

Записки за лекциите по
**Числени методи за диференциални
уравнения**
за спец. Приложна математика, Математика
2018/2019 година

Тихомир Иванов,
Галина Люцканова–Жекова

Съдържание

1	Увод	3
1.1	Защо диференциални уравнения?	3
1.2	Производна на функция – дефиниция и апроксимации. Линеаризация. Формула на Тейлър.	5
1.3	Моделиране с диференциални уравнения	9
1.4	Класове диференциални уравнения и задачи, които те описват	10
1.5	Примери за ОДУ	12
1.6	Числени експерименти и решени задачи (практикум)	14
1.7	Цели на настоящия курс	19
2	Числени методи за ОДУ	20
2.1	Задача на Коши за ОДУ от първи ред. Обща теория.	20
2.2	Идея на диференчните методи за задачата на Коши за ОДУ от първи ред	22
2.3	Методи на Ойлер	23
2.3.1	Явен и неявен метод на Ойлер – въведение	23
2.3.2	Локална грешка на апроксимация	25
2.3.3	Схема с тегло. Подобрен метод на Ойлер.	27
2.3.4	Други подходи за получаване на методите на Ойлер	28
2.3.5	A-устойчивост и монотонност	29
2.3.6	Числени експерименти и решени задачи (Практикум)	37
2.4	Методи на Рунге–Кута	40
2.4.1	Явни методи на Рунге–Кута	40
2.4.2	Изследване на A-устойчивост и монотонност за методите на Рунге–Кута	43
2.4.3	Методи с адаптивен избор на стъпката.	45
2.4.4	Числени експерименти (Практикум)	48
2.5	Методи на Адамс	52
2.5.1	Методи на Адамс–Башфорт	53
2.5.2	Методи на Адамс–Мултон	56
2.5.3	Практически въпроси, свързани с реализацията на методите на Адамс	57
2.5.4	Предикторно-коректорни методи	57
2.5.5	Числени експерименти (Практикум)	57
2.6	Сходимост на методите за решаване на ОДУ. Теорема на Лакс. Метод на Рунге за практическа оценка на грешката и реда на сходимост.	62
2.7	Сравнение на методите за решаване на задача на Коши за ОДУ	62

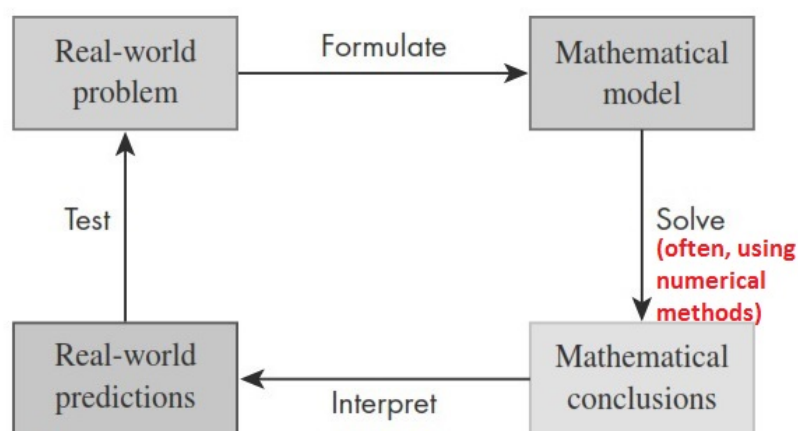
2.7.1	Основни предимства и недостатъци на методите	62
2.7.2	Числени експерименти (Практикум)	63
2.8	Допълнителни задачи	68
3	Диференчни методи за решаване на ЧДУ	73
3.1	Въведение в задачите, които ЧДУ описват. Уравнение на непрекъснатостта.	73
3.2	Диференчни методи за параболични ЧДУ. Линейно 1D нестационарно уравнение на дифузията/топлопроводността.	75
3.2.1	Явна двуслойна диференчна схема при гранични условия на Дирихле	76
3.2.2	Апроксимиране на гранични условия на Нойман и Робин	80
3.2.3	Неявни схеми. Чисто неявна схема. Схема на Кранк–Никълсън.	83
3.2.4	Числени експерименти (Практикум)	86
3.3	Устойчивост на диференчните схеми за решаване на ЧДУ	94
3.3.1	Принцип за максимума. Условие за положителност на коефициентите. Устойчивост в $\ \cdot\ _{h,\infty}$ -норма	97
3.3.2	Метод на хармониките за изследване на устойчивост в мрежова $\ \cdot\ _{h,2}$ -норма	100
3.4	Диференчни методи за хиперболични уравнения	103
3.4.1	Двуслойни монотонни схеми с първи ред на точност за уравнението на преноса	103
3.4.2	Схеми с втори ред на точност за уравнението на преноса, устойчиви в мрежова l_2 -норма	105
3.4.3	Диференчни схеми за уравнението на струната	106
3.4.4	Числени експерименти - практикум	109
3.5	Диференчни схеми за решаване на нестационарни задачи – обобщение	113
3.6	Диференчни схеми за решаване на стационарни задачи	113
3.7	Допълнителни задачи	113
4	Метод на крайните елементи	116
4.1	По части полиноми на една променлива. Интерполация, L_2 -проекция, априорни оценки на грешката.	116
4.1.1	Пространство на линейните полиноми P_1 . Интерполация.	117
4.1.2	Пространство на по части линейните полиноми V_h . Интерполация.	118
4.1.3	L_2 -проекция	120
4.2	МКЕ за 1D задачи с хомогенни гранични условия на Дирихле. Вариационна формулировка. Задача на Ритц–Гальборкин. Дискретизация и извеждане на линейна алгебрична система. Прилагане на метода при по-общи гранични условия.	124
4.2.1	Обща идея на метода	124
4.3	Прилагане на МКЕ при нехомогенни условия на Дирихле и условия на Нойман	127
4.4	Идея на МКЕ в 2D	127

Глава 1

Увод

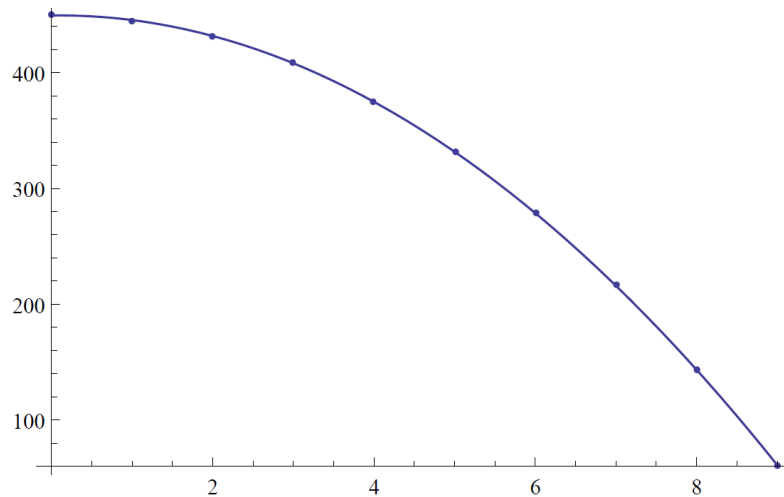
1.1 Защо диференциални уравнения?

Да припомним, че **числените методи са техники за (приближено) решаване на математически задачи с помощта на компютър (т.е. като се използват много на брой аритметични операции)**. Те са основен инструмент при решаването на математическите задачи, които възникват в практиката. Причините за това са, че реалните процеси често водят до решаване на задачи, които нямат аналитично решение и поради това се налага да бъдат решени приближено, което рядко може да стане без помощта на компютъра:



Нека коментираме накратко какво представляват (много често) математическите модели като математически задачи.

В най-простите случаи по дадени експериментални данни можем да намерим функция, която ги апроксимира по метода на най-малките квадрати, чрез интерполация или по друг начин. Например за свободно падане на тяло можем да намерим парабола, която съответства на експерименталните данни, приведени по-долу (означени с точки):

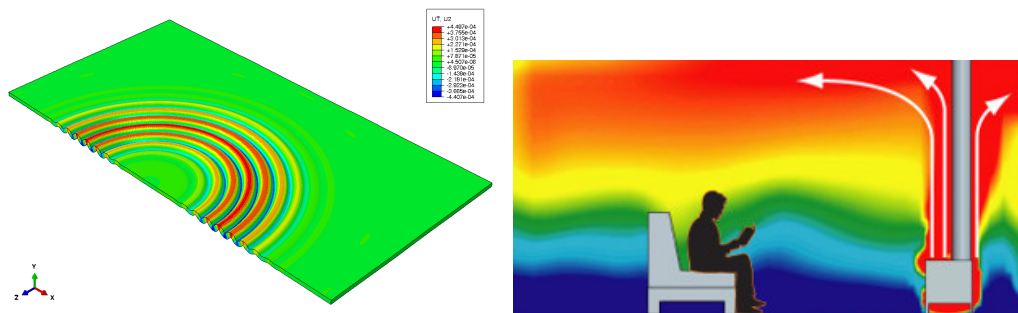


В курса по ЧМЛА (за спец. Приложна математика) направихме и физични експерименти, на база на които установихме, че има линейна зависимост между това колко е разтегната една пружина и силата, която възниква при това.

Впрочем базовите закони на физиката, биологията и т.н. са получени именно по този начин, т.е. чрез приближаване на експериментални данни с функция от даден вид. Това обаче може да стане само за базовите зависимости – например от горната фигура е ясно, че можем да търсим квадратична зависимост.

Макар и прости обаче примерите, които разгледахме, ни показват основното, което искаме при описването на повечето реални процеси – **да намерим функцията**, която отразява зависимостите между величините, съществени за процеса. Най-често ние търсим функция на следните независими променливи:

- времето t ;
- до три пространствени променливи – x, y, z .



N.B. 1

Обикновено, когато искаме да опишем даден реален процес, търсим функция, която описва зависимостта между величините, характеризиращи процеса.

Когато искаме да опишем по-комплексна физическа система обаче, само наблюденията няма да са ни достатъчни. Математическите модели най-често се базират на различни физични закони, т.е. известни зависимости между определени величини. Например закони за запазване, II закон на Нютон и др.

Много често тези закони ни казват **как нещата се изменят**. Не е трудно да се убедим, че нещата около нас се изменят – както във времето, така и в пространството.

N.B. 2

На езика на математиката това **как се изменя дадена величина се описва от нейната производна**.

От казаното дотук можем да заключим, че е естествено в математическите модели да търсим функции и в моделите да участват производните на функциите. Това обяснява защо **ДУ са основният апарат на математическото моделиране**. Огромна част от математическите модели на реални процеси представляват именно ДУ.

N.B. 3

Повечето ДУ не могат да бъдат решени аналитично! Единственият начин за тяхното решаване е приближено с помощта на числени методи.

1.2 Производна на функция – дефиниция и апроксимации. Линеаризация. Формула на Тейлър.

Преди да коментираме по-подробно въпроса за моделирането с ДУ, нека си припомним някои неща за производната.

Дефиниция 1: Производна на функция

Производната на функцията u в точката t се дефинира чрез

$$u'(t) \equiv \frac{du}{dt} = \lim_{h \rightarrow 0} \frac{u(t+h) - u(t)}{h},$$

ако последната граница съществува.

Нека се спрем на тази дефиниция и да я коментираме подробно, тъй като тя е ключова за настоящия курс (и не само). Когато изучаваме дадено математическо понятие, е добре да имаме представа за него от три гледни точки (когато това е възможно):

- интуитивна/практическа – математиката е език и ние трябва да знаем какво описват обектите, с които тя борави; да отбележим, че в исторически план много от математическите понятия, в т.ч. производна, са въведени при изучаването на реални процеси (например Нютон е изучавал движението на телата);
- аналитична (вече дадохме такава дефиниция за производната);
- геометрична.

Н.В. 4

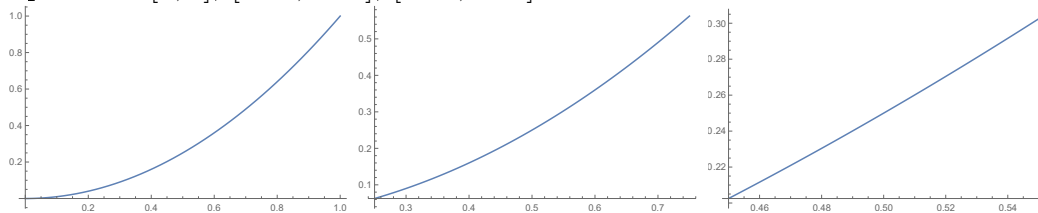
От практическа гледна точка, **производната описва моментната скорост на изменение на величината u за единица изменение на величината t .**

Да разгледаме няколко примера.

Пример 1. Компания, която произвежда столове, има фиксирани разходи 5000 лв. и производствен разход 30лв. на стол. Това означава, че функцията на разходите има вида $cost(q) = 5000 + 30q$. Производната на тази функция е $\frac{d \text{ cost}}{dq} = 30\text{лв./стол}$ или за всеки допълнителен стол разходите на компанията ще нарастнат с 30 лв. Да обърнем внимание, че мерната единица е частното на мерната единица на функцията и мерната единица на аргумента. В случая лв./стол.

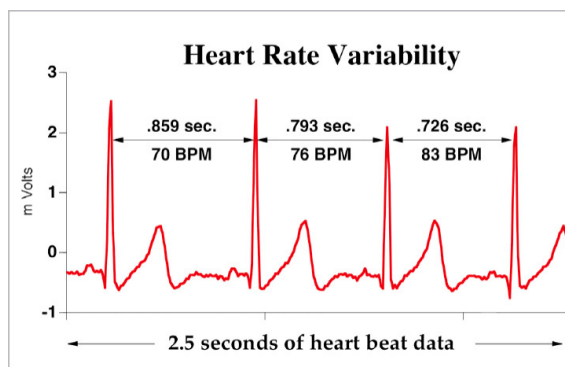
Тъй като функцията в настоящия пример е линейна, то изменението ѝ е еднакво навсякъде, т.е. ако увеличим броя на столовете от q на $q + 1$, разходите ще се увеличат с 30 лв. Ако увеличим броя на столовете от q на $q + 2$, ще се увеличат с 60 лв. и т.н. Изобщо казано, скоростта на изменение е отношението на изменението на стойността на функцията (разходите) и изменението на аргумента (броя столове).

Пример 2. Оказва се, че можем да обобщим идеята от предходния пример и за нелинейни функции. Да разгледаме графиката на параболата $f(x) = x^2$ в интервалите $[0, 1]$, $[0.25, 0.75]$, $[0.45, 0.55]$:



Както виждаме, достатъчно близо до точката $x = 0.5$ графиката на нелинейната функция прилича на права (т.е. на линейна функция) и следователно ние можем да дефинираме **моментната скорост** на изменение в тази точка като отношението на изменението на функцията и изменението на аргумента, както при примера за столовете (но за безкрайно малко изменение на аргумента) – сравни с дефиницията за производна.

Пример 3. Сърдечният ритъм на човек може да бъде измерен по следния начин. Нека $u(t)$ е броят удари на сърцето, а времето t е измерено в минути. Тогава сърдечният ритъм в момента от време t (т.е. броят удари в минута) е $u'(t)$ – с колко удара се изменя за една минута общият брой. Можем да интерпретираме $u''(t)$ като вариативността на сърдечния ритъм – колко бързо се изменя той. Последното също е съществен клиничен показател.



Пример 4. Законът на Мур гласи, че броят на транзисторите в една интегрална схема се удвоява на всеки две години. Тогава ние можем да запишем математически тази зависимост като

$$\frac{du}{dt} = \frac{1}{2}u,$$

където u е броят на транзисторите, а t е времето, измерено в години.

Ще се върнем след малко на геометричната интерпретация на производната. От практическа гледна точка (когато ще работим с компютър) имаме проблем с граничния преход. Тази операция е в основата на много от трудностите в анализа. Ето защо ще разгледаме няколко **формули за числено диференциране**:

- Формула с разлика напред с грешка $O(h)$:

$$u'(t) \approx \frac{u(t+h) - u(t)}{h}; \quad (1.1)$$

- Формула с разлика назад с грешка $O(h)$:

$$u'(t) \approx \frac{u(t) - u(t-h)}{h}; \quad (1.2)$$

- Формула с централна разлика с грешка $O(h^2)$:

$$u'(t) \approx \frac{u(t+h) - u(t-h)}{2h}. \quad (1.3)$$

На практика в горните формули “сме пропуснали” граничния преход от дефиницията за производна, т.е. апроксимираме моментната скорост на изменение със средната скорост за крайно (малко) изменение на аргумента h или $2h$. Засега просто формулирахме остатъчните членове (или грешките на апроксимацията). Те са извеждани в курса “Числени методи на анализа”, но ние също ще обърнем внимание на тяхното получаване малко по-късно в курса, тъй като самият подход за това ще бъде важен за нашите цели.

N.B. 5

Приведените по-горе формули ще заемат централно място в настоящия курс, тъй като ще позволят да заместваме производните в разглежданите ДУ с израз, в който има само аритметични операции.

