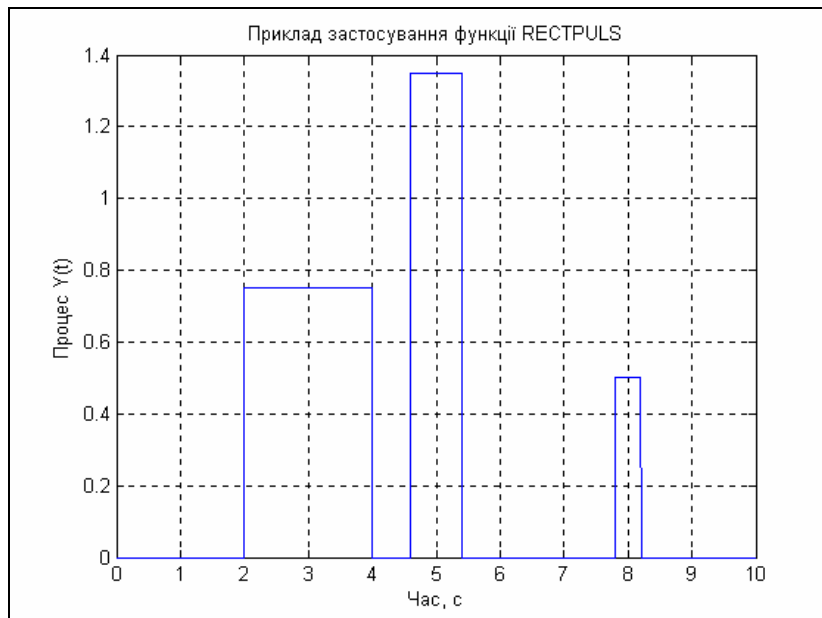


Рис. 9.15. Процес із трьох прямокутних імпульсів

На рис. 9.15. наведений результат утворення процесу, що складається із трьох послідовних прямокутних імпульсів різної висоти й ширини, згідно такої послідовності команд:

```
t = 0:0.01:10;
y = 0.75*rectpuls(t-3, 2)+0.5*rectpuls(t-8, 0.4)+1.35*rectpuls(t-5, 0.8);
plot(t,y), grid, title(' Приклад застосування RECTPULS')
xlabel('Час ( с)'),
ylabel('Процес Y(t)')
```

Формування імпульсу трикутної форми одиничної амплітуди можна здійснити за допомогою процедури *tripuls*, звернення до якої має вид

$$y = \text{tripuls}(t, w, s).$$

Аргументи *y*, *t* і *w* мають той самий зміст. Аргумент *s* ($-1 < s < 1$) визначає нахил трикутника. Якщо $s = 0$, або не зазначений, трикутний імпульс має симетричну форму. Наведемо приклад:

```
t = 0:0.01:10;
y = 0.75*tripuls(t-1, 0.5)+0.5*tripuls(t-5, 0.5, -1)+1.35*tripuls(t-3, 0.8,1);
plot(t,y), grid, title(' Приклад застосування процедури TRIPULS')
xlabel('Час ( с) ) , ylabel('Вихідний процес Y(t)')
Результат показаний на рис. 9.16.
```

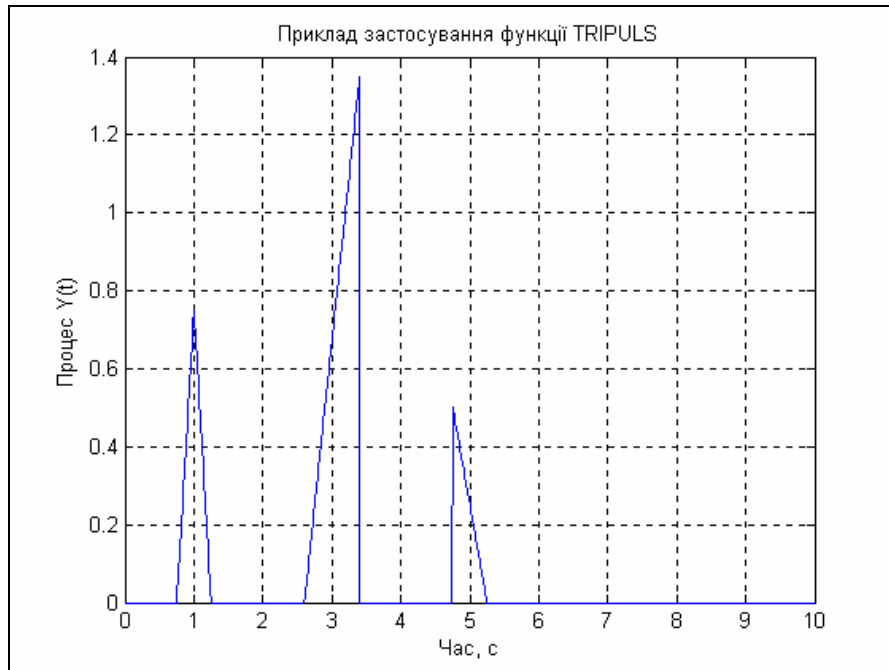


Рис. 9.16. Процес із трьох трикутних імпульсів

Для формування імпульсу, що є синусоїдою, модульованою функцією Гауса, використовується процедура *gauspuls*. Якщо звернутися до неї за формою

$$y = \text{gauspuls}(t, fc, bw),$$

вона утворює вектор значень зазначеного сигналу з одиничною амплітудою, із синусоїдою, що змінюється з частотою *fc* Гц, і із шириною *bw* смуги частот сигналу. У випадку, коли останні два аргументи не зазначені, вони за замовчуванням одержують значення 1000 Гц і 0.5 відповідно.

Наведемо приклад створення одиночного гауссового імпульсу,,

```
t = 0:0.01:10;
y = 0.75*gauspuls(t-3, 1,0.5);
plot(t,y), grid, title(' Приклад застосування GAUSPULS')
xlabel('Час ( с) ) , ylabel('Процес Y(t)')
```

На рис. 9.17. наведено графік результуючого процесу.

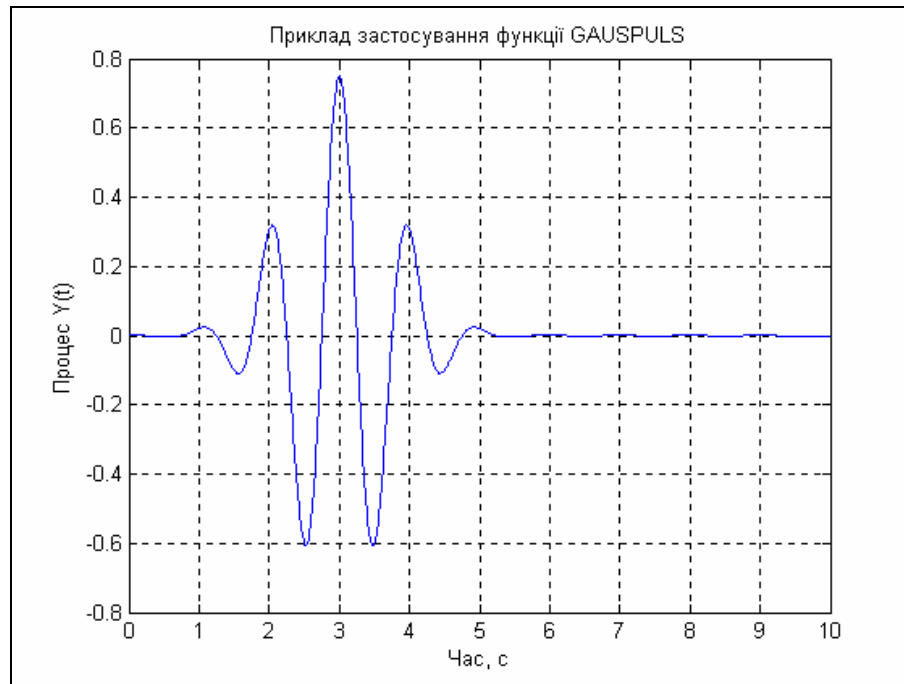


Рис. 9.17. Процес з одного гауссового імпульсу

Нарешті, розглянемо процедуру *sinc*, яка формує вектор значень функції $\text{sinc}(t)$, яка визначається формулами:

$$\text{sinc}(t) = \begin{cases} 1 & t = 0, \\ \frac{\sin(\pi t)}{\pi t} & t \neq 0. \end{cases}$$

Ця функція є зворотним перетворенням Фур'є прямокутного імпульсу шириною 2π і висотою 1:

$$\text{sinc}(t) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{j\omega t} d\omega.$$

На рис. 9.18. поданий графік процесу, сформованого за допомогою цієї процедури:

```
t=0:0.01:50;
y1=0.7*sinc(pi*(t-25)/5);
plot(t,y1), grid, title('Функція Y(t) = 0.7* SINC(pi*(t-25)/5)')
xlabel('Час ( с)'), ylabel('Процес Y(t)')
```

Перейдемо тепер до стандартних блоків – генераторів пакета SIMULINK. Вони подані на рис. 9.2. Тут розглянемо лише три з цих блоків – *Constant*, *Step* і *Ramp*.

Блок *Constant* формує процес, який є незмінним у часі і призначений для встановлення констант (скалярних, векторних або матричних), що використовуються при моделюванні. Він має один параметр налагоджування – **Constant value** (значення константи), який може бути введений і як вектор-рядок із кількох елементів згідно загальних правил MatLab. За замовчуванням він дорівнює 1.

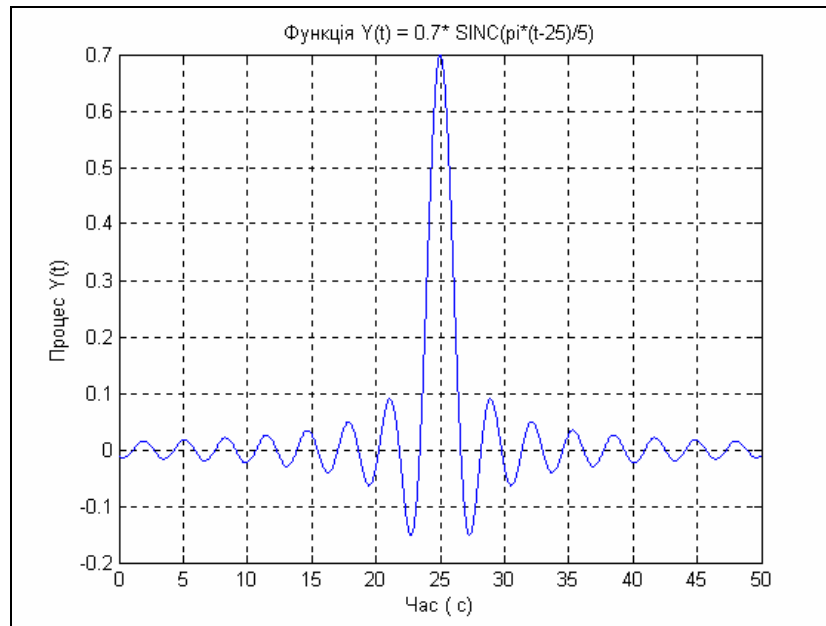


Рис. 9.18. Процес у вигляді функції SINC

Блок *Step* генерує процес у вигляді поодинокі сходинок (стрибкоподібний сигнал) із заданими параметрами (початку сходинок та її висоти). Він містить три параметри налагоджування:

- **Step time** – момент часу, коли здійснюється стрибкоподібне змінювання сигналу; за замовчуванням воно дорівнює 1;
- **Initial value** – початкове (до стрибкоподібного змінювання) значення сигналу; за замовчуванням воно дорівнює 0;
- **Final value** – кінцеве значення сигналу (після стрибкоподібного змінювання); за замовчуванням воно дорівнює 1.

Блок *Ramp* утворює лінійно зростаючий (або спадний) сигнал, тобто генерує процес, який має стрибкоподібне змінювання його швидкості у визначений момент часу. Він має наступні параметри настроювання:

- **Slope** – значення набутої швидкості (початкова швидкість сигналу приймається рівною 0); за замовчуванням швидкість дорівнює 1;
- **Start time** – момент часу, коли сигнал починає лінійно зростати у часі (за замовчуванням 0);
- **Initial output** – значення сигналу у моменти часу, що передують його лінійному зростанню (спаданню); за замовчуванням дорівнює 0.

Складемо блок-схему, подану на рис. 9.19, за якою формуються чотири сигнали:

- сталий сигнал значенням 0,7;
- східчастий сигнал із сходиною висотою 1,5 при $t=2$;
- лінійно зростаючий сигнал із швидкістю зростання 1, початковим значенням сигналу -1.2 і моментом початку зростання сигналу 6;
- суму усіх попередніх сигналів.

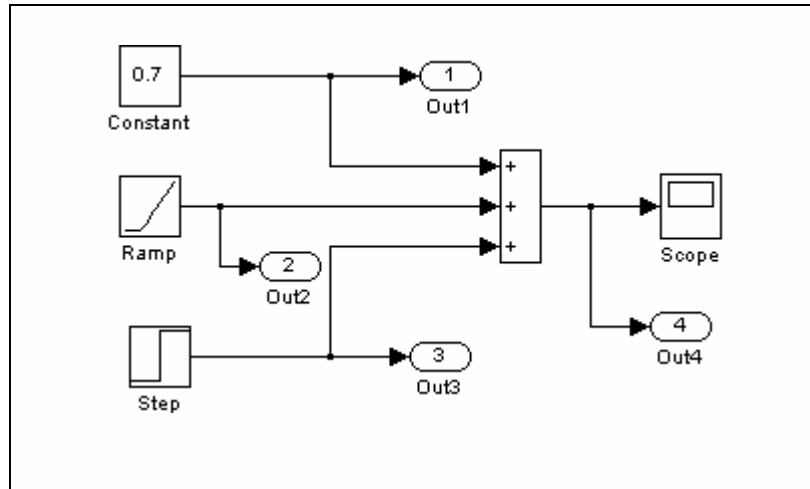


Рис. 9.19. Блок-схема SIMULINK генерування неколивальних процесів

Виведемо графіки усіх цих процесів у єдине графічне поле за допомогою команд:

```
plot(tout,yout(:,1),'k',tout,yout(:,2),'k',tout,yout(:,3),'ko',tout,yout(:,4),'kd'), grid
xlabel('Час t (сек)'), ylabel('Процеси')
legend('CONST','RAMP','STEP',' Сума',2)
title('Процеси, згенеровані блоками CONSTANT, STEP і RAMP')
```

Результат цих дій подано на рис. 9.20.

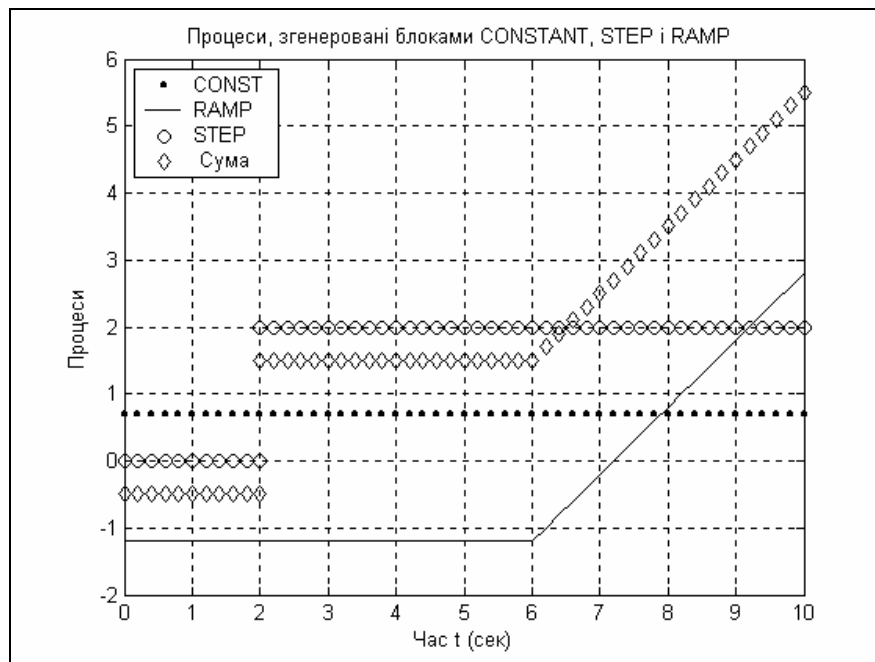


Рис. 9.20. Графіки процесів, згенерованих блоками CONST, RAMP і STEP

На завершення розглянемо процедуру *chirp*, яка формує косинусоїду, частота змінювання якої лінійно змінюється з часом. Загальна форма звернення до цієї процедури є такою:

$$y = chirp(t,F0,t1,F1),$$

де $F0$ - значення частоти в герцах при $t=0$, $t1$ - деяке задане значення моменту часу; $F1$ - значення частоти (у герцах) змінювання косинусоїди в момент часу $t1$. Якщо три останніх аргументи не зазначені, то за замовчуванням їм надасться таких значень: $F0=0$, $t1=1$, $F1=100$. Приклад:

```
t = 0:0.001:1; y = 0.75*chirp(t);
plot(t,y), grid,title(' Приклад процедури CHIRP')
xlabel('Час ( с)'), ylabel('Вихідний процес Y(t)')
```

Результат показано на рис. 9.21.

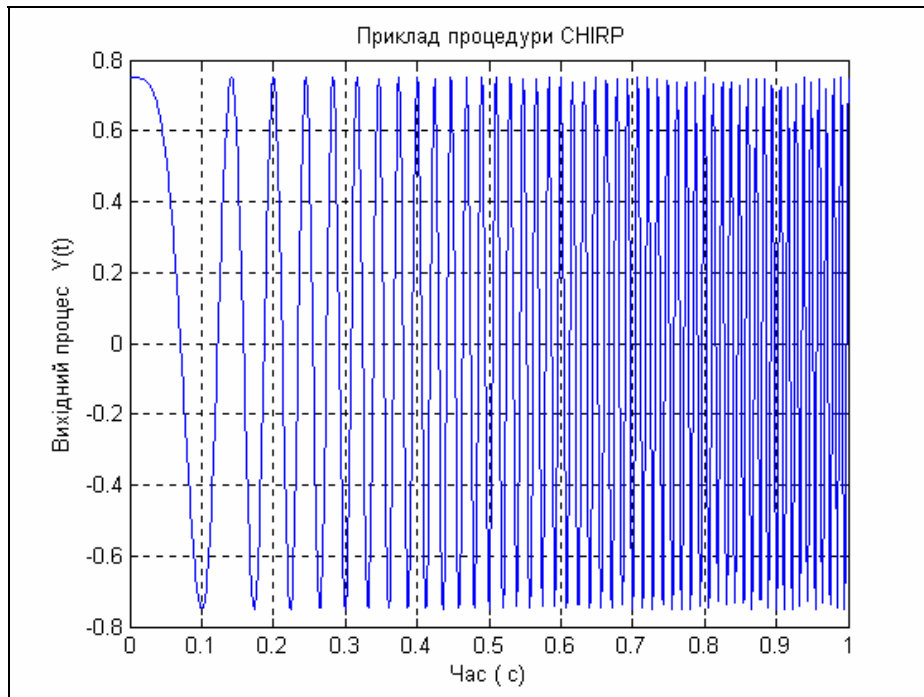


Рис. 9.21. Процес CHIRP

Аналогічний сигнал може бути згенерований і спеціальним блоком **Chirp Signal** (див. рис. 9.2) у середовищі SIMULINK. Цей блок генерує синусоїдальний сигнал одиничної амплітуди змінної частоти, причому значення частоти коливань змінюється з часом за лінійним законом. Відповідно до цього в ньому передбачені такі параметри настроювання (рис. 9.22):

- *Initial frequency (Hz)* - початкове значення (при $t=0$) частоти в герцах;
- *Target time (secs)* - інший (додатний) момент часу (у секундах);
- *Frequency at target time (Hz)* - значення частоти в цей другий момент часу.

Рис. 9.23 демонструє результат використання блока при параметрах, зазначених на рис. 9.22.

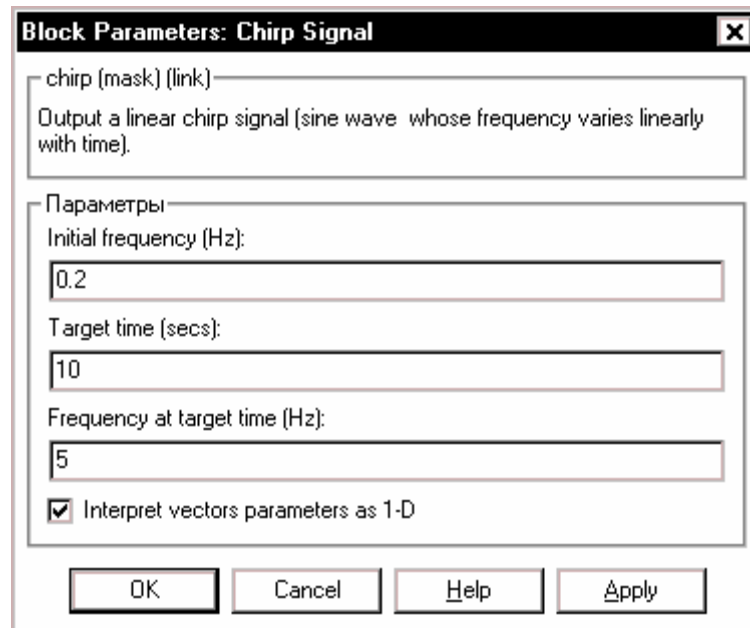


Рис. 9.22. Вікно настроювання блоку Chirp Signal

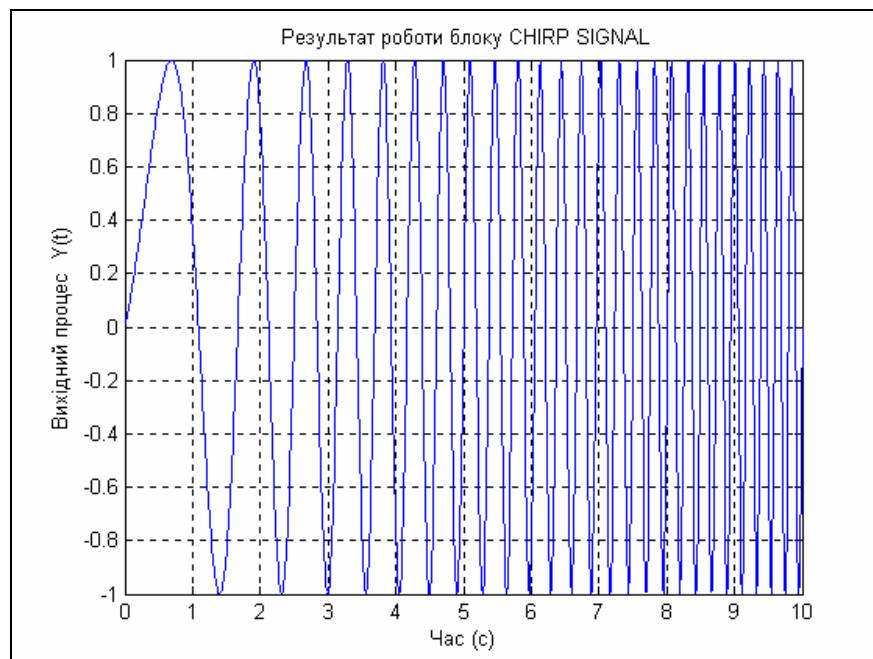


Рис. 9.23. Генерування сигналу блоком CHIRP

9.3.2. Генерування періодичних і стаціонарних коливальних процесів

Формування коливальних процесів, що складаються з певної обмеженої кількості гармонійних складових (тобто так названих *полігармонічних коливальних процесів*), можна здійснити за допомогою звичайних процедур $\sin(x)$ і $\cos(x)$. Розглянемо приклад (рис. 9.24):

```
t=0:0.01:50;    y1=0.7*sin(pi*t/5);
plot(t,y1), grid,set(gca,'FontName','Arial Cyr','FontSize',16)
title('Гармонічні коливання Y(t) = 0.7* SIN(pi*t/5)'),    xlabel('Час ( с)')
ylabel(' Вихідний процес')
```

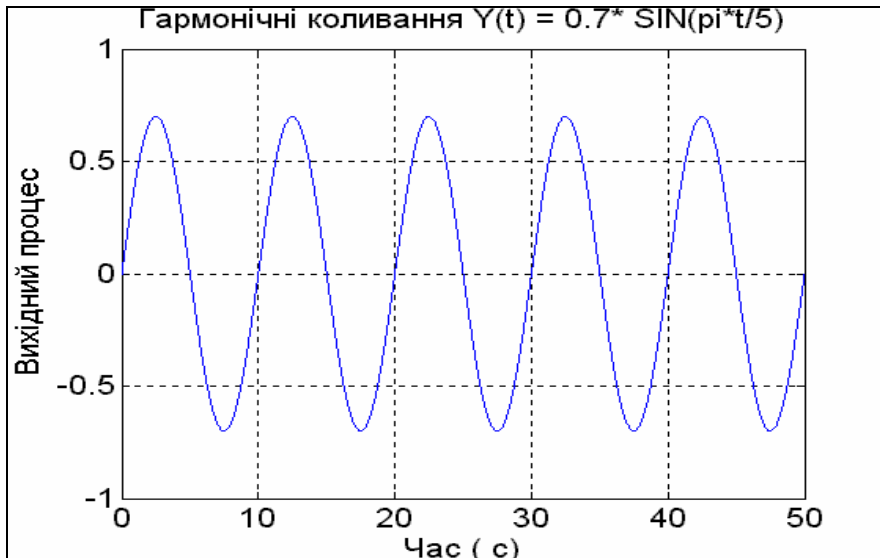



Рис. 9.24. Гармонічний процес

Процес, що є послідовністю прямокутних імпульсів із періодом 2π для заданої у векторі t послідовності відліків часу, "генерується" за допомогою процедури *square*. Звернення до неї відбувається за формою:

$$y = \text{square}(t, \text{duty}),$$

де аргумент *duty* визначає тривалість додатної напівхвилі у відсотках від періоду хвилі. Наприклад, сукупність команд:

```
t=0:0.01:50;    y=0.7*square(pi*t/5, 40);
plot(t,y),      grid, set(gca,'FontName','Arial Cyr','FontSize',16)
title('Прямокутні хвилі Y(t) = 0.7* SQUARE(pi/5*t, 40)')
xlabel('Час (с)'), ylabel('Процес Y(t)')
```

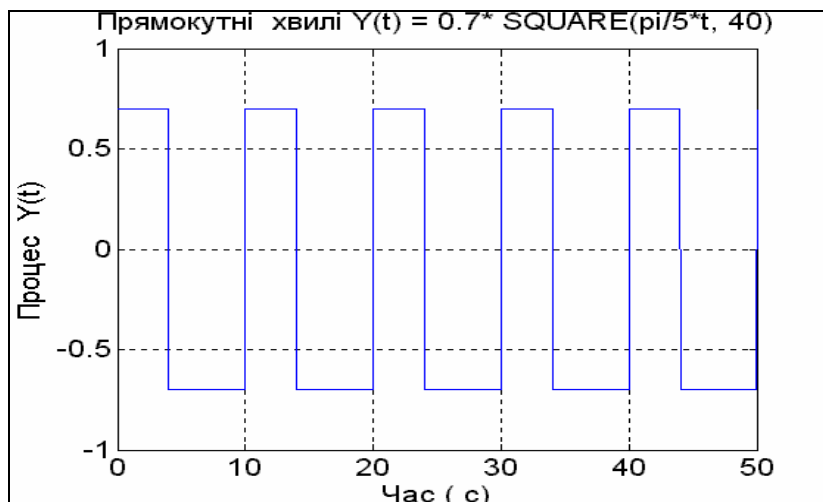


Рис. 9.25. Процес із центрованих прямокутних імпульсів

генерує періодичний (з періодом $10 = \frac{2\pi}{(\pi/5)}$ пізоцес, що складається з

прямокутних хвиль (рис. 9.22) з максимальним відхиленням від нуля у той чи протилежний бік (амплітудою) 0,7, додатна напівхвиля яких має тривалість у

40% від тривалості періоду (4 с), а від'ємна – триває решту 60% періоду (6с). Результат наведений на рис. 9.25.

Аналогічно, генерування пілкоподібних і трикутних коливань можна здійснювати процедурою *sawtooth*. Якщо звернутися до неї так:

$$y = \text{sawtooth}(t, \text{width}),$$

то у векторі y формуються (у моменти часу, що задаються вектором t), значення сигналу, що є пілкоподібними хвилями з періодом 2π . При цьому параметр *width* визначає частину періоду, у якій сигнал збільшується.

Нижче наведений приклад застосування цієї процедури:

```
t=0:0.01:50;
y=0.7*sawtooth(pi*t/5, 0.7);
plot(t,y), grid,set(gca,'FontName','Arial Cyr','FontSize',16)
title('Трикутні хвилі Y(t) = 0.7* SAWTOOTH(pi/5*t, 0.7)')
xlabel('Час ( с)')
ylabel('Вихідний процес Y(t)')
```

У результаті одержуємо процес, зображений на рис. 9.26.

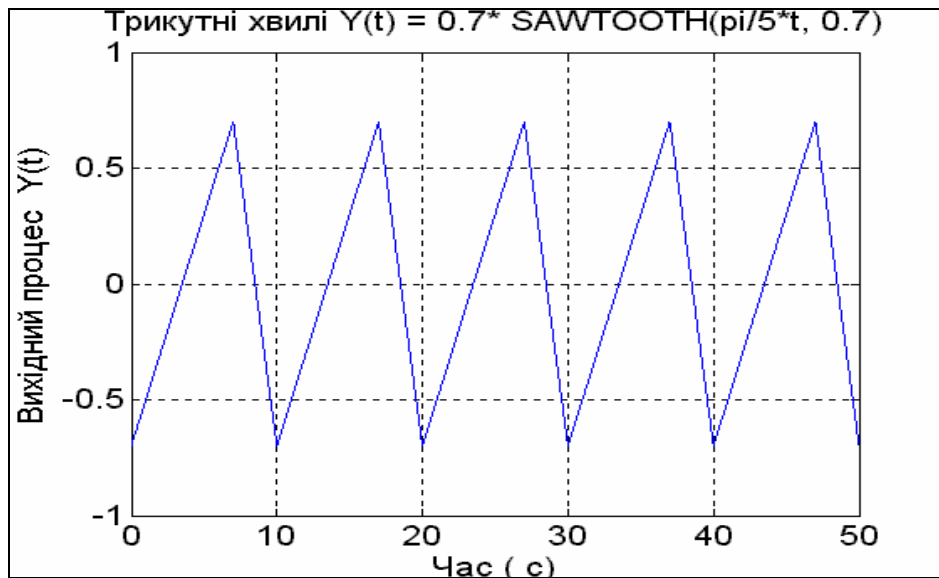


Рис. 9.26. Процес із центрованих трикутних хвиль

До цього ми генерували коливання, які мали однакове максимальне відхилення від нуля в обидва боки. Тепер розглянемо процедуру *pulstran*, яка дозволяє формувати коливання, що є послідовністю або прямокутних, або трикутних, або гауссових імпульсів, сформованих за розглянутими у п. 9.2.1 функціями. Звернення до неї має вид:

$$y = \text{pulstran}(t, d, \text{'func'}, p1, p2, \dots),$$

де d визначає вектор значень тих моментів часу, де мають буми розташовані центри відповідних імпульсів; параметр *func* визначає форму імпульсів і може мати одне з таких значень: *rectpuls* (для прямокутного імпульсу), *tripuls* (для трикутного імпульсу) і *gauspuls* (для гауссового імпульсу); параметри $p1, p2, \dots$ визначають необхідні параметри імпульсу у відповідності з формою звернення до процедури, яка визначає цей імпульс.

Нижче приведені три приклади застосування процедури *pulstran* для різних форм імпульсів.

По-первах, згенеруємо процес, що складається з послідовності трикутних імпульсів, які мають висоту 0,7, ширину імпульсу 5 с, центри цих імпульсів відстоять один від одного на відстань 10 с (див. значення вектора **d**) і з відхиленням вершини трикутника ліворуч від центру імпульсу на (-0,5) напівширини імпульсу:

```
t=0:0.01:50
d=[0:50/5:50]';
y=0.7*pulstran(t, d,'tripuls',5,-0.5);
plot(t,y), grid,set(gca,'FontName','Arial Cyr','FontSize',16)
title('Y(t) = 0.7*PULSTRAN(t,d,"tripuls",5,-0.5)')
xlabel('Час ( с)')
ylabel('Вихідний процес Y(t)')
```

Результат поданий на рис. 9.27.

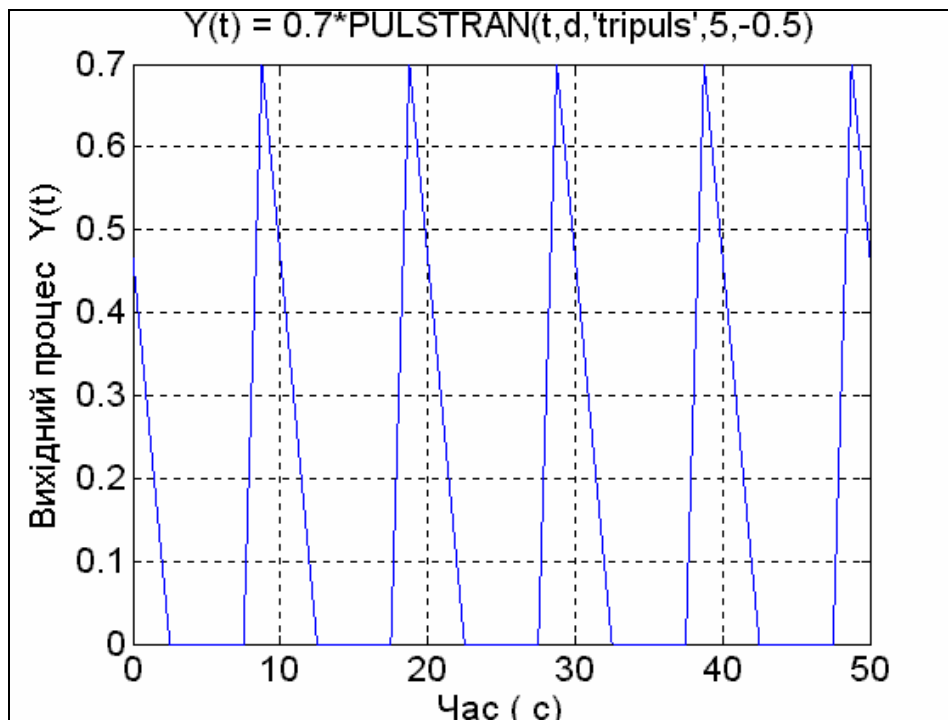


Рис. 9.27. Послідовність трикутних імпульсів

Утворимо процес із послідовності прямокутних імпульсів:

```
t=0:0.01:50
d=[0:50/5:50]';
y=0.75*pulstran(t, d,'rectpuls',3);
plot(t,y), grid,set(gca,'FontName','Arial Cyr','FontSize',16)
title('Y(t) = 0.75*PULSTRAN(t,d,"rectpuls",3)')
xlabel('Час ( с)')
ylabel('Вихідний процес Y(t)')
```

Результат наведено на рис. 9.28.

