ВСТУП

Математичне моделювання з використанням комп’ютерної техніки є загальноприйнятим засобом дослідження різноманітних природних явищ та створення нових технічних пристроїв. Комп’ютерні системи проектування скорочують терміни проектування та підвищують якість нових технічних систем. Теоретичною основою математичного моделювання є фундаментальні закони фізики: збереження енергії, маси та інше. Більшість з них математично можна сформулювати у вигляді диференціальних рівнянь у частинних похідних (ДРЧП). [1] Першим і дуже важливим кроком при чисельному розв’язку задач математичної фізики є правильний вибір чисельного методу для певної задачі. Чинники, що впливають на вибір чисельного методу: тип рівняння з частинними похідними, фізична розмірність задачі, граничні умови, вид координатної системи, задача стаціонарна чи нестаціонарна [2]. Основними методами розв’язання диференціальних рівнянь в частинних похідних є методи скінченних різниць та скінченних елементів. Їх загальними рисами є заміна неперервної області визначення невідомої функції (температури, концентрації речовини та інше) дискретною множиною точок(сіткою), в яких і відшукується розв’язок. Відносно значень невідомої функції в вузлах цієї сітки формується та розв’язується алгебраїчна система рівнянь, що замінює диференціальну задачу у кожний момент часу. Різноманітні варіанти цих методів відрізняються точністю, трудомісткістю, стійкістю та іншими характеристиками. [1]

У даній курсовій роботі використаний адаптивний алгоритм розв’язування крайових задач для диференціальних рівнянь у частинних похідних, який у процесі розв’язування задачі аналізує поведінку функції, контролює похибку та будує змінну за часом різницеву сітку.

Мета курсової роботи – розробити програму, яка розраховує диференціальне рівняння, вказане в умові за допомогою адаптивного алгоритму.

ОСНОВНА ЧАСТИНА

**І. Постановка задачі**

Розв’язати одновимірну нестаціонарну нелінійну задачу для диференційних рівнянь у частинних похідних. Для знаходження крайових та початкових умов необхідно знайти значення точного розв’язку у відповідних точках. Перевірити по можливості підстановкою в точний розв’язок. Значення констант в задачі взяти довільними. Розрахункова область: .[4]

Задача:

, де - деякі константи.

Точний розв’язок:

, де - деяка константа, .

Розробити два екземпляри програми, у яких використати наступні методи розв’язання та сітки:

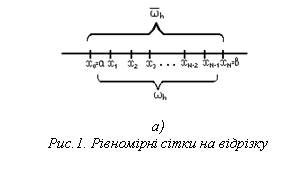
1. Явний метод, стаціонарна сітка.
2. Неявний метод, змінна рівномірна сітка.

**ІІ. Опис чисельних методів**

**2.1 Типи різницевих сіток**

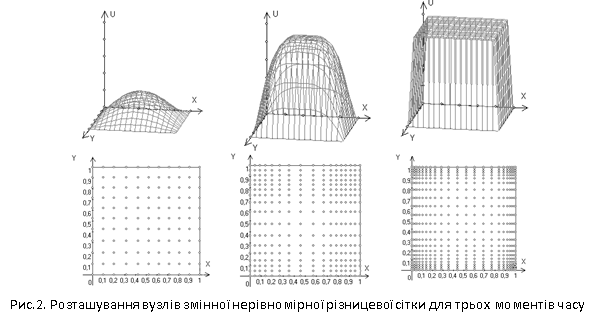
У методі скінченних різниць наближений розв'язок визначається у вузлах сітки, тому спочатку необхідно обрати сітку.

Для одержання рівномірної сітки на відрізку [*a,b*] розіб'ємо його на *n* рівних частин. Відстань між сусідніми вузлами *h=xi*-*xi*-1=(*b-a*)/*n* називають кроком сітки, а точки поділу *xi=a+ih* - вузлами (рис.1а).



Точки *x*1,*x*2,...,*xn*-1, що не належать кінцям відрізка, називають внутрішніми точками сітки, а *x*0=*a* та *xn*=*b* граничними точками.

Отже, в залежності від форми чарунок сітки діляться на трикутні, чотирикутні і т.д. В залежності від типу ліній або поверхонь, що обмежують чарунки, сітки можуть бути криволінійними або прямолінійними. Якщо розміри усіх чарунок однакові, сітки називають рівномірними, інакше – нерівномірними. Якщо з часом розміри чарунок не змінюються, сітки називають фіксованими, інакше – змінними (рис.2). Якщо сітка утворена перетином криволінійних координатних площин, вона зветься структурованою (регулярною), інакше неструктурованою (нерегулярною). Можливі гібридні сітки, коли у частині області сітка структурована, а у іншій – неструктурована. Структуровані сітки можуть бути блочними, коли сітка будується окремо для різних частин області, а стиковка виконується за допомогою інтерполяції (використовують при розв’язанні задач для областей складної форми, наприклад, у аерокосмічній промисловості).



У ряді випадків в деяких частинах області необхідно зробити більш точні розрахунки. Тоді і для простих розрахункових областей використовують нерівномірну сітку, згущуючи вузли у таких зонах. Адаптивними називають такі сітки, які автоматично перебудовується з часом в залежності від поведінки функції для досягнення наперед заданої точності. Вони дають особливо значний ефект, якщо зона великих градієнтів змінює з часом своє положення, наприклад, пляма нафти пересувається з течією і т.д.

Найпростішими є два види змінних сіток:

* змінна рівномірна, коли змінним є крок за часом *t*, а просторові кроки є однаковими в кожному координатному напрямку для фіксованого *t* і змінюються при переході на наступний часовий шар;
* змінна нерівномірна, коли і для просторових змінних відстані між сусідніми точками у кожному напрямку неоднакові.