Впрочем, разглеждайки горните формули, в частност (1.1), можем да припомним и още една централна идея на математическия анализ – **линеаризация на функция**. От (1.1) получаваме

$$u(t+h) \approx u(t) + u'(t)h, \quad (1.4)$$

като остатъчният член е $O(h^2)$. Дясната страна в (1.4) ще наричаме линеаризация на функцията u около точката t .

Н.В. 6

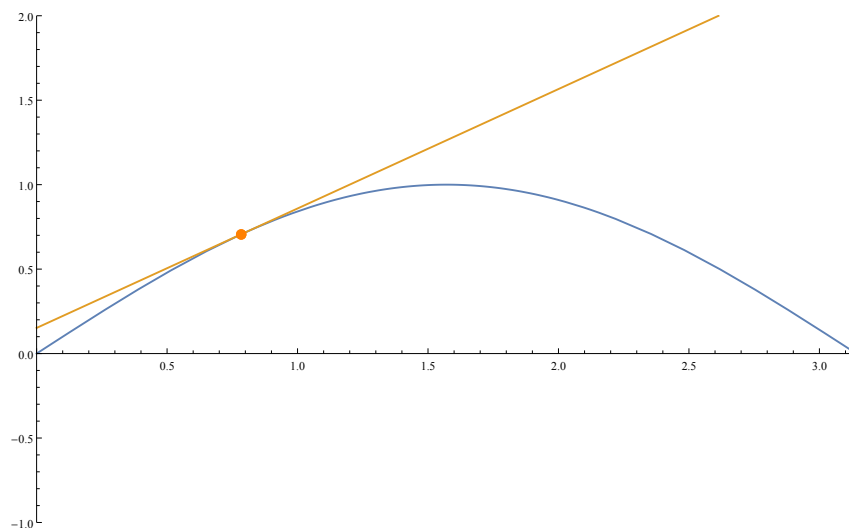
Идеята на линеаризацията е, **че приближаваме (в достатъчно малка околност на точката) нелинейния обект (с който по принцип се работи трудно) с нещо линейно**. На тази идея се базират много съществени резултати от нелинейния анализ.

Геометрично, **графиката на линеаризацията е допирателната към графиката на функцията в точката t , а производната е наклонът на тази допирателна**.

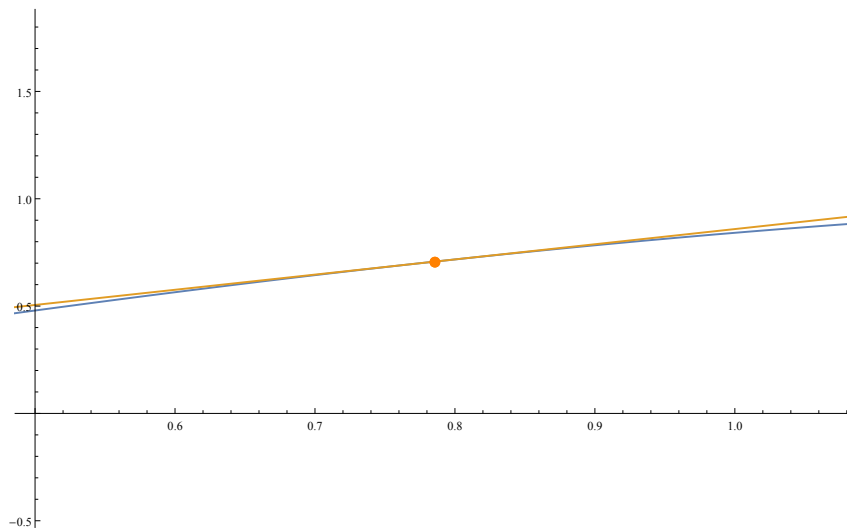
Пример 5. Да разгледаме функцията $u(t) = \sin t$ и да я линеаризираме около точката $t = \pi/4$:

$$u(t+h) \approx \sin \frac{\pi}{4} + h \cos \frac{\pi}{4}.$$

Графиките на функцията и нейната линеаризация изглеждат по следния начин:



Ако ги разгледаме в по-малка околност на точката $t = \pi/4$, виждаме, че двете започват да стават визуално неразличими:



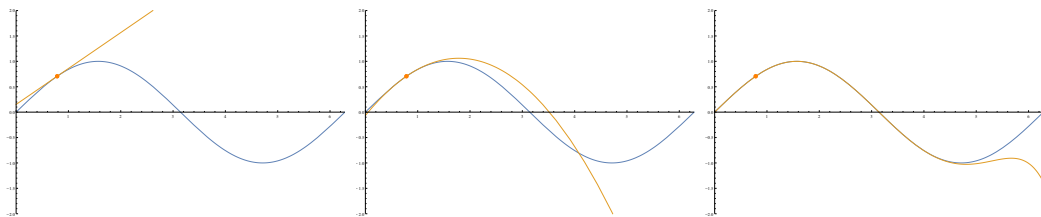
Н.В. 7

Формулата на Тейлър е обобщение, което ни позволява да направим по-добро приближение на функцията в околност на дадената точка, като използваме производни и от по-висок ред:

$$u(t+h) = u(t) + \frac{u'(t)}{1!}h + \frac{u''(t)}{2!}h^2 + \dots + \frac{u^n(t)}{n!}h^n + O(h^{n+1}).$$

Да забележим, че очевидно за да бъде горното в сила, функцията u трябва да е достатъчно гладка – да има поне $n + 1$ непрекъснати производни.

Пример 6. Продължавайки предходния пример, ще визуализираме в интервала $[0, 2\pi]$ графиките на полиномите на Тейлър от степени от 1, 2, 10 за функцията $\sin x$, получени при развитие около точката $t = \pi/4$:

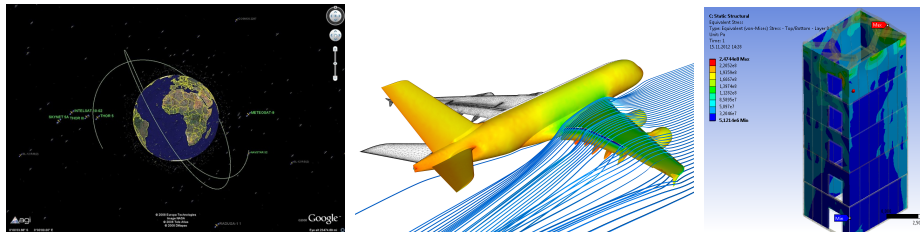


Както лесно можем да се уверим, с увеличаването на степента получаваме все по-точно приближение на оригиналната функция (при това във все по-голям интервал приближението и функцията са визуално неразличими).

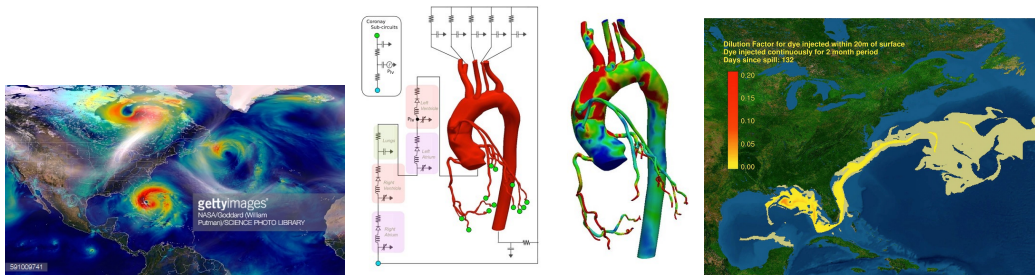
1.3 Моделиране с диференциални уравнения

Да се върнем на въпроса, че в математическите модели много често търсим функция, която описва съответния процес и, тъй като моделите се базират на закони, които ни казват как нещата се изменят, то моделите представляват ДУ.

Но кому е необходимо да разглежда такива модели? Отговорът е, че на практика трудно може да се намери област от науката и технологиите, в която да не се използва апаратът на математическото моделиране.



В реално време се изчисляват траекториите на сателитите. Самолетните инженери симулират обтичането на крилата на самолета от въздушния поток и напреженията, възникващи в неговата структура. Строителните инженери симулират поведението на дадена сграда при различни ситуации. От икономическа гледна точка, тези компютърни симулации позволяват пестенето на милиони, тъй като не е необходимо да се правят прототипи, чрез които да се тестват всички възможни поведения на системата.



Математически модели и компютърни симулации се правят още за предвиждания в областта на метеорологията, за описване на различни процеси в медицината (например потока на кръвта), за опазване на околната среда (например за симулация на развитието на даден петролен разлив) и т.н.

Разбира се, съществуват и други видове математически модели, освен диференциалните уравнения, например в последните години все повече се говори за “big data” и се развиват алгоритми за обработката на тези големи масиви от данни. Все пак обаче диференциалните уравнения описват огромна част от възникващите реални задачи.

Да подчертаем още веднъж, че почти няма математически модел, описван с ДУ, който може да бъде решен точно, с помощта на познатите аналитични техники. Единственият начин е да търсим приближени решения числено, с помощта на компютър.

1.4 Класове диференциални уравнения и задачи, които те описват

Основните два класа диференциални уравнения са обикновените и частните диференциални уравнения.

1. **Обикновени диференциални уравнения (ОДУ).** Това са уравнения, в които участва неизвестна функция на една променлива и нейните производни. Например търсим функция $u(x)$ такава, че

$$u''(x) + u'(x) = x + 1.$$

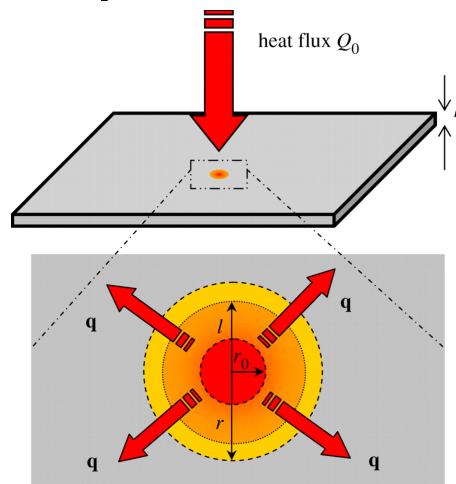
Разглеждането на ОДУ е естествено например в следните случаи.

- Разглеждайки стационарното състояние на даден процес, времето не участва като независима променлива.

- Ако моделираме обект, който може да се разглежда като едномерен (например кабелът на изображението по-долу, който е дълъг и тънък), е естествено неизвестната функция да е на една независима (пространствена) променлива (т.е. търсим $u(x) = ?$).

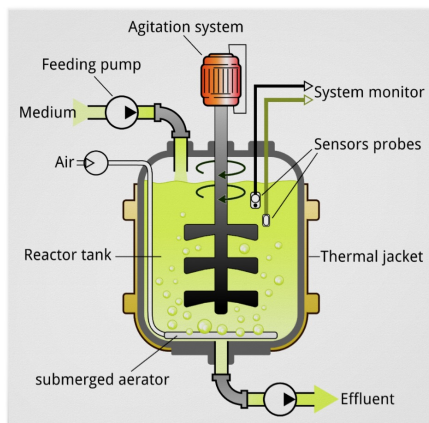


- Същото се отнася и в случая, когато процесът, който описваме, се характеризира с определена симетрия. Например, ако имаме радиална симетрия, както е в случая с разпространение на топлината по-долу, неизвестната функция ще бъде $u(r)$, където r е разстоянието до центъра.



- Другата принципна ситуация, при която неизвестното е функция на една променлива, е случаят, когато моделираме хомогенен процес (или друг процес, който не зависи от пространствени променливи). Ако средата е хомогенна, т.е. изследваната величина не зависи от точката в пространството, единствената независима променлива е времето, т.е. търсим $u(t) = ?$.

Като пример можем да дадем процесите в хомогенен биореактор (вж. схемата по-долу). В него има система за хомогенизиране на средата (например бъркалка), която осигурява равномерно достигане на хранителните вещества до организмите, отглеждани в реактора. Тогава концентрацията им ще зависи само от времето, но не и от точката в пространството.



Schematic Structure of a Ba... by chartsanddiagrams

Zazzle

Често, предположението за хомогенност се прави като първо приближение при моделирането на даден реален процес, който всъщност не е хомогенен.

2. **Частни диференциални уравнения (ЧДУ).** В частните диференциални уравнения, както подсказва името им, неизвестните са функции на няколко променливи и участват техните частни производни. Например търсим функция $u(x, y)$ такава, че

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0.$$

На този етап ще се ограничим до това да направим класификация на ЧДУ, които ще разглеждаме в курса, по отношение на променливите, участващи в неизвестната функция. В зависимост от това дали участва времевата променлива, или не, говорим за нестационарни и стационарни задачи.

- Стационарните задачи (в които не участва времето t) биват 2-мерни (неизвестната функция е $u(x, y)$) и 3-мерни $u(x, y, z)$. Те описват равновесното състояние на даден процес.
- Нестационарните задачи описват времевата еволюция на даден процес. Те могат да бъдат 1-, 2- или 3-мерни. Съответно неизвестната функция е $u(x, t)$, $u(x, y, t)$, $u(x, y, z, t)$.

Съществуват още диференциални уравнения със закъснения, интегро-диференциални уравнения и др., но те остават извън рамките на настоящия курс.

1.5 Примери за ОДУ

Тук ще разгледаме само няколко примера за ОДУ, които ще използваме по-нататък като моделни задачи, върху които ще изложим и изследваме разглежданите числени методи. Повече примери с приложна значимост ще разгледаме в частта "Case Study". Разнообразни примери могат да бъдат намерени и в [8].

- Важна причина за значимостта на диференциалните уравнения е фактът, че голяма част от законите на физиката, химията и т.н. представляват именно диференциални уравнения. Като пример ще дадем фундаменталния закон на Нютон, на който се основава описването на всяко движение в класическата механика:

$$F(t) = ma(t) \implies F(t) = m \frac{d^2 u}{dt^2},$$

където m е масата на дадено тяло, $F(t)$ и $a(t)$ са съответно (резултантната) сила, действаща върху тялото, и ускорението в момента от време t , а $u(t)$ е отместването на тялото спрямо началния момент.

И така, Законът на Нютон представлява едно ОДУ от втори ред (т.е. в него участва втората производна на неизвестната функция u).

- Математическото моделиране навлиза все повече в областта на биологията (за повече информация вж. например класическата монография [7]). Ние ще се спрем на два класически модела, описващи развитието на дадена популация. Нека $u(t)$ е числеността на популацията в момента от време t . Ще разгледаме моделите на Малтус и Верхълст.

- Закон на Малтус за експоненциален растеж. Той се основава на предположението, че изменението на числеността на популацията е пропорционално на нейната численост. Записано в термините на математиката, това представлява едно ОДУ от първи ред по отношение на u :

$$\frac{du}{dt} = ru,$$

където параметърът r е коефициентът на пропорционалност и се нарича коефициент на естествен прираст.

Лесно може да се провери, че

$$u(t) = u(0)e^{rt}$$

е решение на модела на Малтус. То открива един съществен проблем на този модел – при $r > 0$ той предвижда неограничено нарастване на популацията, което, разбира се, не е реалистично.

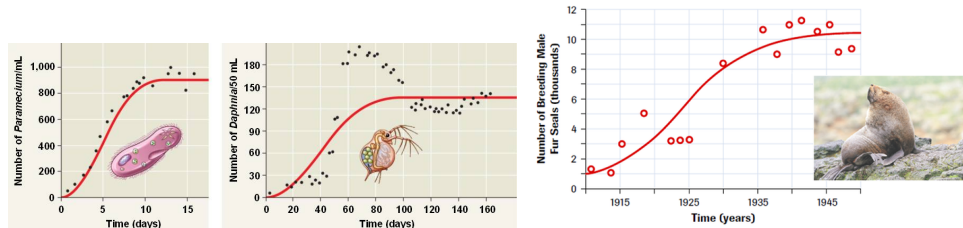
- Закон на Верхълст за логистичен растеж. Той се основава на идеята, че съществува максимална численост K , която жизнената среда може да поддържа. Следователно, ако $u = K$, то популацията не би трябвало да се изменя повече, т.е. $du/dt = 0$ при $u = K$. Тя трябва да намалява при $u > K$ и да расте при $u < K$. Тези наблюдения се описват от модела

$$\frac{du}{dt} = ru(K - u).$$

Точното решение на последното уравнение е

$$u(t) = \frac{Ku(0)e^{rt}}{K + u(0)(e^{rt} - 1)}.$$

Емпиричните сведения потвърждават валидността на този модел, когато популацията се намира при благоприятни условия:



Моделите на Малтус и Верхълст представляват ОДУ от първи ред, като в дясната страна независимата променлива t не участва явно. Такива уравнения се наричат **автономни**.

1.6 Числени експерименти и решени задачи (практикум)

Забележка. В настоящия параграф за стъпката във формулите за числено диференциране е използвано означение Δt вместо h , както е в предишните параграфи. Авторите молят за извинение за липсата на последователност в означенията. В бъдещи версии на записките този недостатък ще бъде отстранен.

Пример 7. Да приближим производната на функцията $f(x) = e^x$ в интервала $[0, 10]$, като използваме формули (1.1), (1.2) и (1.3) при $\Delta t = 0.5$, $\Delta t = 0.001$ и $\Delta t = 10^{-15}$.