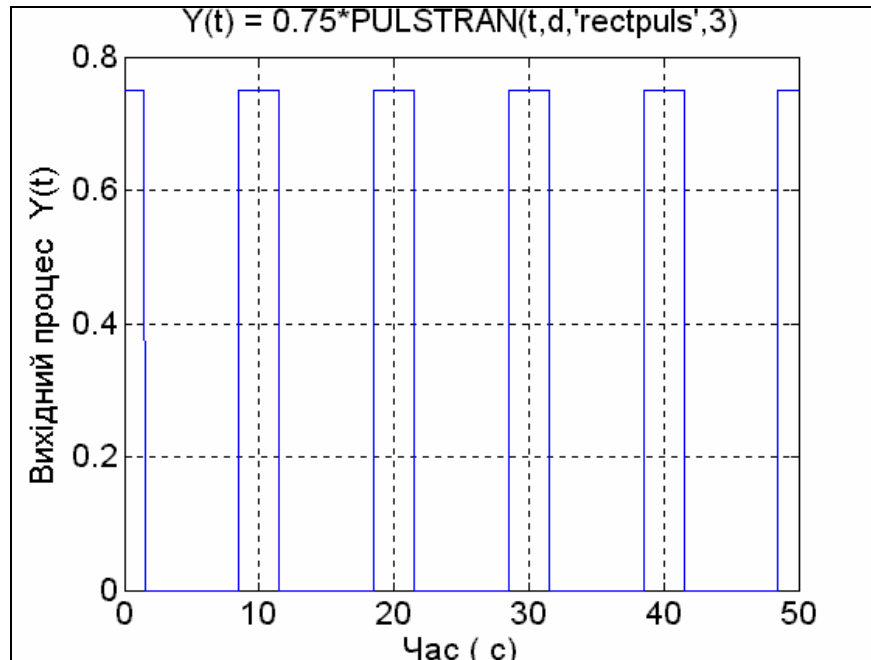


Рис. 9.28. Послідовність прямокутних імпульсів

Тепер утворимо послідовність гауссових імпульсів

```
t=0:0.01:50
d=[0:50/5:50];
y=0.7*pulstran(t, d,'gauspuls',1,0.5);
plot(t,y), grid,set(gca,'FontName','Arial Cyr','FontSize',16)
title('Y(t) = 0.7*PULSTRAN(t,d,"gauspuls",1,0.5)')
xlabel('Час ( с)')
ylabel('Вихідний процес Y(t)')
```

Результат приведений на рис. 9.29.

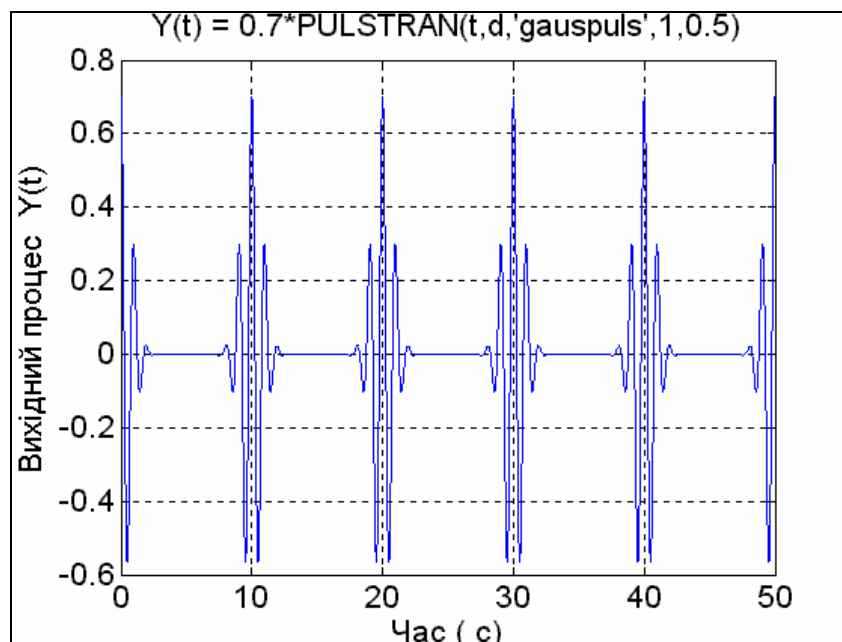


Рис. 9.29. Послідовність гауссових імпульсів

Процедура *diric* формує масив значень так називаної *функції Діріхле*, яка визначається співвідношеннями:

$$diric(t) = \begin{cases} -1^{k(n-1)} & t = 2\pi k, \quad k = 0, \pm 1, \pm 2, \dots \\ \frac{\sin(nt/2)}{n \cdot \sin(t/2)} & \text{при других } t \end{cases}$$

Функція Діріхле є періодичною. За непарних n період дорівнює 2π , за парних - 4π . Максимальне значення її дорівнює 1, мінімальне -1. Параметр n повинний бути цілим додатним числом. Звернення до функції має вид:

$$y = diric(t, n).$$

Нижче приведені оператори, що ілюструють використання процедури *diric* і виводять графіки функції Діріхле для $n=3, 4$ і 5 :

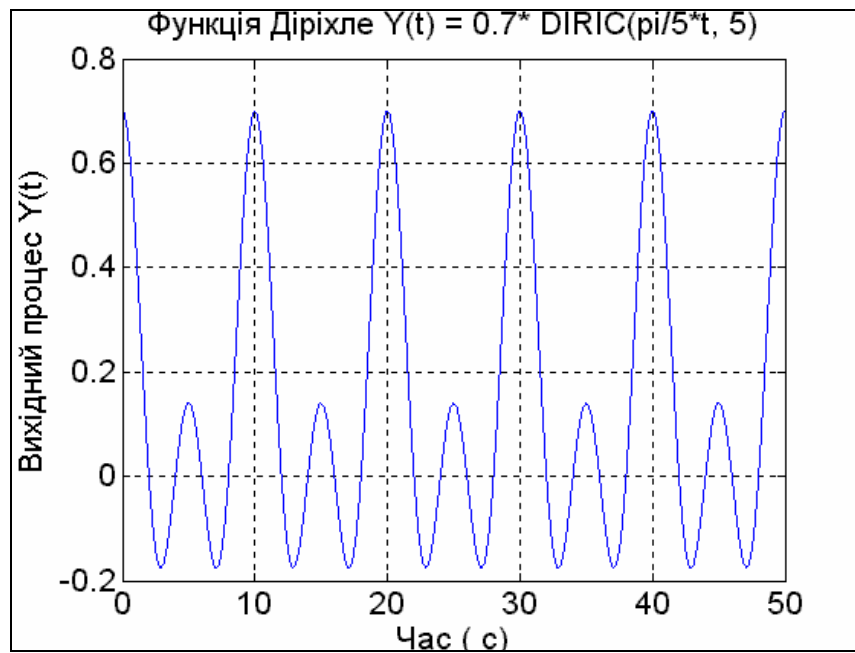


Рис. 9.30. Процес функції Діріхле при $n=5$

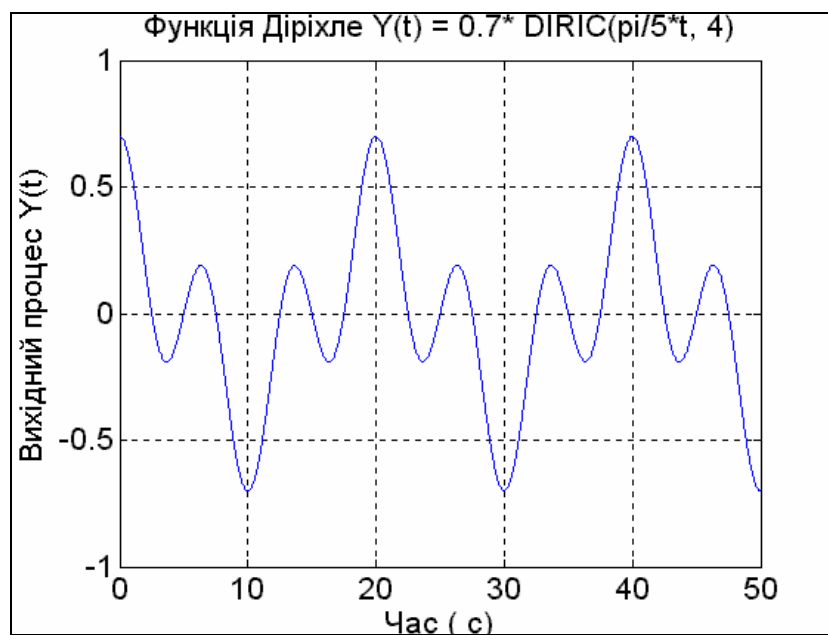


Рис. 9.31. Процес функції Діріхле при $n=4$

```
t=0:0.01:50; y1=0.7*diric(pi*t/5, 3);
plot(t,y1), grid,set(gca,'FontName','Arial Cyr','FontSize',16)
title('Функція Діріхле Y(t) = 0.7* DIRIC(pi/5*t, 3)')
xlabel('Час ( с)'), ylabel('Вихідний процес Y(t)')
```

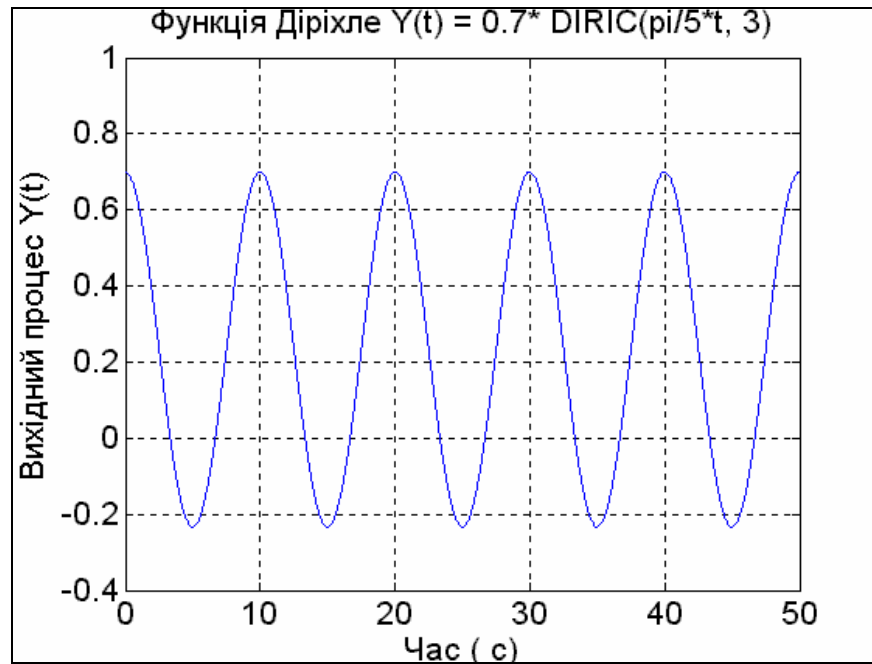


Рис. 9.32. Процес функції Діріхле при $n = 3$

Результати подані на рис. 9.30...9.32.

Бібліотека *Sources* пакета *SIMULINK* надає велику кількість блоків для генерування періодичних і стаціонарних коливань (див. рис. 9.2). У цьому поділі бібліотеки як джерела сигналів, окрім розглянутих у п. 9.2.1, передбачені такі блоки:

- **Signal Generator** - створює (генерує) неперервний коливальний сигнал однієї із хвильових форм на вибір - синусоїдальний, прямокутний, трикутний чи випадковий;
- **Sine Wave** - генерує гармонічний сигнал;
- **Repeating Sequence** - генерує періодичну послідовність;
- **Discrete Pulse Generator** – генератор послідовності дискретних імпульсів;
- **Pulse Generator** - генератор неперервних прямокутних імпульсів;
- **Random Number** - джерело дискретного сигналу, значення якого є випадковою величиною, розподіленою за нормальним (гауссовим) законом;
- **Uniform Random Number** - джерело дискретного сигналу, значення якого є випадковою рівномірно розподіленою величиною;
- **Band-Limited White Noise** - генератор білого шуму з обмеженою смугою частот.

Як і інші блоки бібліотеки SimuLink, блоки-джерела можуть наструюватися користувачем.

Блок *Signal Generator*

Вікно настроювання цього блока показано на рис. 9.33.

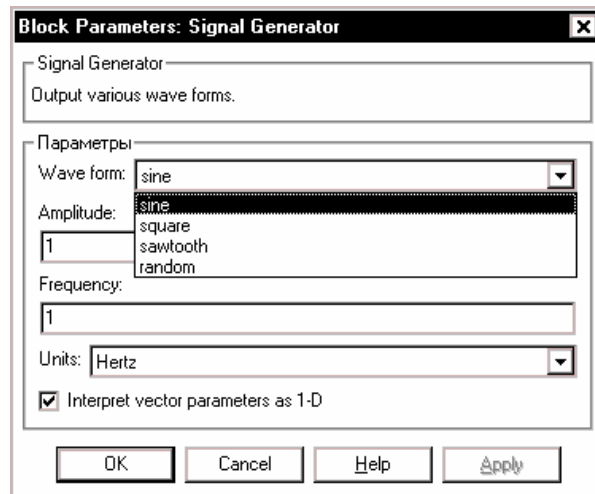


Рис. 9.33. Вікно настроювання блоку *Signal Generator*

Як очевидно, у параметри настроювання входять:

- *Wave form* - форма хвилі - дозволяє обрати одну з таких форм періодичного процесу
 - 1) *Sine* - синусоїдальні хвилі;
 - 2) *Square* - прямокутні хвилі;
 - 3) *Sawtooth* - трикутні хвилі;
 - 4) *Random* - випадкові коливання;
- *Amplitude* - визначає значення амплітуди (напіврозмаху) коливань, що генеруються;
- *Frequency* - задає частоту коливань;
- *Units* - дозволяє обрати одну з одиниць виміру частоти за допомогою спадного меню - *Hertz* (у Герцах) або *Rad/Sec* (у радіанах у секунду).

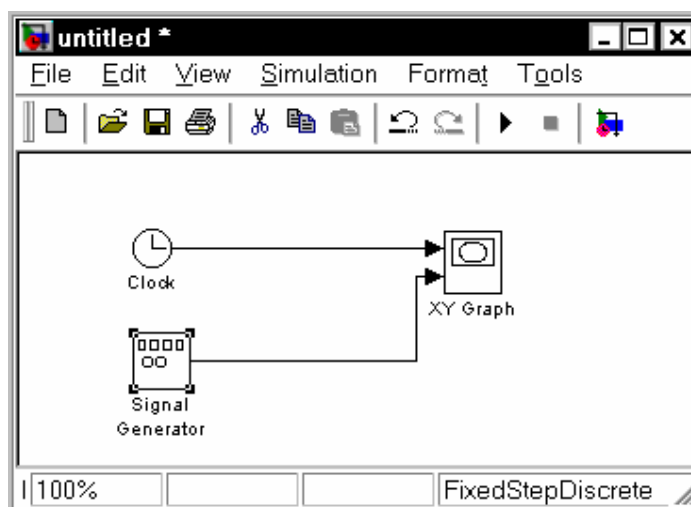


Рис. 9.34. Вікно блок-схеми використання блоку *Signal Generator*

На рис. 9.34 показано найпростішу блок-схему S-моделі, що складається з блока *Signal Generator* і блока відображення *XY Graph*.

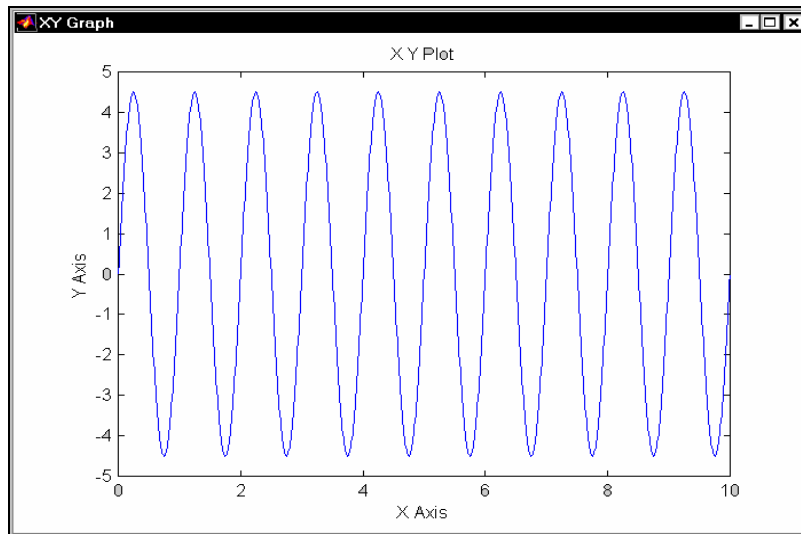


Рис. 9.35. Відображення синусоїдального сигналу блоком XYGraph

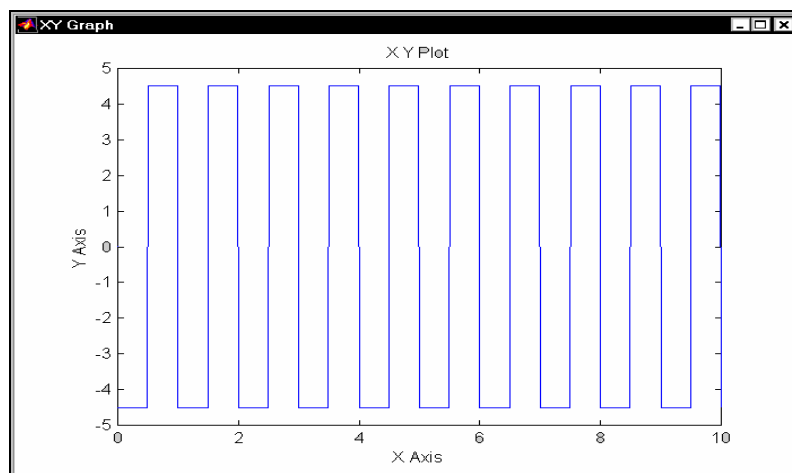


Рис. 9. 36. Процес із прямокутних хвиль

На наступному рис. 9.35 відображений вміст блока відображення після проведення моделювання при таких параметрах настроювання: вид коливань - *Sine*; амплітуда - 4,5; частота - 1 Гц.

На рис. 9.36.. 9.38 подані результати, відображувані у вікні *XYGraph* у випадку вибору відповідно прямокутних, трикутних і випадкових коливань при тих самих інших параметрах настроювання.

Неважко помітити, що з тієї обставини, що кількість параметрів настроювання дорівнює двом (можливо змінювати лише амплітуду хвиль та їхню частоту), впливає обмеженість видів відповідних процесів, що можуть бути згенеровані блоком *Signal Generator*. Так, синусоїдальний сигнал може мати початкову фазу лише рівну 0. Прямокутні хвилі можуть бути лише такими, що додатні й від'ємні напівхвилі є однаковими. Трикутні хвилі

генеруються лише з вертикальним переднім фронтом (див. рис. 9.37). Змінити цю форму неможливо.

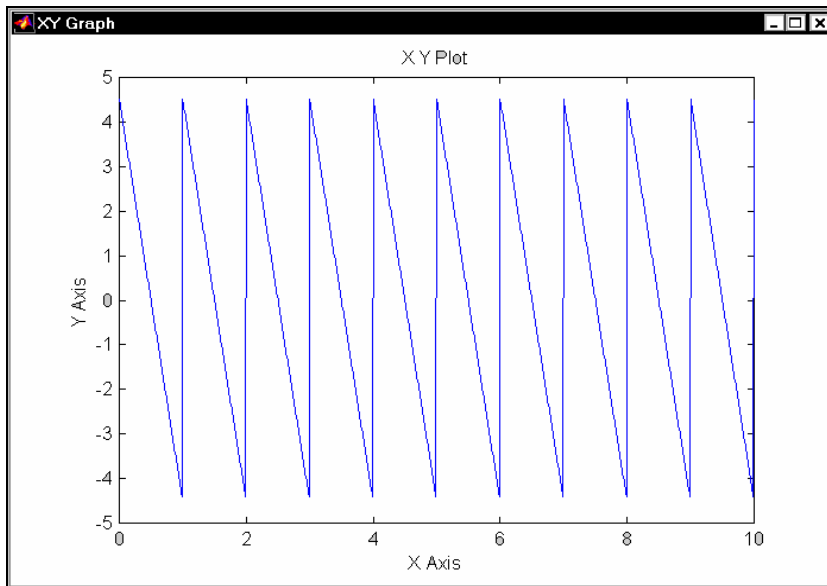


Рис. 9. 37. Процес із трикутних хвиль

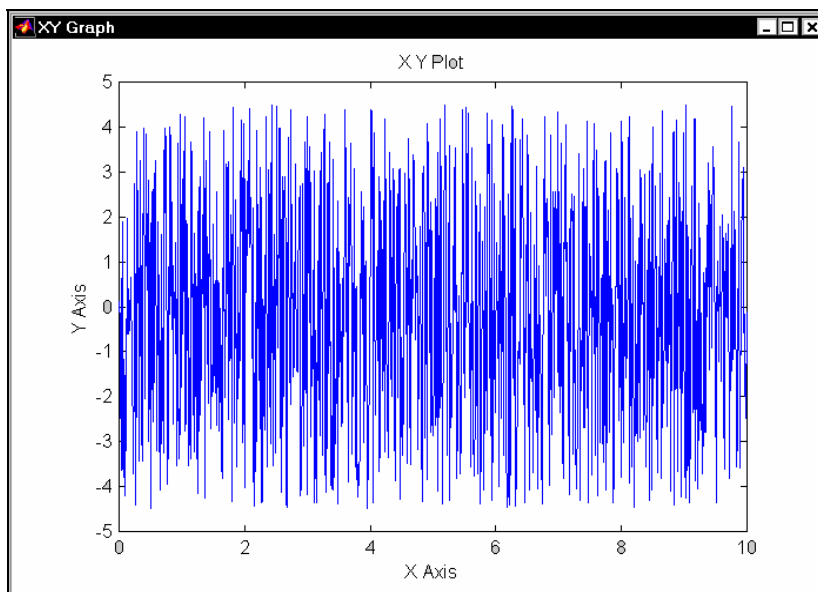


Рис. 9. 38. Випадкові коливання

Блок *Sine Wave*

Блок *Sine Wave* має такі настройки (рис. 9.39):

- *Amplitude* - амплітуда синусоїдального сигналу;
- *Bias* – величина сталої складової сигналу;
- *Frequency (rad/sec)* - задає частоту коливань у радіанах у секунду;
- *Phase (rad)* - початкова фаза в радіанах;
- *Sample time* - величина дискрету часу (у випадку, коли генерується дискретний за часом гармонічний сигнал).

Результат застосування блоку (при значеннях параметрів настроювання: амплітуда - 4,5; частота - 1 рад/с і початкову фазу - $\pi/2$ радіан) показаний на рис. 9.40.

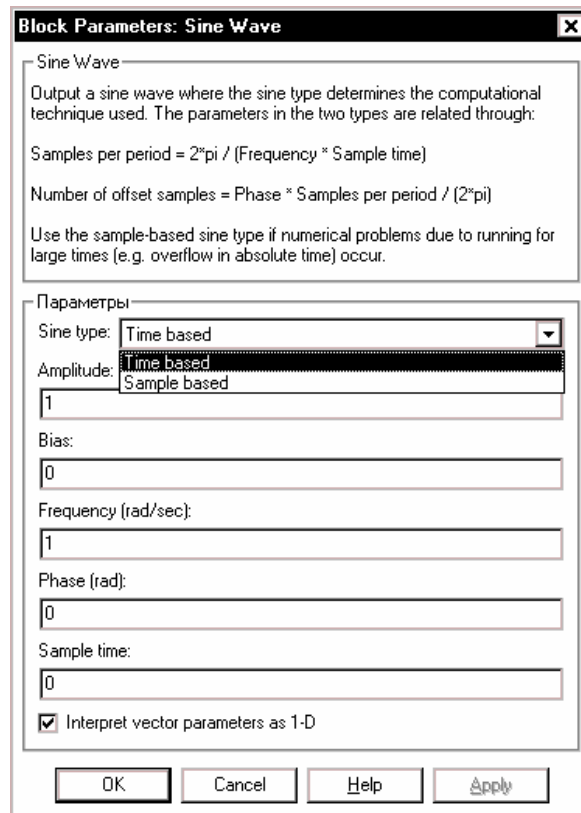


Рис. 9.39. Вікно настроювання блоку Sine Wave

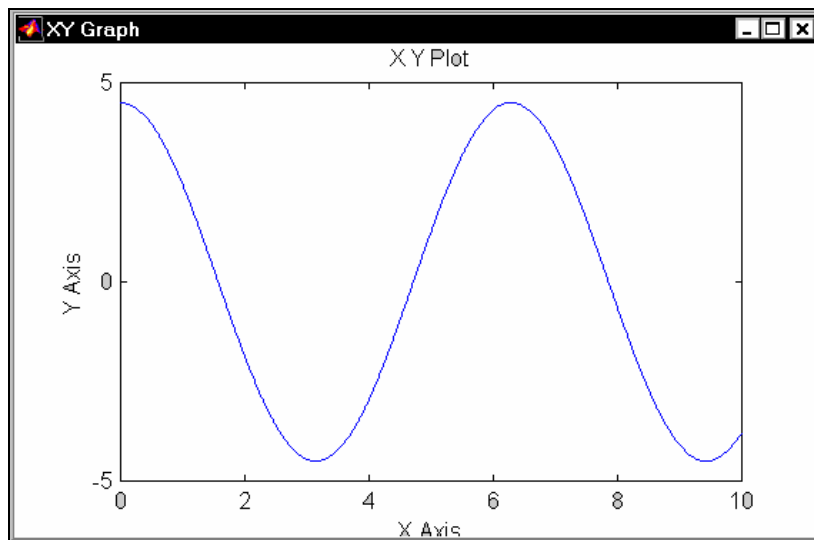


Рис. 9.40. Графічне відображення синусоїдального процесу

Відмінності цього блоку від генератора синусоїдальних коливань у блоці **Signal Generator** полягають у наступному:

- 1) в блоці **Sine Wave** можна встановлювати довільну початкову фазу;
- 2) в ньому можна встановлювати сталу складову сигналу **Bias**;

3) у ньому не можна задати частоту в Герцах.

Блок *Repeating Sequence*

Цей блок містить дві настройки (рис. 9.41):

- *Time values* - вектор значень часу, в які задані значення вихідної величини;
- *Output values* - вектор значень вихідної величини, які вона повинна прийняти в зазначені в першому векторі відповідні моменти часу.

Блок забезпечує генерування коливань із періодом, рівним різниці між останнім значенням вектора *Time values* і значенням першого його елемента. Форма хвилі усередині періоду є ламаною, що проходить через точки із зазначеними у векторах *Time values* і *Output values* координатами.

Як приклад на рис. 9.42 наведене зображення процесу, згенерованого блоком *Repeating Sequence* при параметрах настроювання, зазначених на рис. 9.41.

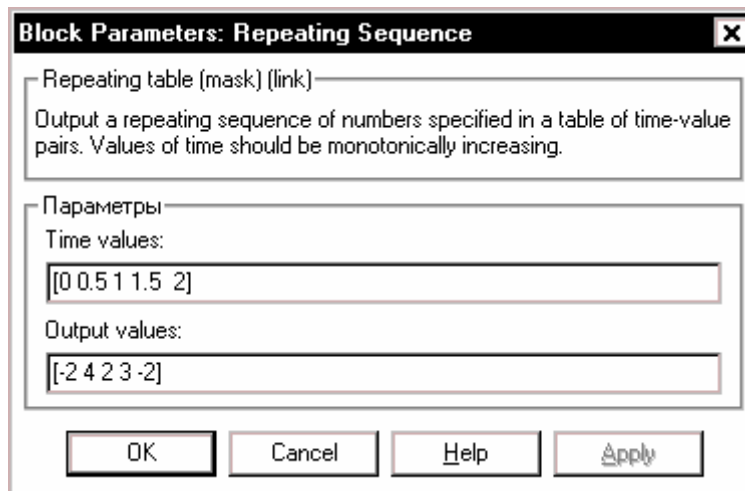


Рис. 9.41. Вікно настроювання блоку *Repeating Sequence*

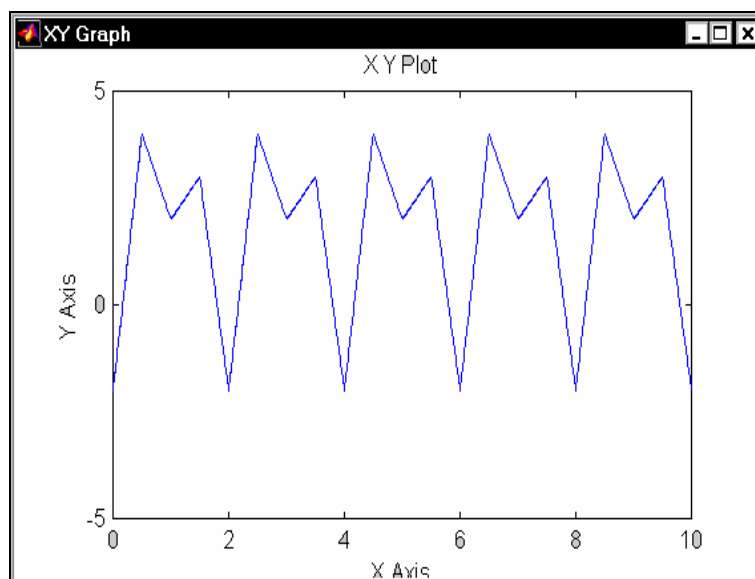


Рис. 9.42. Графічне відображення згенерованого процесу

Блок *Pulse Generator*

Блок генерує процес у вигляді послідовності прямокутних імпульсів. У число параметрів цього блоку, які настраюються, входять (рис. 9.43):

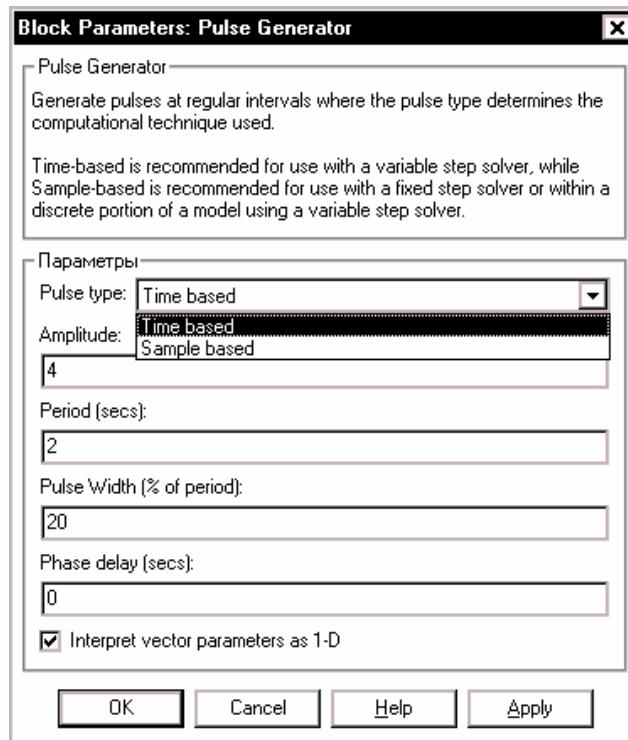


Рис. 9.43. Вікно налаштування блоку *Pulse Generator*

- амплітуда сигналу (*Amplitude*), тобто висота прямокутного імпульсу;
- розмір періоду сигналу (*Period*), в секундах;
- ширина імпульсу (*Pulse width*), у відсотках від періоду;
- розмір затримки імпульсу щодо $t=0$ (*Phase delay*) - у секундах.

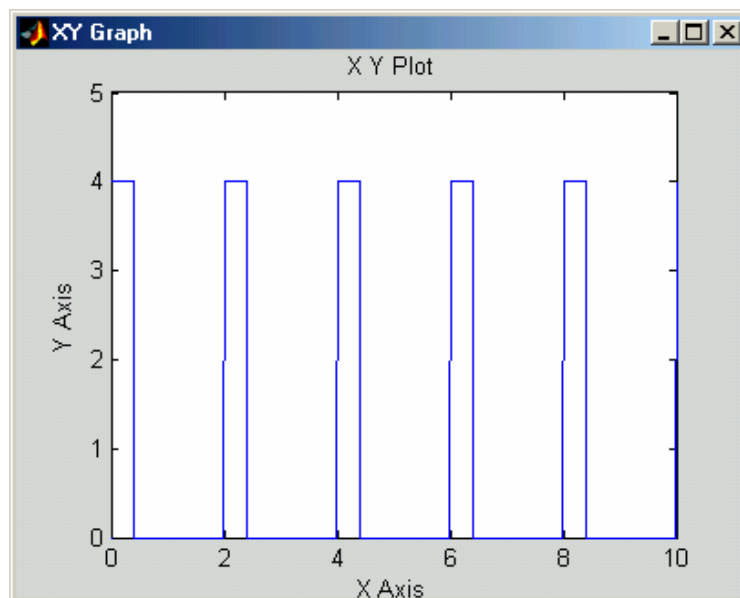


Рис. 9.44. Графічне відображення згенерованої послідовності

Приклад застосування цього блоку при значеннях параметрів, зазначених на рис. 9.43, наведений на рис. 9.44.

Блок *Random Number*

Блок *Random Number* забезпечує формування сигналів, значення яких в окремі моменти часу є випадковою величиною, розподіленою за нормальним (гауссовим) законом із заданими параметрами.

Блок має чотири параметри настроювання (рис. 9.45). Перші два - *Mean* і *Variance* - є параметрами нормального закону (середнє й середньоквадратичне відхилення від цього середнього), третій - *Initial seed* - задає початкове значення для ініціалізації генератора послідовності випадкових чисел.

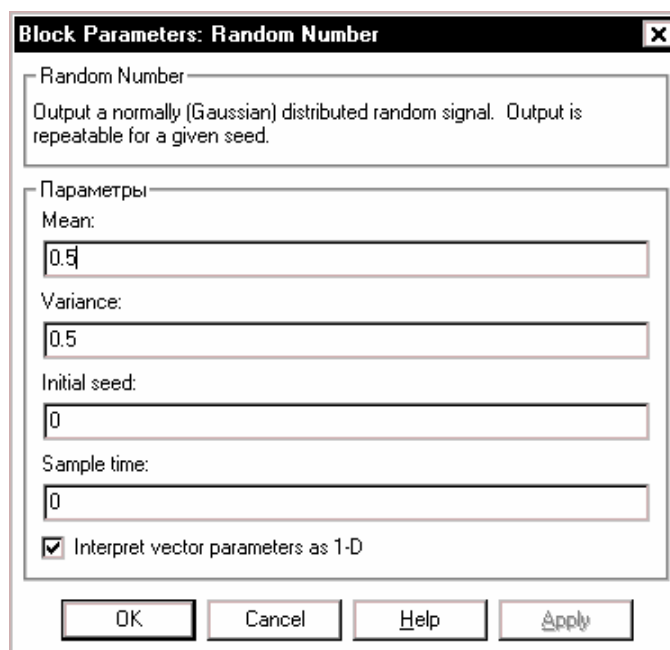


Рис. 9.45. Вікно настроювання блоку *Random Number*

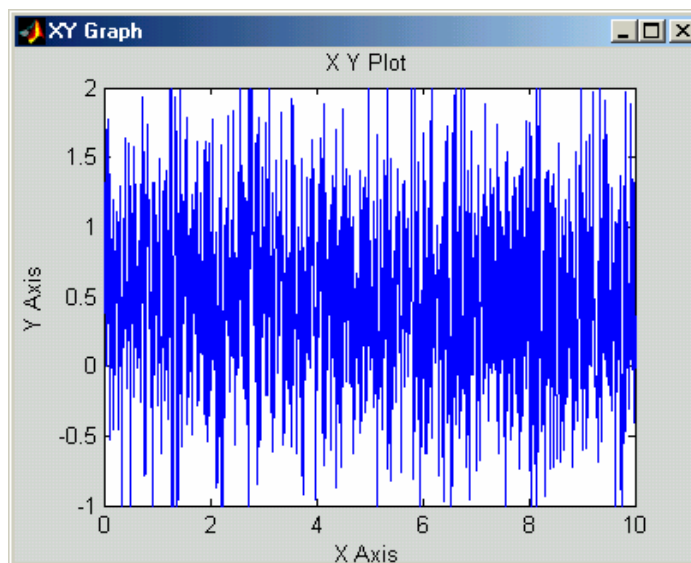


Рис. 9.46. Результат роботи блоку *Random Number*

При фіксованому значенні цього параметра генератор завжди виробляє ту саму послідовність. Четвертий параметр (*Sample time*), задає величину дискрету часу, який визначає інтервал часу, через який змінюється значення сигналу. Якщо цей параметр встановлено рівним -1, значення сигналу змінюється на кожному кроці змінювання модельного часу.