Другий етап методу скінчених різниць - заміна диференційної задачі алгебричною. Заміну можна виконати рядом методів. Наведемо один з них-безпосередню апроксимацію похідних, які входять у рівняння, граничні і початкові умови, кінцевими різницями.

На цьому етапі похідні замінюють кінцево-різницевими співвідношеннями, у результаті чого замість точних значень шуканої функції *U* відшукують значення сіткової функції *u* у вузлах різницевої сітки. Спосіб заміни диференційної задачі різницевою визначає різницеву схему.[2]

**2.2 Явна схема. Стаціонарна рівномірна сітка.**

Маємо диференціальне рівняння :

, де - деякі константи. (2.2.1)

І для контролю реззультатів його точний розв’язок

, де - деяка константа, . (2.2.2)

Покладемо значення констант . Для конкретизації частинного розв’язку також довільно оберемо значення *А* і *В*, наприклад, покладемо , .

Тоді вихідне ДР приймає вигляд:

 (2.2.3)

а точний розв’язок дорівнюватиме

 (2.2.4)

Початкові умови одержимо, якщо у точний розв’язок (2.2.4) підставимо 

(2.2.5)

Граничні умови одержимо, якщо у точний розв’язок (2.2.4) підставимо  та :

(2.2.6)

 (2.2.7)

Перепишемо рівняння (2.2.1) з використанням формули похідної від добутку:

 (2.2.8)

Розіб’ємо область визначення шуканої функції рівнмірною сіткою, . Будемо шукати розв’язок для дискретних . Позначимо через  наближений розв’язок в точці . Для апроксимації похідних по змінній *х* у випадку рівномірної сітки з кроком *h* будемо використовувати формули другого порядку точності:

,  (2.2.9)

Явна схема має наступний вигляд (\*)

h

h

i-1,k

i,k

i+1,k

i-1,k+1

τ

З огляду на це, замінимо похідні в (2.2.8) різницевими апроксимаціями (2.2.9) за схемою (\*):



Виразимо звідси невідому величину і одержимо формулу, яку необхідно використовувати для кожного з (*n*-1) внутрішніх вузлів поточного часового шару:

(2.2.10)

Однак, незмінна сітка не враховує поведінки функції і тому робить обчислення з малим кроком там, де така точність не потрібна, і з порівняно великим кроком там, де даного кроку недостатньо для досягнення заданої точності. В свою чергу, явний метод накладає доволі жорсткі обмеження на кроки часової та просторової змінних. Далі розглядається друга частина задачі, де застосовується адаптивна сітка, яка враховує поведінку функції та неявний метод, який більш невибагливий до співвідношення кроків.

Загальна схема розрахунків за явною схемою має наступний вигляд:

1. Підрахувати початкові точки, при *k=0*, *i* від *0* до *nx*;
2. Покласти *k=0*;
3. Підрахувати крайові точки на новому часовому шарі *k+1*;
4. Прийняти *i=1*;
5. Підрахувати за формулою (2.2.10) значення в точці *(k+1,i)*;
6. Покласти *i = i + 1*;
7. Якщо *i ≠nx*, перейти на п.5;
8. Покласти *k = k + 1;*
9. Якщо *k<nt*, перейти на п.3.

**2.3 Неявна схема. Стаціонарна рівномірна сітка.**

Розглянемо ту ж саму задачу (2.2.1)-(2.2.2), але заміну диференційної задачі на алгебричну зробимо за неявною схемою.

Підготовка задачі до розв’язування виконується так само, як і у випадку явної схеми. Неявна схема має наступний вигляд:

(\*\*)

i-1,k+1

i,k+1

i+1,k+1

i,k

З огляду на це, замінимо похідні в (2.2.8) різницевими апроксимаціями (2.2.9) за схемою (\*\*):



Кожне рівняння містить три невідомих величини , ,  . Величина  відома з попереднього кроку по часу. Отримане рівняння є нелінійним і разом ці рівняння для всіх  утворюють систему  нелінійних алгебричних рівнянь (НАТР) з  невідомими. Але значення  для крайніх точок  відомі з крайових умов. Будемо розв’язувати систему НАТР методом Ньютона. Нагадаємо, що ітераційний метод Ньютона розв’язує систему НАТР, яка приведена до вигляду *F*(*x*)=0 або:



На кожній його ітерації формується і розв’язується система лінійних алгебричних рівнянь (СЛАР)



відносно приростів невідомих, де  — матриця Якобі, *s*=0,1,.... Параметр *s* пов’язаний з номером ітерації: на першій ітерації (*s*=0) з початкового вектору невідомих  одержується перше наближення  і т.д. Запишемо цю формулу в розгорнутому вигляді:

, (2.3.2)

причому для обчислення значень  і  використовується відома точка .

Після розв’язання СЛАР знаходиться чергове наближення до вектора невідомих: .

Для даної задачі невідомим  загальної формули (2.3.2) відповідають значення  шуканої функції у наступний момент часу.

Приведемо рівняння (2.3.1) до канонічного вигляду *F*(*x*)=0 :

.

Додамо в систему рівняння для крайових умов:

**** , ****.

Відповідно до методу Ньютона, на кожній ітерації потрібно сформувати СЛАР відносно приростів невідомих. Для цього необхідно обчислити частинні похідні від  по всім невідомим, тобто по , , : (з метою спрощення у наступних формулах задані значення констант знову позначені літерами)

 ,

,



Частинні похідні для першого та останнього рівнянь, що відповідають крайовим умовам, мають вигляд:

, .