На графиките по-долу са изобразени производната на e^x (т.е. e^x) в синьо и приближенията ѝ, използвайки съответно формули (1.1), (1.2) и (1.3) с $\Delta t = 0.5$, в оранжево:

- $(f[t+\Delta t]-f[t])/\Delta t$

```
In[81]:=  $\Delta t = 0.5;$ 
          $f[t_] = \text{Exp}[t];$ 
          $\text{DerF}[t_] = \text{D}[f[t], t];$ 
          $\text{plotd1} = \text{Plot}\{\text{DerF}[t], (f[t + \Delta t] - f[t]) / \Delta t\}, \{t, 0, 10\};$ 
```

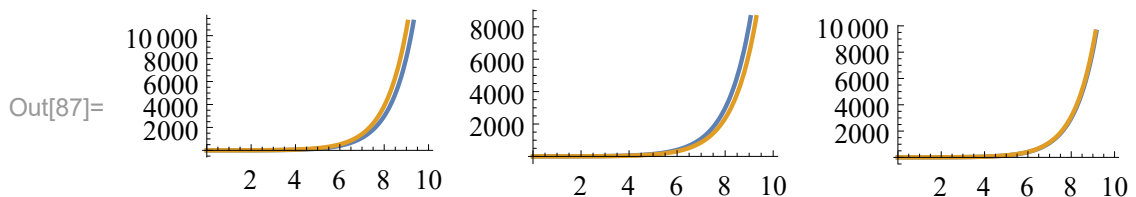
- $(f[t] - f[t-\Delta t])/\Delta t$

```
In[85]:=  $\text{plotd2} = \text{Plot}\{\text{DerF}[t], (f[t] - f[t - \Delta t]) / \Delta t\}, \{t, 0, 10\};$ 
```

- $(f[x+\Delta t] - f[x-\Delta t])/(2\Delta t)$

```
In[86]:=  $\text{plotd3} = \text{Plot}\{\text{DerF}[t], (f[t + \Delta t] - f[t - \Delta t]) / (2 \Delta t)\}, \{t, 0, 10\};$ 
```

```
In[87]:=  $\text{GraphicsRow}\{\text{plotd1}, \text{plotd2}, \text{plotd3}\}$ 
```



Очевидно е, че формулата с централна разлика дава най-добро приближение. Сега да видим какво се случва за различни стойности на Δt . Очакваме при намаляване на Δt да се получи по-добро приближение. Резултатите от експериментите за $\Delta t = 0.5$, $\Delta t = 0.001$ и $\Delta t = 10^{-15}$ са показани по-долу:

- $\Delta t=0.5$

```
In[8]:=  $\text{plotStep5} = \text{Plot}\{\text{DerF}[t], (f[t + 0.5] - f[t]) / 0.5\}, \{t, 0, 10\};$ 
```

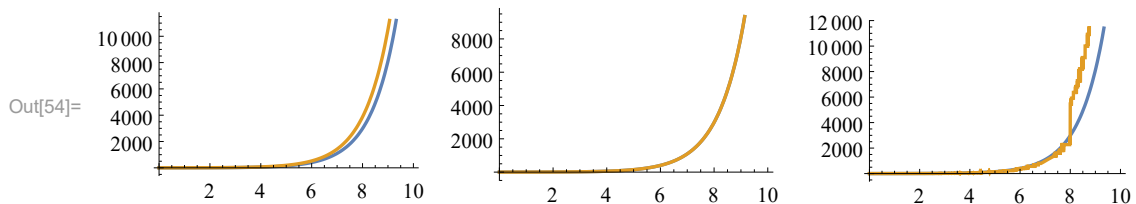
- $\Delta t=0.001$

```
In[9]:=  $\text{plotStep001} = \text{Plot}\{\text{DerF}[t], (f[t + 0.001] - f[t]) / 0.001\}, \{t, 0, 10\};$ 
```

- $\Delta t=10^{(-15)}$

```
In[10]:=  $\text{plotStepMin15} = \text{Plot}\{\text{DerF}[t], (f[t + 10^{(-15)}] - f[t]) / 10^{(-15)}\}, \{t, 0, 10\};$ 
```

```
In[54]:=  $\text{GraphicsRow}\{\text{plotStep5}, \text{plotStep001}, \text{plotStepMin15}\}$ 
```



На втората графика грешката действително е намаляла. При $\Delta t = 10^{-15}$ обаче полученото приближение е по-лошо. Причината за това е, че числата в компютъра се представят с фиксиран брой значещи цифри. По тази причина, когато машинната точност бъде достигната (и следователно намаляването на стъпката не може повече да подобрява точността), грешката ще започне да расте заради увеличаването на броя на операциите. Това е и причината за появата на “шума” на третата графика.

Да припомним следните две дефиниции.

Дефиниция 2: Абсолютна грешка

Абсолютна грешка наричаме разликата между точната и приближената стойност при дадена апроксимация:

$$\varepsilon_a := \text{exact value} - \text{approximation}.$$

Дефиниция 3: Относителна грешка

Относителна грешка дефинираме по следния начин:

$$\varepsilon_r := \frac{\text{exact value} - \text{approximation}}{\text{exact value}} = \frac{\varepsilon_a}{\text{exact value}}.$$

Абсолютната грешка за формулите с разлика напред и разлика назад, както казахме, е $O(\Delta t)$. Това означава, че ако намалим Δt десет пъти, грешката ще намалее от порядъка на 10 пъти. За да верифицираме този теоретичен резултат, нека да начертаем графиките на абсолютните грешки при $\Delta t = 0.001$, $\Delta t = 0.0001$, $\Delta t = 0.000001$,

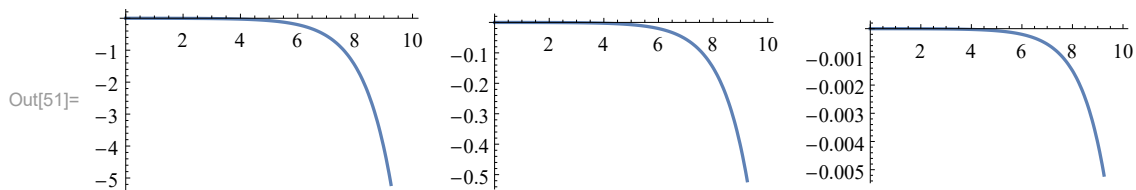
```
In[47]:= f[x_] := E^x
```

```
absErrorD1Step001 = Plot[DerF[t] - (f[t + 0.001] - f[t]) / 0.001, {t, 0, 10}];
```

```
absErrorD1Step0001 = Plot[DerF[t] - (f[t + 0.0001] - f[t]) / 0.0001, {t, 0, 10}];
```

```
absErrorD1Step000001 = Plot[DerF[t] - (f[t + 0.000001] - f[t]) / 0.000001, {t, 0, 10}];
```

```
GraphicsRow[{absErrorD1Step001, absErrorD1Step0001, absErrorD1Step000001}]
```

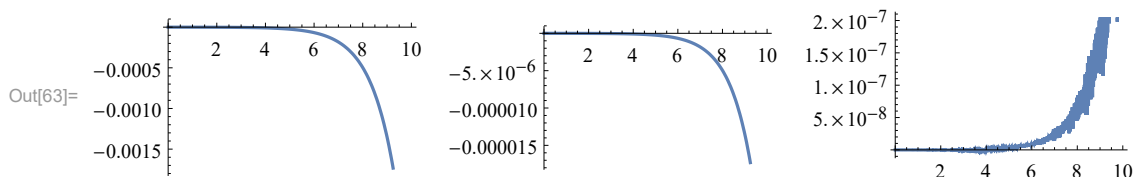


Аналогично, ако намалим Δt във формулата с централна разлика десет пъти, грешката ще намалее от порядъка на 100 пъти. При $\Delta t = 0.001$, $\Delta t = 0.0001$, $\Delta t = 0.000001$ получаваме следните графики за абсолютната грешка:

```

In[60]:= absErrorD3Step001 = Plot[DerF[t] - (f[t + 0.001] - f[t - 0.001]) / 0.002, {t, 0, 10}];
absErrorD3Step0001 = Plot[DerF[t] - (f[t + 0.0001] - f[t - 0.0001]) / 0.0002, {t, 0, 10}];
absErrorD3Step00001 = Plot[DerF[t] - (f[t + 0.00001] - f[t - 0.00001]) / 0.00002, {t, 0, 10}];
GraphicsRow[{absErrorD3Step001, absErrorD3Step0001, absErrorD3Step00001}]

```



Разбира се, ако изберем много малко Δt грешките от закръгляване ще започнат да доминират над точността на апроксимацията, както е видно на последната графика.

Пример 8. Нека приближим функцията $x \sin x \cos x$ в интервала $[0, 10]$, като построим полинома на Тейлър от степен 2, 3, 4 около $x_0 = 5$.

За целта бихме могли да използваме следния код:

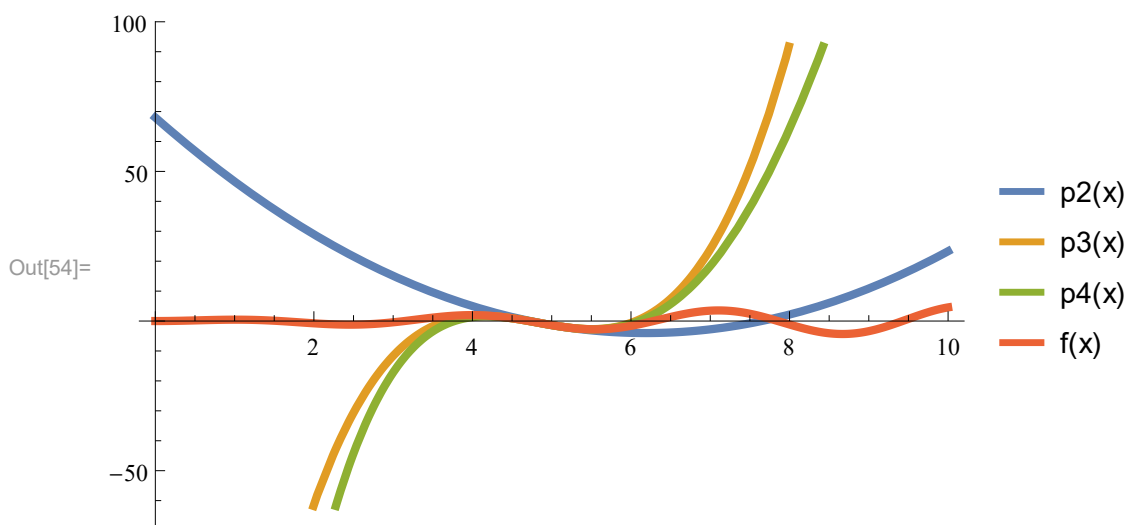
```

In[47]:= f[x_] := Sin[x] * Cos[x] * x
Df1[x_] = D[f[x], {x, 1}];
Df2[x_] = D[f[x], {x, 2}];
Df3[x_] = D[f[x], {x, 3}];

p2[x_, x0_] := f[x0] + f'[x0] (x - x0) + f''[x0] (x - x0)^2 / 2!
p3[x_, x0_] := p2[x, x0] + f'''[x0] (x - x0)^3 / 3!
p4[x_, x0_] := p3[x, x0] + f''''[x0] (x - x0)^4 / 4!

Plot[{p2[x, 5], p3[x, 5], p4[x, 5], f[x]}, {x, 0, 10},
PlotLegends -> {"p2(x)", "p3(x)", "p4(x)", "f(x)"},
PlotStyle -> Table[Thickness[0.01], {i, 4}]]

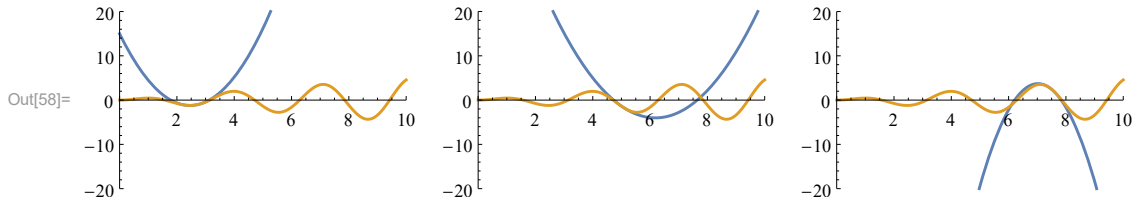
```



Това, което е важно да отбележим, че при увеличаване на степента на полинома, той се „залепва“ все по-добре за функцията около точката, в която развиваме.

Друг интересен въпрос е какво се случва, ако x_0 се променя. Отговор на този въпрос дават следващите графики. На тях е изобразена функцията (в оранжево) полинома на Тейлър от втора степен, построен с развитие около точките $x_0 = 2.5$, $x_0 = 5$ и $x_0 = 7.5$ (в синьо):

```
In[58]= GraphicsRow[{Plot[{p2[x, 2.5], f[x]}, {x, 0, 10}, PlotRange -> {{0, 10}, {-20, 20}}],
    Plot[{p2[x, 5], f[x]}, {x, 0, 10}, PlotRange -> {{0, 10}, {-20, 20}}],
    Plot[{p2[x, 7.5], f[x]}, {x, 0, 10}, PlotRange -> {{0, 10}, {-20, 20}}]}]
```



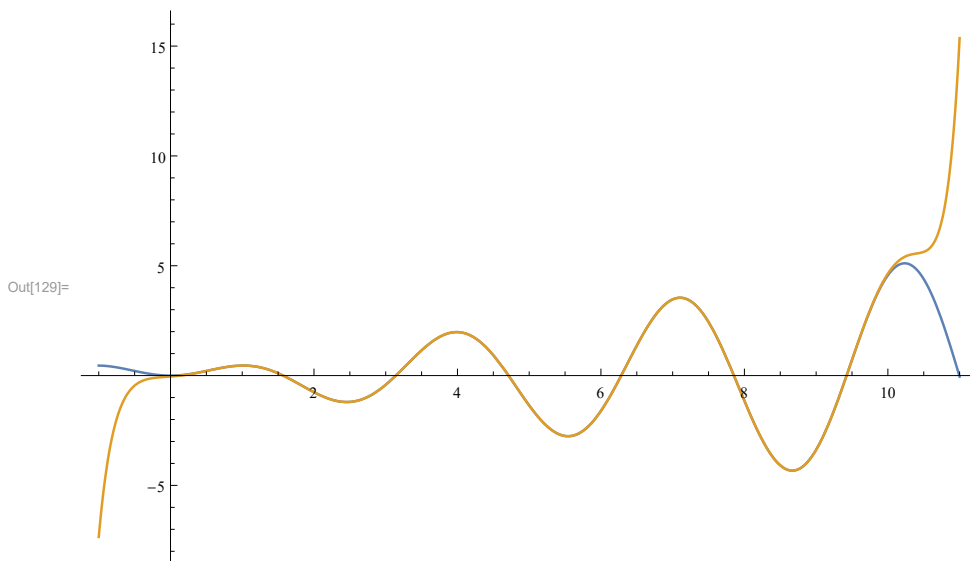
От графиките може да се заключи, че апроксимацията е най-добра за точки, които са близо до точката x_0 (като в нея грешката е 0). Следователно, ако искаме да получим добри резултати в даден интервал, е добра идея да развиваме около средата на този интервал.

Стига да вземем полином от достатъчно висока степен, теоретично можем да получим толкова добро приближение, колкото ни е необходимо. Така например при развитие около точката $x = 5$, нека искаме да приближим $x \sin x \cos x$ в интервала $[0, 10]$ с грешка, по-малка от 0.1?

```
In[126]= f[x_] := Sin[x] * Cos[x] * x
    taylorSeries[x_, order_] := f[5] + Sum[ReplaceAll[D[f[y], {y, k}], y -> 5.] * (x - 5.)^k / k!, {k, 1, order}];
```

```
For[order = 1, order <= 100, order++,
    maxError = Maximize[{Abs[f[x] - taylorSeries[x, order]], 0 <= x <= 10}, x][[1]];
    If[maxError <= 0.1,
        Print[{order, maxError}];
        Break[]];
];
{28, 0.0780946}
```

```
In[129]= Plot[{f[x], taylorSeries[x, 28]}, {x, -1, 11}, PlotRange -> All]
```



Виждаме, че с полином от степен 28 получаваме желаното приближение. Разбира се, точността на практика винаги е ограничена от машинната грешка.

Пример 9. Често формулата на Тейлър се използва за оценяване на грешка на апроксимация. Тук ще изведем оценки на грешката за формулите за числено диференциране. За целта, ще намерим разликата между точното и приближеното решение и ще развием в ред на Тейлър около точката t . По този начин получаваме следните оценки за формули (1.1), (1.2) и (1.3):

$$\begin{aligned} err_1 &= f'(t) - \frac{f(t + \Delta t) - f(t)}{\Delta t} = f'(t) - \frac{f(t) + \Delta t f'(t) + O(\Delta t^2) - f(t)}{\Delta t} = O(\Delta t), \\ err_2 &= f'(t) - \frac{f(t) - f(t - \Delta t)}{\Delta t} = f'(t) - \frac{f(t) - (f(t) - \Delta t f'(t) + O(\Delta t^2))}{\Delta t} = O(\Delta t), \\ err_3 &= f'(t) - \frac{f(t + \Delta t) - f(t - \Delta t)}{\Delta t} = O(\Delta t^2). \end{aligned}$$

1.7 Цели на настоящия курс

В настоящия курс ние ще се занимаем с въпроса за численото решаване на ДУ. Да припомним някои от основните въпроси, които са основни за числените методи изобщо:

- Кой е подходящият метод за дадена конкретна задача?
- Устойчивост на метода – този семестър това понятие ще заема абсолютно централна роля!
- Сходимост на метода (към точното решение).
- Оценка на грешката/точност на метода.