Рис. 9.46 показує результат використання блока при значеннях параметрів, наведених на рис. 9.45.

На рис. 9.47 наведений результат роботи блока при встановленні значення дискрета часу рівним 0.2 сек.

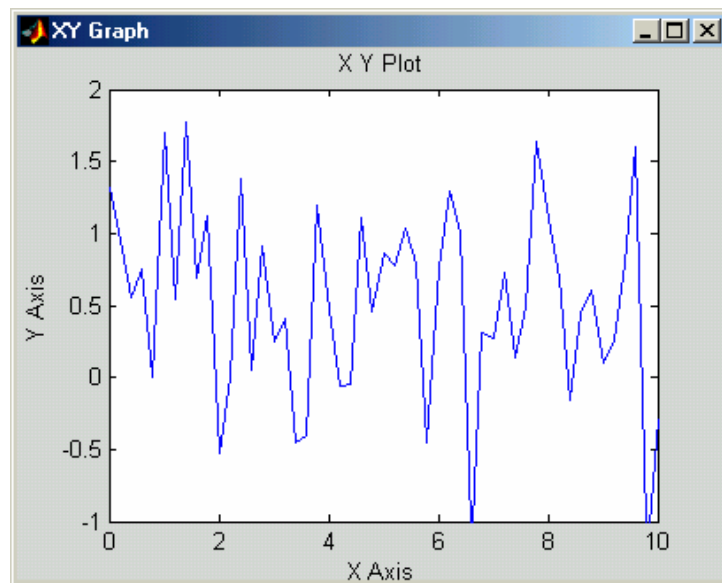


Рис. 9.47. Результат роботи блока *Random Number* при *Sample time=0.2*

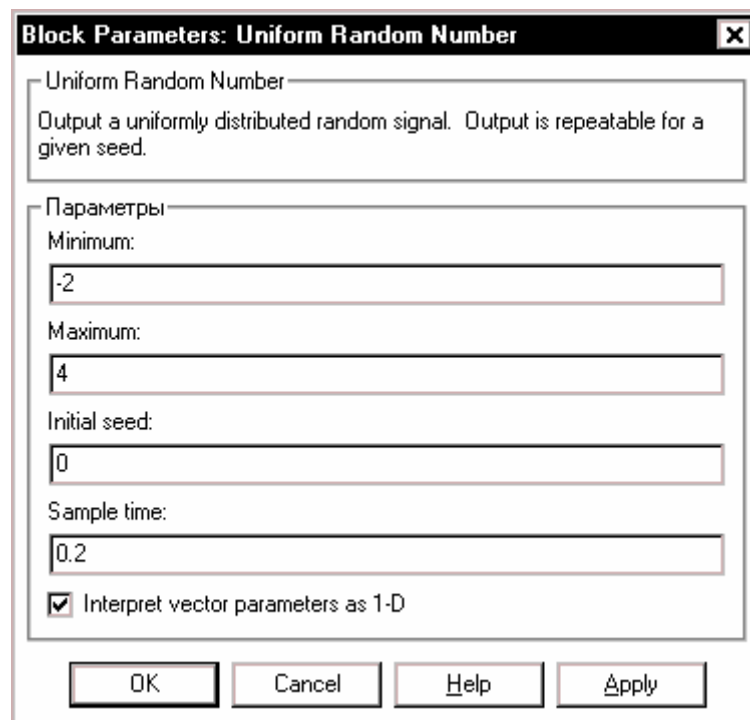


Рис. 9.48. Вікно налаштування блока *Uniform Random Number*

Блок *Uniform Random Number*

Цей блок формує сигнали, значення яких в окремі моменти часу є випадковою величиною, що рівномірно розподілена в заданому інтервалі. У число параметрів налаштування блока входять (рис. 9.48):

- *Minimum* - нижня межа випадкової величини;
- *Maximum* - верхня межа;
- *Initial seed* - початкове значення бази генератора випадкових чисел;
- *Sample time* - дискрет часу.

Приклад процесу, згенерованого блоком по параметрах, зазначених на рис. 9.48, наведений на рис. 9.49. Крок змінювання модельного часу встановлено рівним 0.01 сек.

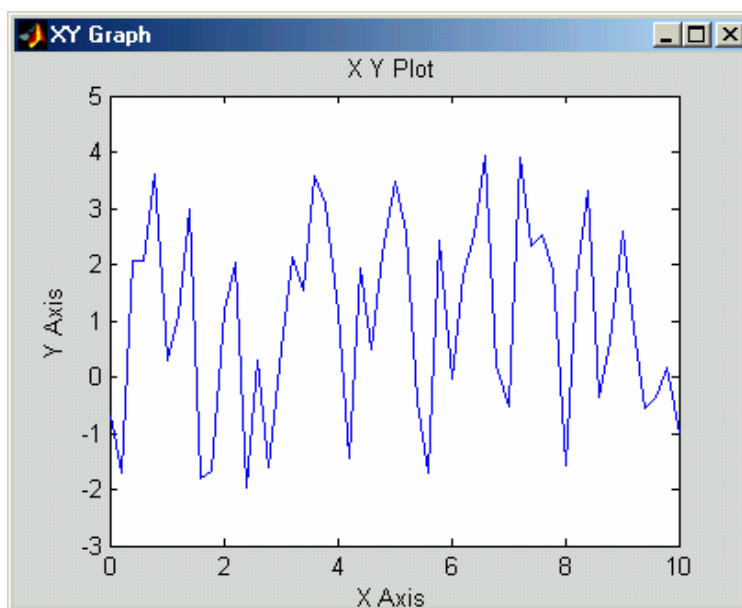


Рис. 9.49. Відображення випадкового процесу з рівномірним розподілом

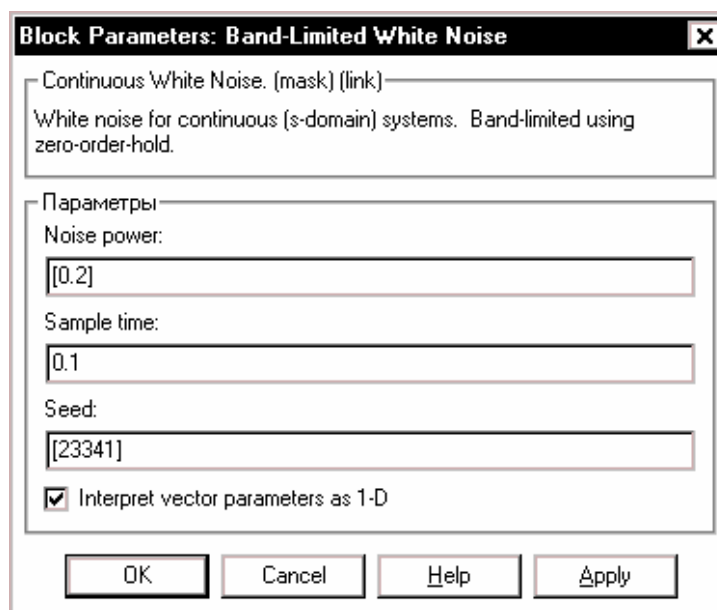


Рис. 9.50. Вікно налаштування блоку *Band-Limited White Noise*

Блок *Band-Limited White Noise*

Цей блок формує процес у виді частотно-обмеженого білого шуму. Параметри настроювання в нього такі (рис. 9.50):

- *Noise power* - значення потужності білого шуму;
- *Sample time* - значення дискрету часу (визначає верхнє значення частоти процесу);
- *Seed* - початкове значення бази генератора випадкової величини.

Нижче (рис. 9.51) наведено приклад реалізації процесу за допомогою блока *Band-Limited White Noise* при параметрах, зазначених на рис. 9.50.

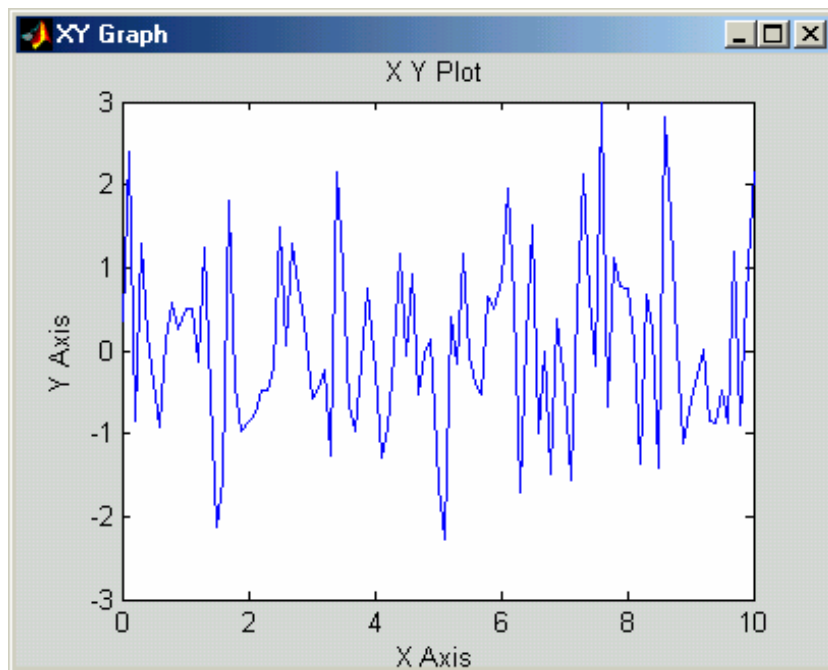


Рис. 9.51. Результат роботи блоку *Band-Limited White Noise*

9.4. Засоби математичної обробки сигналів

Цифрова обробка сигналів традиційно містить в себе створення засобів чисельного перетворення масиву заданого (виміряного у дискретні моменти часу) процесу змінювання деякої неперервної фізичної величини з метою видобуття з нього корисної інформації про іншу фізичну величину, яка міститься у виміряному сигналі.

Загальна схема утворення вимірювального сигналу и процесу його перетворення з метою одержання інформації про величину, яка має бути виміряна, подана на рис. 9.52.

Фізична величина, що є корисною (яка є носієм потрібної інформації), рідко має таку фізичну форму, що може бути безпосередньо виміряна. Зазвичай вона являє собою лише певну складову (сторону, частину, рису) деякої іншої фізичної величини, яка може бути безпосередньо виміряна. Зв'язок між цими двома величинами позначимо введенням ланки, яку назовемо "первісним перетворювачем" (ПП).

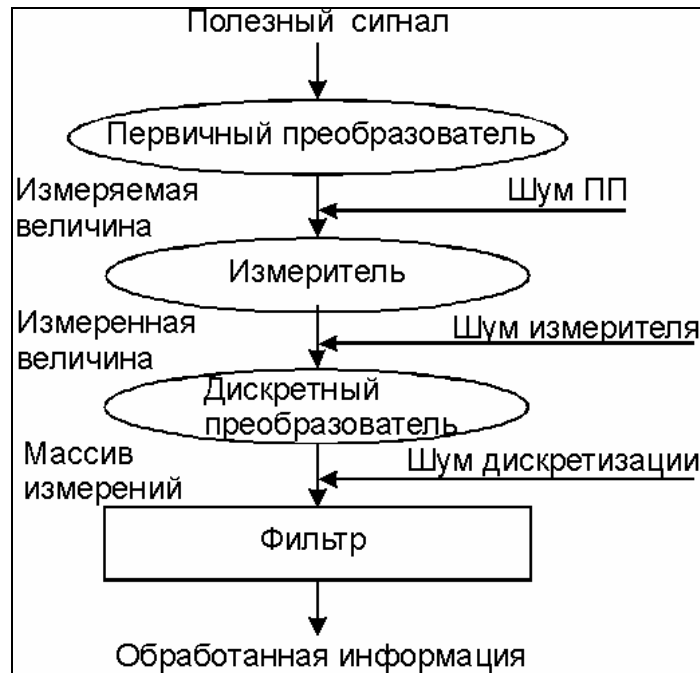


Рис. 9.52. Схема вимірювання і перетворення сигналу

Зазвичай закон перетворювання є відомим заздалегідь, апіорі, інакше відновити інформаційну складову у подальшому було б неможливим. Первісний перетворювач вносить залежність сигналу, який може бути вимірний, від деяких інших фізичних величин. Внаслідок цього вихідна його величина мстить, окрім корисної інформаційної складової, інші, шкідливі складові або риси, які спотворюють корисну інформацію. І хоча залежність виходу ПП від цих інших величин також є відомою, але внаслідок можливого неконтрольованого змінювання останніх з часом, часто важко спрогнозувати їхній вплив на спотворювання корисної інформаційної складової. Назвемо шкідливу складову, що вноситься ПП, *шумом ПП*.

Нехай утворена у такий спосіб безпосередньо вимірювана величина вимірюється деяким вимірювачем. Будь-який реальний вимірювач вносить власні спотворювання у вимірювану величину і додаткові залежності від деяких інших фізичних величин, що не є об'єктом вимірювання. Назвемо ці спотворення *шумами вимірювача*. Не обмежуючі узагальненості, вважатимемо, що вихідною величиною вимірювача є електричний сигнал (*виміряна величина*), який можна у подальшому досить просто перетворювати за допомогою електричних пристроїв.

Задля здійснення цифрової обробки виміряна величина має бути перетворена у дискретну форму за допомогою спеціального пристрою, який складається з *екстраполятора* і *аналого-цифрового перетворювача* (АЦП). Перший здійснює фіксацію поточного значення вимірної величини у окремі моменти часу через визначений сталий проміжок часу, що зветься *дискретом часу*. Другий переводить це значення у цифрову форму, яка дозволяє у подальшому здій-

снювати перетворювання за допомогою цифрових ЕОМ. Хоча обидва пристрої можуть вносити при таких перетвореннях власні спотворювання у вихідний (дискретний) сигнал, однак ними зазвичай нехтують, бо у більшості випадків ці додаткові спотворення є значно меншими за шуми ПП і вимірювача.

Щоб на основі одержаного дискретизованого сигналу отримати корисний сигнал, потрібно розрахувати і створити пристрій (чи програму для ЕОМ), яке здійснювало б такі перетворення вхідного дискретного у часі сигналу, щоб на його виході перетворення, що внесені шумами ПП і вимірювача, були мінімізовані у деякому сенсі. Цей пристрій називають *фільтром*.

У загальному випадку створення (проектування) фільтра є задачею невизначеною, яка конкретизується лише на основі попередньо відомих знань про закономірності утворення вимірюваної величини (моделі ПП), про модель утворення вимірюваної величини з вимірюваної (моделі вимірювача), про характеристики змінювання у часі шкідливих фізичних чинників, що впливають на утворення вимірюваної і вимірюваної величин, і закономірностей їхнього впливу на спотворення корисної інформації.

Через те що моделі ПП і вимірювача можуть бути вельми різноманітними, традиційно задачу фільтрування вирішують лише для деяких найбільш розповсюджених на практиці видів таких моделей, частіше за все - для лінійних моделей.

У загальному випадку процес створення фільтра розпадається на такі етапи:

- на основі априорної інформації про моделі ПП та вимірювача і про характеристики шумів, а також про задачі, які має розв'язувати фільтр, обирається певний тип фільтра з відомих, теорія проектування яких є розробленою;
- на основі конкретних числових даних розраховуються числові характеристики обраного типу фільтра (створюється конкретний фільтр);
- перевіряється ефективність виконання розробленим фільтром поставленої перед ним задачі; для цього необхідно симітувати на ЕОМ дискретний сигнал, що містить корисну (інформаційну) складову з накладеними на неї передбаченими шумами ПП і вимірювача, «пропустити» крізь побудований фільтр і порівняти одержаний на виході сигнал з відомою (у цьому випадку) корисною його складовою; різниця між ними характеризуватиме похибки вимірювання на виході фільтра;
- через те що в реальних умовах деякі характеристики шумів можуть відрізнятися від прийнятих при проектуванні (створенні фільтра), не зайвими становляться *випробування ефективності роботи фільтра* в умовах більш наближених до реальних, ніж ті, що прийняті при проектуванні.

Задля здійснювання операцій за трьома останніми з зазначених етапів призначений пакет Signal Processing Toolbox (у подальшому скорочено - Signal) системи Matlab. Він дозволяє проектувати (розраховувати конкретні числові характеристики) цифрові і аналогові фільтри по потрібних амплітудно- і фазо-частотних їхніх характеристиках, формувати послідовності типових часових сигналів і обробляти їх спроектованими фільтрами. В пакет входять процедури, які здійснюють перетворення Фур'є, Гільберта, а також статистичний аналіз. Пакет дозволяє розраховувати кореляційні функції, спектральну щільність потужності сигналу, оцінювати параметри фільтрів по вимірних відліках вхідної і вихідної послідовностей.

Задача генерування сигналів заданої форми здійснюється засобами, описаними у попередньому поділі.

Необхідним етапом моделювання процесів фільтрації є безпосереднє здійснювання фільтрації. Для цього необхідно спочатку задати самий фільтр як певну динамічну ланку, а потім провести обробку сформованих раніше сигналів за допомогою цієї ланки, застосовуючи спеціальну процедуру фільтрації.

Розглянемо основи лінійної фільтрації на прикладі лінійного стаціонарного фільтра, який в неперервному часі описується диференціальним рівнянням другого порядку:

$$\ddot{y} + 2\zeta\omega_o \cdot \dot{y} + \omega_o^2 \cdot y = A \cdot x, \quad (9.1)$$

де x - заданий процес, який надається до входу цього фільтра другого порядку; y - процес, який одержується на виході фільтра; ω_o - частота власних коливань фільтра, а ζ - відносний коефіцієнт загасання цього фільтра.

Передатна функція фільтра, очевидно, має вид:

$$W(s) = \frac{y(s)}{x(s)} = \frac{A}{s^2 + 2\zeta\omega_o \cdot s + \omega_o^2}. \quad (9.2)$$

Для контролю і графічного подання передатної функції будь-якої лінійної динамічної ланки зручно використовувати процедуру **freqs**. У загальному випадку звернення до неї має вигляд:

$$\mathbf{h} = \text{freqs}(\mathbf{b}, \mathbf{a}, \mathbf{w}).$$

При цьому процедура утворює вектор \mathbf{h} комплексних значень частотної характеристики $W(j\omega)$ по передатній функції $W(s)$ ланки, яка задана векторами коефіцієнтів \mathbf{b} її чисельника і \mathbf{a} знаменника, а також по заданому вектору \mathbf{w} частоти ω . Якщо аргумент \mathbf{w} не вказаний, процедура автоматично обирає 200 відліків частоти, для яких обчислюється частотна характеристика.

Примітка. Якщо не вказана вихідна величина, тобто звернення має вид **freqs(b, a, w)**, процедура виводить у поточне графічне вікно два графіки - АЧХ і ФЧХ.

Наведемо приклад. Нехай для передатної функції (9.2) обрані такі значення параметрів:

$$A = 1; \quad \zeta = 0.05; \quad T_o = 2\pi / \omega_o = 1.$$

Обчислимо значення коефіцієнтів чисельника й знаменника та виведемо графіки АЧХ і ФЧХ:

```
T0=1; dz=0.05; om0=2*pi/T0; A=1;
a1(1)=1; a1(2)=2*dz*om0; a1(3)=om0^2; b1(1)=A;
freqs(b1,a1)
title('А Ч Х и Ф Ч Х фильтра')
```

В результаті одержимо рис. 9.53.

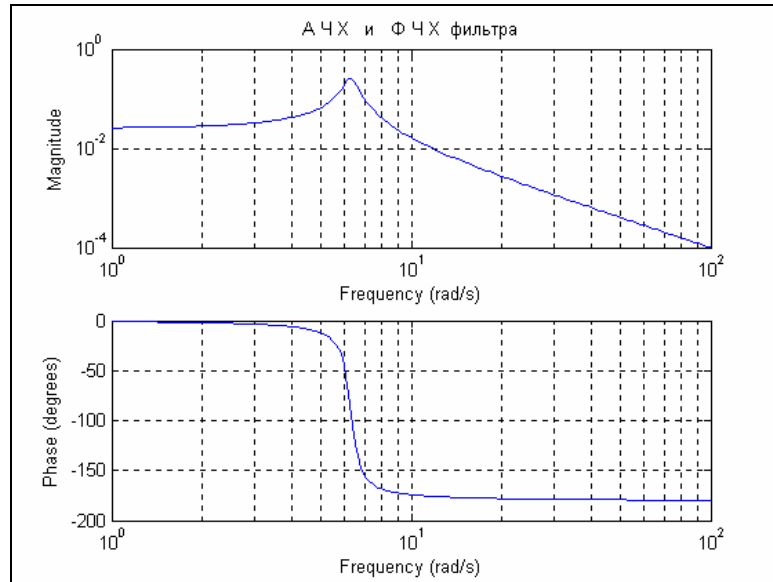


Рис. 9. 53. Результат роботи процедури *FREQS*

Припустимо, що заданий процес $x(t)$ поданий у виді окремих його значень у дискретні моменти часу, які розділені однаковими проміжками T_S часу (дискретом часу). Позначимо через $x(k)$ значення процесу в момент часу $t = k \cdot T_S$, де k - номер виміру з початку процесу.

Запишемо рівняння (9.1) через скінченні різниці процесів x і y , враховуючи, що скінченно-різницеvim еквівалентом похідної $\dot{y} = \frac{dy}{dt}$ є скінченна різниця

$$\frac{\Delta y(k)}{T_S} = \frac{y(k) - y(k-1)}{T_S},$$

а еквівалентом похідної другого порядку \ddot{y} є скінченна різниця другого порядку

$$\frac{\Delta^2 y(k)}{T_S^2} = \frac{\Delta y(k) - \Delta y(k-1)}{T_S^2} = \frac{y(k) - 2y(k-1) + y(k-2)}{T_S^2}.$$

Тому різницеve рівняння

$$(1 + 2\zeta\omega_o \cdot T_S + \omega_o^2 \cdot T_S^2) \cdot y(k) - 2(1 + \zeta\omega_o \cdot T_S) \cdot y(k-1) + y(k-2) = A \cdot T_S^2 \cdot x(k) \quad (9.3)$$

є дискретним аналогом диференційного рівняння (9.1).

Застосовуючи до одержаного рівняння Z -перетворення, отримуємо:

$$y(z) \cdot [a_0 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}] = A \cdot T_S^2 \cdot x(z), \quad (9.4)$$

де $a_0 = 1 + 2\zeta\omega_o \cdot T_S + \omega_o^2 \cdot T_S^2$; $a_1 = -2(1 + \zeta\omega_o \cdot T_S)$; $a_2 = 1$. (9.5)

Дискретна передатна функція фільтра визначається з рівняння (9.4):

$$G(z) = \frac{y(z)}{x(z)} = \frac{A \cdot T_S^2}{a_0 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}}, \quad (9.6)$$

Отже, цифровим аналогом раніше введеної коливальної ланки є цифровий фільтр з коефіцієнтами чисельника і знаменника, розрахованими за формулами (9.4) і (9.5):

```
T0=1; dz=0.05; Ts=0.01;
om0=2*pi/T0; A=1; oms=om0*Ts;
a(1)= 1+2*dz*oms+oms^2; a(2)= -2*(1+dz*oms); a(3)=1;
b(1)=A*Ts*Ts*(2*dz*om0^2);
```

Щоб одержати частотну дискретну характеристику $G(e^{j\omega})$ за дискретною передатною функцією $G(z)$, яка задана векторами значень її чисельника b і знаменника a , зручно використовувати процедуру **freqz**, звернення до якої є аналогічним зверненню до процедури **freqs** :

```
freqz( b,a)
title('А Ч Х и Ф Ч Х дискретного фільтра')
```

Результат наведений на рис. 9.54.

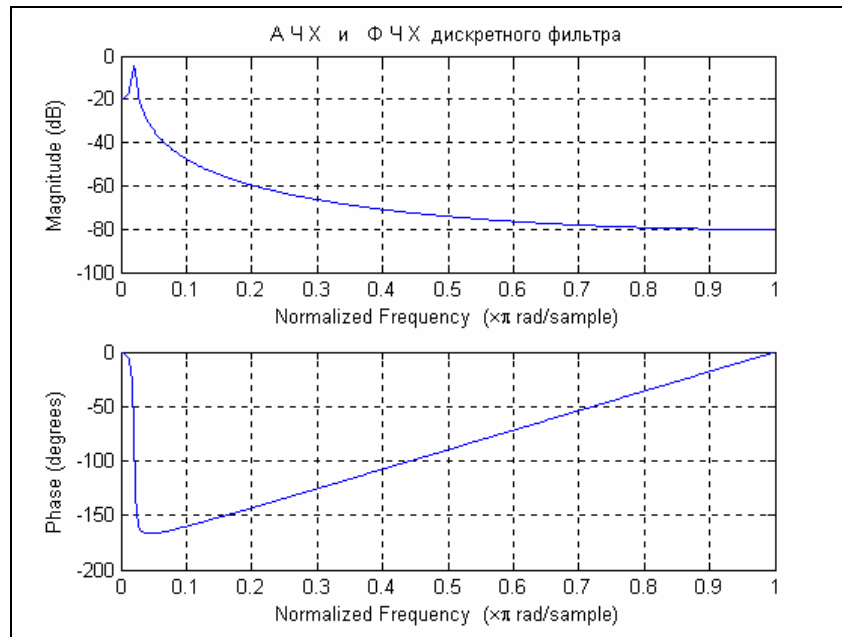


Рис. 9. 54. Результат дії процедури **FREQZ**

В системі MatLAB фільтрування, тобто перетворення заданого сигналу за допомогою лінійного фільтра, який описується дискретною передатною функцією виду

$$G(z) = \frac{y(z)}{x(z)} = \frac{b_0 + b_1 \cdot z^{-1} + \dots + b_m \cdot z^{-m}}{a_0 + a_1 \cdot z^{-1} + \dots + a_n \cdot z^{-n}} \quad (9.7)$$

здійснюється процедурою **filter** наступним образом

$$y = \text{filter}(b, a, x),$$

де x - заданий вектор значень вхідного сигналу; y - вектор значень вихідного сигналу фільтра; b - вектор коефіцієнтів чисельника дискретної передатної функції (9.7) фільтра; a - вектор коефіцієнтів знаменника цієї функції.

Як приклад розглянемо таку задачу. Нехай потрібно одержати достатньо достовірну інформацію про деякий «корисний» сигнал, який має синусоїдальну форму з відомим періодом $T_1 = 1\text{с}$ і амплітудою $A_1 = 0.75$. Сформуємо цей сигнал як вектор його значень у дискретні моменти часу з дискретом $T_s = 0.001\text{ с}$ (рис. 9.55):

```
Ts=0.001; t=0 : Ts : 20; A1=0.75; T1=1;
Yp=A1*sin(2*pi*t/T1);
plot(t(10002:end),Yp(10002:end)),grid,
title("'Корисний' процес '); xlabel('Час (с)'); ylabel('Yp(t)')
```

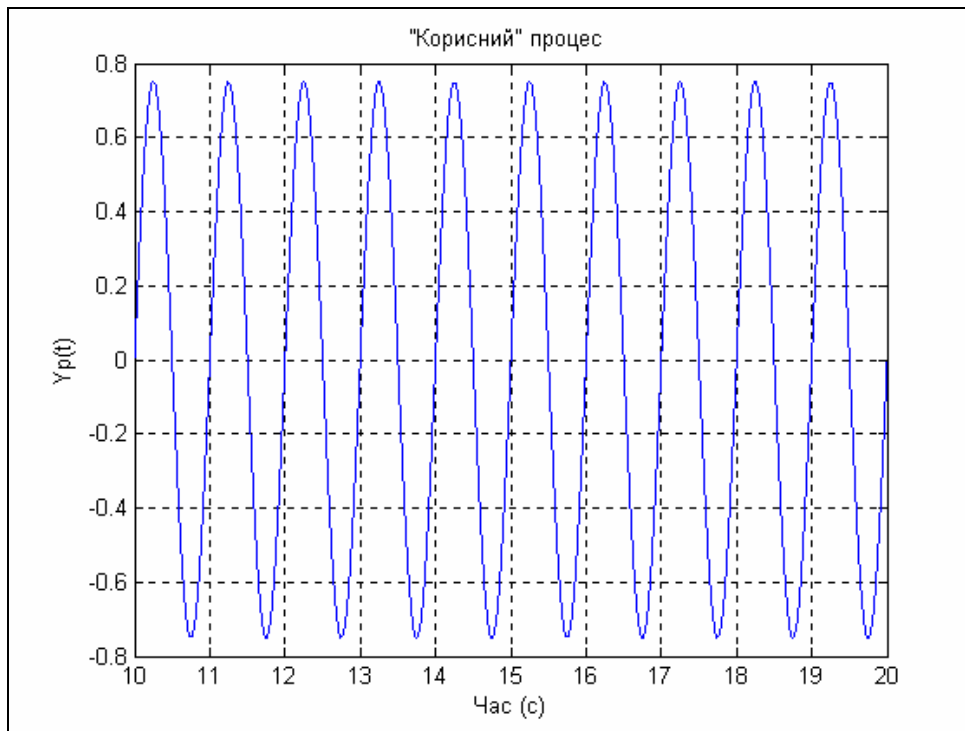


Рис. 9. 55. Вид корисного сигналу

Припустимо, що внаслідок проходження через ПП (первісний перетворювач) до корисного сигналу додався шум ПП у вигляді більш високочастотної синусоїди з періодом $T_2 = 0.2\text{ с}$ і амплітудою $A_2=5$, а в результаті вимірювання до нього ще додався білий гауссовий шум вимірювача з інтенсивністю $A_{\text{ш}} = 5$. В результаті утворився наступний вимірний сигнал $x(t)$ (см. рис. 9.56):

```
T2=0.2;; A2=10; eps=pi/4; Ash=5;
x=A1.*sin(2*pi*t./T1)+A2.*sin(2*pi.*t./T2+eps)+Ash*randn(1,length(t));
plot(t(10002:end),x(10002:end)),grid,
set(gca,'FontSize',12),
title('Вхідний процес '); xlabel('Час (с)');
ylabel('X(t)')
```

Потрібно так обробити виміряні дані x , щоб відновити за ними корисний процес по можливості точніше.

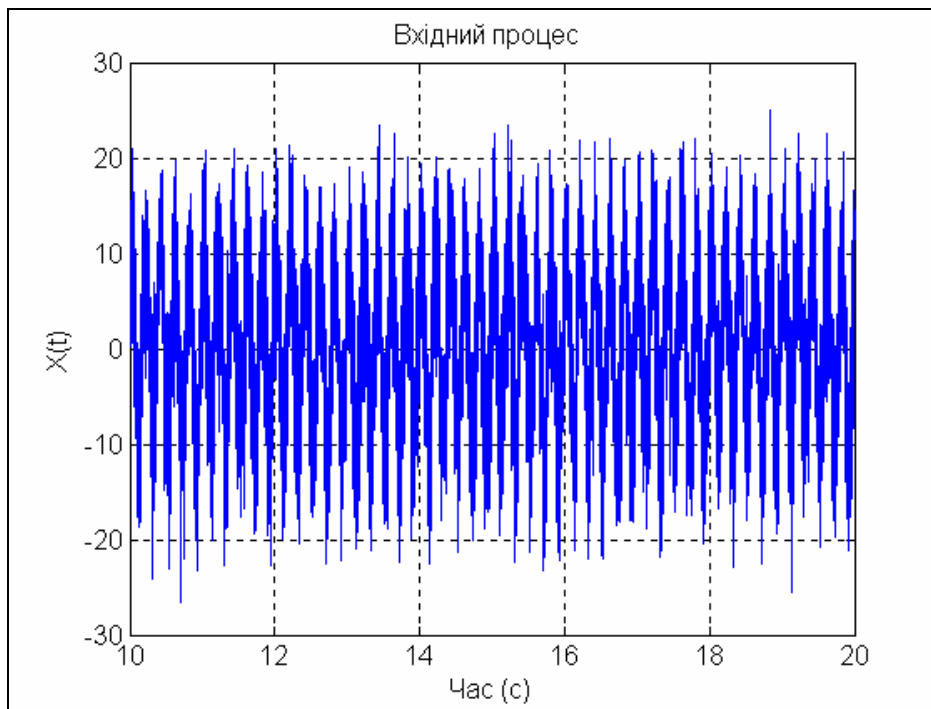


Рис. 9. 56. Вид виміряного сигналу

Через те що частота корисного сигналу заздалегідь відома, його відновлення можна здійснити за допомогою резонансного фільтра зазначеного вище виду. При цьому необхідно утворити такий фільтр, щоб період його власних коливань T_f дорівнював періоду коливань корисного сигналу ($T_f = T_1$). Для того, щоб після проходження через такий фільтр амплітуда відновленого сигналу збіглася з амплітудою корисного сигналу, потрібно вхідний сигнал фільтра помножити на сталу величину $2\zeta\omega_0^2$ (бо при резонансі амплітуда вихідного сигналу «зменшується» саме у стільки раз у порівнянні з амплітудою вхідного сигналу).

Сформуємо фільтр, який описаний вище:

```
T1=1; Tf=T1; dz=0.05;
om0=2*pi/Tf; A=1; oms=om0*Ts;
a(1)= 1+2*dz*oms+oms^2; a(2)= - 2*(1+dz*oms); a(3)=1;
b(1)=A*Ts*Ts*(2*dz*om0^2);
```

і «пропустимо» сформований процес крізь нього

```
y=filter(b,a,x);
plot(t(10002:end),y(10002:end),'.',t(10002:end),Yp(10002:end))
grid, set(gca,'FontSize',12)
title('Процес на виході фільтру (Tf=1; dz=0.05)');
xlabel('Час (с)'); ylabel('Y(t)')
legend('відновлений','первісний',0)
```

В результаті одержуємо відновлений процес (рис. 9.57). Для порівняння на тому самому графіку зображений відновлюваний процес.

Як бачимо, утворений фільтр досить добре відновлює корисний сигнал. Однак більш точному відновленню перешкоджають дві обставини:

1) відновлений процес усталюється на виході фільтра лише деякий час по тому внаслідок нульових початкових умов самого фільтра як динамічної ланки; це проілюстровано нижче, на рис. 9.58;

```
y=filter(b,a,x);
plot(t,y,'.t,Yp),grid, set(gca,'FontSize',12)
title('Процес на виході фільтру (Tf=1; dz=0.05)');
xlabel('Час (с)'); ylabel('Y(t)')
legend('відновлений','первісний',0)
```

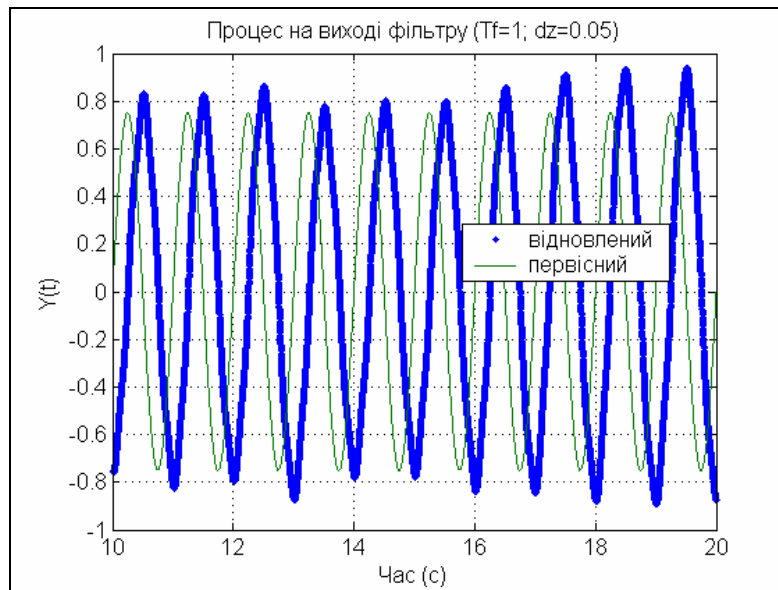


Рис. 9. 57. Результат фільтрування функцією FILTER

2) в усталеному режимі спостерігається значний зсув ($\pi/2$) фаз між відновлюваним та відновленим процесами; це також зрозумілим, бо при резонансі зсув фаз між вхідним (збуджуючим) і вихідним (збуреним) процесами сягає саме такої величини.