Будемо позначати попередній наближений розв’язок , одержаний на *s*-й ітерації методу Ньютона, через , а прирости невідомих через . Тоді *і*-те рівняння СЛАР на (*s*+1)-ій ітерації методу Ньютона має таку форму:

.

Рівняння, що відповідають крайовим умовам мають вигляд:

****, ****.

Остаточний вигляд системи, що формується на (*s*+1)-ій ітерації методу Ньютона запишемо у матричній формі:

, (2.3.3)

де: ,.

Отримана СЛАР має тридіагональну і діагонально домінуючу структуру і для її розв’язання вибраний модифікований метод Гауса.

**2.4** **Алгоритм побудови змінної рівномірної сітки для неявного методу**

Неявна різницева схема збігається зі швидкістю . Її локальна похибка, за умови, що значення функції на попередньому шарі обчислені точно, для (*k*+1)-го шару може бути записана у вигляді:

 (2.4.1)

Розглянемо тепер, як можна оцінити цю похибку. Для знаходження величин *С*1,*С*2 треба для кожного вузла сформувати та розв’язати відповідну систему рівнянь. Її можна одержати, якщо перехід з *k*-го на (*k*+1)-й часовий шар виконати три рази з різними різницевими сітками. Розглянемо один з варіантів послідовності сіток. Нехай розв’язок на *k*-му шарі вже знайдений і треба знайти розв’язок на (*k*+1)-му шарі. Позначимо через *U* значення точного розв’язку в точці (*xi*,*tk*+1). Спочатку за допомогою різницевої схеми знайдемо розв’язок  з кроками *h* і τ, значення якого в точці (*xi*,*tk*+1) позначимо через *uh,τ* . Для нього справедлива рівність:

. (2.4.2)

Далі зробимо перехід з *k*-го на (*k*+1)-й шар з кроками *h* і , двічі застосувавши схему. Одержаний при цьому у точці (*xi*,*tk*+1) розв’язок позначимо через . Оскільки кроки *h* та τ малі, вважатимемо значення *С*1 та *С*2 незмінними в околі точки (*xi*,*tk*+1). Тому справедлива рівність

 (2.4.3)

Повторимо ще раз перехід з *k*-го на (*k*+1)-й шар, але з кроками  і τ. Для цього застосуємо один раз різницеву схему, а значення функції на *k*-му шарі з більш густою сіткою обчислимо за допомогою інтерполяції по числам *uik*. Позначимо знайдений розв’язок через  і одержимо

 (2.4.4)

Рівності (2.4.2)-(2.4.4) утворюють систему алгебричних рівнянь з трьома невідомими величинами *U*,*С*1,*С*2, знайшовши які, можна не тільки визначити похибку , а і уточнити результат *Uik+*1. Запишемо цю систему окремо:

 , (2.4.5)

 (2.4.6)

 (2.4.7)

Її розв’язок має вигляд:

 (2.4.8)

 (2.4.9)

 . (2.4.10)

Підставимо ці вирази у (2.4.5) та одержимо формулу для локальної похибки:

 . (2.4.11)

З похибок *еi*, обчислених для кожного вузла *хi* (*k*+1)-го часового шару, візьмемо максимальну *e*=max . Якщо *e*>*eдоп* , то результати анулюються і виконуються наступні дії:

1. Крок по часу τ зменшується в 2 рази.
2. Крок *h* по координаті *X* зменшується в 2 рази.
3. Повертаємося на шар *tk* .
4. Створюється опорна функція на шарі  з новою сіткою, що ущільнена в 2 рази.

Якщо ж *e*<*eдоп*, то як остаточний результат для (*k*+1)-го шару беруться уточнені за формулою (2.4.10) значення і обираються нові величини кроків *h* та τ для обчислення невідомої функції *U(x,t)* на наступному (*k*+2)-му часовому шарі. Якщо виберемо нові кроки за формулами

 ,  , (2.4.12)

то згідно з (2.4.2) одержимо результат з похибкою

,

яка не повинна перевищувати допустиму *eдоп*. Звідси одержимо рівняння для знаходження коефіцієнтів зміни кроків α та β:

. (2.4.13)

Доцільно вимагати, щоб вклади похибок по часовій та просторовій змінним були однаковими:

.

Звідси можна знайти коефіцієнти зміни кроків для кожного вузла:

 , .

Серед величин α*i* потрібно вибрати єдине значення α, оскільки значення функції *U(x,t)* повинні відшукуватися для однакових *tk*+2 . Поки що будемо розглядати випадок, коли крок *h* при фіксованому *t* теж однаковий, тому потрібно вибрати і єдине значення β. Тут можливі два випадки.

В найпростішому з них можна взяти мінімальні значення по усіх вузлах (α=min α*i* , β=min β*i*), але це призведе до заниження величини кроків та зростання машинного часу. Тому доцільно поступити інакше, враховуючи, що мінімальні значення α та β можуть відповідати різним вузлам. З усіх α*i* виберемо мінімальне, обчислимо залишок допустимої похибки, що припадає на другий доданок в рівнянні (2.4.13) і для кожного з вузлів знайдемо відповідне значення β*i* , з яких теж виберемо мінімальне, тобто реалізуємо такі обчислення:

, , (2.4.14)

,=,β=min β*i* (2.4.15)

Наведемо алгоритм неявного методу зі змінною рівномірною сіткою. Для явної схеми він аналогічний, тільки значення сіткової функції обчислюються без розв’язання СЛАР. Нехай відомі значення функції *U(x,t)* для *t*=*tk* та сітки з кроками *h* і τ. Для переходу на наступний часовий шар *tk*+1=*tk*+τ необхідно виконати такі дії:

1. Обчислити по схемі (\*\*), розв’язуючи систему (2.3.3), з кроками *h*,τ значення *uh,*τ для кожного вузла (*k*+1)-го шару.
2. Обчислити з кроками *h*, значення  , двічі застосувавши схему (\*\*).
3. Обчислити з кроками ,τ значення . Щоб одержати на *k*-му шарі значення опорної функції з кроком  , застосувати інтерполяцію по відомим значенням функції у вузлах з кроком *h* або використати початкову умову на першому кроці.
4. Для кожного вузла за формулою (2.4.11) обчислити *еi* та вибрати найбільшу з них *е* по усім вузлам.
5. Якщо *e*>*e*доп та хоча б один з кроків більший за мінімальний, результати останнього кроку анулювати, кроки *h* та τ зменшити вдвічі, на *k*-му шарі створити нову опорну функцію з сіткою, що ущільнена в два рази. Перейти на п.1.
6. Якщо *e*<*e*доп або кроки дорівнюють мінімальним, результати приймаються. Уточнити результат за формулою (2.4.10). Для кожного вузла за формулою (2.4.14) обчислити коефіцієнт α*i* та з них вибрати мінімальний. Далі за формулою (2.4.15) для кожного вузла обчислити β*і* і з них теж взяти мінімальне.
7. Вибрати нові розміри сітки за формулами (2.4.12).
8. Побудувати на (*k*+1)-му шарі нову опорну функцію з урахуванням нового кроку *h* за допомогою інтерполяції.
9. Прийняти *k*:=*k*+1 та перейти на п.1.

При практичній реалізації алгоритму прийняті деякі обмеження на його параметри. Зокрема, з метою фільтрації високочастотних гармонік розв’язку введені мінімальні значення кроків по часу τmin та по просторовій змінній *h*min. Саме з цих значень починається обчислювальний процес, тобто τпочаткове=τmin та *h*початкове= *h*min. На ці параметри накладено також обмеження зверху: τmax=*t*кінцеве/4, *n*min=5. Крім того, для збереження стійкості процесу введені обмеження на коефіцієнти α та β зміни величин кроків: αmax=βmax=4, αmin=βmin=0.5.[1]

**ІІІ. Результати обчислень**

**3.1 Розрахунок явним методом:**

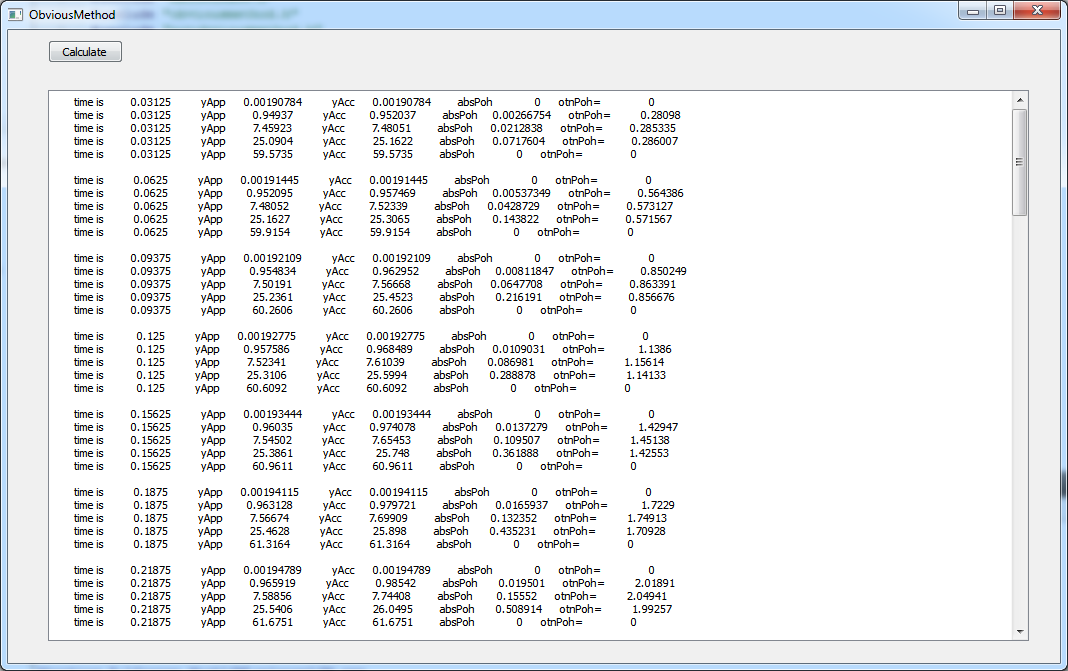
Коефіцієнти

a=0.00001, b=0.00001, A=0.06, B=0.009.

Крок х : 0.25

Крок t: 0.03125;

Результати обчислень : ( перші 7 шарів)