В настоящия курс ние ще разгледаме класически методи за численото решаване на ОДУ и ЧДУ и на базата на известната теория, различни примери и числени експерименти ще коментираме тяхната приложимост за различни класове задачи.

Глава 2

Числени методи за ОДУ

2.1 Задача на Коши за ОДУ от първи ред. Обща теория.

Започваме изучаването на числените методи за решаване на диференциални уравнения с тези за решаване на ОДУ от първи ред, т.е. ще разглеждаме ОДУ от вида

$$\frac{du}{dt} = f(t, u(t)).$$

Уравнения от по-висок ред можем да сведем до такива от първи ред чрез полагане. Например законът на Нютон

$$\frac{d^2u}{dt^2} = \frac{F(t)}{m}$$

може да се сведе до система ОДУ от първи ред чрез полагането $du/dt = v$. Получаваме системата

$$\begin{aligned} \frac{du}{dt} &= v, \\ \frac{dv}{dt} &= \frac{F(t)}{m}. \end{aligned}$$

От друга страна, системи уравнения от първи ред можем да разглеждаме като едно векторно уравнение. Горния пример можем да запишем във вида

$$\frac{d}{dt} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} v \\ F(t)/m \end{bmatrix}$$

или още

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(t, \mathbf{u}(t)),$$

където $\mathbf{u} = (u, v)^T \in \mathbb{R}^2$ е вектор и $\mathbf{f}(t, \mathbf{u}) = (v, F(t)/m)^T$ е векторна функция $\mathbf{f}: \mathbb{R} \times \mathbb{R}^2 \rightarrow \mathbb{R}^2$.

Казаното дотук ни дава основание да разглеждаме именно уравнения от първи ред. Общият им вид е

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(t, \mathbf{u}(t)),$$

където $\mathbf{u} \in \mathbb{R}^n$ и $\mathbf{f}: \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$.

За да бъде решението определено еднозначно, е необходимо (макар и не достатъчно) да знаем поне една точка от него.

И така, **общата задача, която ще разглеждаме, е т.нар. задача на Коши:**

$$\begin{aligned} \frac{d\mathbf{u}}{dt} &= \mathbf{f}(t, \mathbf{u}(t)), \quad t \in (t_0, +\infty) \\ \mathbf{u}(t_0) &= \mathbf{u}_0. \end{aligned} \tag{2.1}$$

Решение на задачата на Коши наричаме непрекъснато диференцируема функция $\mathbf{u}: \mathbb{R} \rightarrow \mathbb{R}^n$, която удовлетворява уравнението и началното условие.

Първият въпрос, на който трябва да отговорим, преди да започнем да решаваме дадена задача, е, разбира се, **има ли тя решение и, ако да, единствено ли е то**. Може да се отговори положително на този въпрос за задачата на Коши при произволни начални данни, ако дясната страна е “достатъчно хубава” функция. **В настоящия курс ние винаги ще приемаме, че това е така!** Това не е съществено ограничение на общността, тъй като изискванията, които се налагат към дясната страна, за да има задачата на Коши единствено решение, обикновено са изпълнени за задачите, възникващи на практика.

Нека D е отворено множество в \mathbb{R}^{n+1} и $f(t, u)$ е дефинирана в D .

Теорема (Peano). *Нека функцията $f(t, u)$ е непрекъсната в областта D . Тогава за всяка точка $(t_0, u_0) \in D$ задачата на Коши (2.1) има решение $u(t)$, което е дефинирано поне за някакъв интервал $t \in (t_0 - \alpha, t_0 + \alpha)$.*

Доказателство. Двете най-популярни доказателства на тази теорема могат да се намерят в [4, 5]. Първото използва основни теореми от областта на функционалния анализ, а второто е директно. \square

Дефиниция. *Казваме, че функцията $f(x)$ е липшицова в областта Ω , ако съществува положително число K такова, че*

$$\|f(x) - f(y)\| \leq K\|x - y\|, \quad \forall x, y \in \Omega.$$

Теорема (Picard-Lindelöf). *Ако $f(t, u)$ е непрекъсната в D и Липшицова по отношение на u в D , т.е.*

$$\|f(t, u) - f(t, v)\| \leq K\|u - v\|, \quad \forall (t, u), (t, v) \in D \text{ и някое } K > 0,$$

тогава за всяко $(t_0, u_0) \in D$ съществува единствено решение $u(t)$ такова, че $u(t_0) = u_0$

Доказателство. Вж. [4, 5]. \square

Забележка. Да обърнем внимание, че теоремите за съществуване и единственост на решението са локални, т.е. решението $u(t)$ може да не е дефинирано за всяко t , а само в някакъв интервал $[t_0, T)$.

Повече подробности относно общата теория на ОДУ (например продължимост на решенията, условия за гладкост на решенията, устойчивост по начални данни, качествена и асимптотичен анализ на решенията и т.н.) могат да се намерят в [4, 5].

2.2 Идея на диференчните методи за задачата на Коши за ОДУ от първи ред

Както казахме, най-общо, задачата на Коши за обикновено диференциално уравнение или система от ОДУ от първи ред има вида

$$\begin{aligned}\frac{d\mathbf{u}}{dt} &= \mathbf{f}(t, \mathbf{u}(t)), \quad t \in (t_0, +\infty), \\ \mathbf{u}(t_0) &= \mathbf{u}_0,\end{aligned}\tag{2.2}$$

където в общия случай неизвестната функция $\mathbf{u}(t)$ е векторна функция $\mathbf{u} : \mathbb{R} \rightarrow \mathbb{R}^n$. Ще приемаме, че функцията $\mathbf{f} : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ е достатъчно гладка, така че съществува единствено решение на задачата на Коши (2.1) и могат да бъдат направени използваните по-нататък Тейлъртови развиятия на \mathbf{f} .

Компютърът, разбира се, не може да прави безкрайни пресмятания. С други думи, трябва да знаем докога да интегрираме.

- В някои задачи ще искаме да намерим решението в предварително известен интервал $t \in [t_0, T]$.
- В други задачи може да искаме да интегрираме, докато е изпълнено предварително зададено условие, например докато стойността на u стане 0.
- В някои проблеми искаме да знаем какво е поведението на решенията при $t \rightarrow \infty$. В този случай трябва да интегрираме дотогава, когато стане ясно какво е поведението за достатъчно големи времена. Засега ще оставим този доста общ отговор за този случай и ще се върнем на него по-късно.

Да обърнем внимание, че променливите t и u са непрекъснати. Нашата цел ще бъде да ги заменим с дискретни, така че задачата да бъде сведена до алгебрична и следователно да може да бъде решена с помощта на стандартни числени методи. Основният въпрос е как тази „замяна“ да се направи така, че да получим добро приближение на решението на оригиналната задача на Коши.

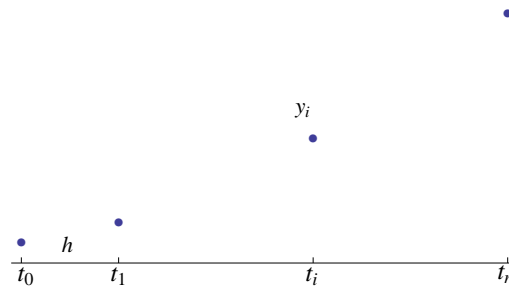
N.B. 8

Въвеждаме мрежата

$$\bar{\omega}_h = \{t_i = t_0 + ih, \quad i = \overline{0, n}, \quad n = (T - t_0)/h\}.$$

Ще търсим приближените стойности на решението именно в точките от тази мрежа. Стойността на приближеното решение в точката t_i ще бележим с y_i , т.е. искаме

$$y_i \approx u(t_i) = u_i.$$



За построяване на числена схема за намиране на стойностите на приближеното решение можем да подходим, като апроксимираме оригиналната диференциална задача (2.1).

Н.В. 9

Така, започвайки с $y_0 = u_0$, нашата цел ще бъде последователно да намерим y_1, y_2, \dots или, изобщо казано, ако сме намерили y_i да намерим y_{i+1} за $i = \overline{0, n-1}$.

2.3 Методи на Ойлер

2.3.1 Явен и неявен метод на Ойлер – въведение

Алгоритъм 1: Явен метод на Ойлер

Явният метод на Ойлер е най-простият метод за числено решаване на ОДУ. Използвайки формулата за числено диференциране с разлика напред в точката t_i

$$u'(t_i) \approx \frac{u(t_{i+1}) - u(t_i)}{h},$$

диференциалната задача (2.1) се апроксимира върху мрежата $\bar{\omega}_h$ с алгебричната задача

$$\frac{y_{i+1} - y_i}{h} = f(t_i, y_i), \quad i = \overline{0, n-1},$$

$$y_0 = u_0.$$

Тогава за пресмятанията на приближеното решение можем да използваме схемата

$$y_0 = u_0, \quad y_{i+1} = y_i + hf(t_i, y_i), \quad i = \overline{0, n-1}. \quad (2.3)$$

Пример 10. Нека използваме явния метод на Ойлер за задачата на Коши

$$\begin{aligned} \frac{du}{dt} &= -u, \quad 0 < t \leq 1, \\ u(0) &= 100 \end{aligned}$$

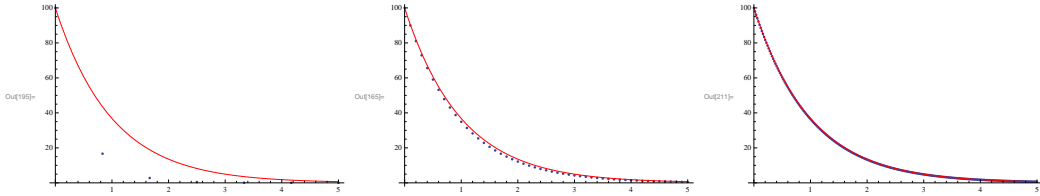
при $n = 4, 50, 250$ и да сравним резултатите с точното решение

$$u(t) = 100e^{-t}.$$

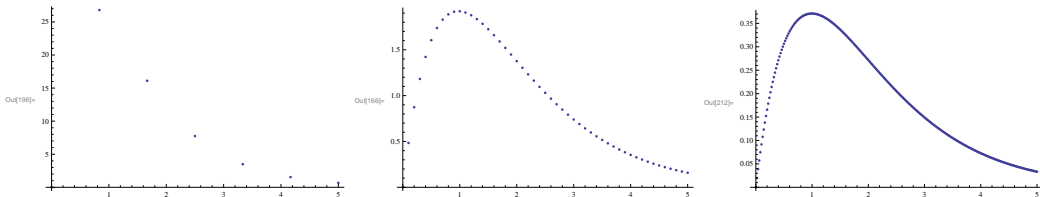
За конкретната задача, (2.3) добива вида

$$y_0 = 100, y_1 = (1 - h)y_0, y_2 = (1 - h)y_1 = 100(1 - h)^2, \dots$$

Решенията за $n = 4, 50, 250$ са показани по-долу, като червената крива изобразява точното решение.



Грешките по абсолютна стойност в точките от мрежата $\bar{\omega}_h$ са изобразени по-долу:



N.B. 10

Виждаме, че с увеличаване на броя на възлите (и следователно намаляване на h) приближеното решение започва да се приближава към точното. Това илюстрира първото важно свойство, което искаме да притежава даден числен метод – да бъде **сходящ към точното решение**, т.е. при $n \rightarrow \infty$ грешката да клони към 0. Ще се върнем отново на този въпрос и тогава ще го формулираме по-точно.

Алгоритъм 2: Неявен метод на Ойлер

Неявният метод на Ойлер се основава на формулата за числено диференциране с разлика назад. Апроксимацията на (2.1) има вида

$$\frac{y_i - y_{i-1}}{h} = f(t_i, y_i), \quad i = \overline{1, n},$$

$$y_0 = u_0.$$

Тогава за пресмятанията на приближеното решение можем да използваме схемата

$$y_0 = u_0, \quad y_{i+1} = y_i + hf(t_{i+1}, y_{i+1}), \quad i = \overline{0, n-1}. \quad (2.4)$$

Дефиниция 4: Неявен метод

На всяка стъпка от алгоритъма, за да получим стойността на приближеното решение в t_{i+1} , **трябва да решим едно (в общия случай нелинейно) уравнение** относно y_{i+1} , тъй като y_{i+1} участва и в дясната страна. По тази причина наричаме метода **неявен**.

Пример 11. Да решим сега задачата на Коши от примера 10, като използваме неявния метод на Ойлер.

Започвайки с началното условие $y_0 = 100$, на всяка следваща стъпка трябва да решим уравнението

$$y_{i+1} = y_i - hy_{i+1}, \quad i = \overline{0, n-1},$$

в което неизвестното е y_{i+1} . В конкретния случай полученото уравнение е линейно и може да бъде решено лесно:

$$y_1 = \frac{100}{1+h}, y_2 = \frac{y_1}{1+h} = \frac{100}{(1+h)^2}, \dots$$

В общия случай обаче ще се наложи да бъдат решавани много на брой нелинейни уравнения.

Пример 12. За задачата на Коши

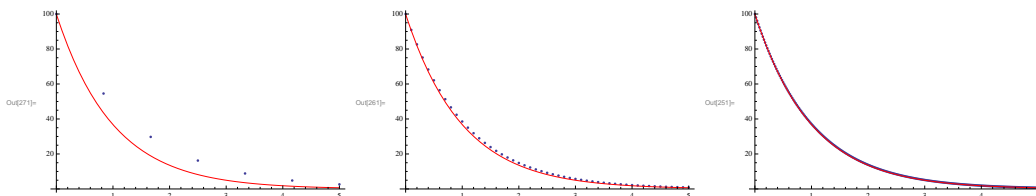
$$\begin{aligned} \frac{du}{dt} &= \sin u + \sqrt{u} - t, \quad t \in (0, T], \\ u(0) &= 100. \end{aligned}$$

на всяка итерация по неявния метод на Ойлер ще трябва да решим нелинейното уравнение

$$y_{i+1} = y_i + h(\sin y_{i+1} + \sqrt{y_{i+1}} - t_{i+1}).$$

За тази цел ще бъде необходимо използването на метод за численото решаване на нелинейни алгебрични уравнения. Най-често използваният метод е методът на Нютон.

Решенията за $n = 6, 50, 250$ са показани по-долу, като червената крива изобразява точното решение.



2.3.2 Локална грешка на апроксимация

Единият основен въпрос, свързан с приложимостта на даден метод, е неговата точност.

При явния метод на Ойлер апроксимираме

$$\frac{u_{i+1} - u_i}{h} = f(t_i, u_i) + O(h) \quad (2.5)$$

с

$$\frac{y_{i+1} - y_i}{h} = f(t_i, y_i), \quad (2.6)$$

а при неявния метод на Ойлер

$$\frac{u_{i+1} - u_i}{h} = f(t_{i+1}, u_{i+1}) + O(h) \quad (2.7)$$

c

$$\frac{y_{i+1} - y_i}{h} = f(t_{i+1}, y_{i+1}), \quad (2.8)$$

т.е. апроксимираме диференциално уравнение с диференчно, като пренебрегваме остатъчния член $O(h)$. Казваме, че тези методи имат локална грешка на апроксимация $O(h)$.

Дефиниция 5: Локална грешка на апроксимация (ЛГА)

ЛГА за един диференчен метод

$$\frac{y_{i+1} - y_i}{h} = \theta(\bar{t}, \bar{y})$$

наричаме разликата между лявата и дясната страна, пресметната за точното решение, т.е.

$$\psi_{i,h} := \frac{u_{i+1} - u_i}{h} - \theta(\bar{t}, \bar{u}),$$

където $\bar{t} = (t_0, t_1, \dots, t_n)$, $\bar{y} = (y_0, y_1, \dots, y_n)$ и $\bar{u} = (u_0, u_1, \dots, u_n)$.

Да обсъдим малко по-внимателно дефиницията. Ясно е защо е грешка на апроксимация – това е грешка при апроксимацията на ДУ с диференчно уравнение, но защо е локална? Това е грешката, която получаваме **локално, за една стъпка от алгоритъма**. Действително, ако до t_i сме смятали абсолютно точно, т.е. $y_i = u_i$, то ЛГА е точно остатъчния член, който пренебрегваме за да получим диференчното уравнение вместо диференциалното.

Разбира се, ние се интересуваме от **глобалната грешка** $u_i - y_i$, където y_i е стойността на приближеното решение, след като са минали всичките предишни $i - 1$ стъпки на алгоритъма.

Оказва се, че ако едностъпков метод има ЛГА $O(h^p)$, то и глобалната грешка е $O(h^p)$ при $h \rightarrow 0$. Ще уточним този въпрос по-късно, като ще имаме специална лекция по темата. Засега е достатъчно да знаем, че ЛГА ни дава информация за сходимостта на метода.