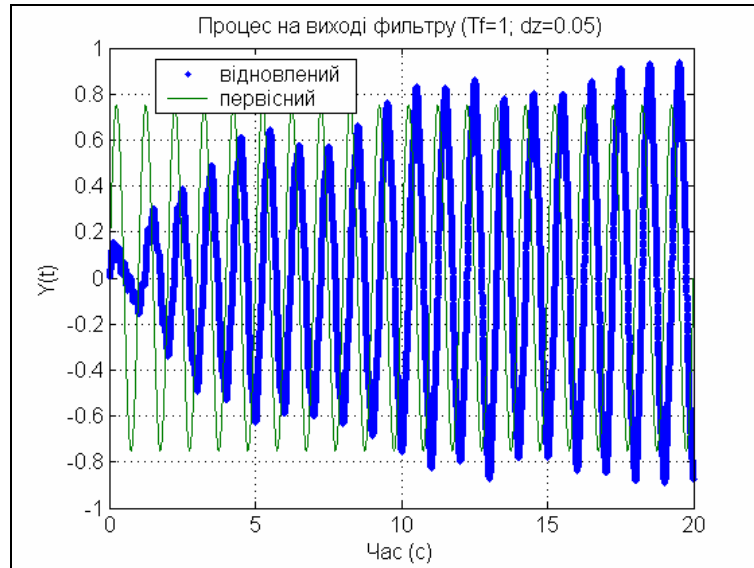


Рис. 9. 58. Переходний процес фільтра

Щоб уникнути фазових спотворень корисного сигналу при його відновленні, можна скористатися процедурою подвійної фільтрації - **filtfilt**. Звернення до неї має ту саму форму, що й до процедури **filter**. На відміну від останньої, процедура **filtfilt** здійснює обробку вектора **x** у два заходи: спочатку в прямому, а потім у зворотному напрямку.

Результат застосування цієї процедури у розглядуваному випадку наведений на рис. 9.59.

```

y=filtfilt(b,a,x);
plot(t,y,'.',t,Yp),grid, set(gca,'FontSize',12)
title('Застосування процедури FILTFILT');
xlabel('Время (с)'); ylabel('Y(t)')
legend('відновлений','первісний',0)
    
```

Як можна впевнитися, у цьому випадку амплітуда й фаза гармонічного сигналу відновлюються задовільно, хоча для цього потрібний досить великий час, протягом якого процес на виході фільтра усталюється.

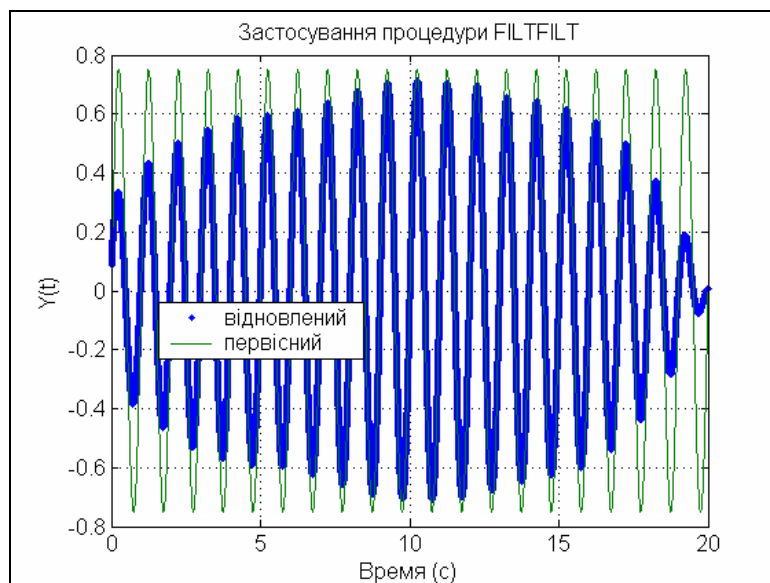


Рис. 9. 59. Результат застосування процедури FILTFILT

9.5. Спектральний і статистичний аналіз процесів

Головною задачею спектрального аналізу процесів є виявлення гармонічного спектра цих процесів, тобто визначення частот гармонічних складових процесу (*виявлення частотного спектра*), амплітуд цих гармонічних складових (*амплітудного спектру*) та їх початкових фаз (*фазового спектра*). До задач статистичного аналізу двох процесів зазвичай відносять визначення характеристик зв'язку цих процесів, таких як взаємна кореляційна функція і пов'язана з нею взаємна спектральна щільність.

В основі спектрального аналізу лежить теорія Фур'є про можливість розкладу будь-якого періодичного процесу з періодом $T = \frac{2\pi}{\omega} = \frac{1}{f}$ (де ω - кругова частота періодичного процесу, а f - його частота в герцах) у нескінченну, але переліковну суму окремих гармонічних складових.

Викладемо деякі основні положення спектрального аналізу.

Перш за все, будь-який періодичний процес з періодом T може бути поданий у вигляді так званого комплексного ряду Фур'є:

$$x(t) = \sum_{m=-\infty}^{+\infty} X^*(m) \cdot e^{j(2\pi mf)t} = \sum_{m=-\infty}^{+\infty} X^*(m) \cdot e^{j(m\omega)t}, \quad (9.8)$$

причому комплексні числа $X^*(m)$, які називають *комплексними амплітудами* гармонічних складових, визначаються по формулах:

$$X^*(m) = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) \cdot e^{-j(2\pi mf)t} dt = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) \cdot e^{-j(m\omega)t} dt. \quad (9.9)$$

Отже, частотний спектр періодичного коливання складається з частот, кратних основній (базовій) частоті f , тобто частот

$$f_m = m \cdot f \quad (m = 0, 1, 2, \dots) \quad (9.10)$$

Дійсні і уявні частини комплексних амплітуд $X^*(m)$ утворюють відповідно *дійсний і уявний спектри* періодичного коливання. Якщо комплексну амплітуду (9.9) подати в експоненціальній формі

$$X^*(m) = \frac{a_m}{2} \cdot e^{j\varphi_m}, \quad (9.11)$$

то величина a_m буде являти собою амплітуду гармонічної складової з частотою $f_m = m \cdot f$, а φ_m - початкову фазу цієї гармоніки, яка має форму косинусоїди, тобто процес можна також записати в виді

$$x(t) = a_0 + \sum_{m=1}^{+\infty} a_m \cdot \cos(2\pi mft + \varphi_m), \quad (9.12)$$

який, власне, и називають *рядом Фур'є*.

Для дійсних процесів є слушними наступні співвідношення

$$\operatorname{Re}\{X(-m)\} = \operatorname{Re}\{X(m)\}; \operatorname{Im}\{X(-m)\} = -\operatorname{Im}\{X(m)\}, \quad (9.13)$$

тобто дійсна частина спектру є парною функцією частоти, а уявна частина - непарною функцією частоти.

Розклади (9.12) и (9.8) дозволяють розглядати сукупність комплексних амплітуд (9.9) як зображення періодичного процесу в частотній області. Прагнення розповсюдити такий підхід на довільні, у тому числі - неперіодичні процеси привело до введення поняття Фур'є-зображення у відповідності до такого виразу:

$$X(f) = \int_{-\infty}^{+\infty} x(t) \cdot e^{-j(2\pi f)t} dt. \quad (9.14)$$

Цей інтеграл, незважаючи на його зовнішню схожість з виразом (9.9) для комплексних коефіцієнтів ряду Фур'є, досить суттєво відрізняється від них.

По-перше, якщо фізична розмірність комплексної амплітуди збігається з розмірністю самої фізичної величини $x(t)$, то розмірність Фур'є-зображення дорівнює розмірності $x(t)$, помноженій на розмірність часу.

По-друге, інтеграл (9.14) існує (збігається до скінченної величини) лише для так званих «двобічно загасаючих» процесів (тобто таких, які прагнуть до нуля як при $t \rightarrow +\infty$, так й при $t \rightarrow -\infty$). Інакше кажучи, його неможна застосовувати до так званих «стаціонарних» (усталених) коливань.

Обернене перетворення Фур'є-зображення у первісний процес $x(t)$ у цьому випадку визначається інтегралом

$$x(t) = \int_{-\infty}^{+\infty} X(f) \cdot e^{j(2\pi f)t} df, \quad (9.15)$$

який є певним аналогом комплексного ряду Фур'є (9.1).

Вказане значне протиріччя дещо згладжується при чисельних розрахунках, бо у цьому випадку можна мати справу лише з процесами обмеженої тривалості, причому сам процес у заданому діапазоні часу має бути заданий своїми значеннями в обмеженій кількості точок. У цьому випадку інтегрування замінюється підсумовуванням, і замість обчислення інтеграла (9.14) обмежуються обчисленням суми

$$X[(k-1) \cdot \Delta f] = \Delta t \cdot \sum_{m=1}^n x[(m-1) \cdot \Delta t] \cdot e^{-j \cdot 2\pi \cdot (k-1) \cdot (m-1) \cdot \Delta f \cdot \Delta t}. \quad (9.16)$$

Тут, у порівнянні з інтегралом (9.14) здійснені такі заміни:

- неперервний інтеграл наближено замінений обмеженою сумою площ прямокутників, одна з сторін яких дорівнює дискрету часу Δt , з яким подані значення процесу, а друга - миттєвому значенню процесу у відповідний момент часу;
- неперервний час t замінений дискретними його значеннями $(m-1) \cdot \Delta t$, де m - номер точки від початку процесу;

- неперервні значення частоти f замінені дискретними її значеннями $(k-1) \cdot \Delta f$, де k - номер значення частоти, а дискрет частоти дорівнює $\Delta f = \frac{1}{T}$, де T - проміжок часу, на якому заданий процес;
- диференціал dt замінений обмеженим приростом часу Δt .

Якщо позначити дискрет часу Δt через T_s , ввести позначення

$$x(m) = x[(m-1) \cdot \Delta t]; \quad X(k) = X[(k-1) \cdot \Delta f].$$

а також врахувати, що кількість n точок, в яких заданий процес, дорівнює

$$n = \frac{T}{\Delta t} = \frac{T}{T_s} = \frac{1}{\Delta f \cdot \Delta t}, \quad (9.17)$$

то співвідношення (9.15) можна подати у більш зручній формі:

$$X(k) = T_s \cdot \sum_{m=1}^n x(m) \cdot e^{-j \cdot (2\pi/n) \cdot (k-1) \cdot (m-1)}. \quad (9.18)$$

Для забезпечення швидкого прямого і зворотного перетворення Фур'є у MatLAB передбачені дві процедури **fft** і **ifft**, які здійснюють обчислення у відповідності до формул:

$$y(k) = \sum_{m=1}^n x(m) \cdot e^{-j \cdot 2\pi \cdot (m-1) \cdot (k-1) / n}; \quad (9.19)$$

$$x(m) = \frac{1}{n} \sum_{k=1}^n y(k) \cdot e^{j \cdot 2\pi \cdot (m-1) \cdot (k-1) / n}. \quad (9.20)$$

Порівнюючи (9.18) з (9.19), можна зробити висновок, що процедура **fft** відшукує дискретне Фур'є-зображення заданого дискретного у часі процесу $x(t)$, поділене на дискрет часу:

$$y(k) = \frac{X(k)}{T_s}. \quad (9.21)$$

Здійснюючи аналогічну операцію дискретизації співвідношення (9.9) для комплексної амплітуди $X^*(k)$, одержимо

$$\begin{aligned} X^*(k) &= \frac{T_s}{T} \cdot \sum_{m=1}^n x(m) \cdot e^{-j \cdot (2\pi/n) \cdot (k-1) \cdot (m-1)} = \\ &= \frac{1}{n} \cdot \sum_{m=1}^n x(m) \cdot e^{-j \cdot (2\pi/n) \cdot (k-1) \cdot (m-1)} = \frac{y(k)}{n}. \end{aligned} \quad (9.22)$$

З цього випливає, що комплексний спектр розкладу стаціонарного процесу дорівнює поділеному на кількість вимірювань результату застосування процедури **fft** до заданого вектору вимірюного процесу.

Якщо ж прийняти до уваги, що для більшості стаціонарних коливальних процесів саме частотний, амплітудний і фазовий спектри не залежать від тривалості T конкретної реалізації і обраного дискрета часу T_s , то слід дійти також висновку, що для спектрального аналізу стаціонарних процесів найбільш

доцільно застосовувати процедуру **fft**, результат якої ділити потім на кількість точок вимірювань.

Перейдемо до визначення спектральної щільності потужності (СПМ), або, скорочено, просто спектральної щільності (СП). Це поняття визначається як Фур'є-зображення так званої кореляційної функції $R_{12}(\tau)$ і застосовується, головним чином, для двох одночасно спливаючих *стаціонарних* процесів $x_1(t)$ і $x_2(t)$. Взаємна кореляційна функція (ВКФ) двох таких процесів визначається співвідношенням:

$$R_{12}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{+T/2} x_1(t) \cdot x_2(t + \tau) \cdot dt, \quad (9.23)$$

тобто ВКФ є середнім у часі значенням добутку першого процесу на зсунутий відносно нього на час затримки τ другий процес.

Отже, взаємна спектральна щільність (ВСП) двох стаціонарних процесів може бути визначена так:

$$S_{12}(f) = \int_{-\infty}^{+\infty} R_{12}(\tau) \cdot e^{-j(2\pi f)\tau} d\tau. \quad (9.24)$$

При числових розрахунках, коли обидва процеси $x_1(t)$ і $x_2(t)$ задані на певному обмеженому проміжку T часу своїми значеннями у певних n точках, розділених дискретом часу Ts , формулу (9.23) можна трансформувати у таку:

$$R_{12}(l) = \frac{1}{n-l} \sum_{m=1}^{n-l} x_1(m) \cdot x_2(m+l-1), \quad (l=1,2,\dots,n/2); \quad (9.25)$$

або у дещо простіший вираз

$$R_{12}(l) = \frac{2}{n} \sum_{m=1}^{n/2} x_1(m) \cdot x_2(m+l-1), \quad (l=1,2,\dots,n/2), \quad (9.26)$$

а замість (9.24) використовувати

$$S_{12}(k) = Ts \cdot \sum_{l=1}^{n/2} R_{12}(l) \cdot e^{-j \cdot (2\pi/n) \cdot (k-1) \cdot (l-1)}, \quad (k=1,2,\dots,n/2). \quad (9.27)$$

Якщо тепер підставити вираз (9.26) у (9.27) і змінити у ньому порядок підсумовування, то можна дійти до такого співвідношення між ВСП і результатами перетворень процедурою **fft** заданих вимірних значень процесів:

$$S_{12}(k) = Ts \cdot \left\{ \frac{2}{n} y_2(k) \right\} \cdot \bar{y}_1(k); \quad (k=1,2,\dots,n/2), \quad (9.28)$$

де риса вгорі позначає комплексне спряжене відповідної величини.

З врахуванням (9.21) і (9.22) вираз (9.28) можна подати також у вигляді:

$$S_{12}(k) = X_2^*(k) \cdot \bar{X}_1(k). \quad (9.29)$$

З цього випливає, що взаємна спектральна щільність двох процесів при будь-якому значенні частоти дорівнює добуткові значення комплексного спектру другого процесу на комплексно-спряжене значення Фур'є-зображення першого процесу на тій самій частоті.

Формули (9.21), (9.22) і (9.28) є основою для обчислень у системі MatLAB відповідно Фур'є-зображення процесу, його комплексного спектру і взаємної спектральної щільності двох процесів.

Щоб применити процедуру **fft** як перетворення процесу, поданого у часовій області, у його подання в частотній області, слід зробити наступне:

- по заданому значенню дискрету часу T_s розрахувати величину F_{\max} діапазону частот (в герцах) за формулою:

$$F_{\max} = 1/T_s; \quad (9.30)$$

- по заданій тривалості заданого процесу T розрахувати дискрет частоти df за формулою:

$$df = 1/T; \quad (9.31)$$

- за обчисленими даними сформувати вектор значень частот, в яких буде обчислене Фур'є-зображення.

Останнє простіше (але не найбільш правильно) зробити у такий спосіб:

$$f1=0 : df : F_{\max}. \quad (9.32)$$

В результаті застосування процедур **fft** буде одержане подання процесу у частотній області. Зворотна процедура **ifft**, якщо її застосувати до результатів першого перетворення, дає можливість відновити первісний процес у часовій області.

Однак процедура **fft** не дає безпосередньо Фур'є-зображення процесу. Щоб отримати Фур'є-зображення, потрібно зробити наступне:

- до результатів дії процедури **fft** застосувати процедуру **fftshift**, яка переставляє місцями першу і другу половини одержаного вектора;
- перебудувати вектор частот за алгоритмом

$$f = -F_{\max}/2 : df : F_{\max}/2. \quad (9.33)$$

Фур'є-зображення прямокутного імпульса

Сформуємо процес, який складається з поодинокого прямокутного імпульса. Задамо дискрет часу $T_s=0.01c$, тривалість процесу $T=100c$, амплітуду імпульса $A=0.75$ і його ширину $w=0.5c$:

```
Ts=0.01; T=100; A=0.75; w=0.5;
t=0 : Ts : T;
y = A*rectpuls(t, w);
plot(t(1:100),y(1:100)), grid, set(gca,'FontSize',12),
title('Процес з поодиноким прямокутним імпульсом');
xlabel('Час (c)'); ylabel('Y(t)')
```

Результат показаний на рис. 9.60.

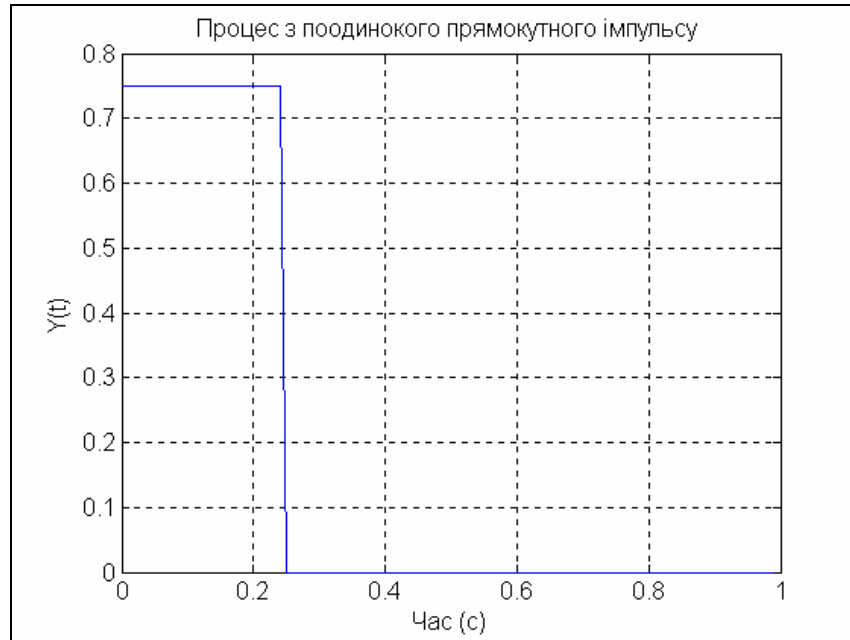


Рис. 9. 60. Поодинокий прямокутний імпульс

Застосуємо до вектора y процедуру **fft** і побудуємо графік залежності модуля результату від частоти. При цьому графіки у частотній області зручніше виводити за допомогою процедури **stem** (див рис. 9.61):

```
x=fft(y); df=1/T; Fmax=1/Ts; f=0 : df : Fmax; a=abs(x);
stem(f,a), grid, set(gca,'FontSize',12),
title('Модуль FFT-перетворення прямокутного імпульсу ');
xlabel('Частота (Гц)'); ylabel('Модуль')
```

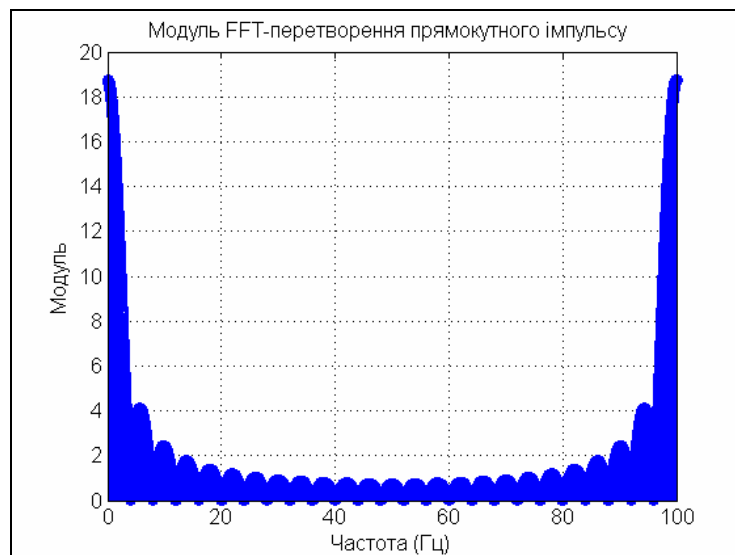


Рис. 9. 61. Модуль FFT-перетворення прямокутного імпульсу

Тепер побудуємо графік модуля Фур'є-зображення процесу:

```
xp=fftshift(x); f1=-Fmax/2 : df : Fmax/2; a=abs(xp);
stem(f1,a),grid,set(gca,'FontSize',12),
title('МодульФур'є-зображення прямокутного імпульсу');
xlabel('Частота (Гц)'); ylabel('Модуль')
```

Одержимо результат, наведений на рис. 9.62.

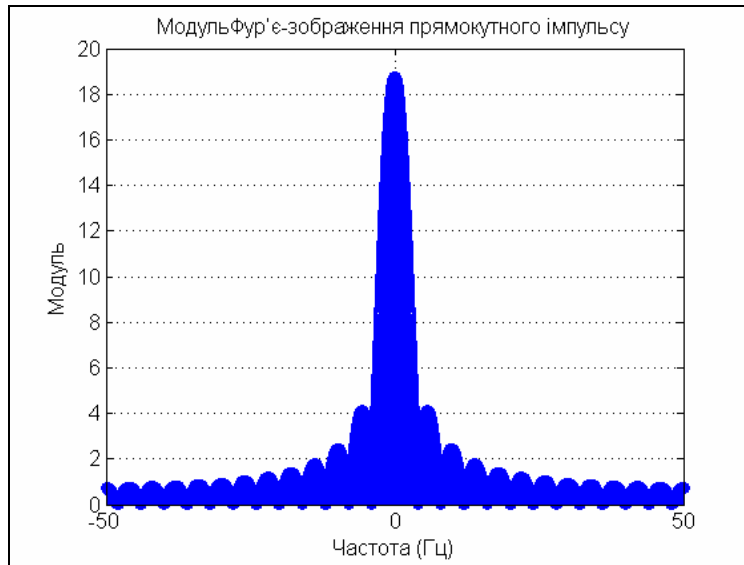


Рис. 9. 62. Модуль Фур'є-зображення прямокутного імпульсу

На останок побудуємо графіки дійсної і уявної частин Фур'є-зображення прямокутного імпульсу:

```
dch=real(xp); mch=imag(xp);
plot(f1,dch,'.',f1,mch), grid, set(gca,'FontSize',12),
title('Фур'є-зображення прямокутного імпульсу ');
ylabel('Дійсна та Уявна частини'),
xlabel('Частота (Гц)');
legend('Дійсна','Уявна',0)
```

Вони подані на рис. 9.63.

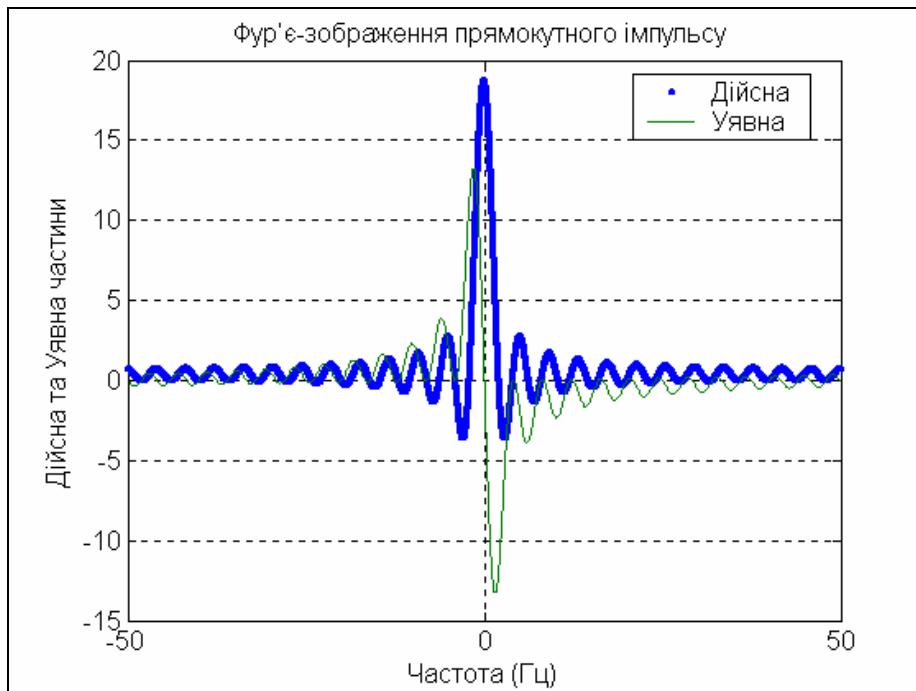


Рис. 9. 63. Дійсна й уявна частини Фур'є-зображення прямокутного імпульсу

Фур'є-зображення полігармонічного процесу

Розглянемо приклад тричастотних гармонічних коливань - с частотою $1/\pi$, 1 та 3 Гц та амплітудами відповідно 0.6, 0.3 та 0.7:

$$y(t) = 0.6 \cdot \cos(2t) + 0.3 \cdot \sin(2\pi \cdot t) + 0.7 \cdot \cos(6\pi \cdot t + \pi/4).$$

Знайдемо Фур'є-зображення цього процесу і виведемо графіки самого процесу, модулю його Фур'є-зображення, а також дійсну і уявну частини:

```
Ts = 0.02; T = 100; t = 0 : Ts : T;
Y = 0.6*cos(2*t)+0.3*sin(2*pi*t)+0.7*cos(6*pi*t+pi/4);
plot(t,Y), grid, set(gca,'FontSize',12),
title(' Тричастотний полігармонічний процес ');
xlabel('час (с)'); ylabel('Y(t)')
```

Графік процесу показаний на рис. 9.64.

Відшукуємо модуль Фур'є-зображення цього процесу:

```
df = 1/T; Fmax = 1/Ts; dovg=length(t);
f = - Fmax/2 : df : Fmax/2;
X = fft(Y); Xp = fftshift(X); A = abs(Xp);
s1 = dovg/2 - 400; s2 = dovg/2 + 400;
stem(f(s1:s2),A(s1:s2)), grid,
set(gca,'FontSize',12),
title('Модуль Фур'є-зображення полігармонічного процесу');
xlabel('Частота (Гц)'); ylabel('Модуль')
```

Результат поданий на рис. 9.65.

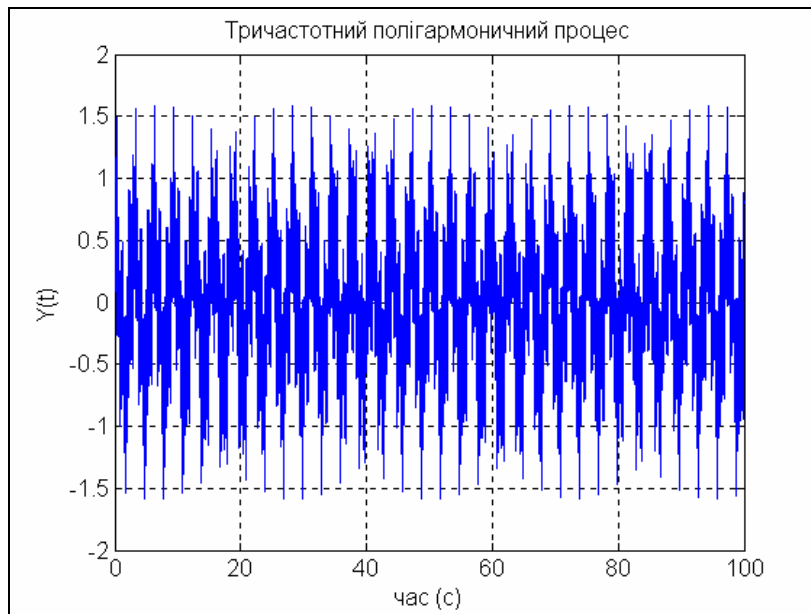


Рис. 9. 64. Графік тричастотного полігармонічного процесу

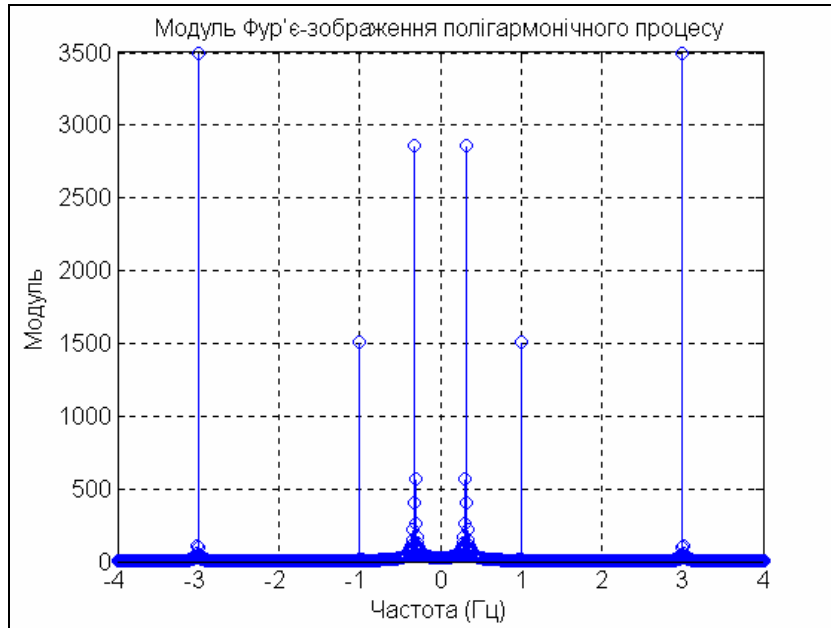


Рис. 9. 64. Модуль Фур'є-зображення полігармонічного процесу при $T_s=0.01$ с

Якщо змінити дискрет часу на $T_s=0.02$, одержимо результат, зображений на рис. 9.65.

Як бачимо, результат Фур'є-перетворення у значній мірі залежить від величини дискрета часу і мало що говорить про амплітуди гармонічних складових. Це обумовлено різницею між визначеннями Фур'є-зображення і комплексного спектра. Тому для незагасаючих (усталених, стаціонарних) коливань будь-якого виду значно зручніше відшукувати не Фур'є-зображення, а його величину, поділену на кількість точок у реалізації. У попередньої частини програми це є еквівалентним заміні оператора $X=fft(Y)$ на $X = fft(Y)/dovg$, де $dovg$ - довжина вектора t .

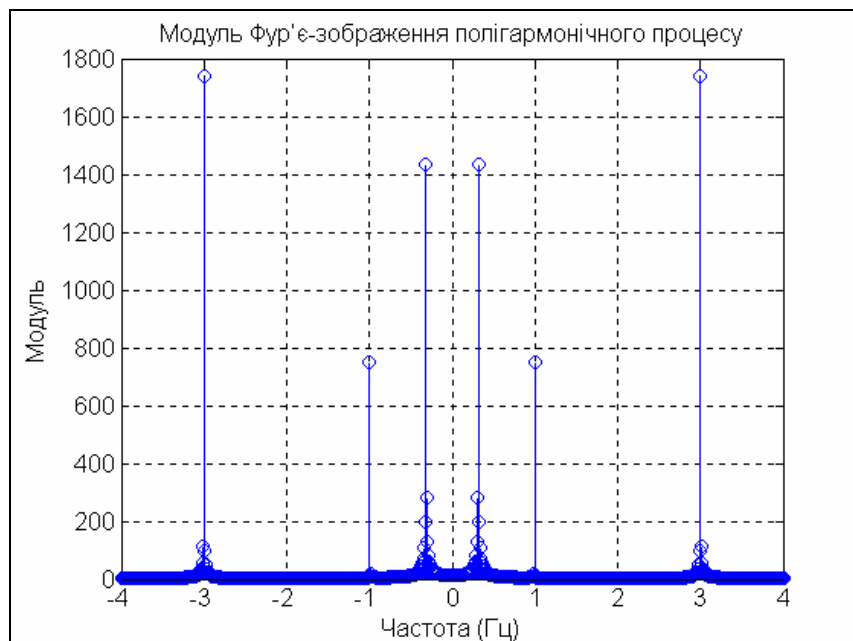


Рис. 9. 65. Модуль Фур'є-зображення полігармонічного процесу при $T_s=0.02$ с

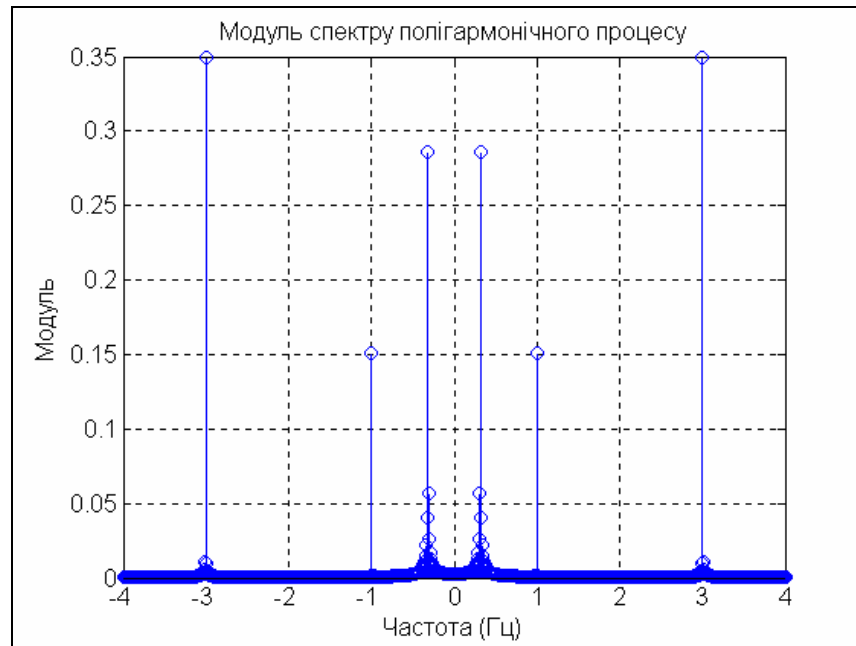


Рис. 9. 66. Модуль спектру полігармонічного процесу

В результаті виходить модуль комплексного спектру (рис. 9.66), який повністю відповідає коефіцієнтам комплексного ряду Фур'є.