Максимальна абсолютна похибка(на всьому проміжку): 3.31706

Максимальна відносна похибка(на всьому проміжку): 0.66437%

**3.2 Розрахунок неявним методом:**

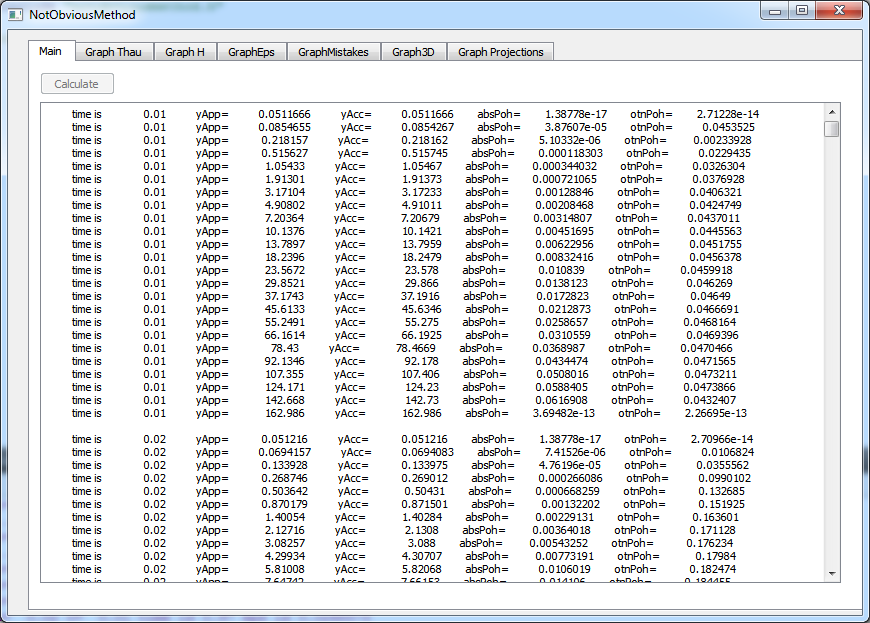
Коефіцієнти

a=0.00001, b=0.00001, A=0.06, B=0.009.

Крок x початковий: 0.0526316

Крок t початковий: 0.010101

Результати обчислень ( перші 2 шари)



Максимальна абсолютна похибка(на всьому проміжку): 5.14874

Максимальна відносна похибка(на всьому проміжку): 3.7548 %

**Висновки:**

Розрахунок неявним методом з адаптивною сіткою дає потрібну точність з значно меншою кількістю ітерацій, через що він більш доцільний у використанні ніж явний метод з фіксованою сіткою. Порівнюючи похибку, можна зробити висновок, що при легкому рівнянні явний метод з фіксованою може дати більшу точність ніж неявний з адаптивною сіткою. Але для складних функцій, що мають різкі перепади функції для розрахунку з фіксованою сіткою потрібно брати дуже малий крок, що збільшує навантаження на ЕОМ. Тому доцільніше використовувати неявний метод з адаптивною сіткою.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Лук’яненко С.О. **Адаптивні обчислювальні методи моделювання об’єктів з розподіленими параметрами — К.**: ІВЦ “Видавництво «Політехніка»”**, 2004. — 236 с.**
2. Лук’яненко С.О. **Числові методи в інформатиці: Навч.посіб. — К.**: НТУУ «КПІ»**, 2007. — 140 с.**
3. Лук’яненко С.О. **Числові методи в розв’язування диференційних рівнянь: Навч.посіб. — К.**: Знання України**, 2010. — 136 с.: іл.. —** Бібліогр **.: с.132-133**
4. Лук’яненко С.О. **Числові методи в інформатиці: Методичні вказівки для виконання курсових робіт / Лук’яненко С.О., Михайлова І.Ю., Третяк В.А. — НТУУ «КПІ» ВПІ ВПК «Політехніка», 2011. — 52 с.**
5. **Qwt - Qt Widgets for Technical Applications — Mode of access : URL :** <http://qwt.sourceforge.net/> **Title from the screen.**
6. **В поддержку Qt, статьи в помощь разработчику – Режим доступа : URL :** <http://qtlinux.narod.ru/> **Название с экрана**
7. **Qt Documentation – Mode of access : URL :** <http://qt-project.org/doc/> **Title from the screen**

**Додаток 1. Опис програми**

Програма написана на мові C++ з застосуванням Qt SDK 5.0.0 – засобів кроссплатформової розробки.

В проекті можна виділити три типи файлів:

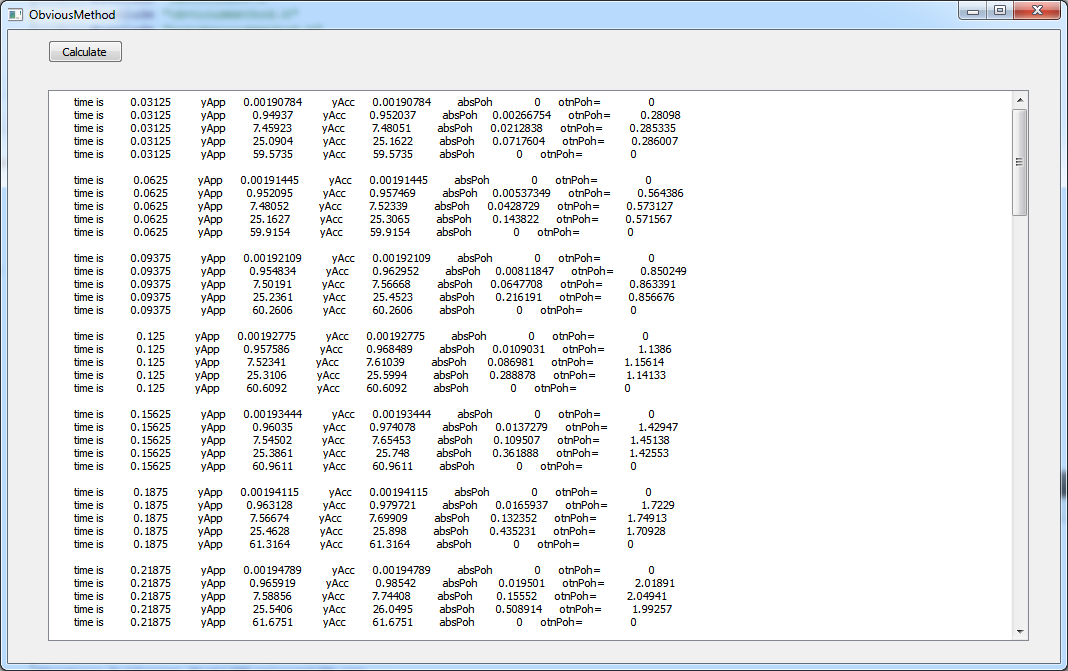
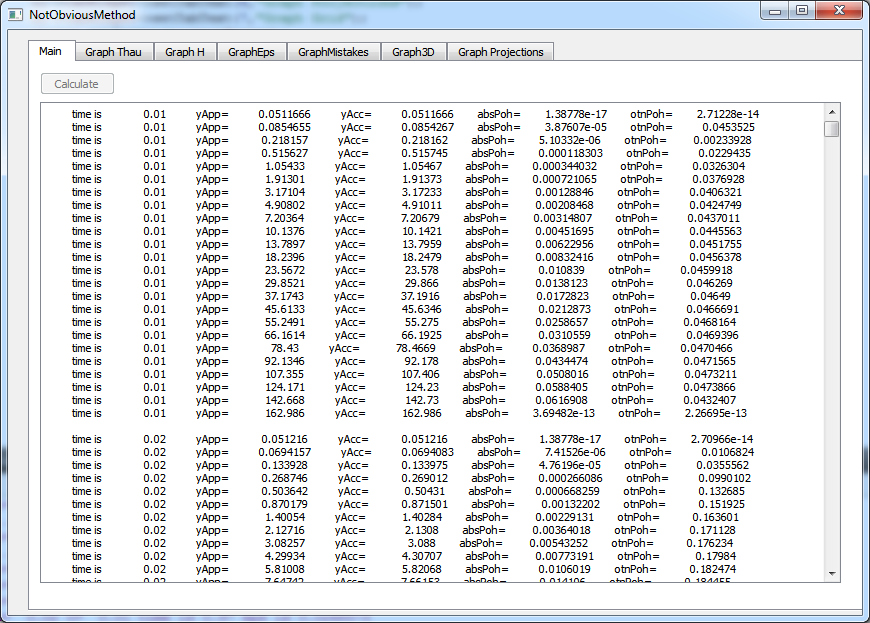
1. class.h – містить оголошення класу та його методів
2. class.cpp – містить реалізацію класу та його методів
3. class.ui – містить графічне зображення форми, яке зберігається в XML-розмітці та може бути редаговано в редакторі Qt Designer.

Програма складається з 4 класів

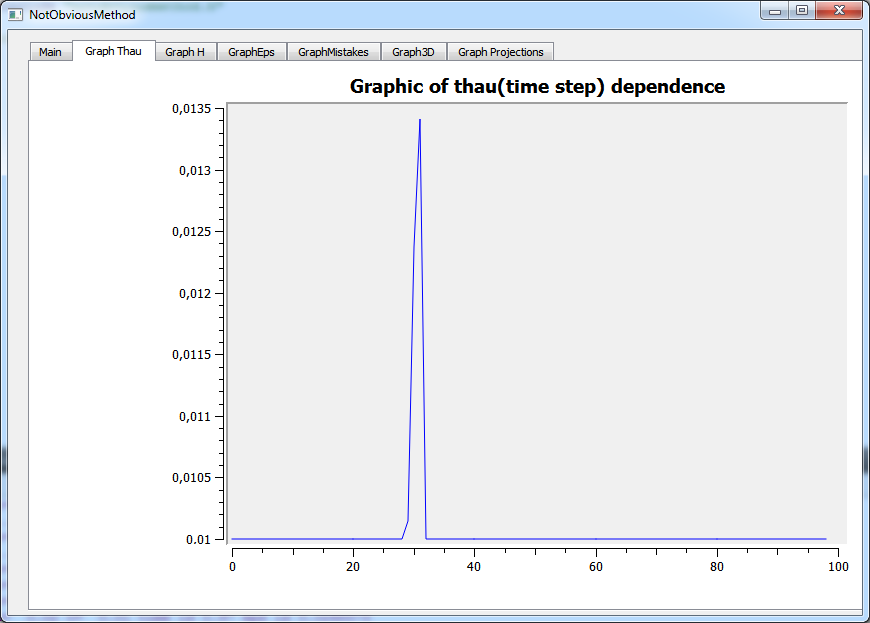
1. Клас головного вікна Main Window, яке користувач бачить при вході в програму
2. Клас ObviousMethod, в якому реалізовані усі необхідні підрахунки для явного методу
3. Клас NotObviousMethod, в якому реалізовані усі необхідні підрахунки для неявного методу
4. Клас QFunc3D, за допомогою методів якого реалізовано виведення 3D графіки

**Ілюстрація роботи програми**

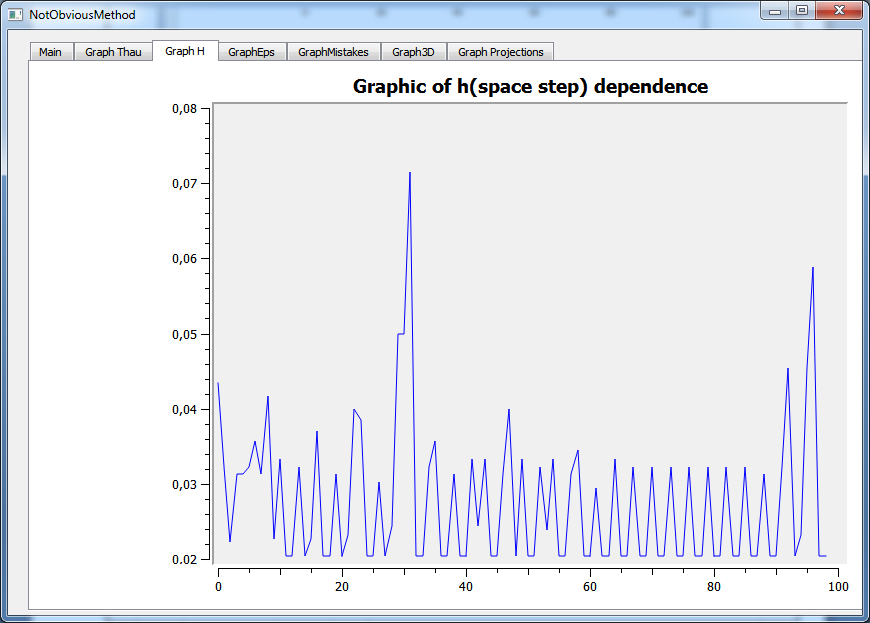
Малюнок1.Головне вікно програми, розв’язок неявним методом.