Сега да изведем ЛГА за явния и неявния метод на Ойлер директно (без да използваме оценката на грешката за формулите за числено диференциране, която всъщност не сме доказали). От една страна самият подход ще бъде много важен за нас в курса нататък. От друга страна, така ще изведем и оценките на грешката за формулите за числено диференциране.

ЛГА за явния метод на Ойлер има вида

$$\begin{aligned} \psi_{h,i} &:= \frac{u_{i+1} - u_i}{h} - f(t_i, u_i) \\ &= \frac{1}{h} \left(u_i + \frac{u'_i}{1!} h + \frac{u''_i}{2!} h^2 + O(h^3) - u_i \right) - u'_i \\ &= \frac{u''_i}{2} h + O(h^2) = O(h). \end{aligned} \quad (2.9)$$

Следователно ЛГА на явния метод на Ойлер (при предположение за непрекъснатата втора производна на търсеното решение) е $O(h)$. ЛГА за неявния метод

на Ойлер има вида

$$\begin{aligned}
\psi_{h,i} &:= \frac{u_{i+1} - u_i}{h} - f(t_{i+1}, u_{i+1}) \\
&= \frac{1}{h} \left(u_{i+1} - \left(u_{i+1} - \frac{u'_{i+1}}{1!}h + \frac{u''_{i+1}}{2!}h^2 + O(h^3) \right) \right) - u'_{i+1} \\
&= \frac{u''_{i+1}}{2}h + O(h^2) = O(h).
\end{aligned} \tag{2.10}$$

Последният извод е направен при предположение за непрекъснатата втора производна на търсеното решение.

С други думи, и в двата случая

$$\|\bar{\psi}_h\|_C = O(h),$$

където $\bar{\psi}_h = (\psi_1, \psi_2, \dots, \psi_n)$, а $\|\bar{\psi}_h\|_C := \max_{1 \leq i \leq n} |\psi_i|$. Това означава, че на всяка стъпка, ако прилагаме методите на явния или неявния Ойлер върху точното решение на задачата ще внесем най-много грешка от порядъка на стъпката h .

2.3.3 Схема с тегло. Подобен метод на Ойлер.

Да припомним, че явният метод на Ойлер апроксимира задачата на Коши (2.1) с разлика напред

$$\frac{y_{i+1} - y_i}{h} = f(t_i, y_i), \tag{2.11}$$

а неявният – с разлика на назад

$$\frac{y_{i+1} - y_i}{h} = f(t_{i+1}, y_{i+1}). \tag{2.12}$$

Ако вземем с тегло σ апроксимацията по явния метод на Ойлер, а с тегло $1 - \sigma$ – тази по неявния метод на Ойлер, то ще получим следната диференчна схема:

$$\frac{y_{i+1} - y_i}{h} = \sigma f(t_i, y_i) + (1 - \sigma)f(t_{i+1}, y_{i+1}). \tag{2.13}$$

Тази схема се нарича диференчна схема с тегло за задачата на Коши. Въпросът е дали можем да изберем σ така, че да получим по-добра ЛГА.

$$\begin{aligned}
\psi_{h,i} &= \frac{u_{i+1} - u_i}{h} - \sigma f(t_i, u_i) - (1 - \sigma)f(t_{i+1}, u_{i+1}) \\
&= \frac{1}{h}(u_{i+1} - u_i) - \sigma u'_i - (1 - \sigma)u'_{i+1} \\
&= \frac{1}{h} \left(u_i + \frac{u'_i}{1!}h + \frac{u''_i}{2!}h^2 + \frac{u'''_i}{3!}h^3 + O(h^4) - u_i \right) - \sigma u'_i \\
&\quad - (1 - \sigma) \left(u'_i + \frac{u''_i}{1!}h + \frac{u'''_i}{2!}h^2 + O(h^3) \right) \\
&= \left(\frac{h}{2} - (1 - \sigma)h \right) u''_i + \left(\frac{h^2}{6} - (1 - \sigma)\frac{h^2}{2} \right) u'''_i + O(h^3).
\end{aligned}$$

Следователно грешката на всички методи с тегло е $O(h)$. За да бъде грешката от по-висок ред е достатъчно

$$\frac{h}{2} - (1 - \sigma)h = 0 \Rightarrow \sigma = \frac{1}{2}.$$

Да обърнем внимание, че при това положение членовете, в които участва третата производна не могат да се унищожат.

Алгоритъм 3: Подобрен метод на Ойлер

Методът, който се получава от схемата с тегло при $\sigma = \frac{1}{2}$, е известен като **подобрен метод на Ойлер**:

$$\frac{y_{i+1} - y_i}{h} = \frac{f(t_i, y_i) + f(t_{i+1}, y_{i+1})}{2}.$$

Той има втори ред на апроксимация.

Забележка. Можем (макар и с малко повече съобразителност) да използваме формулата за централна разлика при извеждането на подобрения метод на Ойлер.

N.B. 11

Явният и неявният методи на Ойлер имат ЛГА $O(h)$ (и първи ред на сходимост). Подобреният метод на Ойлер има ЛГА $O(h^2)$ и втори ред на сходимост.

2.3.4 Други подходи за получаване на методите на Ойлер

Апроксимиране на еквивалентно интегрално уравнение

Нека да разгледаме задачата на Коши (2.1). Ако интегрираме диференциалното уравнение в граници $[0, t]$, ще получим

$$u(t) = u_0 + \int_0^t f(s, u(s)) ds$$

или аналогично можем да запишем

$$u(t_{i+1}) = u(t_i) + \int_{t_i}^{t_{i+1}} f(s, u(s)) ds, \quad (2.14)$$

което директно ни дава връзка между стойността в i -тата $i+1$ -вата точка, което е нашата цел.

По този начин съждаме решаването задачата на Коши (2.1) до решаване на интегрално уравнение. Това обаче не опростява съществено проблема, тъй като е известно, че повечето интегрални не могат да се сметнат аналитично. Затова в общия случай и е необходимо да апроксимираме интеграла в (2.14) по подходящ начин. Ако използваме елементарната квадратурна формула на левите правоъгълници, на десните правоъгълници или на трапеците ще получим съответно явния, неявния или подобрения метод на Ойлер.

За тези извеждания прочети [3], стр.12.

Геометричен подход

Геометрично, явният метод на Ойлер се основава на класическата идея в анализа – линеаризацията. Това означава, че геометрично приближаваме графиката на функцията u с нейната допирателна, която дава приближение в достатъчно малка околност на точката, в която линеаризираме. **Тук прочети стр.12, т.1 в параграф 1.2 в [3].**

За добра визуална илюстрация на геометричния смисъл на явния метод на Ойлер, вж.

<https://www.khanacademy.org/math/ap-calculus-bc/bc-differential-equations-new/bc-7-5/v/eulers-method>.

2.3.5 А-устойчивост и монотонност

Казахме, че ЛГА ни позволява да определим реда на сходимост за даден едностъпков метод за задачата на Коши за ОДУ от първи ред. От друга страна, **това, че за достатъчно малко h грешката клони към 0, далеч не означава, че за по-малки стойности на h поведението на метода ще е добро.** Да разгледаме следния пример.

Пример 13. Решаваме задачата на Коши

$$\begin{aligned}\frac{du}{dt} &= 10u(1 - u), \quad t \in (0, 6], \\ u(0) &= 0.01\end{aligned}\tag{2.15}$$

с явния и неявния метод на Ойлер за $n = 10, 20, 35, 75, 500$.

Привеждаме резултатите от решението на задачата по-долу.

n	Неявен Ойлер	Явен Ойлер
10		
20		
35		
75		
500		

И така, очевидно явният метод на Ойлер няма добро поведение за малки стойности на n (т.е. голяма стъпка h). Такъв проблем не се наблюдава при неявния метод на Ойлер. Това е свързано с понятията абсолютна устойчивост (А-устойчивост) и монотонност на методите.

За да въведем тези понятия, първо трябва да изясним някои свойства на решенията на едно ОДУ.

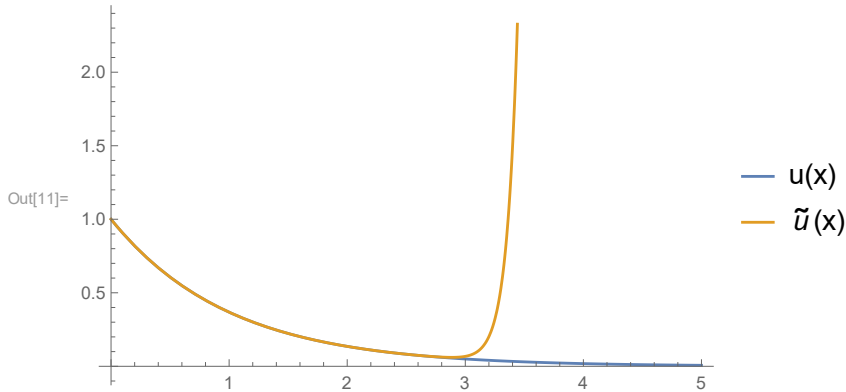
Устойчивост на задачата срещу устойчивост на метода

Ще започнем с един пример.

Пример 14. Да разгледаме задачата

$$\begin{aligned}u'' - 10u' - 11u &= 0, \\ u(0) &= 1, u'(0) = -1.\end{aligned}$$

Решението ѝ е $u(x) = e^{-x}$. Нека сега малко променим първото начално условие на $u(0) = 1 + \varepsilon$. Решението на така получената задача е $\tilde{u}(x) = \left(1 + \frac{11}{12}\varepsilon\right)e^{-x} + \frac{\varepsilon}{12}e^{11x}$. На следващата фигура са построени графиките на функциите $u(x)$ и $\tilde{u}(x)$ при $\varepsilon = 10^{-15}$.



Виждаме, че двете графики започват бързо да се различават съществено. Казваме, че **решението на задачата е неустойчиво**. Малка промяна на входните данни води до голяма промяна в **точното решение** на задачата.

N.B. 12

Намирането на неустойчиви решения е **лошо обусловена задача**. Грешките от закръгляване и апроксимация при използването на числени методи неминуемо ще доведат до ефект, какъвто наблюдавахме в предходния пример.

Много от процесите в природата обаче водят до модели, които се характеризират с устойчиви решения, тъй като нещата в природата обикновено “се стремят” към достигането на някакво устойчиво състояние. Да отбележим, че все пак има процеси, които не са такива. Например това е причината, поради която метеорологичните прогнози не могат да се правят с висока точност за дълъг период от време. Тъй като за пресмятането на метеорологичната прогноза се решава система от частни диференциални уравнения, която се характеризира с неустойчиви решения, численото им намиране дава резултат, който е (относително) надежден само в краткосрочен план.

Предвид казаното дотук, ние ще се интересуваме в настоящия параграф от намирането на устойчиви решения.

По принцип могат да се пресмятат с известен успех и неустойчиви решения (и това се прави, както в примера за метеорологичните прогнози, който дадохме!), но не можем да се надяваме численият метод да е устойчив в пълния смисъл на думата по причини, които вече уточнихме (тогава се дефинира специален вид устойчивост на методите, но ние няма да се спираме на този въпрос). Все пак, ако разгледаме предходния пример, виждаме, че поне в някакъв интервал (например $x \in [0, 2.5]$) бихме могли да се надява-

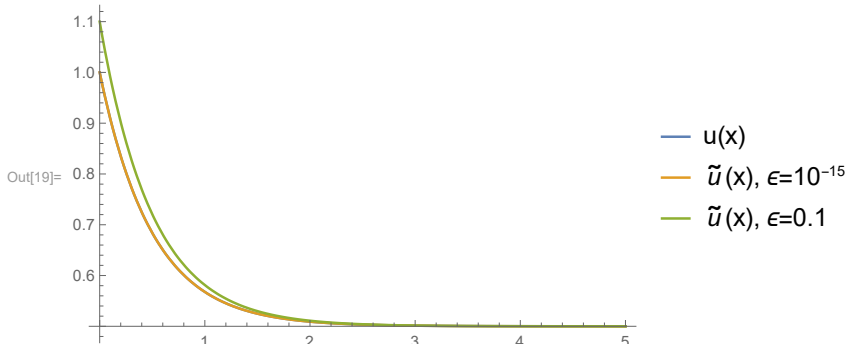
ме да получим числено с добра точност и неустойчиво решение, но това е невъзможно да стане в произволно голям интервал.

Да разгледаме още един пример, преди да дадем точната дефиниция за устойчиво решение.

Пример 15. Да разгледаме задачата на Коши

$$\begin{aligned} u' &= -2u + 1, \\ u(0) &= 1. \end{aligned}$$

Нейното решение е $u(x) = \frac{1}{2}e^{-2x} + \frac{1}{2}$, а това на задачата, която се получава, като разгледаме начално условие $u(0) = 1 + \varepsilon$, е $\tilde{u}(x) = (\frac{1}{2} + \varepsilon)e^{-2x} + \frac{1}{2}$. Тук с увеличаване на t разликата между двете решения с времето не се увеличава, а напротив – намалява и клони към 0. На следващата графика са визуализирани решенията при $\varepsilon = 10^{-15}$ и $\varepsilon = 0.1$:



Дефиниция 6: Устойчивост по Ляпунов

Казваме, че решението на задачата на Коши е **устойчиво (по Ляпунов/по начални данни)**, ако малки промени в началните условия не водят до големи промени в решението. **Вж. точната дефиниция за устойчивост по Ляпунов на стр. 10 от [3].**

Следователно ние първо трябва да видим (в някакъв смисъл) как изглеждат задачите, които имат устойчиви решения, и тогава да видим какво да поискаме от числените методи за тяхното решаване.

Ще изложим идеите първо върху автономни уравнения (т.е. при които независимата променлива не участва експлицитно в дясната страна), тъй като там те са по-лесни да бъдат възприети.

Равновесните точки на едно автономно диференциално уравнение

$$\frac{du}{dt} = f(u)$$

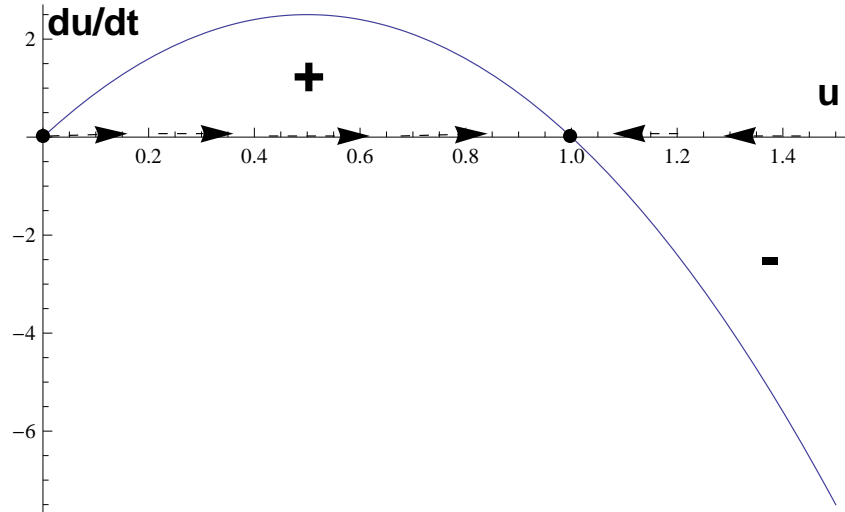
са точките $u = \xi$, за които $f(\xi) = 0$.

Равновесните точки определят еднозначно асимптотичното поведение на решенията (т.е. какво се случва с решенията в крайна сметка, след достатъчно дълъг период от време).

Пример 16. Разглеждаме логистичното уравнение

$$\frac{du}{dt} = 10u(1 - u).$$

За него очевидно равновесните точки са $u = 0$ и $u = 1$. Нека разгледаме графиката на du/dt , т.е. параболата $10u(1 - u)$:



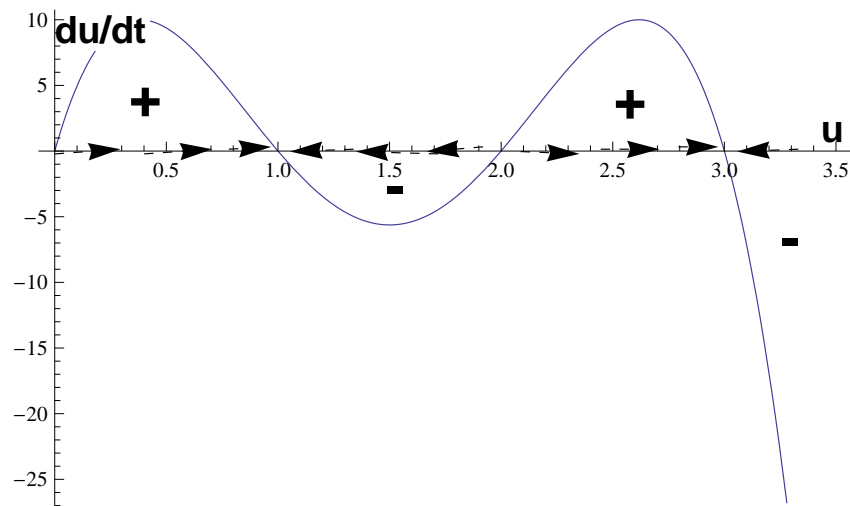
Тъй като производната на u е положителна за $u \in (0, 1)$, то ако u е в този интервал функцията е растяща. В противен случай тя е намаляваща.