Виділимо дійсну і уявну частини комплексного спектру:

```
dch = real(Xp);    mch = imag(Xp);
s1 = dovg/2 - 400; s2 = dovg/2 + 400;
subplot(2,1,1), plot(f(s1:s2),dch(s1:s2)), grid,
set(gca,'FontSize',12),
title('Комплексний спектр полігармонічних коливань');
ylabel('Дійсна частина');
subplot(2,1,2), plot(f(s1:s2),mch(s1:s2)), grid,
set(gca,'FontSize',12),
xlabel('Частота (Гц)'); ylabel('Уявна частина')
```

По одержаних графиках (рис. 9.67) можна судити не тільки про частоти і амплітудах, а й про початкові фази окремих гармонічних складових.

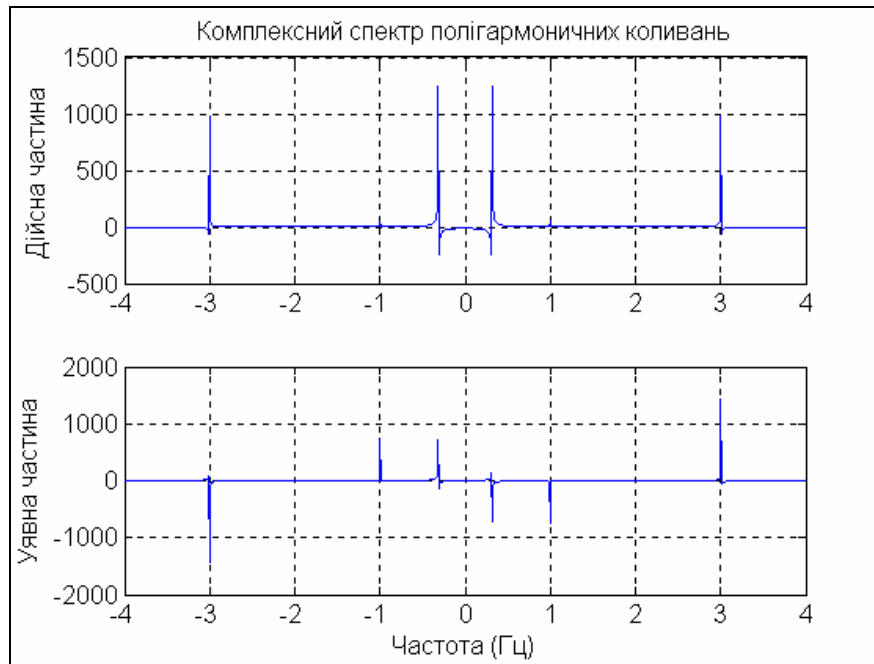


Рис. 9.67. Комплексний спектр полігармонічних коливань

Фур'є-зображення випадкового процесу

На останок розглянемо Фур'є-перетворення випадкового стаціонарного процесу. Сформуємо процес у вигляді білого гауссового шуму з кроком у часі 0.01 і тривалістю у 100 с, створимо формуючий фільтр, «пропустимо» через нього білий шум і результат виведемо на рис. 9.68:

```
Ts=0.01; T=100; t=0:Ts:T;
x1=randn(1,length(t));
om0=2*pi; dz=0.05; A=1; oms=om0*Ts;
a(1)=1+2*dz*oms+oms^2; a(2)=-2*(1+dz*oms); a(3)=1; b(1)=A^2*dz*oms^2;
y1=filter(b,a,x1);
plot(t,y1),grid, set(gca,'FontSize',12)
title('Процес на виході фільтра (T=1; dz=0.05, Ts= 0.01)');
xlabel('Час (с)'); ylabel('Y1(t)')
```

Обчислимо Фур'є-зображення (ФЗ) для процесу-шуму з врахуванням заваження, зробленого для усталених процесів і побудуємо на рис. 9.69 графіки модуля ФЗ і спектральної щільності потужності (СПМ):

```
% Формування масиву частот
df = 1/T; Fmax = 1/Ts; f = -Fmax/2 : df : Fmax/2; dovg = length(f);
% Розрахунок зкорегованих масивів Фур'є-зображень
Fu1 = fft(x1)/dovg; Fu2 = fft(y1)/dovg;
Fu1p = fftshift(Fu1);Fu2p = fftshift(Fu2);
% Формування масивів модулів ФЗ
A1 = abs(Fu1p); A2 = abs(Fu2p);
```

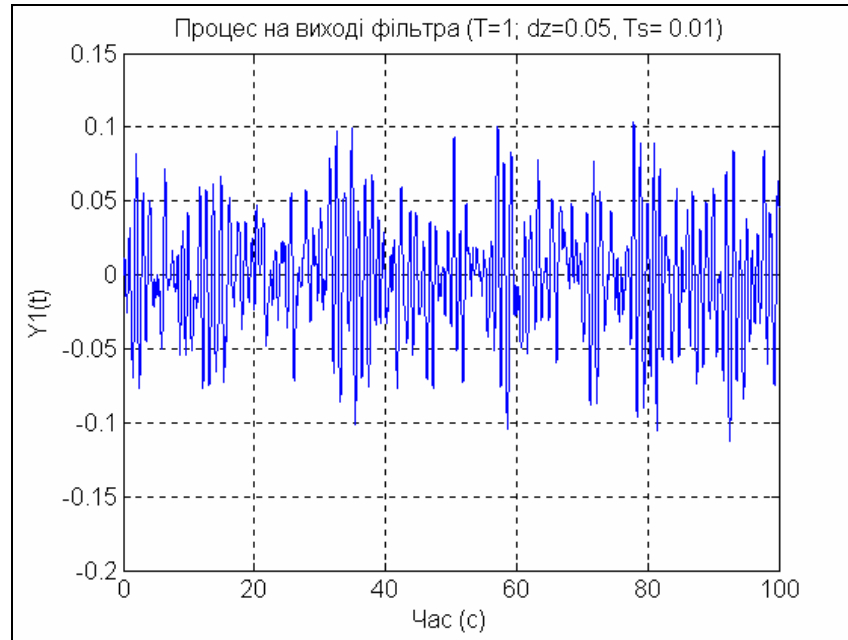


Рис. 9. 68. Випадковий нормальний процес з переважною частотою 1 Гц

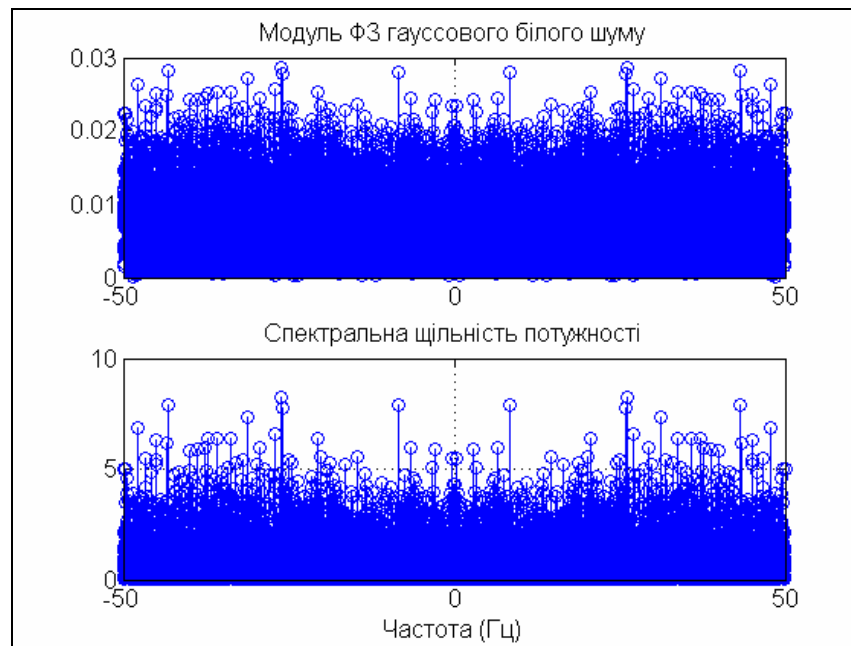


Рис. 9. 69. Модуль Фур'є-зображення і СПМ білого шуму

```
% Обчислення спектральних щільностей потужності
S1 = Fu1p.*conj(Fu1p)*dovg; S2 = Fu2p.*conj(Fu2p)*dovg;
% Виведення графіків білого шуму
subplot(2,1,1); stem(f,A1),grid,
set(gca,'FontSize',12)
title('Модуль ФЗ гауссового білого шуму');
subplot(2,1,2); stem(f,S1), grid,
set(gca,'FontSize',12)
title('Спектральна щільність потужності');
xlabel('Частота (Гц)');
```

Розглядаючи рис. 9.69, можна впевнитися, що спектральна щільність практично однакова за величиною в усьому діапазоні частот, чим й обумовлено назву процесу - "білий шум".

Аналогічну процедуру проведемо і з "профільтрованим" процесом (рис. 9.70):

```
% Виведення графіків профільованого процесу
c1 = fix(dovg/2)-200, c2 = fix(dovg/2)+200, length(f)
subplot(2,1,1); stem(f(c1:c2),A2(c1:c2)),grid,
set(gca,'FontSize',12)
title('Модуль ФЗ випадкового стаціонарного процесу');
subplot(2,1,2); stem(f(c1:c2),S2(c1:c2)),grid,
set(gca,'FontSize',12)
title('Спектральна щільність потужності');
xlabel('Частота (Гц)');
```

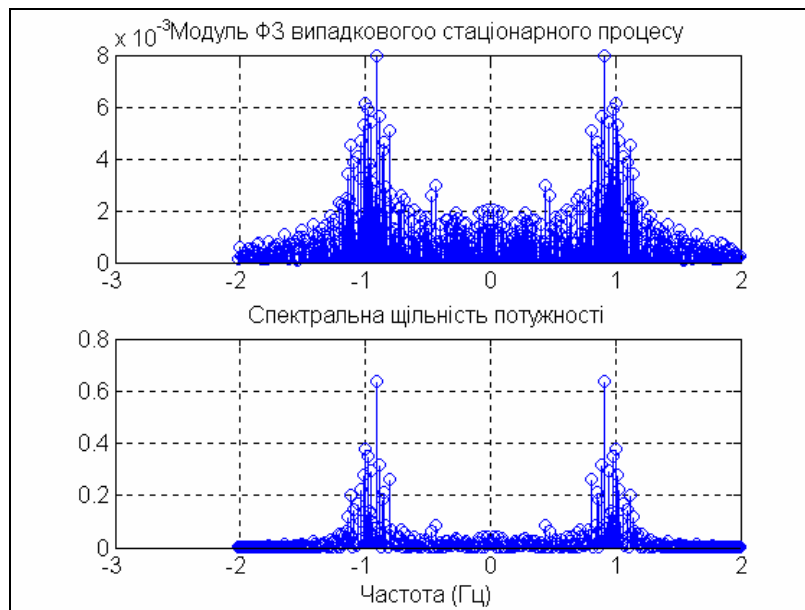


Рис. 9. 70. Модуль ФЗ і СПМ випадкового процесу з переважною частотою 1 Гц

Проводячи ці обчислення ще раз з новою тривалістю процесу $T=20$ с, можна наочно впевнитися, що величини ФЗ і СПМ практично при цьому не змінюються.

В Signal Processing Toolbox передбачено окрему спеціальну процедуру *psd*, яка дозволяє одразу знаходити СП сигналу. Звернення до неї має вид

```
[S,f] = psd(x, nfft, Fmax),
```

де x - вектор заданих значень процесу, $nfft$ - кількість елементів вектора x , які оброблюються процедурою *fft*, $F_{max} = 1/T_s$ - значення частоти дискретизації сигналу, S - вектор значень СП сигналу, f - вектор значень частот, яким відповідають знайдені значення СП. У загальному випадку довжина останніх двох векторів дорівнює $nfft/2$.

Наведемо приклад застосування процедури *psd* для відшукування СП попереднього випадкового процесу з переважною частотою 1 Гц:

```
[C,f] = psd(y1,dovg,Fmax); stem(f(1:200),C(1:200)),grid,
```

`title('Спектральна щільність потужності'); xlabel('Частота (Гц)');`
 В результаті одержимо картину, подану на рис. 9.71

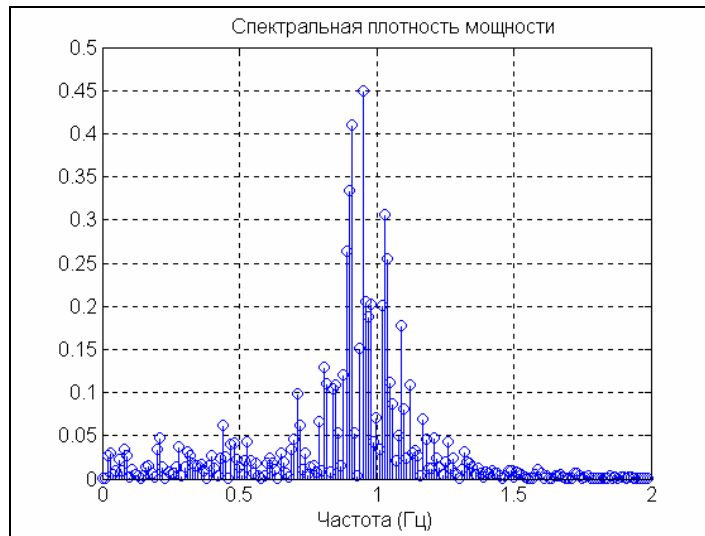


Рис. 9. 71. СПМ одержана застосуванням процедури PSD

Безпосереднє звернення виду `psd(y1,dovg,Fmax)` приводить до появи графіка, поданого на рис. 9. 72.

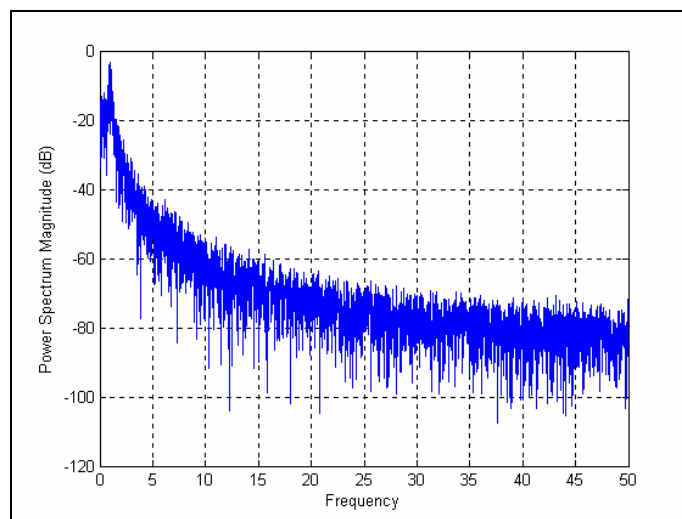


Рис. 9. 72. Графік СПМ одержаний процедурою PSD

Група функцій `xcorr` обчислює оцінку взаємною кореляційної функції (ВКФ) двох послідовностей x і y . Звернення $c = \text{xcorr}(x,y)$ обчислює і видає вектор c довжини $2N-1$ значень ВКФ векторів x і y довжини N . Звернення $c = \text{xcorr}(x)$ дозволяє обчислити АКФ (автокореляційну функцію) послідовності, заданої у векторі x .

Обчислимо АКФ випадкового процесу, сформованого раніше:

```
R=xcorr(y1); tau=-10+Ts : Ts : 10; lt=length(tau);
s1r=round(length(R)/2)-lt/2; s2r=round(length(R)/2)+lt/2-1;
```

```
plot(tau,R(s1r:s2r)),grid
set(gca, 'FontSize', 12)
title('АКФ випадкового процесу') , xlabel('Запізнення (с)')
```

На рис. 9.73 подано результат застосування процедури *xcorr*.

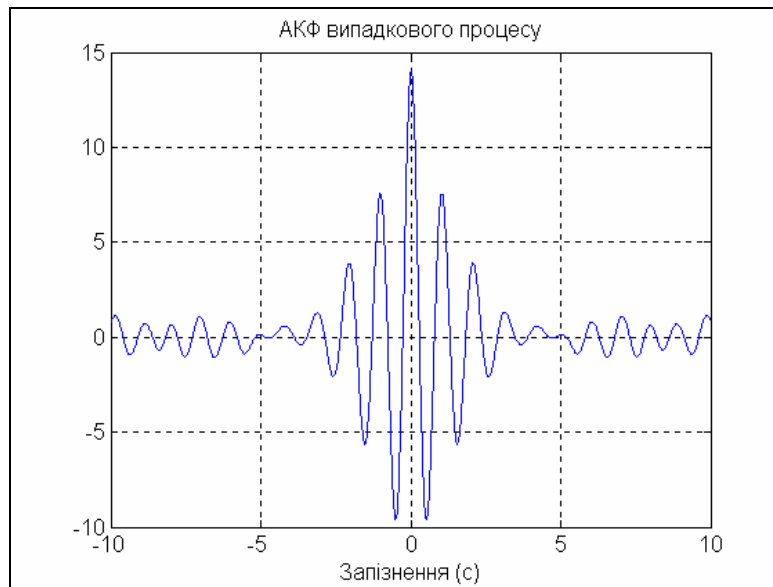


Рис. 9. 73. Автокореляційна функція випадкового процесу, одержана процедурою XCORR

9.6. Моделювання процесів у динамічних системах

Динамічні системи - це системи, поведження яких можна задовільно описати за допомогою звичайних диференційних рівнянь, незалежним аргументом в яких є єдина величина, зазвичай – час.

Пакет візуального програмування *Simulink*, що входить у склад сучасної комплектації комп'ютерної системи Matlab, дозволяє здійснювати досліджування (моделювання у часі) поведження динамічних лінійних і нелінійних систем, причому складання «програми» і введення характеристик досліджуваних систем здійснювати у діалоговому режимі, шляхом графічного "збирання" на екрані дисплею схеми з'єднань елементарних (стандартних або користуваческих) блоків. В результаті такого складання виходить модель досліджуваної системи, яку у подальшому називатимемо S-моделлю і яка зберігається у файлі з розширенням **.mdl**. Такий процес утворення обчислювальних програм прийнято називати візуальним програмуванням.

Створення моделей в пакеті Simulink ґрунтується на використанні технології Drag-and-Drop (*Перетягни і залиш*). Як «кирпичики» при побудові S-моделі використовуються візуальні блоки (модулі), які зберігаються в бібліотеках Simulink. S-модель може мати ієрархічну структуру, тобто складатися з моделей більш низького рівня, причому кількість рівнів ієрархії практично не обмежена. У процесі моделювання є можливість спостерігати за процесами, які відбуваються в системі. Для цього використовуються спеціальні блоки («оглядові вікна»), що входять до складу бібліотеки Simulink. Бібліотека

Simulink може бути поповнена користувачем за рахунок розробки власних блоків.

Основою пакету Simulink, його ядром є бібліотека SIMULINK, в якій зосереджені усі головні засоби, графічні блоки і програми, що дозволяють складати моделі складних динамічних систем, які описуються складними нелінійними диференційними рівняннями.

Складання блок-схеми S-моделі полягає у тому, що графічні зображення обраних блоків за допомогою миші перетягаються з вікна розділу бібліотеки у вікно блок-схеми, а потім виходить одних блоків у вікні блок-схеми з'єднуються зі входами інших блоків також за допомогою миші. Технологія перетягування зображення блоку є такою: курсор миші потрібно встановити на зображенні обраного блоку у вікні розділу бібліотеки, потім натиснути ліву клавішу миші і, не відпускаючи її, пересунути курсор на поле блок-схеми, після чого відпустити клавішу. Аналогічно здійснюються з'єднання лініями виходів одних блоків зі входами інших блоків у блок-схемі: курсор миші підводять до потрібного виходу певного блоку (при цьому курсор має набути форми хрестика), натискають ліву клавішу і, не відпускаючи її, курсор переміщують до потрібного входу іншого блоку, а потім відпускають клавішу. Якщо з'єднання здійснено вірно, на вході останнього блоку виникне зображення чорної затушованої стрілки.

Будь-яка блок-схема S-моделі обов'язково має містити один чи кілька блоків-джерел, що генерують сигнали, які, власне, і забезпечують "рух" моделюваної системи, та один чи кілька блоків-приймачів, які дозволяють одержати інформацію про вихідні сигнали цієї системи (побачити результати моделювання). Блоки-джерела містяться у розділі **Sources** бібліотеки Simulink і описані у п. 9.3. Блоки-приймачі зосереджені у розділі **Sinks** цієї бібліотеки (див. п. 9.2).

У випадку утворення моделі динамічної системи у блок-схемі має бути хоча б один блок з розділу **Continuous** бібліотеки Simulink (див. п. 9.2).

Розглянемо блоки розділу **Sinks**, бо саме вони забезпечують візуалізацію результатів, які одержують при моделюванні, і без них неможливо проконтролювати правильність роботи того чи іншого блоку або моделі у цілому.

Склад розділу **Sinks** зображений на рис. 9.3 (п. 9.2). З його розгляду випливає, що в ньому містяться три групи блоків:

1) блоки, які при моделюванні відіграють роль оглядових вікон; до них відносяться:

Scope	блок з одним входом, який виводить у графічне вікно графік залежності величини, що подається на його вхід, від модельного часу;
Floating Scope	блок з одним входом і з аналогічними функціями;
XYGraph	блок з двома входами, який забезпечує побудову графіка залежності однієї моделюваної величини (другий згори вхід) від іншої (першій вхід);
Display	блок з одним входом, призначений для відображення чисельних значень

вхідної величини;

2) блоки для пересилання і зберігання результатів:

Out вихідний порт для виведення результатів назовні моделі;
Terminator (заглушка) порт для виведення результату «у нікуди»;
To File блок, що забезпечує зберігання результатів моделювання на жорсткому диску у виді MAT-файла (з розширенням .mat);
To Workspace блок, який зберігає результати у робочому просторі Matlab;

3) блок керування моделюванням - **Stop Simulation**, який дозволяє переривати моделювання при виконанні тих чи інших умов; блок спрацьовує тоді, коли до його входу надходить ненульовий сигнал.

Блок Scope

Цей блок дозволяє у процесі моделювання спостерігати, як змінюються у часі сигнали (процеси), які цікавлять дослідника.

Для настроювання параметрів блоку необхідно *після встановлення зображення блоку у вікно блок-схеми подвійно клацнути мишою на цьому зображенні*. В результаті на екрані з'явиться віно Scope (рис. 9.74).

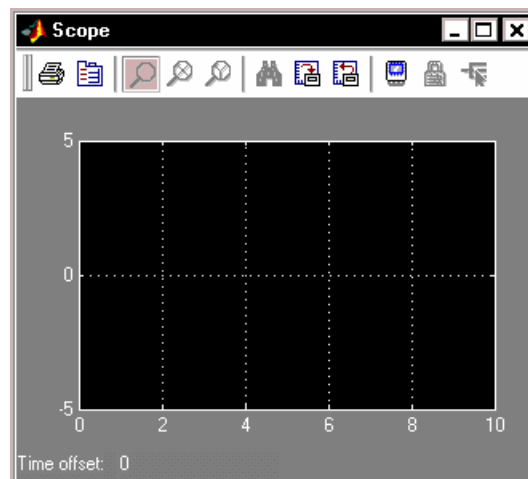


Рис. 9. 74. Вікно блоку Scope

Вздовж горизонтальної осі відкладаються значення модельного часу, а вздовж вертикальної - значення вихідної величини, що відповідають цим моментам часу. Якщо вхідна величина блоку **Scope** є вектором, у вікні відбудовуються графіки змінювання усіх елементів цього вектора, тобто стільки кривих, скільки елементів у вхідному векторі, причому кожна – власного кольору. Одночасно у вікні може відображатися до 30 кривих.

Продемонструємо роботу блоку **Scope** на простішому прикладі.

Перетягнемо у вікно майбутньої блок-схеми з вікна розділу **Sources** блок **Sine Wave** (Синусоїдальна хвиля), а з вікна розділу **Sinks** – блок **Scope** і з'єднаємо вихід першого блоку зі входом другого. Одержимо схему, показану на рис. 9.75.

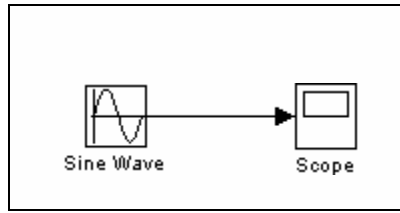


Рис. 9. 75. Найпростіша блок-схема з блоком Scope

Викличемо у вікні цієї блок-схеми **Simulation** ► **Start**, а потім двічі клацнемо на зображенні блока **Scope**. На екрані виникне графічне вікно цього блоку з зображенням графіка змінювання у часі гармонічно змінюваного сигналу (рис. 9. 76).

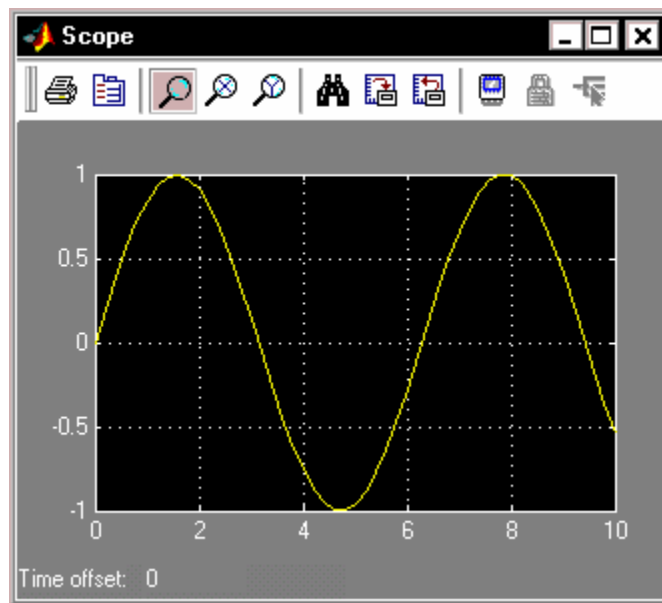


Рис. 9. 76. Вікно Scope із зображенням синусоїди

Блок XY Graph

Цей блок також є оглядовим вікном. На відміну від **Scope**, він має два входи: на першій (верхній) подається сигнал, значення якого відкладаються вздовж горизонтальної осі графіка, а на другий (нижній) - вздовж вертикальної осі.

Якщо перетягнути цей блок на поле блок-схеми, а потім на зображенні його двічі клацнути мишою, то на екрані виникне вікно налаштування блоку (рис. 9.77), яке дозволяє встановити межі змінювань обох вхідних величин, всередині яких буде побудований графік залежності другої величини від першої.

Наведемо приклад використання блоку **XY Graph**. Для цього перетягнемо у вікно блок-схеми зображення цього блоку, а з вікна **Library:Simulink/Sources** - два блоки-джерела **Clock** та **Sine Wave**. З'єднаємо виходи блоків-джерел із входами блоку **XY Graph**. Одержимо блок-схему, наведену на рис. 9.78.

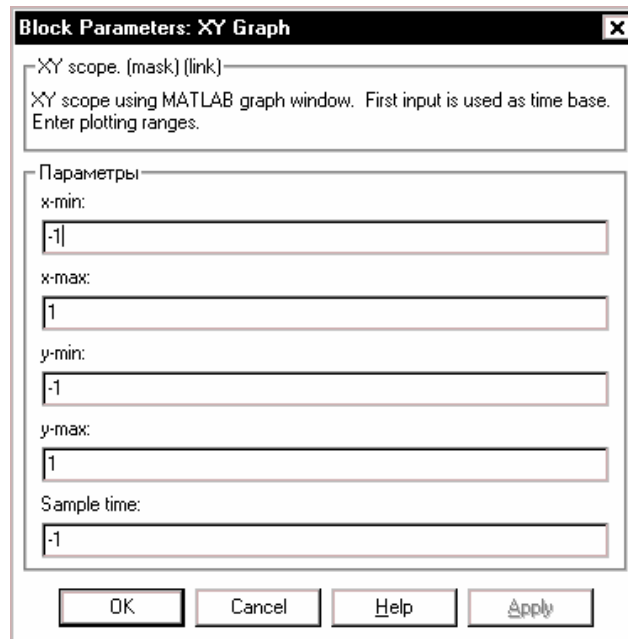


Рис. 9. 77. Вікно налаштування блоку *XY Graph*

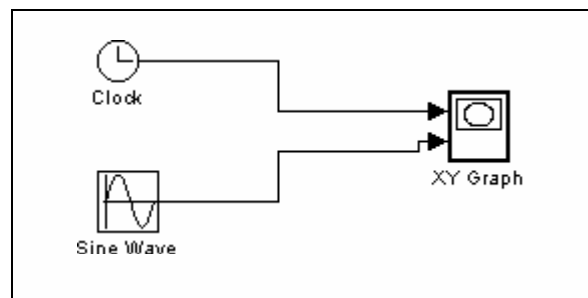


Рис. 9. 78. Блок-схема перевірки роботи блоку *XY Graph*

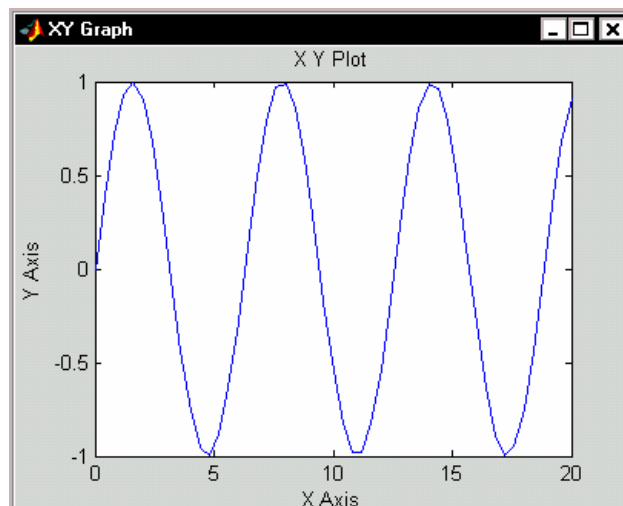


Рис. 9. 79. Оглядове вікно блоку *XY Graph* з графіком синусоїди

Перш ніж запустити цю схему на моделювання, необхідно налагодити блок **XY Graph**, вводячи в його вікні налаштування (рис. 9.77) очікувані діапазони змінювання величини x (у нашому випадку – часу: $x\text{-min}=0$, $x\text{-$

max=10) і y (синусоїдального сигналу: $y_{\min} = -1$, $y_{\max} = 1$). Зазначимо, що у випадку використання блоку **Scope** у введенні діапазонів змінювання величин не має потреби, - вони встановлюються автоматично.

Викликаючи **Simulation ► Start**, одержимо на екрані оглядове вікно **XY Graph**, і у ньому буде побудоване зображення, подане на рис. 9.79.

Блок To File

Блок забезпечує запис значень величини, що подана на його вхід, у MAT-файл даних для використання їх у подальшому в інших S-моделях або програмах.

Блок має такі параметр настроювання (див. рис. 9.80):

- Filename* Ім'я MAT-файла, в який будуть записуватися значення вхідної величини (за замовчуванням - *untitled.mat*); ім'я файла виводиться на зображенні блока у блок-схемі;
- Variable name* Ім'я змінної, за яким можна буде звертатися до даних, записаних у файлі (для того, щоб переглянути або змінити їх у командному вікні MatLAB); за замовчуванням використовується системне ім'я *ans*;
- Decimation* Проріджування (через скільки кроків часу) запису даних у файл;

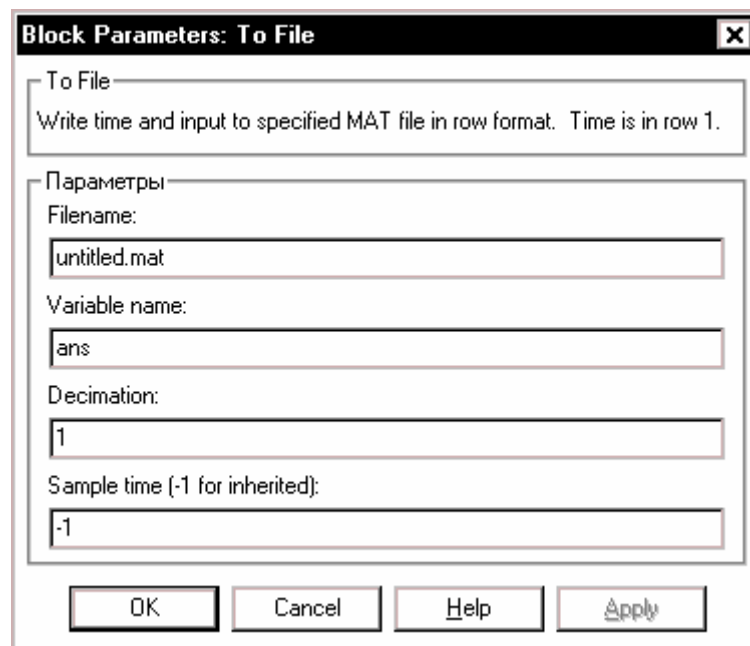


Рис. 9. 80. Вікно настроювання блоку **To File**

Слід зазначити, що значення даних, які подаються до входу блоку записуються у вихідну змінну (наприклад, *ans*) у такий спосіб: перший рядок матриці утворюють значення відповідних моментів модельного часу; другий рядок містить відповідні значення першого елемента вхідного вектора, третій рядок - значення другого елемента вектора і т.д. В результаті записується матриця розміром $(k+1)*N$, де k - кількість елементів вхідного вектора, а N - кількість точок вимірювання (або кількість моментів часу, у які здійснені вимірювання).

Блок To Workspace

Блок призначений для зберігання даних у робочому просторі системи MatLAB. Дані зберігаються у виді матриці розміром ($N \times k$), структура якої відрізняється від структури даних в MAT-файлі тим, що: 1) значення величин, які зберігаються, розташовуються по стовпцях, а не по рядках; 2) не записуються значення модельного часу.

Блок має 4 параметри настроювання (див. рис. 9.81):

Variable name	Ім'я, під яким дані зберігаються у робочому просторі (за замовчуванням - <i>simout</i>);
Maximum number of rows	максимально припустима кількість рядків, тобто значень даних, що записуються; за замовчуванням вона задається константою <i>inf</i> , тобто дані реєструються на всьому інтервалі моделювання;
Decimation	(проріджування) має той самий зміст, що й раніше.

Поле Save format (Формат записи) дозволяє обрати один з трьох варіантів запису даних: *Array* (Масив, матриця), *Structure* (Структура) та *Structure with time* (Структура з часом).

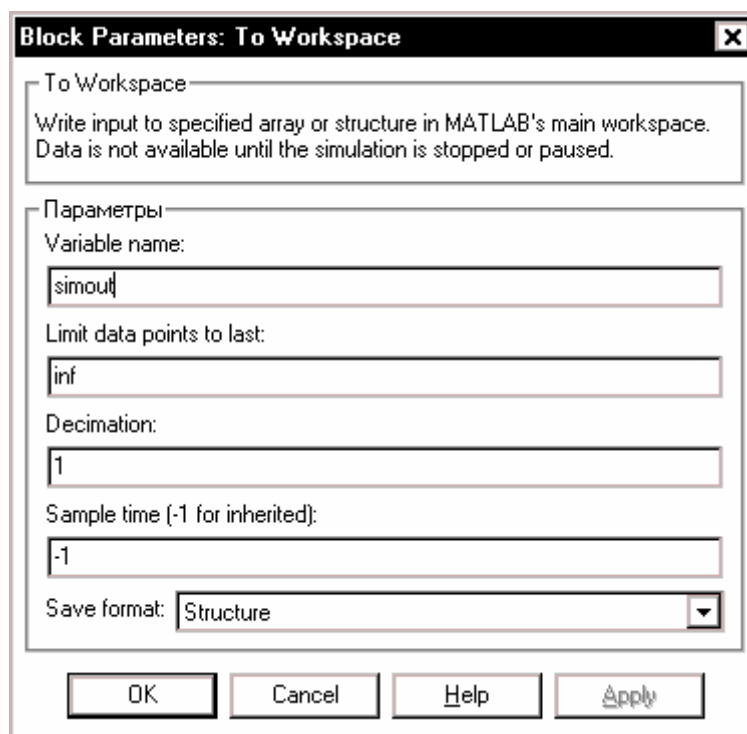


Рис. 9. 81. Вікно настроювання блоку *To Workspace*

Перейдемо до засобів перетворення сигналів, які передбачені пакетом візуального програмування Simulink.