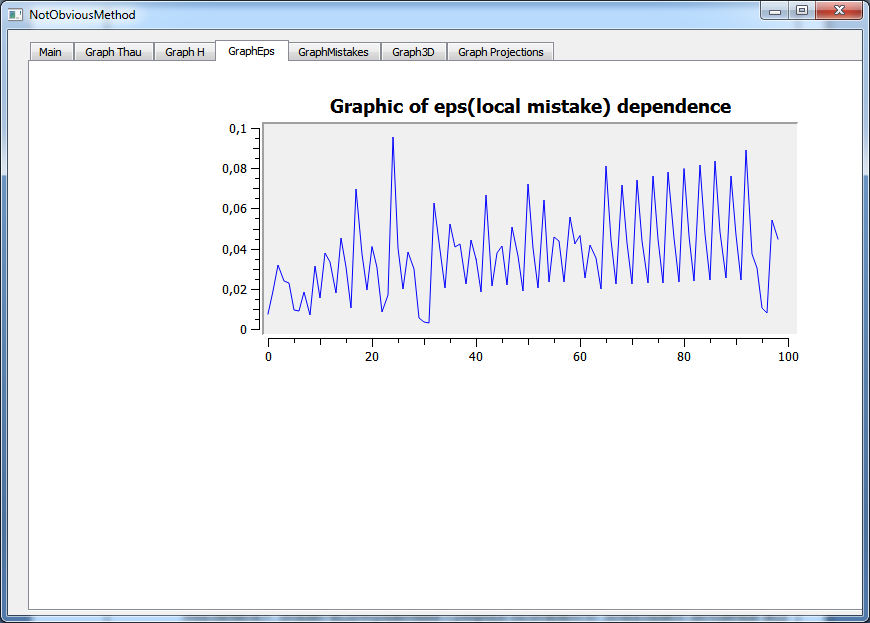
Малюнок2 . Головне вікно програми, розв’язок явним методом.



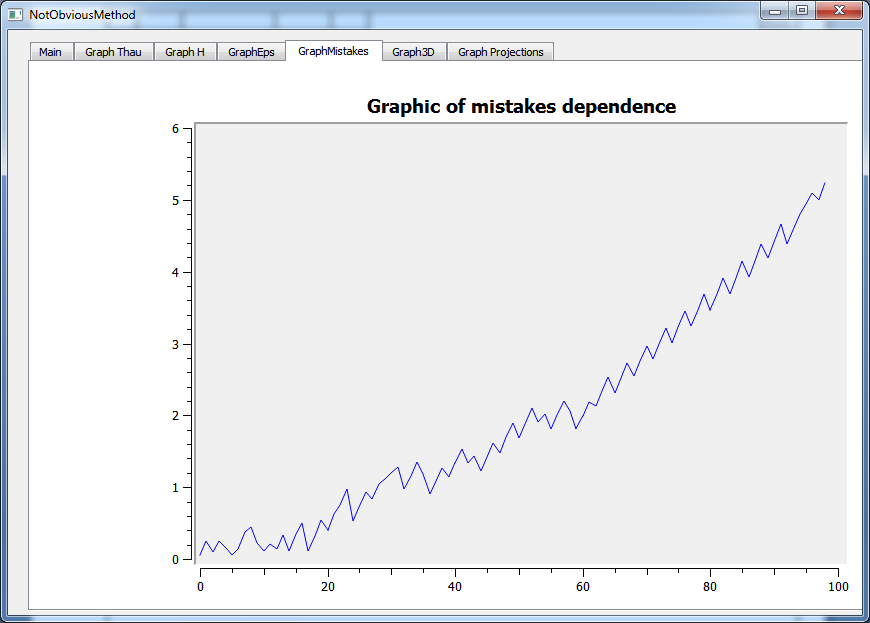
Малюнок3. Вікно відображення графіка залежності кроку по часу від номеру кроку.



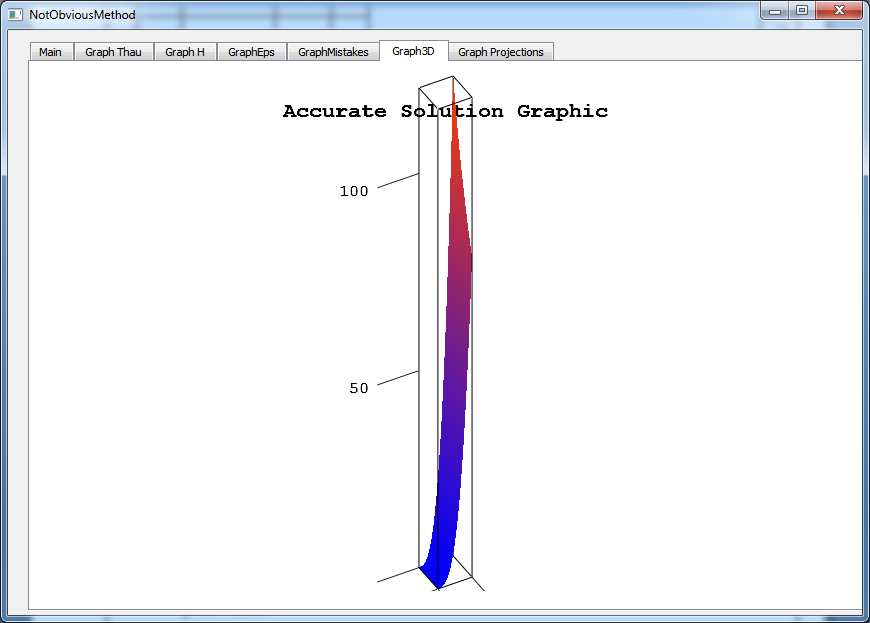
Малюнок4. Вікно відображення графіка залежності кроку по просторовій змінній від номеру кроку.