От горната фигура лесно се вижда, че всички решения клонят към равновесната точка $u = 1$.

Казваме, че една равновесна точка ξ е **асимптотично устойчива**, ако за всички достатъчно близки до нея начални условия u_0 следва, че решението u клони към ξ . В противен случай равновесната точка се нарича неустойчива.

За логистичното уравнение точката $u = 0$ е неустойчива, защото всяко изменение на $u > 0$ води до решение, клонящо към $u = 1$. От друга страна точката $u = 1$ е асимптотично устойчива, тъй като при изменение на u решението “се връща” към единицата.

Лесно се съобразява, че за едно автономно уравнение решенията винаги клонят към асимптотично устойчива равновесна точка. Да разгледаме следната диаграма:



От фигурата е ясно, че решенията клонят или към $u = 1$, или към $u = 3$.

От гледна точка на числените методи, това поведение е от съществено значение. Да подчертаем още веднъж, че възстановяването на неустойчиви решения с помощта на числени методи е лошо обусловена задача, тъй като всяка грешка от закръгляване означава, че решението, което ще се получи, ще е различно (ще клони към устойчива равновесна точка).

От друга страна, грешките от закръгляване не би трябвало да имат голямо влияние върху решенията, клонящи към устойчивите равновесни точки, тъй като $|u - \xi| \rightarrow 0$ при $t \rightarrow \infty$. С други думи, грешките от закръгляване, допуснати в даден момент от време, които са неизбежни, би трябвало с времето да стават незначителни.

N.B. 13

Ето защо искаме числените методи да запазват асимптотичната устойчивост на решенията. Това свойство на числените методи се нарича абсолютна устойчивост или A-устойчивост. Ако численото решение също клони към равновесната точка, то ще има “правилното” поведение, а няма да отива към безкрайност.

Изследване на A-устойчивост и монотонност за даден числен метод

За да видим как ще изследваме устойчивостта на числения метод, първо трябва да коментираме как аналитично изследваме устойчивостта на една равновесна точка.

Нека разгледаме уравнението

$$\frac{du}{dt} = f(u),$$

за което ξ е равновесна точка, т.е. $f(\xi) = 0$.

Линеаризирайки дясната страна в точката $u = \xi$, получаваме

$$\frac{du}{dt} = f(\xi) + f'(\xi)(u - \xi) + O(|2|) \approx f'(\xi)(u - \xi).$$

Използвайки, че $f(\xi) = 0$ и че членовете от втори и по-висок ред могат да се пренебрегнат за достатъчно малки стойности на $u - \xi$, получаваме, че поведението на решението в достатъчно малка околност на точката ξ се определя от уравнението

$$\frac{du}{dt} = f'(\xi)(u - \xi).$$

Означавайки $\lambda := f'(\xi)$ и $\bar{u} := u - \xi$, записваме уравнението във вида

$$\frac{d\bar{u}}{dt} = \lambda\bar{u}, \tag{2.16}$$

където $\lambda < 0$, тъй като равновесната точка ξ е устойчива.

N.B. 14

Последното уравнение е прието за **моделно уравнение** за изследване на устойчивостта на даден метод. Това на практика е **уравнение за грешката** \bar{u} . В идеалния случай искаме тя да клони към 0 при $t \rightarrow \infty$.

Дефиниция 7: Абсолютна устойчивост (А-устойчивост)

Казваме, че даден числен метод е **абсолютно устойчив (А-устойчив)**, ако, приложен върху задачата (2.16), решението му удовлетворява

$$|y_i| \leq |y_0|.$$

С други думи, ако в даден момент от време е допусната грешка (например от закръгляване) y_0 , искаме численият метод да не я увеличава с времето и във всеки следващ момент y_i да остава не по-голяма по абсолютна стойност от y_0 . Ние ще поискаме малко по-силното условие $|y_i| \rightarrow 0$.

Тогава, ако можем да представим метода във вида $y_{i+1} = Ry_i$, лесно се вижда, че условието е изпълнено, ако $|R| < 1$.

Действително, имаме

$$y_{i+1} = Ry_i = R^2 y_{i-1} = \dots = R^{i+1} y_0,$$

което клони към 0 точно тогава, когато $|R| < 1$.

Забележка. В зависимост от решаваната задача можем да поискаме методът, който използваме, да запазва различни важни свойства на търсеното решение. А-устойчивостта е свързана със запазване на асимптотичната устойчивост.

Нека сега изследваме А-устойчивостта на методите на Ойлер. За явния метод на Ойлер имаме

$$y_{i+1} = y_i + hf(t_i, y_i) = y_i + h\lambda y_i = (1 + h\lambda)y_i.$$

Следователно, за да бъде явният метод на Ойлер А-устойчив, трябва да е изпълнено

$$|1 + h\lambda| < 1 \implies h < -2/\lambda$$

За неявния метод на Ойлер имаме

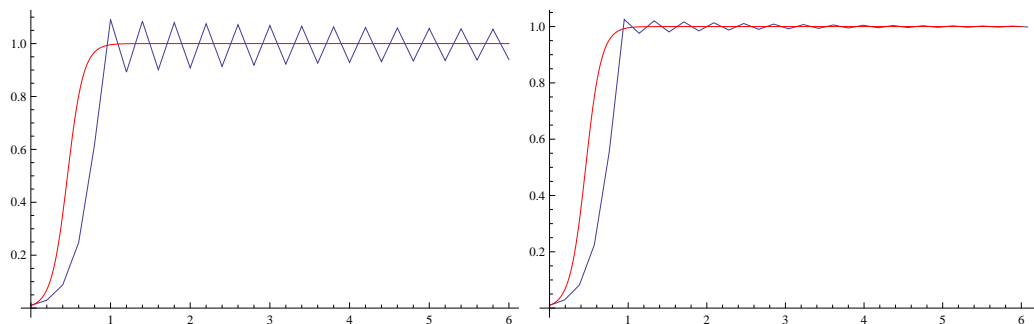
$$y_{i+1} = y_i + hf(t_{i+1}, y_{i+1}) = y_i + h\lambda y_{i+1} \implies y_{i+1} = \frac{1}{1 - \lambda h} y_i.$$

Вземайки предвид, че $\lambda < 0$ е ясно, че $|\frac{1}{1 - \lambda h}| < 1$ за всяко h , т.е. явният метод на Ойлер е А-устойчив за всяко h . **За подобрения метод на Ойлер прочети т.2 от параграф 1.6 на стр. 18 в [3].**

N.B. 15

Явният метод на Ойлер е условно А-устойчив при $h < -2/\lambda$. Неявният и подобреният методи на Ойлер са А-устойчиви (за всяко h).

Пример 17. Да илюстрираме казаното върху задачата (2.15). Единствената устойчива равновесна точка е $u = 1$ и $\lambda = f'(1) = -10$. Следователно методът е А-устойчив точно тогава, когато $h \leq 2/10$, като очакваме грешката да клони към 0, ако $h < 2/10$. Действително, нека разгледаме решенията при стъпка $h = 0.2$, $h = 0.19$:



В първия случай грешката осцилира около равновесната точка без да расте, но и без да намалява. Във втория случай грешката клони към 0.

Резултатът при $h = 0.19$ в последния пример обаче ни показва, че А-устойчивостта на метода не е достатъчна, за да има решението добро поведение. Въпреки че грешката клони към 0, се появява нежелано осцилиране около решението. Следователно е добре да поискаме методът да няма такова поведение.

Дефиниция 8: Монотонност на метода

Казваме, че даден числен метод е **монотонен**, ако, приложен върху задачата (2.16), решението му не сменя знака си, т.е.

$$\operatorname{sgn}(y_i) = \operatorname{sgn}(y_0).$$

Забележка. С други думи, искаме грешката да не осцилира около нулата.

Ако запишем метода във вида

$$y_{i+1} = Ry_i,$$

е ясно, че последното е изпълнено точно тогава, когато $R > 0$.

За явния метод на Ойлер имаме

$$y_{i+1} = (1 + h\lambda)y_i$$

и следователно методът е монотонен точно тогава, когато $h < -1/\lambda$.

Лесно се вижда, че неявният метод на Ойлер е монотонен за всяко h , вземайки предвид, че

$$y_{i+1} = \frac{1}{1 - \lambda h} y_i.$$

За подобрения метод на Ойлер прочети т.2 от параграф 1.6 на стр. 18 в [3].

N.B. 16

Явният метод на Ойлер е условно монотонен при $h < -1/\lambda$. Подобреният метод на Ойлер е условно монотонен при $h < -2/\lambda$. Неявният метод на Ойлер е монотонен (за всяко h).

Впрочем моделната задача (2.16) е естествено да се разглежда, за да се заключи за устойчивостта и монотонността и при неавтономни задачи **прочети стр 10 от (1.4) до (1.7) включително от [3]**

2.3.6 Числени експерименти и решени задачи (Практикум)

Пример 18. Нека използваме явния метод на Ойлер за логистичното уравнение

$$\begin{aligned} \frac{du}{dt} &= 10u(1-u), \quad 0 < t \leq 1, \\ u(0) &= 0.01 \end{aligned}$$

при $n = 4, 50, 250$ и да сравним резултатите с

$$u(t) = \frac{0.01}{0.01 + 0.99e^{-10t}}.$$

За конкретната задача, (2.3) добива вида

$$y_0 = 0.01, \quad y_{i+1} = y_i + 10hy_i(1 - y_i), \quad i = \overline{0, n-1}.$$

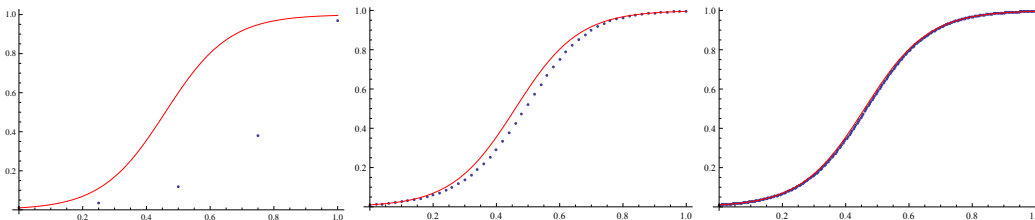
Имплементираме метода в Mathematica:

```
In[2]:= explicitEulerLogistic[n_] := (
  h = 1 / n;
  y = Table[0, {n+1}];
  y[[1]] = 0.01;
  For[i = 1, i < n+1, i++,
    y[[i+1]] = y[[i]] + 10 h y[[i]] (1 - y[[i]])
  ];
  y
)
```

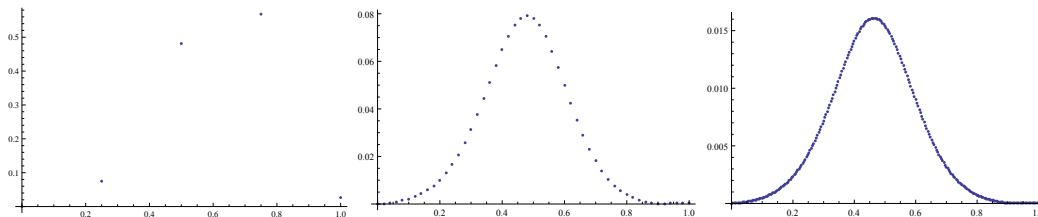
Като използваме реализираната функция, ще изобразим решенията за различни стойности на n .

```
In[3]:= Manipulate[
  y = explicitEulerLogistic[n];
  plotAppr = ListPlot[Table[{i h, y[[i+1]]}, {i, 0, n}]];
  plotExact = Plot[ $\frac{0.01}{0.01 + (1 - 0.01) E^{-10 t}}$ , {t, 0, 1}, PlotStyle -> Red];
  GraphicsRow[{Show[plotAppr, plotExact, PlotRange -> All],
    ListPlot[Table[{i h, Abs[y[[i+1]] -  $\frac{0.01}{0.01 + (1 - 0.01) E^{-10 i h}}$ ]}],
      {i, 0, n}], PlotRange -> All]}],
  {n, 4, 800, 10}]
```

Решенията за $n = 4, 50, 250$ са показани по-долу, като червената крива изобразява точното решение.



Грешките по абсолютна стойност в точките от мрежата $\bar{\omega}_h$ са изобразени по-долу:



Виждаме, че с увеличаване на броя на възлите приближеното решение започва да се приближава към точното. Това илюстрира първото важно свойство, което искаме да притежава даден числен метод – да бъде **сходящ към точното решение**, т.е. при $n \rightarrow \infty$ грешката да клони към 0.

Пример 19. Да решим сега задачата на Коши от пример 18, като използваме неявния метод на Ойлер.

Започвайки с началното условие $y_0 = 0.01$, на всяка следваща стъпка трябва да решим уравнението

$$y_{i+1} = y_i + 10hy_{i+1}(1 - y_{i+1}), \quad i = \overline{0, n-1},$$

в което неизвестното е y_{i+1} . Засега ще използваме вградената в Mathematica функция FindRoot, за да решим съответните уравнения.

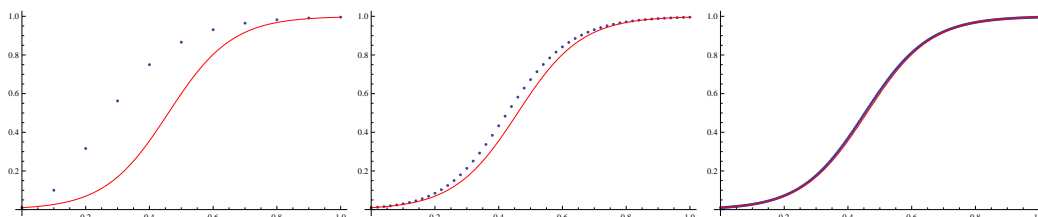
Повечето методи за решаване на нелинейни уравнения са итерационни и се нуждаят от добро начално приближение. Един възможен начин да получим такова е, като използваме явния метод на Ойлер:

$$y_{i+1,init} = y_i + 10hy_i(1 - y_i).$$

Имплементираме числената схема в Mathematica.

```
implicitEulerLogistic[n_] := (
  h = 1 / n;
  y = Table[0, {n + 1}];
  y[[1]] = 0.01;
  For[i = 1, i < n + 1, i++,
    initialGuess = y[[i]] + 10 h y[[i]] (1 - y[[i]]);
    y[[i + 1]] =
      yNew /. FindRoot[yNew == y[[i]] + 10 h yNew (1 - yNew), {yNew, initialGuess}]
  ];
  y
)
```

Решенията за $n = 10, 50, 250$ са показани по-долу, като червената крива изобразява точното решение.



Нека разгледаме още два примера за определяна на ЛГА.

Пример 20. Да пресметнем ЛГА за следните диференчни уравнения:

- $\frac{y_{i+1}-y_i}{h} = \frac{1}{3h}(y_i - y_{i-1}) + \frac{2}{3}f_i,$
- $y_{i+1} = y_i + \frac{h}{2}(3f_i - f_{i-1}).$

Решение. • В първия случай ЛГА е

$$\psi_h = \frac{1}{h}(u_{i+1} - u_i) - \frac{1}{3h}(u_i - u_{i-1}) + \frac{2}{3}f_i.$$

Развиваме около i -тата точка и получаваме последователно

$$\begin{aligned} \psi_h &= \frac{1}{h} \left(\underbrace{u_i + \frac{u'_i}{1!}h + \frac{u''_i}{2!}h^2 + O(h^3)}_{u_{i+1}} - u_i \right) \\ &\quad - \frac{1}{3h} \left(u_i - \underbrace{\left(u_i - \frac{u'_i}{1!}h + \frac{u''_i}{2!}h^2 + O(h^3) \right)}_{u_{i-1}} \right) + \frac{2}{3}f_i \\ &= \frac{2h}{3}u''_i + O(h^2) = O(h). \end{aligned}$$

- За да пресметнем ЛГА във втория пример, първо трябва да приведем уравнението във вида:

$$\frac{y_{i+1} - y_i}{h} = \frac{1}{2}(3f_i - f_{i-1}).$$

Тогав ЛГА е

$$\begin{aligned} \psi_h &= \frac{1}{h}(u_{i+1} - u_i) - \frac{1}{2}(3u'_i - u'_{i-1}) \\ &= \frac{1}{h} \left(\underbrace{u_i + \frac{u'_i}{1!}h + \frac{u''_i}{2!}h^2 + \frac{u'''_i}{3!}h^3 + O(h^4)}_{u_{i+1}} - u_i \right) \\ &\quad - \frac{1}{2} \left(3u'_i - \underbrace{\left(u'_i - \frac{u''_i}{1!}h + \frac{u'''_i}{2!}h^2 + O(h^3) \right)}_{u'_{i-1}} \right) \\ &= \frac{5h^2}{12}u'''_i + O(h^3) = O(h^2). \end{aligned}$$

□

2.4 Методи на Рунге–Кута

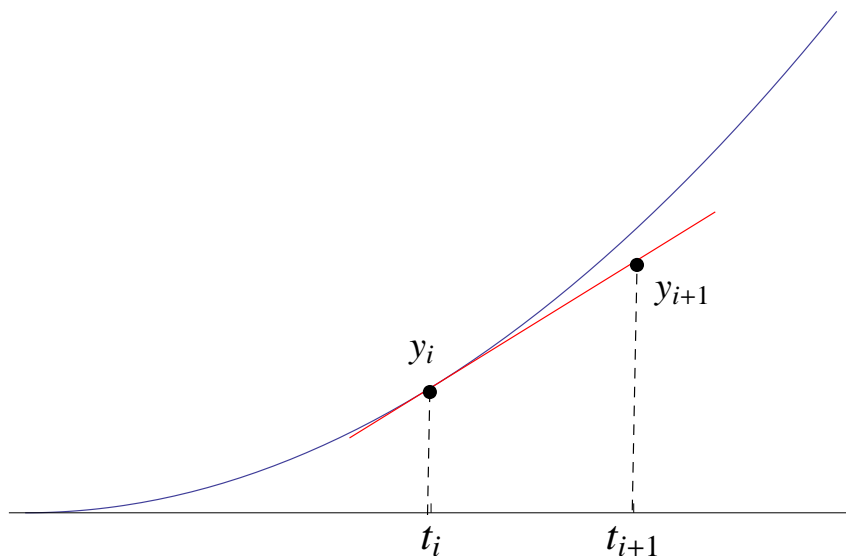
Основен недостатък на методите на Ойлер е, че те са бавно сходящи – да припомним, че явният и неявният методи на Ойлер имат първи ред на сходимост, тъй като локалната грешка на апроксимация е $O(h)$. Това означава, че ако искаме да получим висока точност, трябва да работим с много малка стъпка, т.е. да правим голям брой операции, което, разбира се, е нецелесъобразно. Сега ще разгледаме първия от двата основни класа методи, които се използват, когато е необходим по-висок ред на сходимост.