Головні блоки, які забезпечують перетворення сигналів, що надаються до їхніх входів, зосереджені в наступних розділах бібліотеки:

- *Continuous* (Неперервні перетворення), де містяться блоки, які реалізують перетворення у виді лінійних стаціонарних ланок (рис. 9.4);

- **Math Operations** (Математичні операції); тут розташовані блоки, які реалізують певні математичні операції над сигналами, що подаються на їхній вхід (рис. 9.6);

- **Discontinuities** (Розривні перетворення) містить блоки, що реалізують певні типи (люфт, зона нечутливості, сухе тертя, насичення тощо) кусково-лінійні перетворення сигналів (рис. 9.9).

Для утворення чисельних моделей динамічних систем неперервного часу найбільш важливими є наступні блоки:

- **Integrator** – блок інтегратора, який здійснює інтегрування за модельним часом сигналу, що надається до його входу;
- **Derivative** – блок похідної, який здійснює диференціювання за часом вхідного сигналу;
- **State-Space** – блок, що реалізує динамічну ланку шляхом задання матриць її рівнянь у просторі станів;
- **Transfer Fcn** – блок, що реалізує динамічну ланку шляхом задання передатної її функції;
- **Zero-Pole** – блок, що реалізує динамічну ланку шляхом задання векторів її нулів і полюсів.

Ці блоки дозволяють утворити моделі лінійних стаціонарних динамічних систем.

Щоб утворити моделі динамічних систем, в які входять нелінійні або не-стаціонарні ланки, використовуються блоки з розділів **Math Operations** та **Discontinuities**:

- **Product** (Добуток), який здійснює множення або ділення двох сигналів, що подаються на його два входи;
- **Dot Product** (Скалярний добуток) – здійснює множення двох вхідних величин, якщо вони є скалярами, або обчислює суму добутків елементів двох вхідних векторів однакової довжини;
- **Abs** (Модуль) утворює сигнал, який дорівнює абсолютному значенню вхідного сигналу
- **Sign** (Сигнум-функція) утворює вихідний сигнал, який приймає усього три можливих значення +1, 0 і -1 у залежності від того, є значення вхідного сигналу додатним, рівним нулю чи від'ємним;
- **Math Function** (Математична функція) дозволяє сформувати значення вихідного сигналу як одну з математичних функцій вхідного сигналу u з такого набору: $\exp(u)$, $\log(u)$, 10^u , $\log_{10}(u)$, $\text{sqrt}(u)$, $\text{square}(u)$ та інших;
- **Trigonometric Function** (Тригонометрична функція) дозволяє сформувати значення вихідного сигналу як одну з тригонометричних або гіперболічних функцій вхідного сигналу u з такого набору: $\sin(u)$, $\cos(u)$,

- $\tan(u)$, $\text{asin}(u)$, $\text{acos}(u)$, $\text{atan}(u)$, $\text{atan2}(u1,u2)$, $\sinh(u)$, $\cosh(u)$, $\tanh(u)$, $\text{asinh}(u)$, $\text{acohs}(u)$, $\text{atanh}(u)$ та інших;
- **Saturation** (Насичення), **Dead Zone** (Мертва зона), **Rate Limiter** (Обмежувач швидкості), **BackLash** (Люфт), **Relay** (Реле), **Coulombic and Viscous Friction** (Кулонове і в'язке тертя) дозволяють забезпечити кусково-лінійну залежність певного виду вихідної величини від вхідної.

Для забезпечення підсумовування кількох сигналів передбачений блок **Sum**, для забезпечення підсилювання сигналів – блоки **Gain** (Підсилювач) та **Matrix Gain** (Матричний підсилювач), які розташовані у розділі **Math Operations** бібліотеки **Simulink**.

Щоб об'єднати кілька сигналів в один векторний сигнал, потрібно використовувати блок **Mux**, а щоб, навпаки, розділити векторний сигнал на декілька окремих скалярних або векторних меншої довжини, - блок **Demux**, які розташовані у розділі **Signal Routing**. У тому ж розділі містяться блоки, що забезпечують різного роду перемикання сигналів. Такі блоки можуть стати у нагоді для створення складних моделей з перемиканнями сигналів за певних умов.

9.7. Приклади утворення S-моделей

Відзначимо переваги і деякі недоліки моделювання динамічних систем за допомогою пакета **Simulink** у порівнянні з аналогічними дослідженнями за допомогою програм у системі MatLab:

- складання блок-схеми рівнянь руху замість запису тексту процедури правих частин диференціальних рівнянь у формі Коші є значно більш наочним, дозволяє контролювати правильність складання моделі шляхом усвідомлення фізичного змісту окремих блоків і їхніх взаємозв'язків;
- при проведенні процесу моделювання у середовищі **Simulink** зникає потреба в організації самого процесу чисельного інтегрування диференціальних рівнянь і навіть виведення результатів у графічній формі;
- однак форма виведення результатів у графічній формі в **Simulink** виявляється недостатньо гнучкою: неможливо додати власні написи у заголовок і по осях графіка, неможливо встановити сітку координатних ліній у графічні вікна блоків XY Graph; особливо незручним є те, що тут не передбачені засоби виведення текстової інформації на поле графіка, що робить графічне подання безадресним.

Останній недолік є суттєвим. Він може бути усунений існуючими у пакеті **Simulink** засобами. Наприклад, можна записати одержані значення вихідних величин в MAT-файл (посилаючи їх на блок To File), а потім створити і використати програму, яка б здійснювала зчитування даних,

записаних в MAT-файлі, і формування на цій основі графічного зображення у вікні фігури.

Незручністю застосовування оглядового вікна XY Graph є також те, що попередньо потрібно встановлювати діапазони змінювання обох вхідних величин. Якщо ці діапазони встановлені невірні, в оглядовому вікні може взагалі не виникнути ніякого зображення, або виникне такий фрагмент графіка, за яким неможливо зробити правильний висновок про поведінку досліджуваної системи. А при досліджуванні часто неможливо заздалегідь передбачити можливі діапазони змінювання величин, або зробити це занадто складно.

9.7.1. Модель руху планет

Тут розглядається класична задача небесної механіки – визначення руху трьох матеріальних точок, між якими діють сили гравітації (всесвітнього тяжіння) – з точки зору чисельного моделювання його у середовищі *Simulink*.

Нехай існують три ізольовані матеріальні точки з масами відповідно m_1 , m_2 та m_3 . Позначимо радіус-вектори цих точок відносно деякої нерухомої в інерціальному просторі точки O через \mathbf{R}_1 , \mathbf{R}_2 і \mathbf{R}_3 .

Тоді диференціальні рівняння руху цих трьох точок можуть бути записані у наступному векторному виді:

$$\begin{cases} m_1 \frac{d^2 \mathbf{R}_1}{dt^2} = G \cdot \frac{m_1 \cdot m_2}{R_{21}^3} \cdot \mathbf{R}_{21} - G \cdot \frac{m_1 \cdot m_3}{R_{31}^3} \cdot \mathbf{R}_{13}; \\ m_2 \frac{d^2 \mathbf{R}_2}{dt^2} = -G \cdot \frac{m_1 \cdot m_2}{R_{21}^3} \cdot \mathbf{R}_{21} + G \cdot \frac{m_2 \cdot m_3}{R_{32}^3} \cdot \mathbf{R}_{32}; \\ m_3 \frac{d^2 \mathbf{R}_3}{dt^2} = -G \cdot \frac{m_3 \cdot m_2}{R_{32}^3} \cdot \mathbf{R}_{32} + G \cdot \frac{m_3 \cdot m_1}{R_{31}^3} \cdot \mathbf{R}_{13}; \end{cases} \quad (9.34)$$

де позначено $\mathbf{R}_{21} = \mathbf{R}_2 - \mathbf{R}_1$; $\mathbf{R}_{32} = \mathbf{R}_3 - \mathbf{R}_2$; $\mathbf{R}_{13} = \mathbf{R}_1 - \mathbf{R}_3$.

Досліджування рівнянь руху (і, особливо, чисельне) завжди зручніше здійснювати за рівняннями, приведеними у безрозмірний вид, - у цьому випадку скорочується кількість параметрів, від яких залежить розв'язок, тому дослідження стає набагато простішим.

Приведемо рівняння (9.34) до безрозмірної форми. Для цього як базові використаємо такі фізичні величини:

G - гравітаційна стала розмірності $L^3 M^{-1} T^{-2}$ (L – одиниця довжини, M – одиниця маси, T – одиниця часу);

m_1 - маса першого тіла, яке будемо вважати головним; ним зазвичай є найбільш масивне тіло (наприклад, Сонце, яке знаходиться під дією гравітаційного тяжіння Землі (друге, середнє за масою тіло) і Місяця (третє, найменш масивне тіло)); розмірність M ;

R_{210} - початкова відстань другого тіла від першого, розмірністю L .

Тепер введемо безрозмірні величини:

- 1) безрозмірна маса першого тіла, очевидно, буде дорівнювати одиниці;
- 2) безрозмірні маси другого і третього тіл дорівнює

$$\mu_2 = \frac{m_2}{m_1}; \quad \mu_3 = \frac{m_3}{m_1}; \quad (9.35)$$

- 3) безрозмірні довжини радіус-векторів

$$\rho_1 = \frac{R_1}{R_{210}}; \quad \rho_2 = \frac{R_2}{R_{210}}; \quad \rho_3 = \frac{R_3}{R_{210}}; \quad (9.36)$$

- 4) безрозмірні радіус-вектори

$$\mathbf{r}_i = \frac{\mathbf{R}_i}{R_{210}}; \quad \mathbf{r}_{ij} = \frac{\mathbf{R}_{ij}}{R_{210}}; \quad (9.37)$$

- 5) безрозмірний час визначимо формулою

$$\tau = \sqrt{\frac{G \cdot m_1}{R_{210}^3}} \cdot t; \quad (9.38)$$

Остання формула означає, що як одиниця вимірювання часу використовується величина

$$T_o = 2\pi \sqrt{\frac{R_{20}^3}{G \cdot m_1}}. \quad (9.39)$$

У випадку, наприклад, Сонця, Землі і Місяця ця величина дорівнює рокові. Прийнятий безрозмірний час є таким, що *безрозмірний період кругового обертання другого тіла навколо першого дорівнює 2π* .

З врахуванням цього рівняння (9.34) у безрозмірній формі на будуть виду:

$$\begin{cases} \frac{d^2 \mathbf{r}_1}{d\tau^2} = \frac{\mu_2}{\rho_{21}^3} \cdot \mathbf{r}_{21} - \frac{\mu_3}{\rho_{31}^3} \cdot \mathbf{r}_{13}; \\ \frac{d^2 \mathbf{r}_2}{d\tau^2} = -\frac{1}{\rho_{21}^3} \cdot \mathbf{r}_{21} + \frac{\mu_3}{\rho_{32}^3} \cdot \mathbf{r}_{32}; \\ \frac{d^2 \mathbf{r}_3}{d\tau^2} = -\frac{\mu_2}{\rho_{32}^3} \cdot \mathbf{r}_{32} + \frac{1}{\rho_{31}^3} \cdot \mathbf{r}_{13} \end{cases} \quad (9.40)$$

де позначено

$$\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j; \quad \rho_{ij} = |\mathbf{r}_{ij}|. \quad (9.41)$$

Три вектори (9.41) пов'язані між собою співвідношенням

$$\mathbf{r}_{13} + \mathbf{r}_{32} + \mathbf{r}_{21} = 0, \quad (9.42)$$

Рівняння руху зручніше повністю виразити у векторах (9.41), які характеризують положення тіл одне відносно одного. Для цього віднімемо по черзі

рівняння (9.40) одне з іншого. З одержаних трьох рівнянь вилучимо вектор \mathbf{r}_{13} , використовуючи (9.42). Залишаться два рівняння

$$\begin{cases} \frac{d^2 \mathbf{r}_{21}}{d\tau^2} = -\left(\frac{1 + \mu_2}{\rho_{21}^3} + \frac{\mu_3}{\rho_{31}^3}\right) \cdot \mathbf{r}_{21} + \mu_3 \left(\frac{1}{\rho_{32}^3} - \frac{1}{\rho_{31}^3}\right) \cdot \mathbf{r}_{32}; \\ \frac{d^2 \mathbf{r}_{32}}{d\tau^2} = -\left(\frac{\mu_2 + \mu_3}{\rho_{32}^3} + \frac{1}{\rho_{31}^3}\right) \cdot \mathbf{r}_{32} + \left(\frac{1}{\rho_{21}^3} - \frac{1}{\rho_{31}^3}\right) \cdot \mathbf{r}_{21} \end{cases} \quad (9.43)$$

Ці рівняння і покладемо в основу моделювання. При цьому слід зважити на те, що $\rho_{21}(0) = 1$.

При складанні S-моделі будемо використовувати наступні змінні робочого простору:

У формулі	У робочому просторі	Примітка
$\mu_2 = \frac{m_2}{m_1}$	mu2	відношення мас другого і першого тіл
$\mu_3 = \frac{m_3}{m_1}$	mu3	відношення мас третього і першого тіл
$x_{21}(0), y_{21}(0), z_{21}(0)$	X210, Y210, Z210	Початкові безрозмірні координати другого тіла відносно першого
$x_{32}(0), y_{32}(0), z_{32}(0)$	X320, Y320, Z320	Початкові безрозмірні координати третього тіла відносно другого
$V_{21x}(0), V_{21y}(0), V_{21z}(0)$	V21x, V21y, V21z	Початкові безрозмірні проекції вектора швидкості другого тіла відносно першого
$V_{32x}(0), V_{32y}(0), V_{32z}(0)$	V32x, V32y, V32z	Початкові безрозмірні проекції вектора швидкості третього тіла відносно другого

Варіант S-моделі під назвою **GR_3_TEL.mdl** приведений на рис. 9. 82. Вона включає в себе 6 підсистем (створених на ґрунті блоків **Subsystem**)—Uravnenye 21, Uravnenye 32, Prav 21, Prav 32, RV13 и Absolute Coordinates і блок **To File**. Останній забезпечує запис результатів моделювання у MAT-файл під ім'ям GR_3_TEL.mat, в якому вони зберігаються у матриці під ім'ям RV.

Головні дії з чисельного інтегрування диференціальних рівнянь (9.43) зосереджені у підсистемах Uravnenye 21 та Uravnenye 32. Тут (рис. 9. 83) здійснюється подвійне інтегрування прискорень, а також обчислюється поточна відстань (довжина вектора) між другим і першим тілами. Вхідною величиною обох підсистем є вектор, елементи якого являють собою проекції вектора прискорення на осі X, Y і Z. Вихідні величини дві. Одна з них – вектор з трьох елементів, що є поточними проекціями радіус-вектора на ті самі осі. Другий вихід – скалярна величина – довжина цього радіус-вектора.

Блок, який реалізує обчислення довжини вектора за його складовими, створений на основі використання блоку **Fcn** з розділу **User-Defined Functions**.

Примітка.

При написанні арифметичних виразів у вікнах настроювання блоків Simulink використовуються правила М-мови. Однак вказівка індексів для векторів і матриць має здійснюватися у квадратних дужках (а не в круглих, як у М-мові)

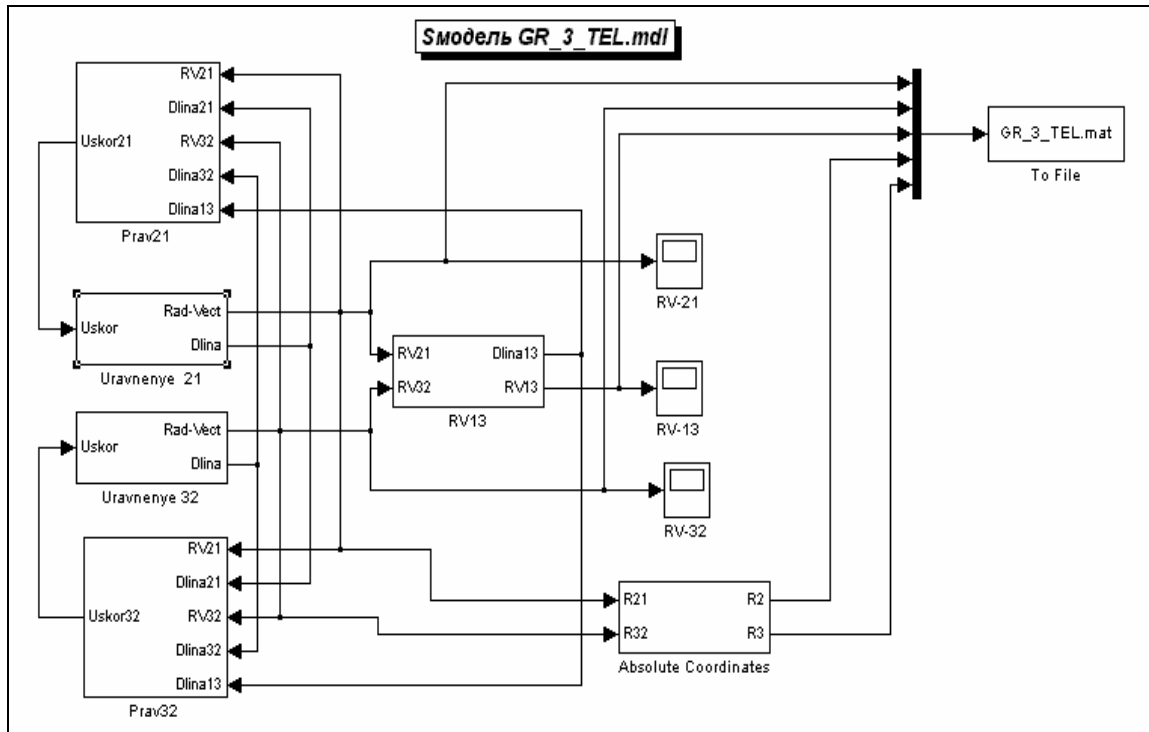


Рис. 9. 82. Блок-схема S-моделі *GR_3_TEL.mdl*

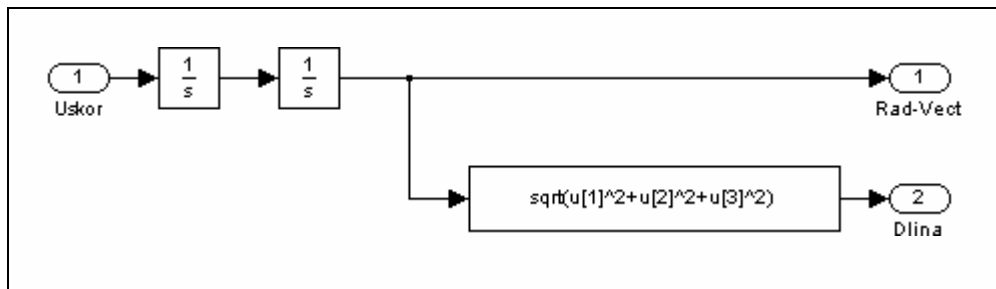


Рис. 9. 83. Блок-схема підсистеми *Urvnenye 21*

Важливою і характерною особливістю цих підсистем, є те, що в них інтегрується не скалярна функція часу, а векторна, яка складається одночасно з трьох скалярних величин. Наприклад, у підсистемі *Urvnenye 21* здійснюється одночасне інтегрування прискорень x''_{21} , y''_{21} і z''_{21} . Досягається це за допомогою лише двох блоків **Integrator**. Для цього достатньо у блоці настроювання інтеграторів ввести як початкову умову не одну величину, а вектор з трьох величин, кожна з яких є початковою умовою відповідного елемента вектора, що подається на вхід інтегратора. Наприклад, вектором початкових умов для першого інтегратора є $[V21x, V21y, V21z]$ (см. рис. 9.84).

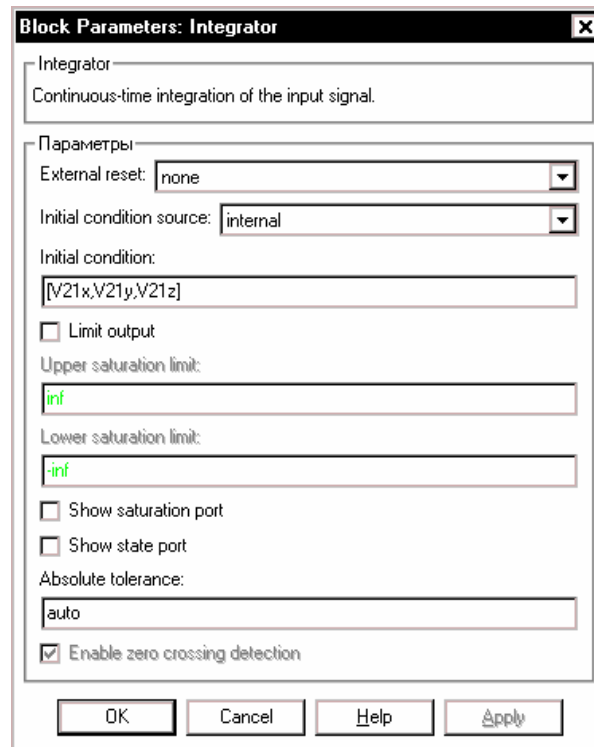


Рис. 9. 84. Вікно налаштування першого блоку *Integrator* підсистеми *Uravnenye 21*

Аналогічно, у другому інтеграторі встановлений вектор $[X210, Y210, Z210]$ початкових умов.

Те саме зроблено у підсистемі *Uravnenye 32*, яка обчислює поточні складові вектора \mathbf{r}_{32} , одержані подвійного інтегрування прискорення \mathbf{r}_{32}'' . Тут в інтеграторах встановлені початкові умови у виді векторів $[V32x, V32y, V32z]$ і $[X320, Y320, Z320]$.

Підсистеми *Prav 21* і *Prav 32* формують праві частини першого і другого рівнянь (9.43), тобто прискорення у виді вектора з трьох елементів. Вихідна величина у цих підсистем – одна. Наприклад, у підсистемі *Prav 21* (рис. 9.85) виходом є вектор *Uskor21*, елементами якого є проєкції прискорення на осі *X*, *Y* та *Z*.

Входами є два обчислених раніше вектори *RV21* і *RV32*, що складається з проєкцій радіус-векторів \mathbf{r}_{21} і \mathbf{r}_{32} , та три скалярні величини, які дорівнює поточним значенням довжин трьох радіус-векторів, що з'єднує три тіла.

В підсистемі *Prav 21* використані три блоки типу **Fcn** для обчислення коефіцієнтів при векторах у правій частині першого рівняння (9.43) і два блоки **Product** для множення вектора на скалярний коефіцієнт.

Блок-схема підсистеми *RV13*, яка обчислює вектор \mathbf{r}_{13} і його довжину за відомими векторами \mathbf{r}_{21} і \mathbf{r}_{32} , наведена на рис. 9.86.

На рис. 9.87 показано блок-схему підсистеми *Absolute Coordinates*. Вона здійснює обчислення абсолютних координат другого і третього тіл відносно центру мас системи трьох тіл.

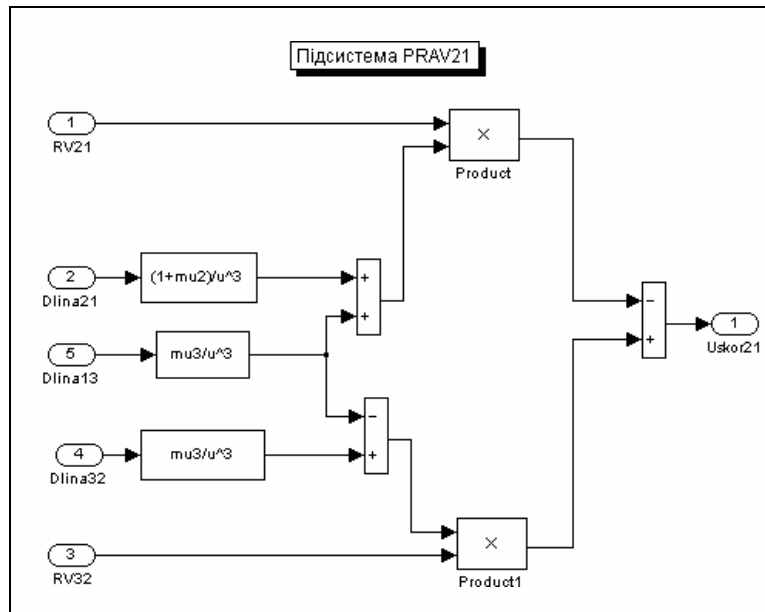


Рис. 9. 85. Блок-схема підсистеми Prav 21

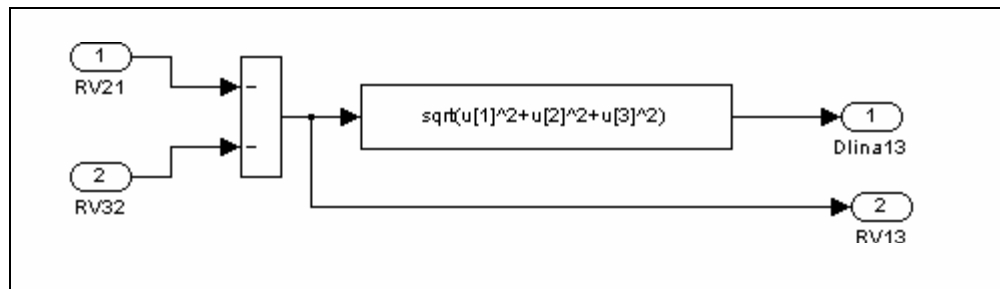


Рис. 9.86. Блок-схема підсистеми RV13

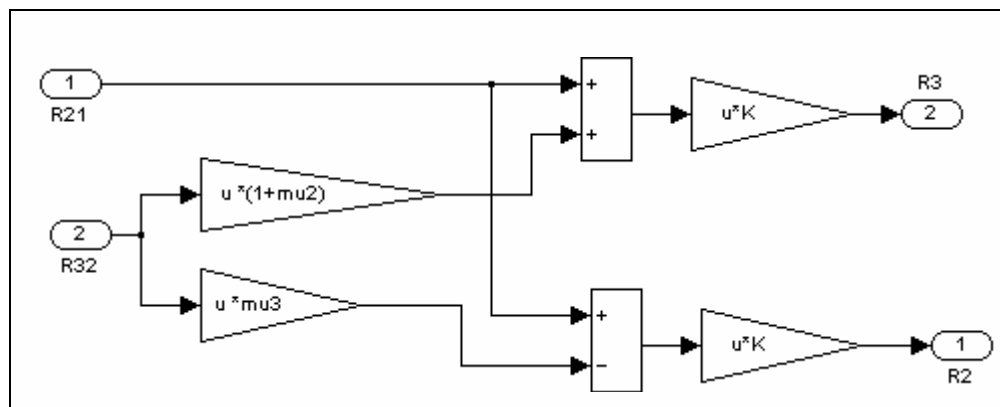


Рис. 9.87. Блок-схема підсистеми Absolute Coordinates

Блок **To File** здійснює запис даних, що подаються на його вхід, у MAT-файл GR_3_TEL.mat. Ці дані записуються на диск у виді матриці під ім'ям RV (рис. 9.88) у такому порядку:

- перший рядок утворює масив значень модельного часу, для яких є обчисленими величини, що надаються до входу блоку (хоча, як видно з рис. 9.81, до входу не надається час);

- другий рядок утворюють значення першого елемента вхідного векторного сигналу (йому відповідає верхній сигнал, що подається до входу блоку **Mux**, вихідний сигнал якого поступає на вхід блоку **To File**), які відповідають моментам часу, записаним у першому рядку;

- решта рядків заповнюються значеннями решти елементів векторного сигналу входу у порядку слідування елементів цього вектора.

Цю обставину слід враховувати при зчитуванні даних з сформованого MAT-файла.

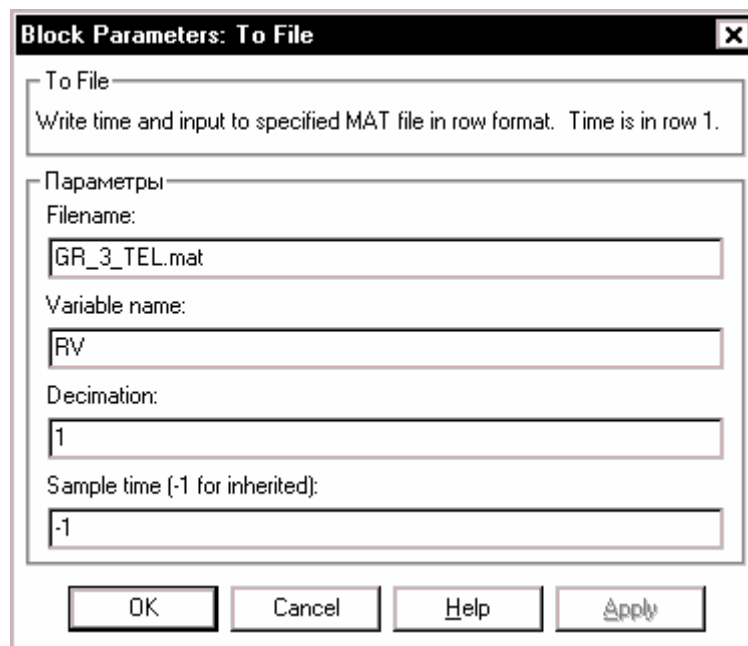


Рис. 9.88. Заповнення вікна настроювання блоку **To File**

Для того, щоб провести моделювання за створеною S-моделлю і оформити графічно одержані потрібно здійснити наступні операції:

- 1) ввести первісні дані у робочий простір;
- 2) запустити S-модель GR_3_TEL.mdl на моделювання; при цьому результати будуть записані у MAT-файл GR_3_TEL.mat;
- 3) завантажити вміст MAT-файла GR_3_TEL.mat у робочий простір, вводячи у командному вікні MatLab команду

load GR_3_TEL

при цьому усі дані, записані у цьому MAT-файлі, будуть записані у робочий простір у вигляді матриці під ім'ям RV розміром $(16 \times n)$, де n – розмір вектору значень часу;

- 4) використовуючи дані матриці RV, забезпечити графічно і текстове оформлення результатів моделювання у графічному вікні MatLab.

Усі ці операції можна автоматизувати, запрограмувавши їх в окремій М-програмі, яку зручно оформити у виді М-файлу *GR_3_TEL_upr* (керуюча програма для S-моделі GR_3_TEL). При цьому викликати S-модель до моделювання з програми можна за допомогою команди

sim('имя_S-модели')

Нижче наводиться текст керуючої програми.

```
% GR_3_TEL_upr
% Програма керування для S-моделі GR_3_TEL
% Лазарев Ю. Ф. 25-01-2004
% 1. Введення даних у робочий простір
mu2=0.1; mu3=0.01;
X210=1; Y210=0; Z210=0;
V21x=0; V21y=1; V21z=0;
X320=0.1; Y320=0.0; Z320=0;
V32x=0; V32y=-1; V32z=0;
% 2. Запуск S-моделі
sim('GR_3_TEL')
% 3. Завантаження MAT-файлу
load GR_3_TEL;
% 4. Присвоювання значень з даних MAT-файлу новим змінним
t=RV(1,:);
X21=RV(2,:); Y21=RV(3,:); Z21=RV(4,:);
X32=RV(5,:); Y32=RV(6,:); Z32=RV(7,:);
X13=RV(8,:); Y13=RV(9,:); Z13=RV(10,:);
X2=RV(11,:); Y2=RV(12,:); Z2=RV(13,:);
X3=RV(14,:); Y3=RV(15,:); Z3=RV(16,:);
n=length(t);
% 5. Побудова графика залежності координат второго тела відносно першого
% від часу
subplot(2,2,1)
plot(t,X21,t,Y21,'--',t,Z21,'. '), grid
title('Движение второго тела относительно первого')
xlabel('Время (безразмерное)')
ylabel('Координаты (безразмерные)')
legend('X','Y','Z',0)
% 6. Побудова графика залежності координат третього тіла відносно другого
% від часу
subplot(2,2,3)
n1=round(n);
plot(t(1:n1),X32(1:n1),t(1:n1),Y32(1:n1),...
'—',t(1:n1),Z32(1:n1),'. '),grid
title('Движение третьего тела относительно второго')
xlabel('Время (безразмерное)')
ylabel('Координаты (безразмерные)')
legend('X','Y','Z',0)
% 7. Побудова просторових траєкторій
% другого і третього тіл відносно першого
subplot(4,4,[3,4,7,8,11,12])
plot3(X2,Y2,Z2,'. ',X3,Y3,Z3), grid
title('Движение трех гравитирующих тел','FontSize',14)
xlabel('Координата X')
ylabel('Координата Y')
zlabel('Координата Z')
legend('первое тело','второе тело',0)
% 8. Вывод текстового оформлення
subplot(4,4,[15,16])
axis('off')
h= text(0,1.0,'Массовые параметры (относительные):');
```



```

h= text(-0.1,0.9,['mu1 = 1; ',sprintf('mu2 = %g; ',mu2),sprintf('mu3 = %g',mu3)]);
h= text(0.0,0.8,'Начальные координаты (безразмерные):');
h= text(-0.1,0.7,['второго тела относительно первого:',...
    sprintf('X21 = %g; ',X210),sprintf('Y21 = %g; ',Y210),...
    sprintf('Z21 = %g; ',Z210)]);
h= text(-0.1,0.6,['третьего тела относительно второго: ',
    sprintf('X32 = %g; ',X320),sprintf('Y32 = %g; ',Y320),...
    sprintf('Z32 = %g; ',Z320)]);
h= text(0.0,0.5,'Начальные скорости (безразмерные):');
h= text(-0.1,0.4,['второго тела относительно первого: ',...
    sprintf('V21x = %g; ',V21x),sprintf('V21y = %g; ',V21y),...
    sprintf('V21z = %g; ',V21z)]);
h= text(-0.1,0.3,['третьего тела относительно второго: ',...
    sprintf('V32x = %g; ',V32x),sprintf('V32y = %g; ',V32y),...
    sprintf('V32z = %g; ',V32z)]);
h= text(-0.1,0.1,'_____');
h= text(-0.1,-0.1,'Программа GR-3-TEL-upr Автор - Лазарев Ю. Ф. 25-01-2004');
h= text(-0.1,-0.2,'_____');
    
```

Запуск цієї програми дозволяє одразу одержувати результати моделювання у виді, наведеному на рис. 9.89...9.91.