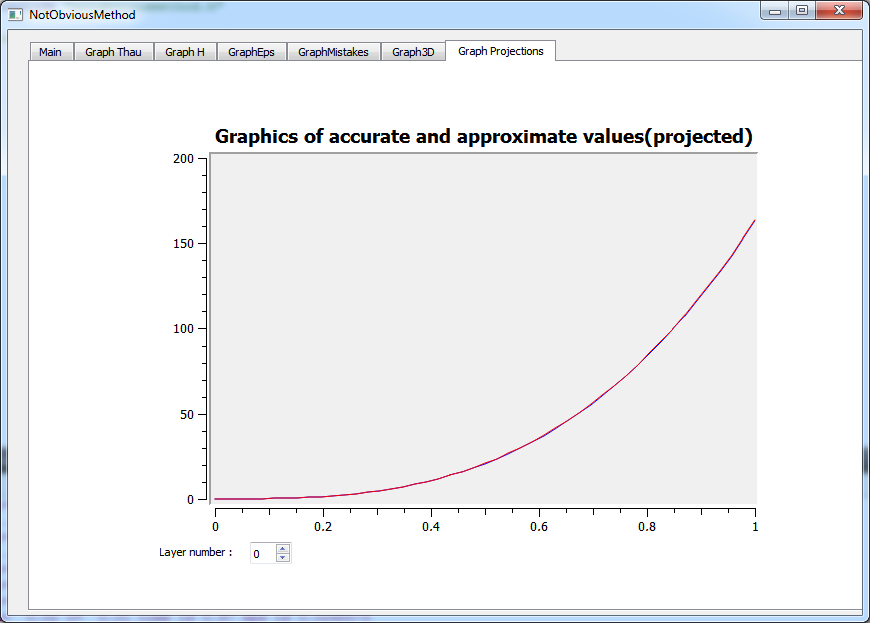
Малюнок5. Вікно відображення графіка залежності локальної похибки від номеру кроку.



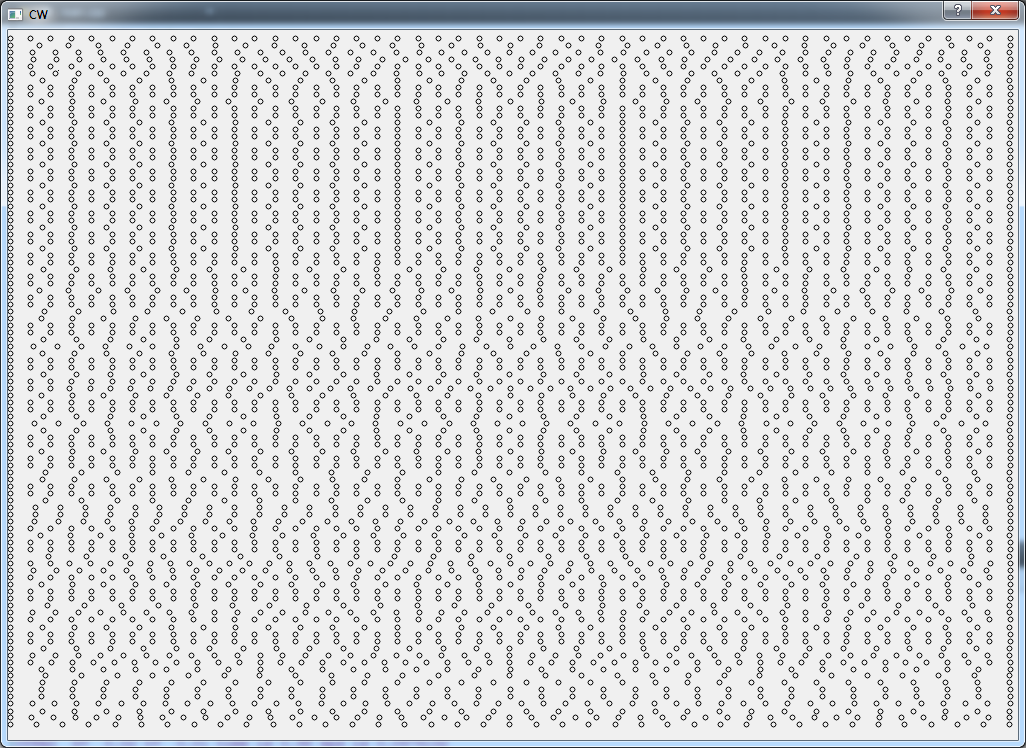
Малюнок6. Вікно відображення графіка залежності похибки накопичення від номеру кроку.



Малюнок7. Вікно відображення 3D графіків точного на наближеного розв’язків.



Малюнок8. Вікно відображення графіків проекцій точного та наближеного розв’язків з вибором номеру часового шару для проекції.



Малюнок9. Відображення сітки розрахункової області.