2.4.1 Явни методи на Рунге–Кута

Явните методи на Рунге–Кута са най-често използваният клас методи за решаване на ОДУ. За да мотивираме тяхното използване, нека първо разгледаме от геометрична гледна точка методите на Ойлер. При явния метод на Ойлер имаме

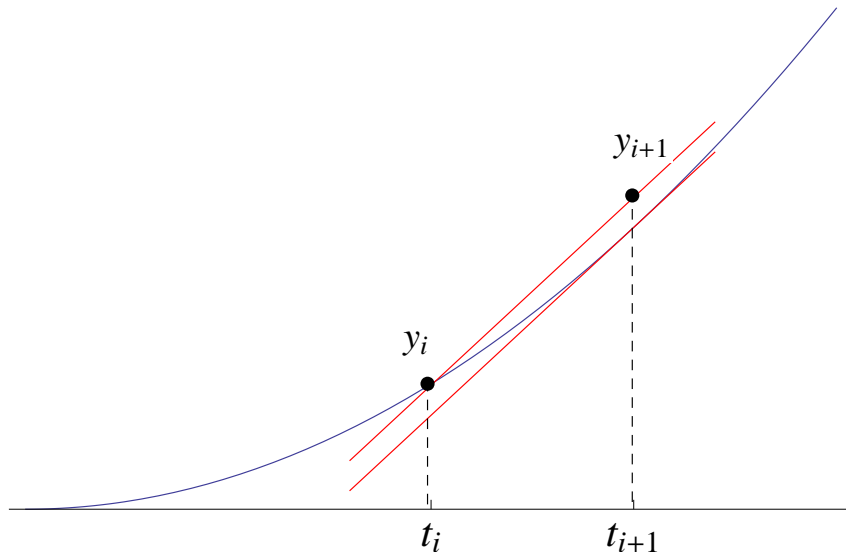
$$y_{i+1} = y_i + hf(t_i, y_i),$$

т.е. правим една стъпка по направление на допирателната в точката t_i , за да получим приближението в точката t_{i+1} :



Както знаем, ЛГА при това е от първи ред, т.е. $O(h)$.

При неявния метод на Ойлер използваме направлението, зададено от допирателната в точката t_{i+1} :

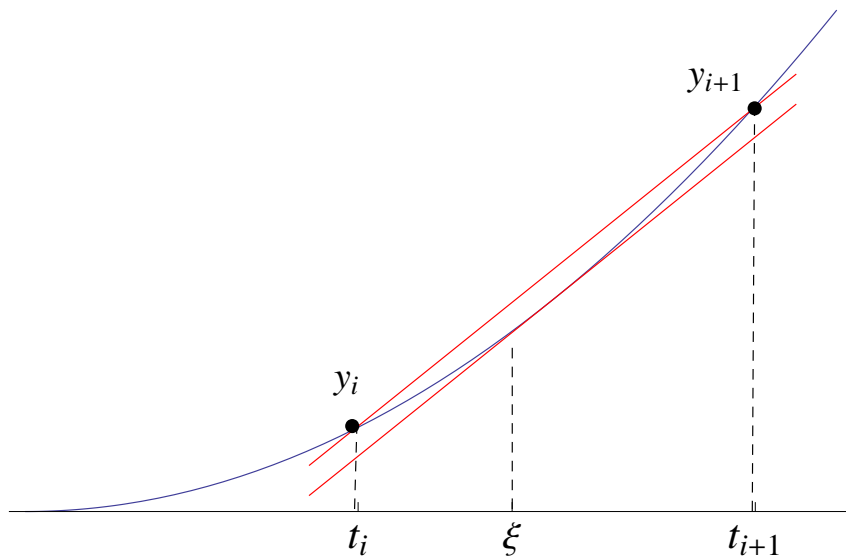


ЛГА и в този случай е $O(h)$.

От друга страна, според Теоремата за крайните нараствания, съществува ξ , така че

$$u(t_i + h) - u(t_i) = u'(\xi)(t_{i+1} - t_i) = u'(\xi)h.$$

С други думи, ако използваме направлението, зададено от допирателната в тази точка, ЛГА би била 0:



Ние, разбира се, няма как да намерим точката ξ но нейното съществуване ни дава основание да вземем производните в различни точки в интервала $[t_i, t_{i+1}]$ и да ги усредним с някакви тегла, така че да получим стойност, близка (в някакъв смисъл) до стойността в ξ . Нека означим стойностите на производните в тези точки (умножени по h) с k_1, k_2, \dots, k_s . Общият вид на s -етапните методи на Рунге-Кута е

$$\frac{y_{i+1} - y_i}{h} = \frac{1}{h}(p_1 k_1 + p_2 k_2 + \dots + p_s k_s),$$

където p_1, p_2, \dots, p_s са теглата, с които вземаме всяка от производните. Ще ги определим така, че ЛГА да е възможно най-малка.

k_1 е производната в точката t_i . За нея имаме

$$k_1 = hf(t_i, y_i).$$

k_2 е производната в някоя друга точка $t_i + \alpha_2 h$. За да я пресметнем обаче (т.е. да пресметнем дясната страна в диференциалното уравнение), ни е необходима стойността на решението в тази точка, което ни е неизвестно. Ще го апроксимираме на база на известната информация, като за неговата стойност вземем $y_i + \beta_{2,1} k_1$. Тогава

$$k_2 = hf(t_i + \alpha_2 h, y_i + \beta_{2,1} k_1).$$

Параметрите $\alpha_2, \beta_{2,1}$ отново подлежат на определяне, така че ЛГА да е възможно най-малка.

Разсъждавайки аналогично и за производните в следващите точки, получаваме следния общ вид на методите на Рунге-Кута:

Алгоритъм 4: Методи на Рунге-Кута

Даден s -етапен метод на Рунге-Кута има следния общ вид:

$$\begin{aligned} \frac{y_{i+1} - y_i}{h} &= \frac{1}{h}(p_1 k_1 + p_2 k_2 + \dots + p_s k_s), \\ k_1 &= hf(t_i, y_i) \\ k_2 &= hf(t_i + \alpha_2 h, y_i + \beta_{2,1} k_1) \\ k_3 &= hf(t_i + \alpha_3 h, y_i + \beta_{3,1} k_1 + \beta_{3,2} k_2) \\ k_j &= hf(t_i + \alpha_j h, y_i + \beta_{j,1} k_1 + \dots + \beta_{j,j-1} k_{j-1}), \quad j = 4, \dots, s. \end{aligned}$$

Да обърнем внимание, че методите на Рунге-Кута са **едностъпкови методи**, тъй като при намирането на y_{i+1} използват само стойността на y_i , но не и на стойностите на приближеното решение в предходните точки.

В литературата могат да се открият коефициентите на различни методи от този тип с различен ред на сходимост. Често коефициентите се записват в т.нар. Таблица на Butcher:

$$\begin{array}{c|cccc} 0 & & & & \\ \alpha_2 & \beta_{2,1} & & & \\ \vdots & \vdots & \ddots & & \\ \alpha_s & \beta_{s,1} & \cdots & \beta_{s,s-1} & \\ \hline & p_1 & p_2 & \cdots & p_s \end{array}$$

Тук прочети стр. 21–23 от [3].

По подобен начин на извежданията в [3] могат да бъдат получени и методи от по-висок ред, но извеждането им от техническа гледна точка става много по-трудно. Така например за извеждането на метод от 12 ред условията, които се получават са 5972 на брой. Методи дори от толкова висок ред се налага понякога да се използват на практика. Добрата новина е, че те са известни и могат да бъдат открити в литературата. Интересен факт, който допълнително усложнява нещата, е, че методите, които имат ред на сходимост $p > 4$ за скаларни уравнения, може да имат ред на сходимост, по-малък от p за системи уравнения. Много по-подробна информация за

методите на RK могат да бъдат намерени в монографията на Butcher [1].

Най-често използваният метод за решаване на ОДУ при равномерна мрежа е методът RK4, който има таблица на Butcher

$$\begin{array}{c|ccc} 0 & & & \\ 1/2 & 1/2 & & \\ 1/2 & 0 & 1/2 & \\ 1 & 0 & 0 & 1 \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array}$$

Записан подробно, методът има следния вид:

$$\begin{aligned} \frac{y_{i+1} - y_i}{h} &= \frac{1}{h}(1/6k_1 + 1/3k_2 + 1/3k_3 + 1/6k_4), \\ k_1 &= hf(t_i, y_i), \\ k_2 &= hf(t_i + 1/2h, y_i + 1/2k_1), \\ k_3 &= hf(t_i + 1/2h, y_i + 1/2k_2), \\ k_4 &= hf(t_i + h, y_i + k_3). \end{aligned}$$

2.4.2 Изследване на A-устойчивост и монотонност за методите на Рунге–Кута

Ще коментираме въпроса за устойчивостта и монотонността на явните методи на Рунге–Кута на базата на два примера.

Пример 21. Да изследваме A-устойчивостта и монотонността на метода на Рунге–Кута

$$\begin{aligned} \frac{y_{i+1} - y_i}{h} &= \frac{1}{2h}(k_1 + k_2), \\ k_1 &= hf(t_i, y_i), \\ k_2 &= hf(t_i + h, y_i + k_1). \end{aligned}$$

Изследваме A-устойчивост и монотонност върху моделната задача. Получаваме последователно

$$\begin{aligned} k_1 &= h\lambda y_i, \\ k_2 &= h\lambda(y_i + k_1) = h\lambda(y_i + h\lambda y_i) = h\lambda y_i + (h\lambda)^2 y_i, \\ y_{i+1} &= y_i + \frac{1}{2}(h\lambda y_i + h\lambda y_i + (h\lambda)^2 y_i) = \left(1 + h\lambda + \frac{1}{2}(h\lambda)^2\right) y_i. \end{aligned}$$

Следователно, за да бъде методът A-устойчив, трябва да е изпълнено условието

$$\left|1 + \lambda h + \frac{1}{2}(h\lambda)^2\right| < 1.$$

Решавайки последното уравнение по отношение на $z := h\lambda$, получаваме

$$-2 < z < 0 \implies h < -2/\lambda.$$

Условието за монотонност

$$1 + z + \frac{1}{2}z^2 > 0$$

е винаги изпълнено, т.е. методът е монотонен за всяко $h > 0$.

Пример 22. Да се изследва за А-устойчивост и монотонност методът на Рунге–Кута

$$\begin{aligned}\frac{y_{i+1} - y_i}{h} &= \frac{1}{6h}(k_1 + 4k_2 + k_3), \\ k_1 &= hf(t_i, y_i), \\ k_2 &= hf(t_i + h/2, y_i + k_1/2), \\ k_3 &= hf(t_i + h, y_i - k_1 + 2k_2).\end{aligned}$$

Изследваме А-устойчивост и монотонност върху моделната задача. Получаваме последователно

$$\begin{aligned}k_1 &= h\lambda y_i, \\ k_2 &= h\lambda(y_i + k_1/2) = h\lambda(y_i + h\lambda y_i/2) = h\lambda y_i + \frac{(h\lambda)^2}{2}y_i, \\ k_3 &= h\lambda(y_i - k_1 + 2k_2) = h\lambda(y_i - h\lambda y_i + 2h\lambda y_i + (h\lambda)^2 y_i) \\ &= h\lambda y_i + (h\lambda)^2 y_i + (h\lambda)^3 y_i \\ y_{i+1} &= y_i + \frac{1}{6} \left[h\lambda y_i + 4 \left(h\lambda y_i + \frac{(h\lambda)^2}{2} y_i \right) + h\lambda y_i + (h\lambda)^2 y_i + (h\lambda)^3 y_i \right], \\ &= \left(1 + h\lambda + \frac{1}{2}(h\lambda)^2 + \frac{1}{6}(h\lambda)^3 \right) y_i.\end{aligned}$$

Следователно, за да бъде методът А-устойчив, трябва да е изпълнено условието

$$\left| 1 + h\lambda + \frac{1}{2}(h\lambda)^2 + \frac{1}{6}(h\lambda)^3 \right| < 1.$$

Решавайки последното уравнение по отношение на $z := h\lambda$, получаваме

$$-2.51275 < z < 0 \implies h < -2.51275/\lambda.$$

Условието за монотонност

$$1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 > 0$$

е изпълнено за $z > -1.59607$, т.е. $h < -1.59607/\lambda$.

Забележка. Може да се покаже, че за всички явни методи на Рунге–Кута условието за А-устойчивост има вида

$$|1 + z + O(z^2)| < 1.$$

Последното очевидно няма как да бъде изпълнено за всяко z , т.е. **явните методи на Рунге–Кута са условно А-устойчиви.**

N.B. 17

Всички явни методи на Рунге–Кута са условно А-устойчиви.

Съществуват и неявни методи на РК, които няма да разглеждаме в настоящия курс.

2.4.3 Методи с адаптивен избор на стъпката.

Разгледаните дотук методи използват равномерна мрежа. Това обаче невинаги е целесъобразно. Нека разгледаме следната задача на Коши.

$$\begin{aligned} \frac{du}{dt} &= -b(t)u + t, \quad t \in (0, 10], \\ u(0) &= 1, \end{aligned} \tag{2.17}$$

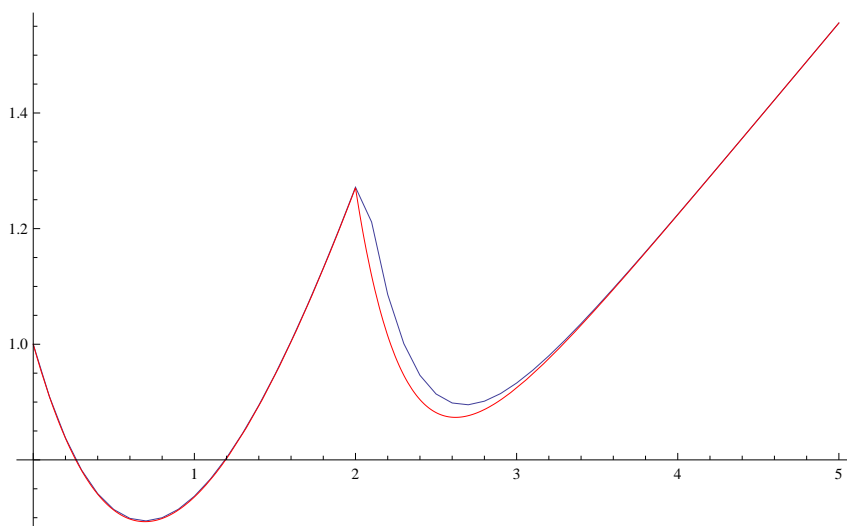
където

$$b(t) = \begin{cases} 1 & 0 \leq t \leq 2 \\ 3 & 2 < t \end{cases}.$$

имаща точно решение

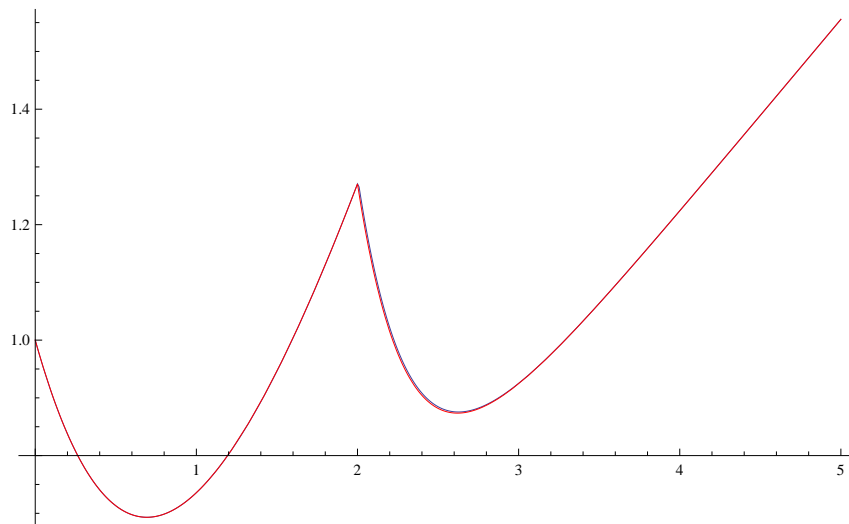
$$u(t) = \begin{cases} t - 1 + 2e^{-t} & 0 \leq t \leq 2 \\ \frac{1}{3}t - \frac{1}{9} + e^{-3t} \left(\frac{4}{9}e^6 + 2e^4 \right) & t > 2 \end{cases}.$$

Решавайки задачата с метод на Рунге-Кута от трети ред със стъпка $h = 0.1$, получаваме следния резултат;



Както виждаме от графиката, методът дава добър резултат до особеността на решението в $x = 2$, след което обаче графиките на приближеното и точното решение сериозно се разминават.