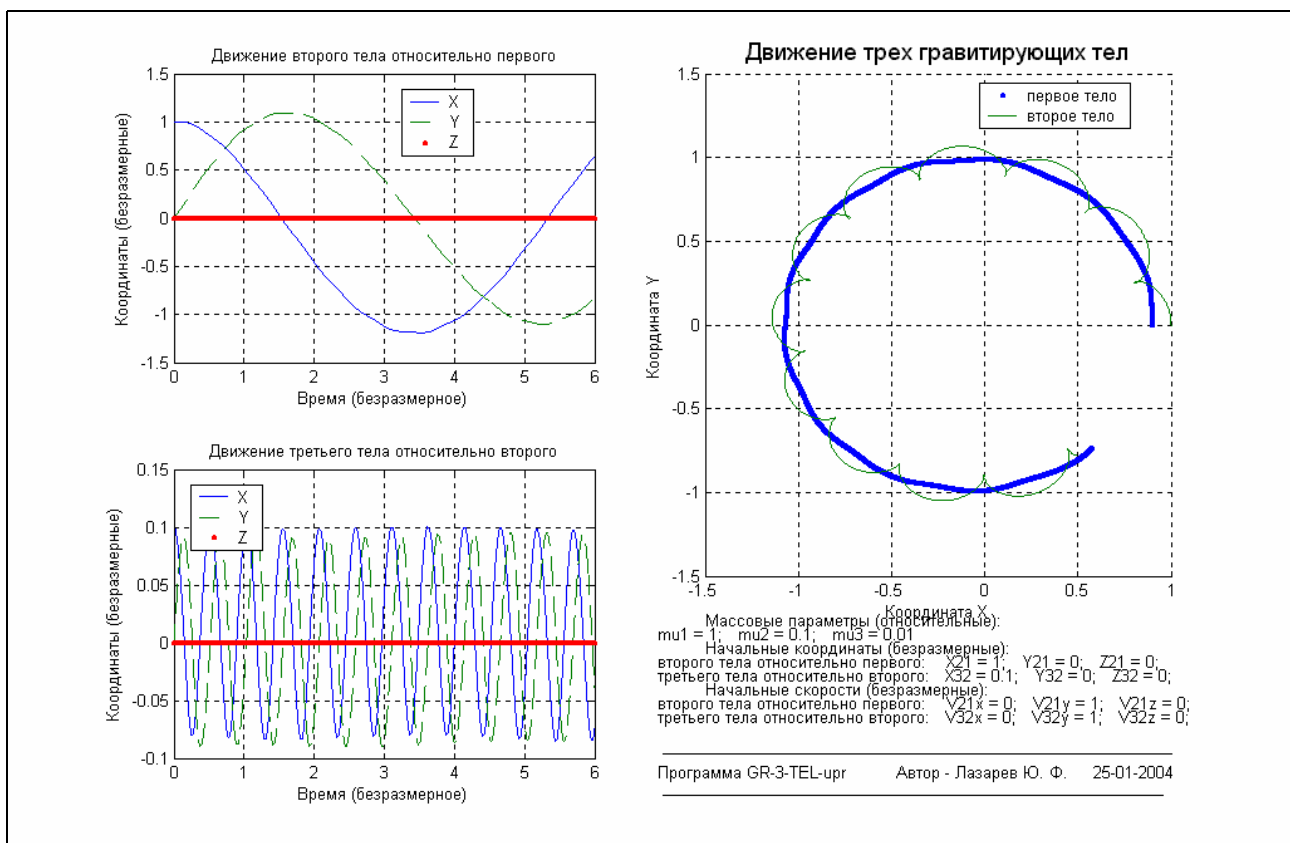


Рис. 9.89. Плоский орбитальный рух двох тіл в одному напрямку

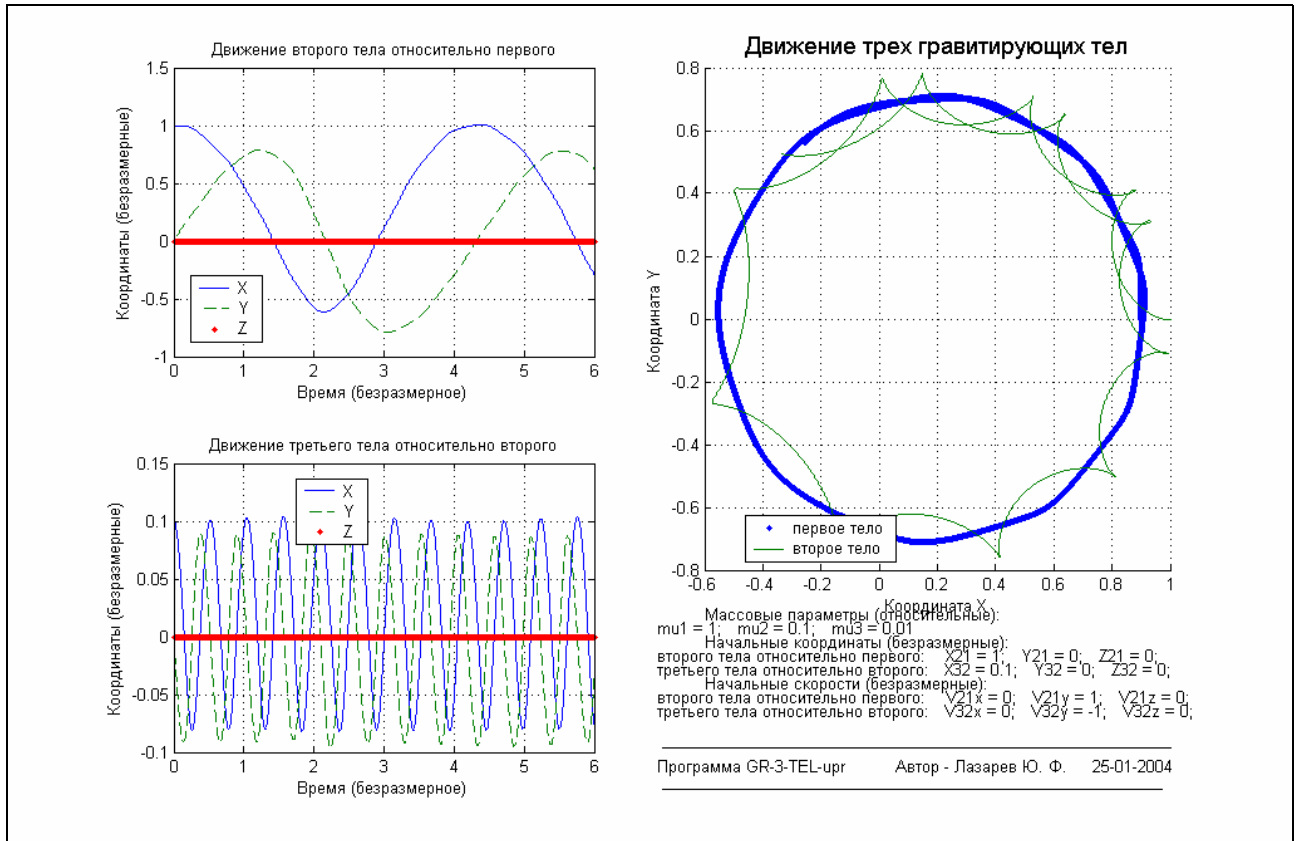


Рис. 9.90. Плоский орбитальный рух двох тіл у протилежних напрямках

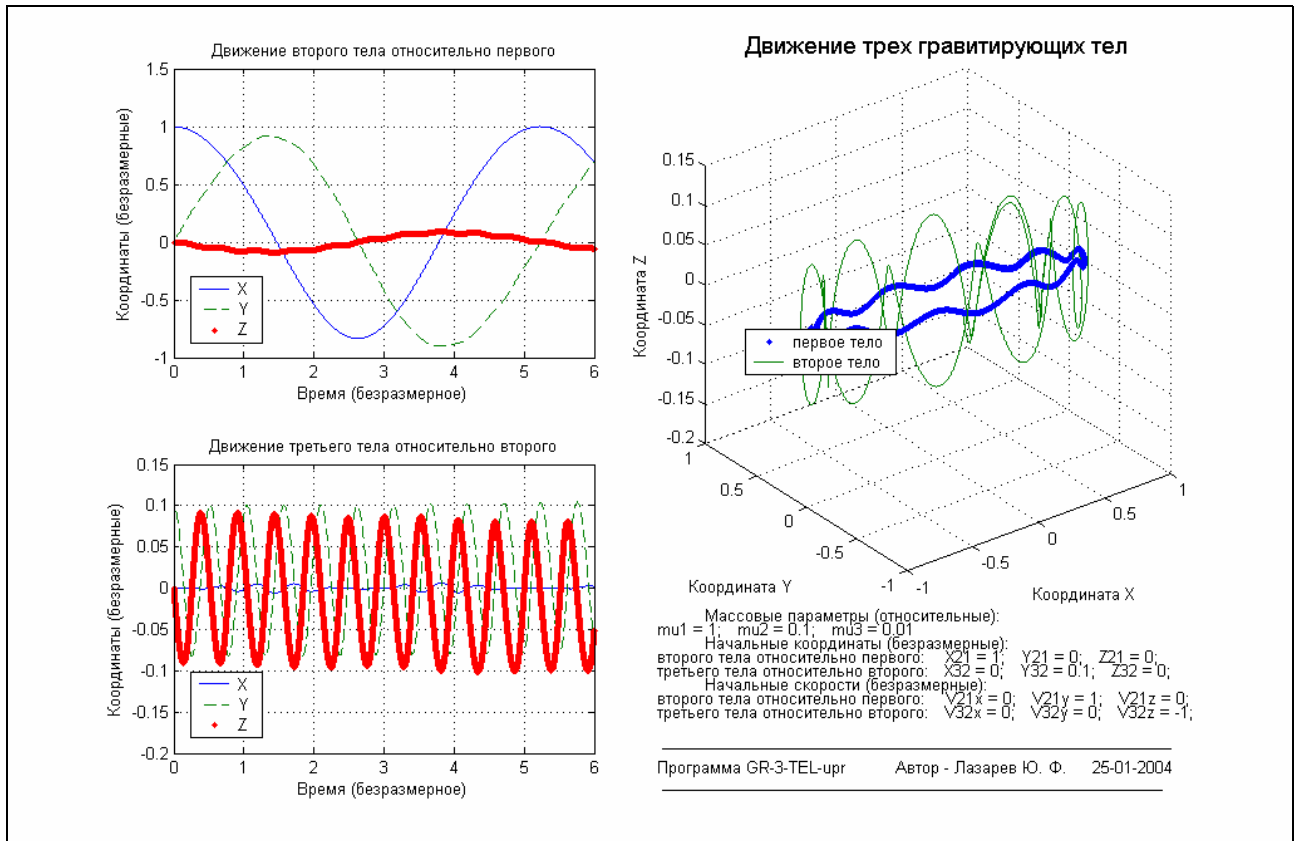


Рис. 9.91. Орбітальний рух у випадку перпендикулярності площин орбіт

Подані результати відповідають трьом випадкам:

- 1) відносні орбітальні рухи другого тіла відносно першого і третього відносно другого здійснюються в одній площині (XY) в одному напрямку (орбітальні моменти однонапрямлені – вздовж осі Z);
- 2) зазначені орбітальні рухи здійснюються в одній площині (XY) у протилежних напрямках (орбітальні моменти спрямовані протилежно);
- 3) вказані орбітальні рухи здійснюються у взаємно-перпендикулярних площинах.

В усіх випадках відношення мас тіл першого, другого і третього тіл дорівнює 100:10:1. Відносні початкові швидкості тіл в усіх свипадках прийняті однаковими за величиною ($V_{21} = 1$ і $V_{32} = 1$).

Зверніть увагу на різницю у параметрах орбіти середнього (другого) тіла і періода цього орбітального руху.

9.7.2. Модель руху маятника з сухим тертям

Розглянемо тепер задачу дослідження руху фізичного маятника, на який діє момент сил сухого тертя в опорах його осі обертання, в умовах обертання навколо цієї осі й самої основи, на якій встановлено маятник.

Позначимо α - кут повороту маятника відносно основи. Тоді рівняння руху (обертання навколо його осі) маятника можна записати у такому безрозмірному виді:

$$\varphi'' + \sin \varphi = \mu_{tr}(\alpha'), \quad (9.44)$$

де φ - кут відхилення маятника від вертикалі. Величина $\mu_{tr}(\alpha')$ являє собою безрозмірний момент сил тертя, тобто відношення момента тертя до так званого опорного маятникового моменту mgL .

Позначимо кут повороту основи відносно вертикалі навколо осі обертання маятника через ϑ . Тоді три кути ϑ , α і φ пов'язані між собою співвідношенням

$$\alpha = \varphi - \vartheta. \quad (9.45)$$

Запишемо рівняння (9.44) з врахуванням (9.45) у виді:

$$\alpha'' = -\vartheta'' - \sin(\alpha + \vartheta) + \mu_{tr}(\alpha'). \quad (9.46)$$

Координата α характеризує відносне переміщення (кутове) маятника і основи.

Утворення універсального блоку сухого тертя

Почнемо з утворення блоку, який здійснював би дії по формуванню впливу сухого тертя на поведінку динамічної системи. Через те, що головні особливості сухого тертя є відомими, можна створити універсальний (узагальнений) блок, який можна застосовувати в усіх механічних системах.

Усім добре відомі голоні властивості сили тертя, що виникають при відносному переміщенні двох тертьових одне по одному тіл:

- сила тертя завжди спрямована у бік, протилежний відносно швидкості цих тіл;
- величина сили тертя не залежить від величини цієї відносної швидкості, залишаючись тією самою, як би не змінювалася ця швидкість.

Ці властивості достатньо добре описуються математично, якщо використати відому разривну елементарну математичну функцію *sign*:

$$F_{TR} = -F_T \cdot \text{sign}(V),$$

де F_T - деякий додатний коефіцієнт, рівний величині сили сухого тертя, а V - величина відносної швидкості взаємного переміщення тертьових тіл.

Але відома ще одна властивість сухого тертя:

- якщо тертьови тіла є нерухомими одне відносно одного, то прикладення зовнішньої сили до одного з них не викличе відносного руху тіл доти, поки діюча сила (назвемо її «активною» - F_a) не перевищить за величиною так звану силу тертя спокою $F_p > F_T > 0$.

У цьому випадку сила тертя вже визначається не величиною і напрямком швидкості, а величиною і напрямком прикладеної активної сили, приймаючи таке значення і напрямком, що вона повністю компенсує дію цієї сили: $F_a + F_{TR} = 0$, якщо $V = 0$ і $|F_a| \leq F_p$.

Ця особливість сухого тертя є причиною цілої низки незвичайних властивостей систем з сухим тертям, зокрема такого, як явища «захоплення», «зчеплення» одного тіла з другим, коли обидва тіла починають рухатися як одно, залишаючись нерухомими один відносно одного.

Теоретичне вивчення цієї властивості сил сухого тертя наштовхується на значні труднощі, пов'язані з розривним характером залежності сили тертя від швидкості і зазначеною складною залежністю сили сухого тертя від швидкості і активної сили.

Сформулюємо задачу опису руху механічної системи з сухим тертям.

Введемо у розгляд узагальнену координату q (це може бути лінійне переміщення або кут при обертальному русі), яка відповідає відносному переміщенню тертьових тіл. Складемо узагальнене диференціальне рівняння за цією координатою і поділимо усі члени цього рівняння на три частини:

- 1) узагальнену силу інерції, до якої віднесемо лише члени, пропорційні відносному узагальненому прискоренню; тому її можна подати у вигляді ($-M_q \cdot \ddot{q}$), де M_q має сенс узагальненої маси і може залежати лише від узагальненої координати q ;
- 2) узагальнену силу тертя $Q_{TR}(\dot{q})$, до якої віднесемо усі члени рівняння, що визначають вплив сухого тертя;

3) «активну» узагальнену силу Q_a , до якої включимо усі решту членів рівняння.

Тоді рівняння руху за цією координатою можна подати у виді:

$$\ddot{q} = \frac{1}{M_q} (Q_a + Q_{TR}(\dot{q})). \quad (9.47)$$

Лише після такої операції можливо сформулювати математичну залежність узагальненої сили сухого тертя від усіх чинників, що впливають на неї у відповідності до встановлених властивостей тертя:

$$Q_{TR}(\dot{q}) = \begin{cases} -M_{TD}, & \text{якщо } \dot{q} > 0 \\ M_{TD}, & \text{якщо } \dot{q} < 0 \\ -Q_a, & \text{якщо } \dot{q} = 0 \text{ та } |Q_a| < M_{TP} \\ M_{TP}, & \text{якщо } \dot{q} = 0 \text{ та } |Q_a| \geq M_{TP} \end{cases}, \quad (9.48)$$

де сталі додатні величини M_{TD} і M_{TP} визначають величини узагальненої сили тертя руху і спокою відповідно. При цьому зазвичай виконується співвідношення $M_{TP} \geq M_{TD}$.

Як бачимо, описати повністю усі зазначені особливості сили сухого тертя можна тільки після того, як задані (відомі) рівняння руху і в ньому виділена так звана активна сила.

Утворимо універсальний блок, який здійснює однократне інтегрування рівняння (9.47). Входом цього блоку мусить бути поточне значення активної сили, а виходом – поточне значення узагальненої швидкості \dot{q} . Параметрами, що визначають процес інтегрування, є:

У формулі	У програмі	Примітка
M_q	Mq	узагальнена маса
M_{TD}	TrDvig	величина тертя руху
M_{TP}	TrPoc	величина тертя спокою
\dot{q}_0	qt0	початкове значення узагальненої відносної швидкості

Оформимо блок у виді підсистеми, що зображена на рис. 9.92 і назвемо його **Suhoe Trenye**.

Основний елемент блоку – інтегратор. Він здійснює інтегрування сигналу відносного прискорення, що подається на нього, видаючи сигнал, рівний поточному значенню узагальненої швидкості. Початкове значення узагальненої швидкості задається зовнішнім блоком **IC**, на який подається сталий сигнал з блоку **Constant**, рівний початковому значенню узагальненої швидкості.

Сигнал узагальненого прискорення форується на суматорі (блок **Sum**), де підсумовуються активна сила із силою тертя після проходження нею через блок **Gain**, який здійснює ділення сумарного сигналу на узагальнену масу.

Формування сили тертя здійснено у нижній частині блок-схем.

Якщо швидкість \dot{q} не дорівнює нулеві, то перемикач у блоці **Ключ** знаходиться у нижньому положенні, і сигнал узагальненої швидкості проходить через блок **Sign** (нижня права частина схеми), множиться на постійний коефіцієнт $TrDvig$ і передається на суматор як сила тертя зі зворотним знаком. Цим реалізуються перші два співвідношення (9.48).

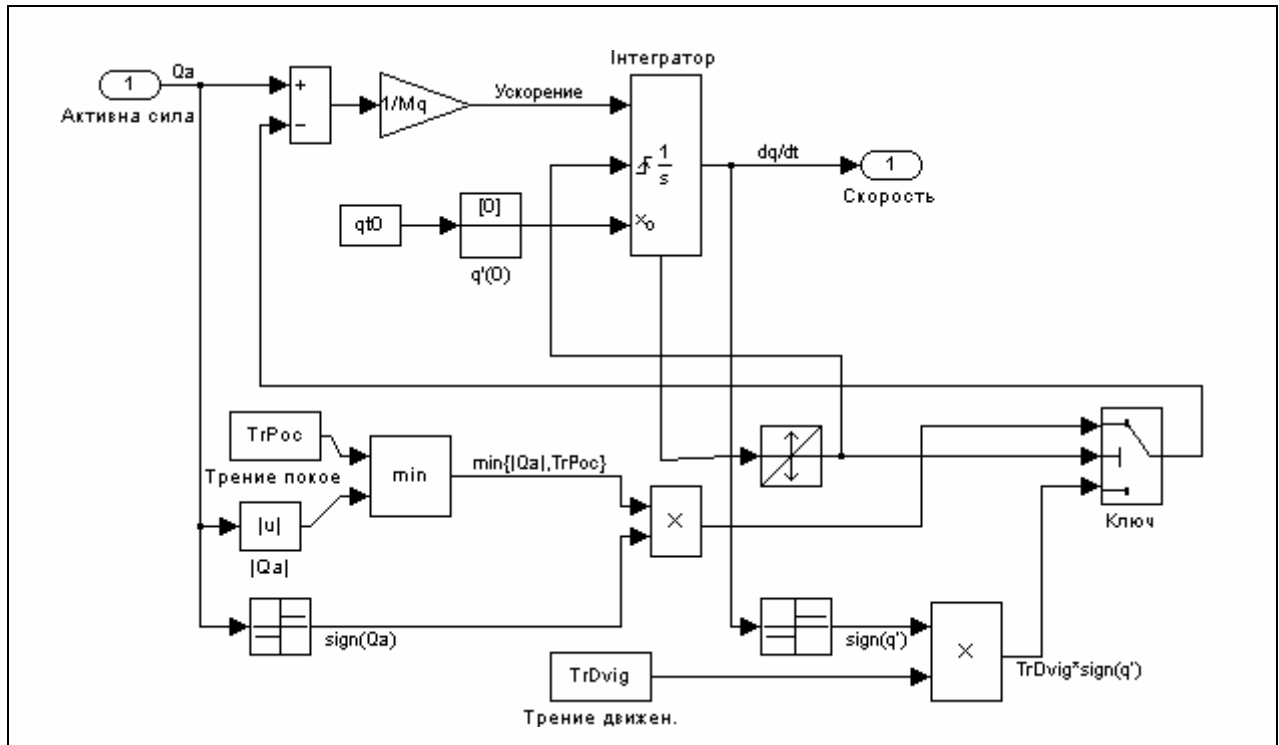


Рис. 9.92. Блок-схема блоку *Suhoє Trenye*

Значно складніше реалізувати останні дві умови у (9.48). Для цього, перш за все, потрібно якомога точніше визначити момент часу, коли значення відносної швидкості переходить через нуль. Це досягається трьома засобами:

- 1) у блоці інтегратора слід відчинити порт стану *Show state port*; при цьому на зображенні блоку внизу виникне додатковий вихід – порт стану; окрім того, слід під'єднати зовнішнє керування роботою інтегратора, встановивши значення *Rising* цього параметру *External reset*; при цьому з лівого боку блоку виникне зображення ще одного порта (керування);
- 2) до порту стану треба підключити вхід блоку **Hit Crossing**, який здійснюватиме фіксацію точного моменту часу переходу швидкості через нуль і видаватиме у цей момент часу керуючий одиничний сигнал;
- 3) вихід блоку **Hit Crossing** з'єднаємо з керуючим входом блоку **Switch** (другий згори вхід); при цьому як поріг (параметр *Threshold*) у цьому блоці слід встановити значення 0,5; окрім того вихід блоку **Hit Crossing** треба з'єднати з портом керування блоку **Інтегратор**.

Сукупність цих блоків працює у такий спосіб. Якщо швидкість не перевищує нульового значення, вихідний сигнал блоку **Hit Crossing** дорівнює нулеві. Він менше за величину порогу блоку **Switch** (0,5). Тому перемикач з'єднує з виходом третій (нижній) вхід і на суматор подається сила тертя руху. Як тільки блок **Hit Crossing** региструє перетинання швидкістю нуля, на його виході сигнал стає рівним 1, він стає більшим за поріг блоку **Switch**, і пермикає на суматор гілку блок-схеми, яка формує тертя спокою (ліва нижня частина блок-схеми). Одночасно сигнал блоку **Hit Crossing** поступає на керуючий вхід блоку **Інтегратор**. За ним інтегратор починає інтегрування спочатку з моменту перетинання швидкістю нуля з початковою умовою, встановленою у блоці ІС (у нашому випадку $\dot{q}=0$).

Якщо при подальшому інтегруванні \dot{q} продовжує залишатися рівною нулеві, цей стан залишається незмінним. Якщо ж \dot{q} на якомусь кроці інтегрування набуде значення відмінного від нуля, блок **Hit Crossing** скине значення свого вихідного сигналу до нуля, перемикач перекине «рубильник» у нижнє положення і знову «запрацює» тертя руху.

Гілка схеми, що формує тертя спокою, здійснює наступні функції. Спочатку визначається модуль активної сили. Потім він порівнюється зі значенням тертя спокою. Визначається мінімальна з цих двох додатних величин. Потім їй присвоюється знак активної сили (блоки **Sign** і **Product**). Ця величина й складає тертя спокою і спрямовується на першій вхід перемикача.

Сконструюємо маску блоку **Suhoe trenye** (рис. 9.93).

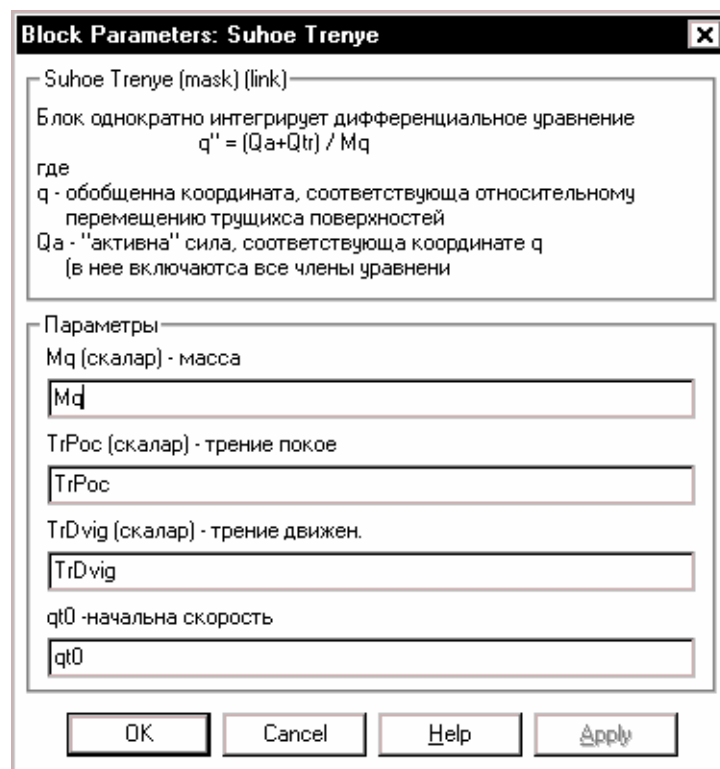


Рис. 9.93. Вікно налаштування (маска) блоку **Suhoe Trenye**

Порівнюючи (9.46) з (9.47) можна дійти висновку, що у розглядуваному випадку

$$q = \alpha, \quad M_q = 1, \quad Q_a = -\vartheta'' - \sin(\alpha + \vartheta) = -\vartheta'' - \sin \varphi.$$

Блок-схема **TRENYE** S-моделі, яка реалізує інтегрування рівняння (9.46) маятника, наведена на рис. 9. 94.

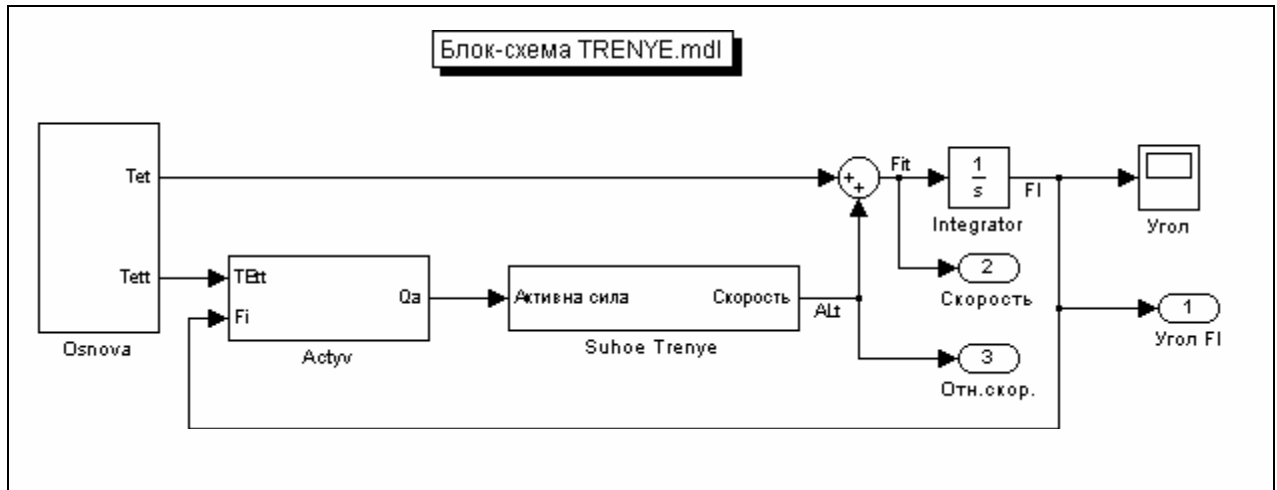


Рис. 9.94. Блок-схема S-моделі **TRENYE**

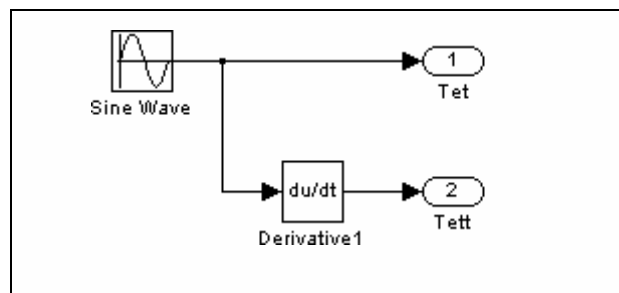


Рис. 9.95. Блок-схема підсистеми **Osнова**

Блок **Osнова** у цій S-моделі (рис. 9.95) формує сигнали кутової швидкості (Tet) і кутового прискорення (Tett) обертання основи.

Як бачимо, кутова швидкість основи формується за законом

$$\vartheta'(\tau) = \theta'_o + \theta'_m \sin(\omega_\vartheta \cdot \tau + \varepsilon_\vartheta).$$

Значення констант, які використовуються, вводяться у вікні налаштування блоку **Sine Wave** (рис. 9.96).

Нижче наводиться таблиця позначень у формулах і програмі.

У формулі	У програмі	Примітка
θ'_o	Tet0	Стала складова кутової швидкості
θ'_m	Tetm	Амплітуда кутової швидкості
ω_ϑ	omt	Частота змінювання кутової швидкості
ε_ϑ	et	Початкова фаза кутової швидкості

Блок-схема другої підсистеми **Activ**, яка формує сигнал активної сили, є надзвичайно простою і подана на рис. 9.97.

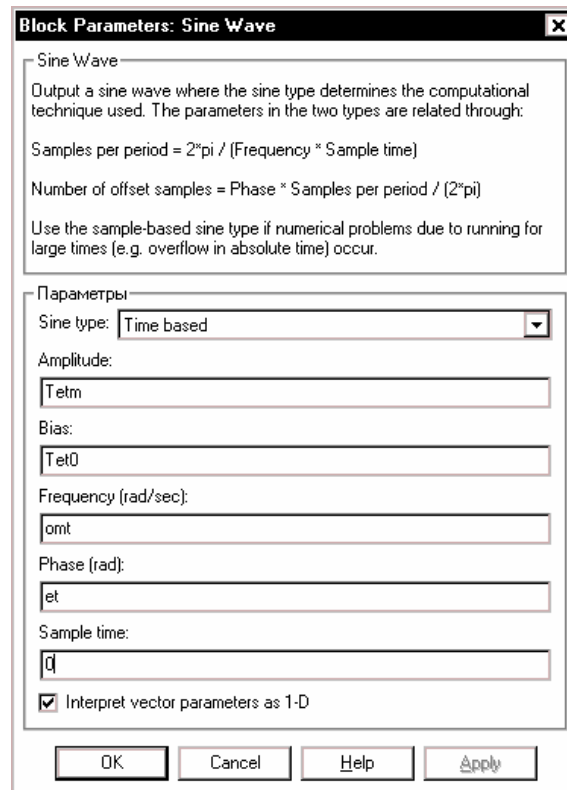


Рис. 9.96. Вікно настроювання блоку *Sine Wave*

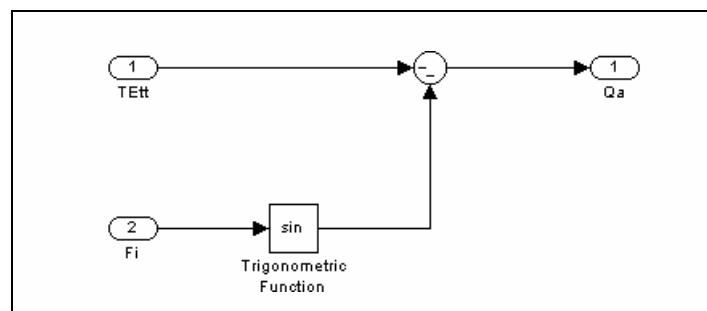


Рис. 9.97. Блок-схема підсистеми *Activ*

Як і раніше, утворимо на мові Matlab керуючу програму TRENYE_upr. m, яка виконуватиме наступні функції:

- 1) виведення значень усіх параметрів, які однозначно визначають рух системи (маятника);
- 2) запуск S-моделі TRENYE.mdl до моделювання;
- 3) виведення результатів моделювання у графічне вікно MatLab.
Текст програми наведений нижче.

```
% Trenye_upr
```

```

% Керуюча програма для роботи з моделлю TRENYE.mdl
% Лазарев Ю.Ф. 5-2-2004
clear all, clc
%      1. Завдання маси і характеристик тертя
Mq=1; TrPoc=0.2; TrDvig=0.01;
%      2. Завдання параметрів обертання основи
% Teta' = Tet0+Tetm*sin(omt*t+et)
Tetm=0; Tet0=0; omt=0; et=0;
%      3. Завдання початкових умов
fi0=30*pi/180; fit0=0;
%      4. Розрахунок початкової відносної швидкості
qt0=fit0-Tet0-Tetm*sin(et);
%      5. Запуск моделі до модельовання
sim('TRENYE'); % МОДЕЛЮВАННЯ на S-моделі
%      6. Формування даних для виведення ГРАФІКІВ
Fl=yout(:,1)*180/pi; Flt=yout(:,2); ALt=yout(:,3); t=tout;
%      7. Виведення графіків
subplot(2,2,1)
plot(Fl,Flt,'.',Fl,ALt), grid
set(gca,'fontsize',12)
xlabel('Угол (градусы)'), ylabel('Угловая скорость (б/р)')
legend('относит.','абсолютн.',0)
set(gca,'fontsize',14), title('Фазовый портрет')
subplot(2,2,[3 4])
plot(t,Fl), grid
set(gca,'fontsize',12)
xlabel('Время (б/р)'), ylabel('Угол (градусы)')
set(gca,'fontsize',14), title('Угол отклонения от вертикали')
subplot(2,2,2)
axis('off')
h=text(0,1,'Маятник с сухим трением','fontsize',16);
h=text(-0.2,0.8,'Вращение основания: Teta"(t)=Tet0+Tetm*sin(omt*t+et)','fontsize',12);
h=text(-0.2,0.7,['где: ',...
    sprintf('Tet0 = %g; ',Tet0),sprintf('Tetm = %g;',Tetm),...
    sprintf('omt = %g; ',omt),...
    sprintf('et =% g градусов',et*180/pi)]);
h=text(-0.2,0.5,'Характеристики трения','fontsize',12);
h=text(-0.1,0.4,[sprintf('Трение покоя = %g; ',TrPoc),...
    sprintf('Трение движения = %g; ',TrDvig)]);
h=text(-0.2,0.2,sprintf('Начальная абс. угл. скор. = %g;',fit0),...
    'fontsize',12);
h=text(-0.2,0.0,'-----');
h=text(-0.2,-0.1,'Программа TRENYE-upr 5-02-2004 Лазарев Ю. Ф. ');
h=text(-0.2,-0.2,'-----');

```

Запускаючи цю програму, одержимо результат, показаний на рис. 9.98.

З рисунку стають наочними три головні нелінійні властивості маятника:

- *твірна вільних коливань є прямою;*
- *вільні коливання загасають за скінченний проміжок часу;*
- *маятник зупиняється не у положенні вертикалі, а у зміщеному відносно неї положенні.*

Наступні два рисунки (9.99 і 9.100) ілюструють поведінку маятника при рівномірному обертанні основи навколо осі маятника (такий маятник називають маятником Фроуда).

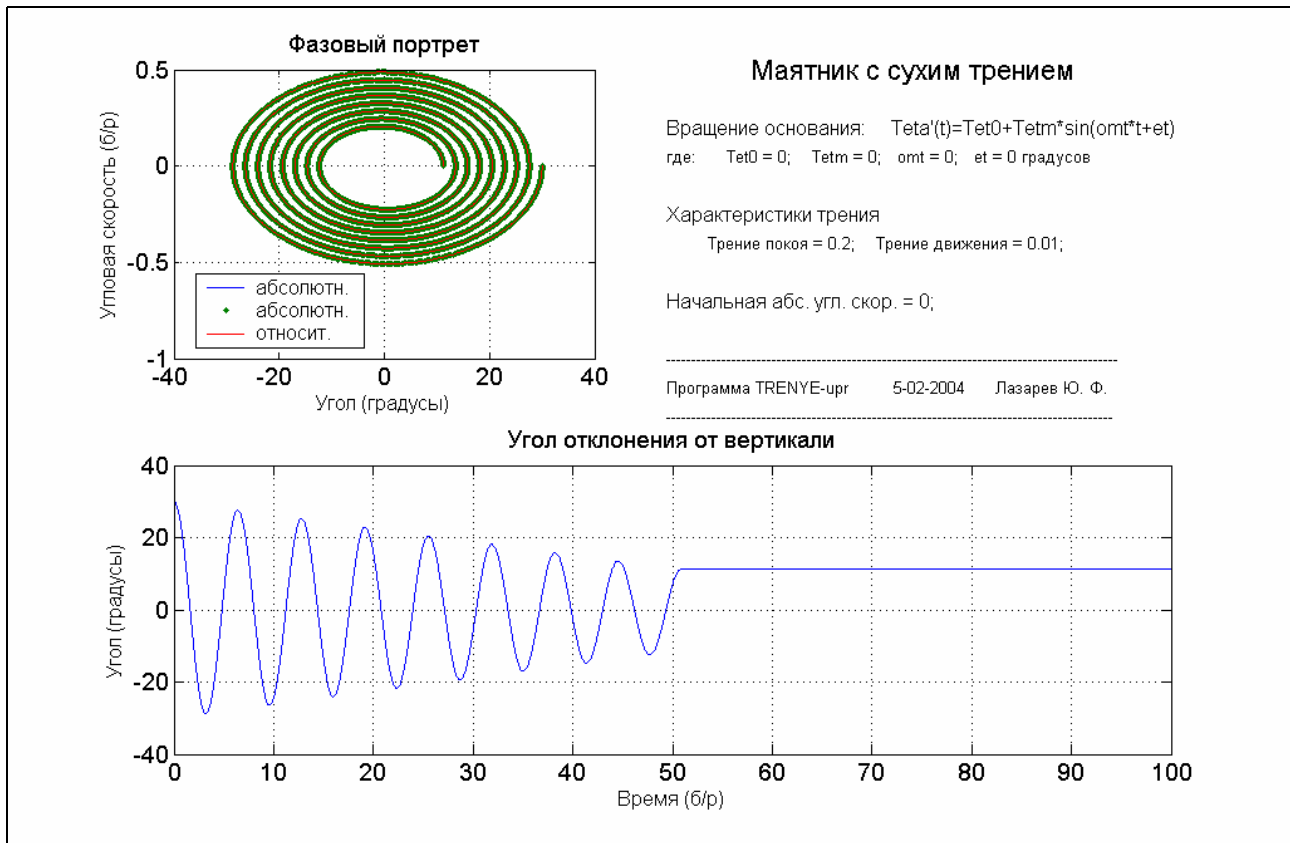


Рис. 9.98. Вільні коливання маятника з сухим тертям

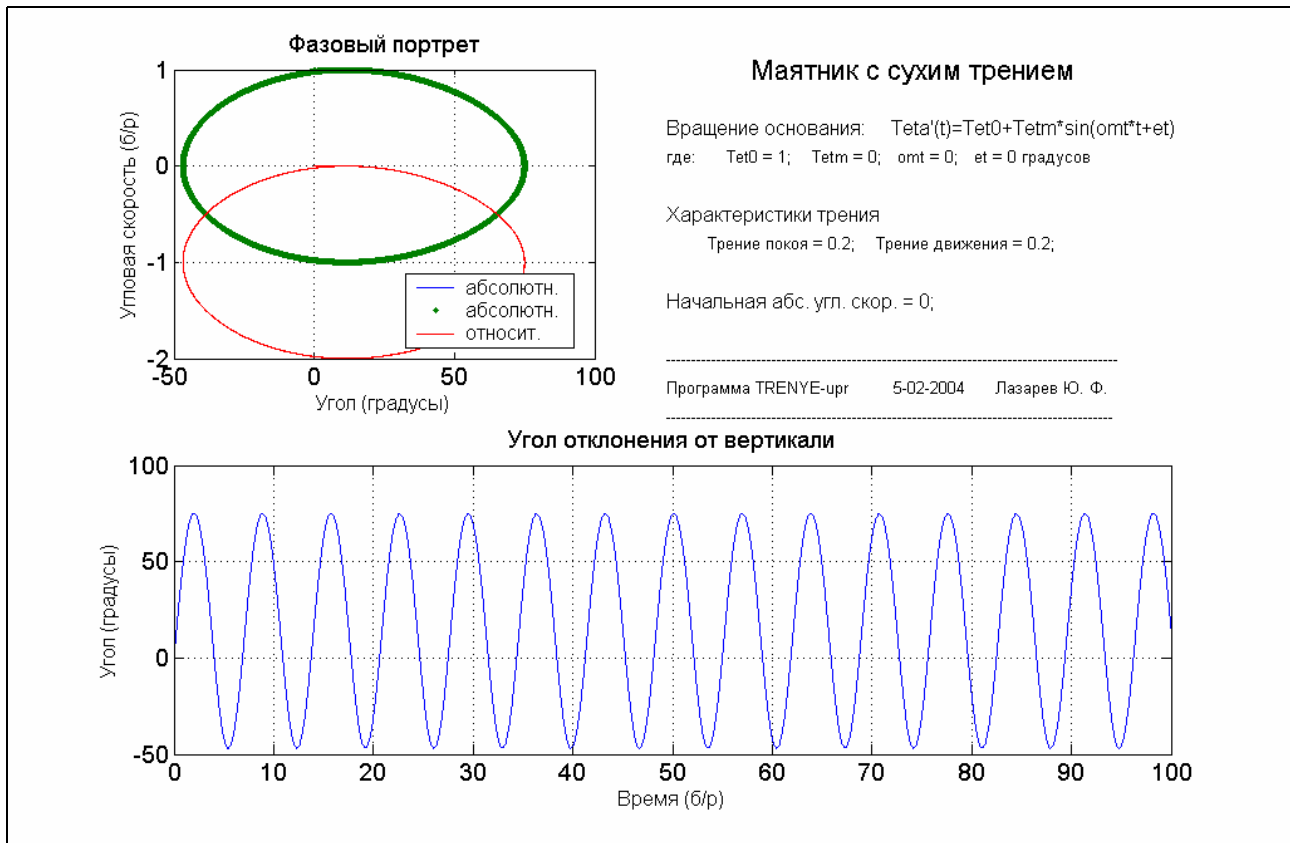


Рис. 9.99. Маятник Фроуда при значній (1 1/с) кутовій швидкості основи

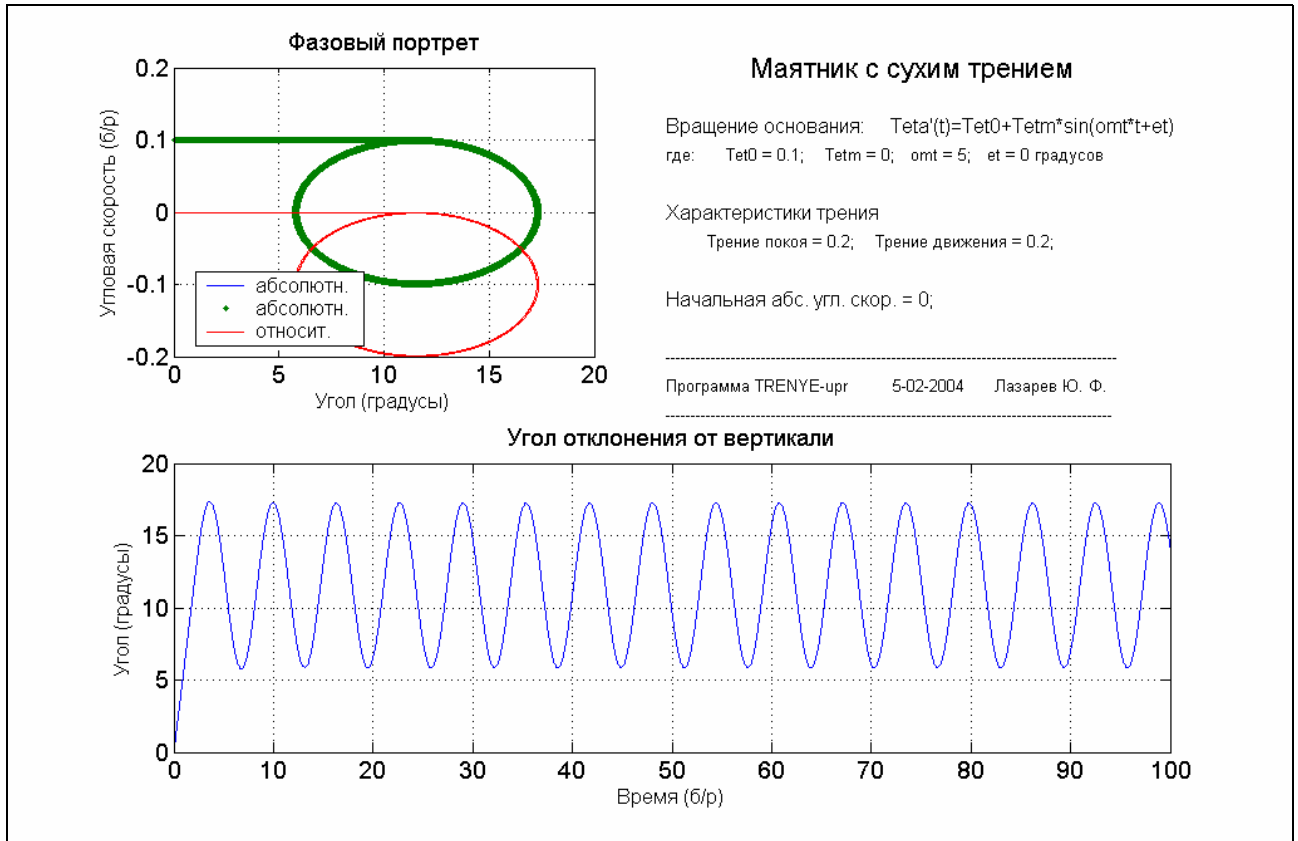


Рис. 9. 100. Маятник Фроуда при малій (0,1 1/с) кутовій швидкості основи

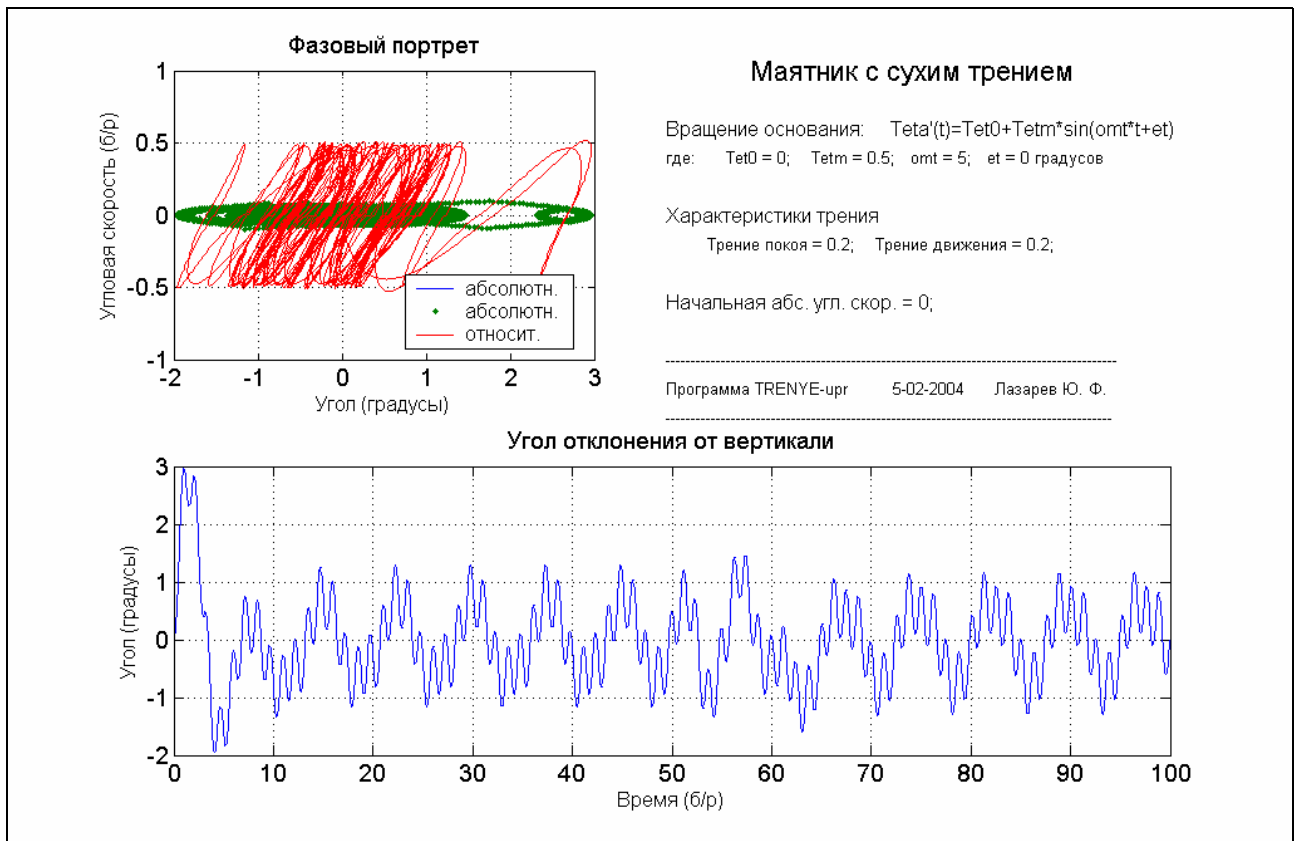


Рис. 9. 101. Маятник при коливаннях основи з великою амплітудою

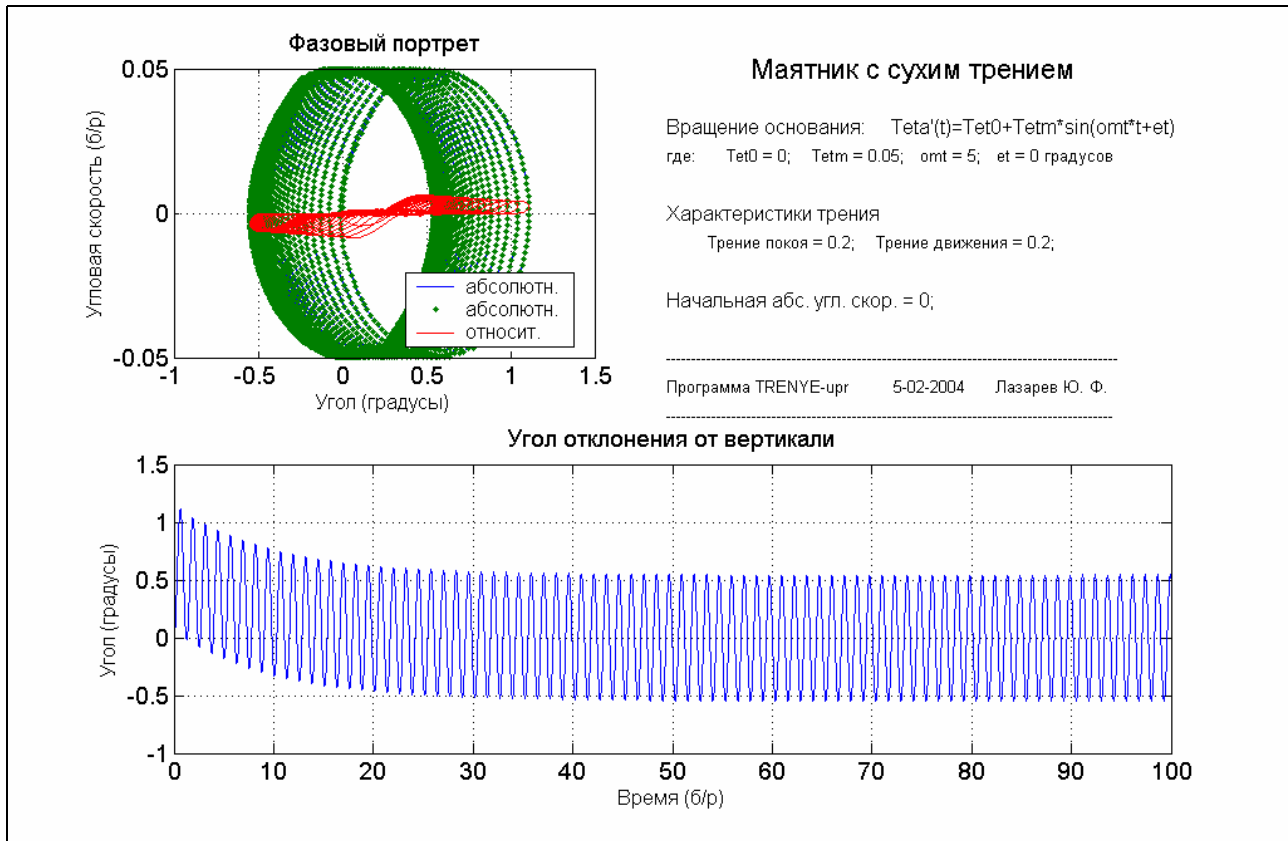


Рис. 9.102. Маятник при колебаниях основы с малою амплітудою

З розгляду рисунків випливає, що при обертанні основи з постійною кутовою швидкістю під дією сил сухого тертя маятник здійснює незгасаючі коливання з частотою його власних (вільних) коливань відносно середнього положення, зсунутого відносно вертикалі на 11,5 градусів у бік обертання основи. Від величини кутової швидкості основи залежить лише амплітуда цих коливань.

Вплив коливань основи навколо осі маятника продемонстровано на рис. 9.101 і 9.102.

Цікаво, що амплітуда вимушених коливань маятника практично не залежить від амплітуди коливань основи. Але за значних амплітуд коливань основи, на вимушені коливання накладаються незагасаючі власні коливання маятника.

9.8. Контрольні запитання

1. Які переваги використання пакета **Simulink** для розв'язування обчислювальних задач у порівнянні з програмуванням їх безпосередньо у середовищі MatLab?
2. Присутність блоків яких розділів бібліотеки **SIMULINK** є необхідною у блок-схемі будь-якої S-моделі і чому?
3. Яке головне призначення блоків розділу **Source** бібліотеки **SIMULINK**?

3. Яке головне призначення блоків розділу **Sinks** бібліотеки **SIMULINK**?
4. Блоки якого розділу бібліотеки **SIMULINK** забезпечують можливість користувачеві утворювати власні блоки?
5. Що таке підсистема і як її утворити?
6. Якими засобами можна забезпечити передавання даних з середовища MatLab у S-модель і зворотно?
7. У чому головна функціональна відмінність блоку Fcn від блоку MATLAB Fcn?
8. У чому полягає головна перевага блоку S-Function у порівнянні з усіма рештою блоків бібліотеки **SIMULINK**, які дозволяють користувачеві утворювати власні блоки?
9. У чому полягають переваги використанні підсистем?
10. Чи можна забезпечити одночасне інтегрування кількох процесів одним блоком Integrator?
11. Як всередині блоків позначаються вхідні, вихідні величини блоку та його змінні стану?
12. Що таке функції перетинання нуля, для чого вони прислужуються і в які блоки входять?
13. Опишіть засоби пакета **SIMULINK**, які забезпечують зв'язок даних з робочого простору MatLab і даних, що містяться в S-моделі.
14. Що таке S-функції, для чого вони призначені і як утворити S-блоки на їхній основі?
15. Як забезпечити запуск S-моделі з програми MatLab?
16. Як забезпечити запуск програми MatLab з S-моделі?
17. Що таке маска блоку, і для чого вона призначена?
18. Як утворити вікно настроювання блоку?
19. Як утворити власну бібліотеку S-блоків?



Рекомендована література

1. Барановская Г.Г., Любченко И.Н. Микрокалькуляторы в курсе высшей математики: Учебное пособие. - К.: Вища школа, 1987. - 288 с.
2. Вержбицкий В. М. Численные методы (линейная алгебра и нелинейные уравнения): Учеб. пособие для вузов. – М.: ОНИКС 21 век, 2005. – 432 с.
3. Вержбицкий В. М. Численные методы (математический анализ и обыкновенные дифференциальные уравнения): Учеб. пособие для вузов. – М.: ОНИКС 21 век, 2005. – 400 с.
4. Гринчишин Я.Т. и др. Алгоритмы и программы на Бейсике: Учебное пособие. - М.: Просвещение, 1988. - 160 с.
5. Гутер Р.С. Элементы численного анализа и математической обработки результатов опыта: Учебное пособие. – М.: "Наука", 1979. – 432 с.
6. Крылов В. И., Бобков В. В., Монастырский П. И. Вычислительные методы: Учебное пособие. – М.: "Наука", 1976. –304 с.
7. Лазарев Ю.Ф. Початки програмування в середовищі MatLAB: Навчальний посібник. - К.: "Корнійчук", 1999. - 160 с.
8. Лазарев Ю.Ф. MatLAB 5.x. - К.: "Ирина", ВНУ, 2000. - 384 с.
9. Лазарев Ю.Ф. Моделирование процессов и систем в MatLAB. Учебный курс. – СПб.: Питер; К.: Издательская группа ВНУ, 2005. - 512 с.
10. Лященко М. Я., Головань М. С. Чисельні методи: Підручник. – К.: "Либідь", 1996. – 288 с.
11. Лебедев А.Н. Моделирование в научно-технических исследованиях. - М.: Радио и связь, 1989. - 224 с.
12. Маликов В.Т., Кветный Р.Н. Вычислительные методы и применение ЭВМ: Учеб. пособие. - К.: Вища школа, 1989. - 213 с.
13. Половко А.М., Бутусов П.Н. MATLAB для студента. – СПб.: БХВ-Петербург, 2005. – 320 с.
14. Потемкин В.Г. Система MATLAB-5 для студентов. Справочное пособие. – М.: ДИАЛОГ-МИФИ, 1998. – 314 с.
15. Ракитин В.И., Первушин В.Е. Практическое руководство по методам вычислений с приложением программ для персональных компьютеров: Учебное пособие. - М.: Высш.шк., 1998. - 383 с.
16. Сулима И.М. и др. Основные численные методы и их реализация на микрокалькуляторах: Учебное пособие. - К.: Вища школа, 1987. - 312 с.
17. Чеблаков Б.Г. Сборник задач по программированию на языке Паскаль с решениями: Учебное пособие. - Новосибирск: НГУ, 1991. - 68 с.

Предметний покажчик

Автокореляційна функція	242	- дійсних чисел	49
Алгоритм	26	- комплексних чисел	61
- Горнера	45	- матриць	95
- методу	26	Відшукання коренів	
Амплітуда комплексна	66, 238	- поліномів	44
Амплітудно-частотна		- системи лінійних	
характеристика	72	алгебричних рівнянь	40
Аналіз розмірностей	13	- функції	92
Апроксимування функцій	131, 141	Власні значення	
- поліноміальне	143	- матричного полінома	101
- - квадратичне	144	- матриці	100
- - кубічне	144		
- - лінійне	144	Генерування випадкових величин	
- рядом Фур'є	144	- нормально розподілених	28, 77
Арифметичні дії	10, 49	- рівномірно розподілених	28
		Генерування процесів	206
Безрозмірні степеневі комплекси	12, 14	- імпульсних	206
		- неколивальних	206
Введення у системі Matlab		- нестационарних	206
- дійсних чисел	49	- періодичних	212
- векторів	49	- стаціонарних коливальних	212
- комплексних чисел	49		
- лінійних стаціонарних		Дискрет	
систем	184	- частоти	239
- матриць	49	- часу	239
Взаємна кореляційна функція	238, 250	Дискретна передатна функція	233
Взаємна спектральна щільність	238, 279	Диференційні рівняння	61
Величини		Ділення зліва направо	
- екстенсивні	9	- дійсних чисел	49
- інтенсивні	9	- комплексних чисел	61
- кількісні	9	- матриць	96
- якісні	9	Ділення поліномів	51
Визначення		Ділення справа наліво	
- власних векторів матриці	100	- дійсних чисел	49
- власних значень		- комплексних чисел	61
матричного полінома	100	- матриць	96
- власних значень матриці	100	Добуток	
- характеристичного		- матриць	95
полінома матриці	100	- поліномів	49
Визначник		Додавання	
- Грама	142	- дійсних чисел	49
- матриці	52	- електричних зарядів	10
Віднімання		- комплексних чисел	61
		- мас	10

- матриць	96	- визначувані	13
- переміщень	9		
- фізичне	7		
- часів	9	Л інійна стаціонарна система	65
Д ріб		Логарифм	
- ланцюговий	55	- від матриці	102
- придатний	56	- двійковий	60
		- десятковий	60
З гортка векторів	50	- натуральний	60
- обернена	51		
З мінні		М -файл	93, 94
- глобальні	101	Мантиса числа	38
- локальні	94	Математика	7, 10
- робочого простору	94	Математична обробка	
- стану	91, 123	масивів даних	103
- фазові	91	Математична обробка сигналів	228
З начущі цифри числа	32	Матрична експонента	100
- вірні	32	Матричний логарифм	101
- сумнівні	32	Матричні дії	96
		М етод	
Е кспонента	60	- Гаусса	88
- матрична	102	- - єдиного ділення	89
		- - роздільного елемента	91
І нваріантність системи		- Гаусса-Жордана	92
диференційних рівнянь	13	- дихотомії	110
І нтеграл еліптичний	16	- дотичних	113
І нтегрування		- Ейлера	169
- диференціальних рівнянь	91, 118	- - модифікований	172
- методом трапецій	49	- ітерацій	98, 116
- методом квадратур	90	- комбінований	115
І нтерполювання	130	- Крамера	87
- поліноміальне	131	- лівих прямокутників	
- - за Ермітом	133	- Ліна	122
- - за Лагранжем	132	- найменших квадратів	141
- - за Ньютоном	133	- правих прямокутників	
- сплайнами	135	- Рунге-Кутта	176
		- Сімпсона	
К омплексне спряження	61	- трапецій	
К орінь квадратний	60	- Хойне	174
- з комплексного числа	61	- хорд	111
- з матриці	101	М етоди	
К орінь характеристичного		- багатокрокові	176, 178
рівняння	63	- з автоматичним	
К ритерії подібності	13, 14	обранням кроку	91
- визначальні	13	- наближені	26
		- неявні	184

- однокрокові	167	вектора	102
- точні	25	- мінімуму функції	92
- прогнозу-корекції	177	- нулів функції	117
- Рунге-Кутта	167	- - передатної функції	58
- явні	182	- оберненої матриці	95
Мінімум функції		- полюсів передатної функції	58
Множення		- похідної від полінома	51
- дійсних чисел	49	- середнього значення	
- комплексних чисел	61	елементів вектора	48
- матриць	96	- середньоквадратичного	
- поліномів	44	відхилення елементів	
Модель	5	вектора від їх середнього	
- аналогова	6	значення	48
- знакова	7	- суми елементів вектора	48
- математична	7, 21, 22-25		
- обчислювальна	21, 26-27	Параметри	
- програмна	7	- базові	13
- фізична	6	- зовнішніх дій	11
- чисельна	7	- нелінійні	11
Моделювання	5, 25	- однозначності	11
- математичне	7, 25	- процесу	11
- фізичне	6	- системи	11
Норма матриці	52	Передатна функція	64, 65
Нормалізований вид числа		- - частотна	65
Нуль функції		Перетворення	
- полінома		- Лапласа	64
Обернення матриці	97	- Фур'є	238
Обчислення		- - обернене	239
- визначника матриці	94	Піднесення до степеня	
- добутку елементів вектора	48	- дійсних чисел	49
- значення матричного		- комплексних чисел	61
полінома	59	- матриць	96
- значення полінома	45-50	Подібність	
- значення функції	45, 51	- явищ	11, 12
- - на ЕОМ	57	- умов однозначності	13
- - - мовою BASIC	58	Поліном	
- - - мовою MATLAB		- Ерміта	135, 143
59		- Лагранжа	132
- - - мовою PASCAL	58	- Лежандра	143
- інтеграла	90	- характеристичний	62
- максимального елемента		- Чебишева	143
вектора	102	Порядок числа	38
- мінімального елемента		Похідна від полінома	46, 47
		Похибка	
		- абсолютна	31
		- - гранична	31
		- відносна	31
		- - гранична	32

- глобальна	181	- дійсний	238
- локальна	180	- комплексний	246
Похибки		- уявний	238
- ділення	36	- фазовий	237
- добутку	36	- частотний	237
- математичної моделі	30	Спектральна щільність	238, 249
- методу	31, 168	Спектральний аналіз	237-250
- неусувні	30	Співвідношення рекурентне	53
- обчислення функції	36	Статистичний аналіз	240-251
- обчислювальні	31	Сухе тертя	354
- округлення	31, 37, 168		
- первісних даних	30	Т еорія подібності	10, 12-13
- підсумовування	34	Транспонування	
- поширення	34, 37	- вектора	34
- усікання	31	- матриці	100
Простір стану	188, 298		
Р анг матриці	52	У мови	
Розкладення матриці		- межові	11
- сингулярне	55	- однозначності	11
- Холецького	53	- подібності	12, 13
- LU	53	- початкові	11
- QR	54		
Ряд		Ф азо-частотна характеристика	72
- біноміальний	52	Фільтр	230
- знакоперемінний	53	Фільтрація	
- степеневий	51	- векторна	233
- Фур'є	237	- подвійна	237
- - комплексний	237	Форма Коші диференціальних рівнянь	166
С игнум-функція	60	Формування	
Сингулярні числа матриці	101	- випадкових процесів	225-228
Система		- імпульсних процесів	206
- багатовимірна	67	- коливань	212
- вільна	61	Функції	
- динамічна	200	- Бесселя	60
- жорстка	184	- бета	60
- лінійна стаціонарна	65	- гамма	60
- лінійних алгебричних рівнянь	87	- гіперболичні	59
- нелінійна	11	- експоненціальні	60
- нестійка	63	- елементарні	59
- одновимірна	67	- еліптичні	60
- стійка	63	- комплексного аргументу	61
Скінченна різниця	132	- логарифмічні	59
Сортування комплексних чисел	18	- округлення	59
Спектр		- ортогональні	142
- амплітудний	237	- перетворення координат	60

- похибок	60
- спеціальні	59
- тригонометричні	59
- цілочислові	60
Функція	
- Лежандра	60
- функції	90, 117-123
Фур'є-зображення	238
- випадкового процесу	247
- дискретне	240
- полігармонійних коливань	244
- прямокутного імпульсу	242-244
Фур'є-перетворення	
- дискретне	240
- зворотне	239
- зворотне дискретне	240
- пряме	238
Характеристичне рівняння	62
Характеристичний поліном	65
Цифри числа значущі	32
- вірні	32
- сумнівні	32
Частотна передатна функція	71
Чисельне інтегрування системи диференціальних рівнянь	91, 118-123
Число наближене	30
- запис	32
- - у пам'ять ЕОМ	38
- - - мовою MATLAB	39
- - - мовою PASCAL	38
- - - мовою FORTRAN	38
- нормалізований вид	38

Показчик операторів і функцій MatLAB

	+						
-		49, 61, 96					
\		49, 61, 96					
*		49, 61, 96					
'		96, 102					
.*		104					
/		104					
.^		104					
/		49, 61, 96					
^		49, 61, 96					
+		49, 61, 96					
	A						
abs		60, 61					
Abs		258					
acos		59					
acosh		59					
acot		59					
acoth		59					
acsc		59					
acsch		59					
angle		61					
asec		59					
asech		59					
asin		59					
asinh		59					
atan		59					
atan2		59					
atanh		59					
	B						
BackLash		259					
Band-Limited							
White Noise		218, 228					
besseli		60					
besselj		60					
besselk		60					
bessely		60					
beta		60					
betainc		60					
betaln		60					
bicg		98					
bicgstab		98					
	C						
cart2pol		60					
cart2sph		60					
ceil		60					
cgs		98					
chirp		211					
Chirp Signal		211					
Clock		255					
cond		103					
conj		61					
Constant		209, 272					
conv		50					
corrcoeff		51					
cos		59					
cosh		59					
cot		59					
coth		59					
Coulomb & Viscous							
Friction		259					
csc		59					
csch		59					
cumprod		102					
cumsum		102					
	D						
Dead Zone		259					
deconv		51					
Demux		259					
Derivative		258					
det		52					
diag		103					
diff		103					
diric		217					
Discrete Pulse							
Generator		218					
Display		253					
Dot Product		258					
	E						
eig		100					
ellipj		60					
ellipke		60					
eps		103					
erf		61					
erfc		61					
erfcx		61					
erfinv		61					
exp		60					
expint		60					
expm		102					
expm1		102					
expm2		102					
expm3		102					
eye		103					
	F						
Fcn		262, 264					
fft		240					
fftshift		241					
filter		233					
filtfilt		237					
fix		60					
floor		60					
fmin		126					
fmins		127					
freqs		231					
freqz		233					
fsolve		119					
fzero		118					
	G						
Gain		260, 273					
gamma		60					
gammainc		60					
gammainl		60					
gauspuls		207, 214					
gcd		61					
global		101					
gmres		98					
grid		73					
gtext		81					
	H						
help		60					
Hit Crossing		273					

I		P		sech	59
i	49	pcg	97	sign	60
IC	272	pol2cart	60	Sign	258, 273
ifft	240	poly	50, 100	Signal	
imag	61	polyder	51	Generator	218, 219
In	253	polyeig	101	sim	154, 268
Integrator	258, 263	polyfit	144	sin	59
interp1	139	polyval	50	sinc	208
inv	97	polyvalm	101	Sine Wave	218, 221, 254
J		pow2	61	sinh	59
j	49	prod	102	size	102
L		Product	258, 273	solve	119
lcm	61	psd	249	sort	48, 49
legendre	61	Pulse Generator	218, 224	sph2cart	60
log	60	pulstran	214	spline	61
log10	60	Q		sqrt	60, 61
log2	61	qmr	98	sqrtm	42
logm	102	qr	54	Square	213
lu	53	quad	165	State-Space	337
M		quad8	165	std	102
Math Function	258	R		Step	209
Matrix Gain	259	Ramp	209	Subsystem	265
max	102	rand	103	sum	102
mean	102	randn	103	Sum	259, 272
min	102	Random Number	225	svd	101
Mux	259	rank	103	Swith	273
N		rat	61	T	
NaN	13	Rate Limiter	259	tan	59
norm	103	rcond	103	tanh	59
O		real	61	To File	253, 256
ode23	185	rectpuls	206, 214	To Workspace	253, 257
ode45	185	Relay	258	Transfer Fcn	258
ode5	195	rem	60	trapz	163
odeset	91	Repeating Sequence	218, 223	Trigonometric	
ones	103	Function	258	tripulse	207, 214
Out	253	roots	50, 72, 125	U	
O		round	60	Uniform Random	
O		S		Number	218, 227
O		Sawtooth	214	X	
O		Saturation	259	xcorr	250
O		Scope	149, 253	XY Graph	219, 253, 254
O		sec	59		

Z

Zero-Pole	258
zeros	103