Значително по-добър резултат получаваме със стъпка $h = 0.01$:



Възниква обаче логичният въпрос защо е необходимо да използваме толкова малка стъпка в целия интервал, след като в част от него 10 пъти по-голяма стъпка дава добри резултати.

При използването на равномерна мрежа стъпката трябва да се съобрази с най-лошия случай.

Идеята на методите с адаптивен избор на стъпката е да използваме малка стъпка там, където това е необходимо (обикновено, където решението се изменя бързо) и по-голяма стъпка, където поведението на решението го позволява.

Най-общо, идеята на методите с адаптивен избор на стъпката можем да изложим със следния алгоритъм.

Нека означим $h_i := t_{i+1} - t_i$, $i = 0, 1, 2, \dots$

АЛГОРИТЪМ:

Избираме първоначална стойност за h_0 ; инициализираме $y_0 = u_0$, $t_0 = 0$.

Докато $t_i < T$:

1. Оценяваме локалната грешка, err_i , която бихме допуснали със стъпка h_i .
2. Докато $err_i > tol$, намаляваме стъпката h_i и отново оценяваме err_i .
3. Пресмятаме y_{i+1} , t_{i+1} , както и първоначална стойност за h_{i+1} ; увеличаваме i с 1.

Нека разгледаме последователно стъпки 1, 2 и 3.

1. Оценката на грешката зависи от конкретния метод, който използваме.

Един възможен начин за практическа оценка на грешката е да решим задачата с два различни метода от различен ред, например втори и трети. Съответните приближени решения ще означим с $y_i^{<2>}$ и $y_i^{<3>}$. За оценка на грешката вземаме

$$err_i := y_i^{<3>} - y_i^{<2>}. \quad (2.18)$$

Да отбележим, че тази оценка е апостериорна, защото използва получения от метода резултат. Числените методи за диференциални уравнения върху адаптивни мрежи се базират именно на апостериорни оценки на грешката.

Неудобството при оценката (2.18) е, че трябва да се пресмята приближеното решение по два различни начина, тоест да се правят допълнителни операции. Съществуват обаче двойки методи на Рунге-Кута, при които този проблем бива решен, тъй като методът от по-висок ред преизползва всички пресмятания на метода от по-нисък ред. Да разгледаме следната двойка методи на Рунге-Кута от втори и трети ред:

0			
1	1		
1/2	1/4	1/4	
	1/6	1/6	2/3
	1/2	1/2	0

За метода от втори ред се използват коефициентите в първите два реда, а за метода от трети ред – в първите три.

Коефициентите в първия ред под чертата се отнасят за метода от трети ред, а във втория ред – за метода от втори ред.

Така получаваме

$$\begin{aligned} y_{i+1}^{<3>} &= y_i + 1/6k_1 + 1/6k_2 + 2/3k_3, \\ y_{i+1}^{<2>} &= y_i + 1/2k_1 + 1/2k_2. \end{aligned}$$

Следователно за оценката на грешката err_i имаме

$$err_i = y_{i+1}^{<3>} - y_{i+1}^{<2>} = -\frac{1}{3}(k_1 + k_2 - 2k_3).$$

2. Сега ще изведем формула, по която ще променяме стъпката. Нека текущата стъпка е $h_{i,temp}$. Тъй като използваме метод от трети ред, имаме

$$err_i \approx Ch_{i,temp}^3 \tag{2.19}$$

за някоя константа C .

Искаме да изберем нова стойност на стъпката, $h_{i,new}$ така, че грешката да е приблизително равна на зададения толеранс tol , т.е.

$$tol \approx Ch_{i,new}^3 \implies h_{i,new} = \left(\frac{tol}{C}\right)^{1/3}.$$

Изразявайки C от (2.19), получаваме окончателно

$$h_{i,new} = h_{i,temp} \left(\frac{tol}{err_i}\right)^{1/3}.$$

Да обърнем внимание, че ако $err_i < tol$, то горната формула води до увеличаването на стъпката. С други думи, ако сме използвали в текущата итерация по-малка стъпка, отколкото е било необходимо, за следващата итерация стъпката ще се увеличи. Затова същата формула използваме и за определянето на начална стойност за h_{i+1} .

2.4.4 Числени експерименти (Практикум)

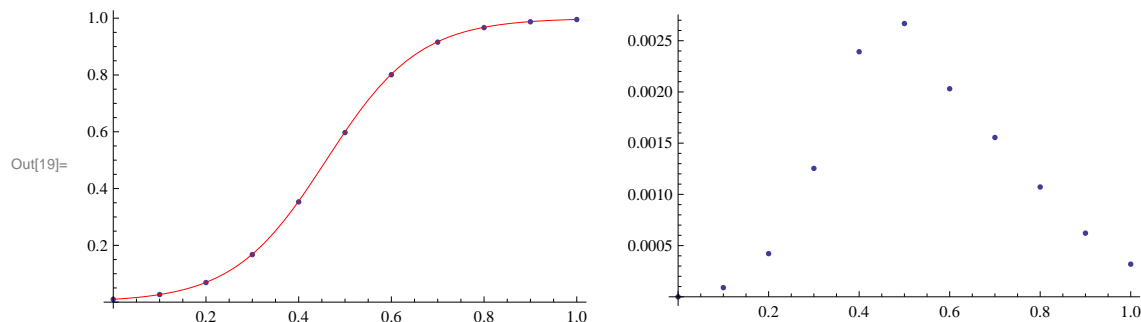
Пример 23. Прилагаме функция в Mathematica, реализираща метода на Рунге-Кута от четвърти ред за произволна задача на Коши (2.1).

```
In[6]:= RK4[f_, n_, t0_, T_] := (
  h = (T - t0) / n // N;
  t = Table[i h, {i, 0, n}];
  y = Table[0, {n + 1}];
  y[[1]] = 0.01;
  For[i = 1, i ≤ n, i++,
    k1 = h f[t[[i]], y[[i]]];
    k2 = h f[t[[i]] + h / 2, y[[i]] + k1 / 2];
    k3 = h f[t[[i]] + h / 2, y[[i]] + k2 / 2];
    k4 = h f[t[[i]] + h, y[[i]] + k3];
    y[[i + 1]] = y[[i]] +  $\frac{1}{6}$  (k1 + 2 k2 + 2 k3 + k4)
  ];
  y
)
```

Нека я използваме, за да решим задачата на Коши (18).

```
In[20]:= n = 10;
f[t_, y_] := 10 y (1 - y);
y = RK4[f, n, 0, 1];
plotAppr = ListPlot[Table[{t[[i + 1]], y[[i + 1]]}, {i, 0, n}],
  PlotRange → All];
plotExact = Plot[ $\frac{0.01}{0.01 + (1 - 0.01) E^{-10 t}}$ , {t, 0, 1}, PlotStyle → Red];
GraphicsRow[{Show[plotAppr, plotExact, PlotRange → All],
  ListPlot[Table[{i h, Abs[y[[i + 1]] -  $\frac{0.01}{0.01 + (1 - 0.01) E^{-10 i h}}$ ]}],
    {i, 0, n}], PlotRange → All]}]
```

Графика с приближеното и точното решение и графика на грешката прилагаме по-долу.

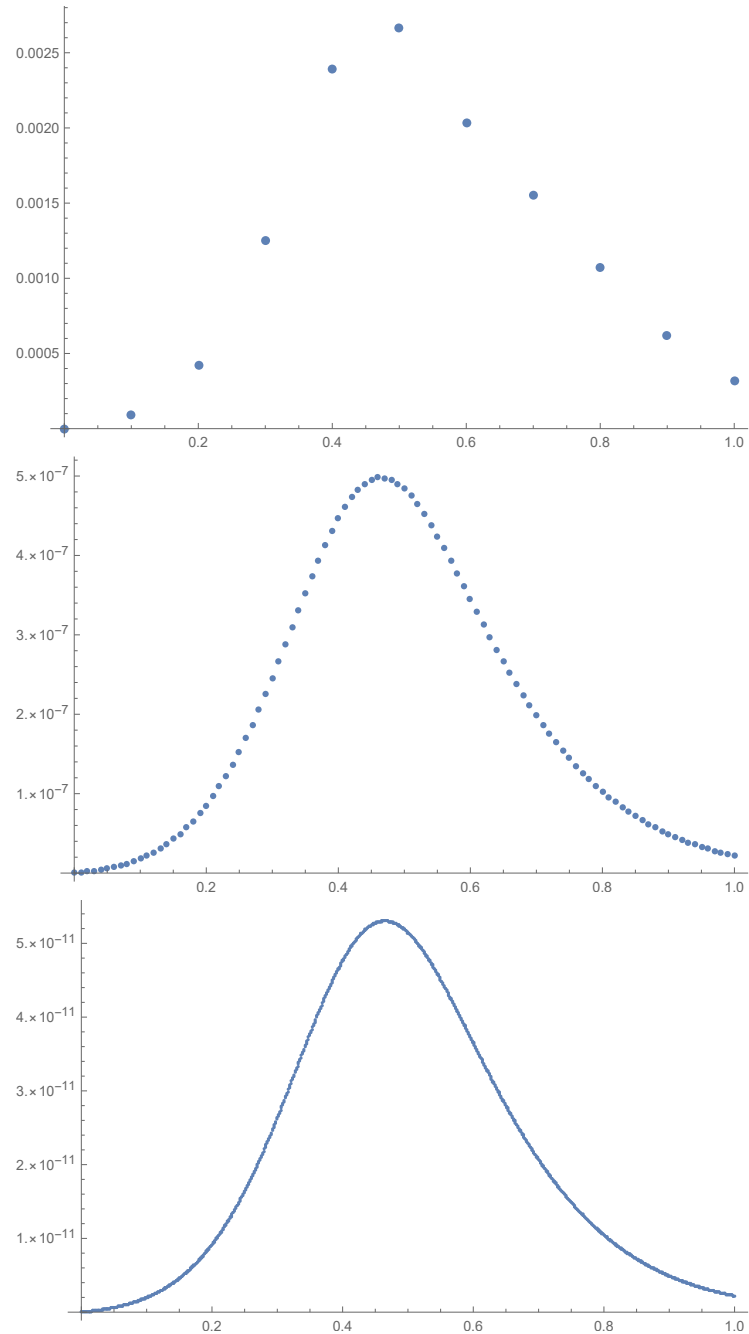


Визуално, решението при $n = 10$ изглежда много добре (сравнете с решени-

ето при методите на Ойлер).

Както може да се очаква, при намаляване на стъпката 10 пъти грешката намалява от порядъка на 10^4 пъти

На следващите графики е визуализирана грешката във възлите от мрежата за $h = 0.1, 0.01, 0.001$:



Пример 24. Привеждаме примерна реализация на метода на Рунге–Кута с адаптивен избор на стъпката, който изведохме в предишния параграф.

```

In[48]:= RK23[f_, t0_, T_, h0_, tol_] := (
    h = h0;
    t = t0;
    y = {0};
    y[[1]] = 1;
    i = 1;
    t = {0};
    While[t[[i]] < T,
        k1 = h f[t[[i]], y[[i]];
        k2 = h f[t[[i]] + h, y[[i]] + k1];
        k3 = h f[t[[i]] +  $\frac{h}{2}$ , y[[i]] +  $\frac{1}{4}$  k1 +  $\frac{1}{4}$  k2];
        While[Abs[ $\frac{1}{3}$  (k1 + k2 - 2 k3)] > tol,
            h = h  $\left( \frac{tol}{Abs[\frac{1}{3} (k1 + k2 - 2 k3)]} \right)^{1/3}$ ;
            k1 = h f[t[[i]], y[[i]];
            k2 = h f[t[[i]] + h, y[[i]] + k1];
            k3 = h f[t[[i]] +  $\frac{h}{2}$ , y[[i]] +  $\frac{1}{4}$  k1 +  $\frac{1}{4}$  k2];
        ];
        y = Append[y, y[[i]] +  $\frac{1}{6}$  k1 +  $\frac{1}{6}$  k2 +  $\frac{2}{3}$  k3];
        t = Append[t, t[[i]] + h];
        h = h  $\left( \frac{tol}{\frac{1}{3} Abs[(k1 + k2 - 2 k3)]} \right)^{1/3}$ ;
        i++;
    ];
    {t, y}
)

```

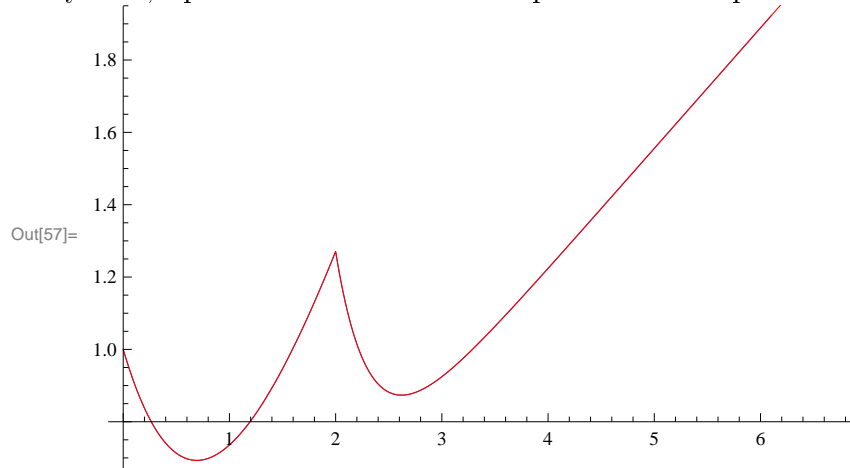
Прилагаме метода за примерната задача, с която го мотивирахме в предходния параграф, като задаваме толеранс 10^{-5}

```

In[50]:= b[t_] := If[0 ≤ t ≤ 2, 1, 3]
f[t_, y_] := -b[t] y + t;
t = RK23[f, 0, 10, 0.1, 10-5][[1]];
y = RK23[f, 0, 10, 0.1, 10-5][[2]];
plotAppr = ListLinePlot[Table[{t[[i + 1]], y[[i + 1]]}, {i, 0, Length[y] - 1}]];
exactSolution[t_] := If[0 ≤ t ≤ 2, t - 1 + 2 E-t,  $\frac{1}{3} t - \frac{1}{9} + E^{-3t} \left( \frac{4}{9} E^6 + 2 E^4 \right)$ ];
plotExact = Plot[exactSolution[t], {t, 0, 10}, PlotStyle → Red];
Show[plotAppr, plotExact]

```

Визуално, приближеното и точното решение са неразличими:

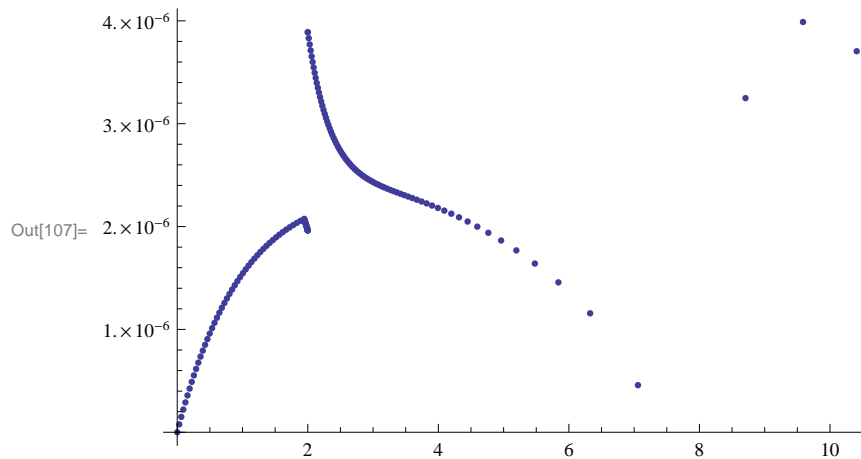


Можем да видим, че грешката действително е ограничена от 10^{-5} :

```

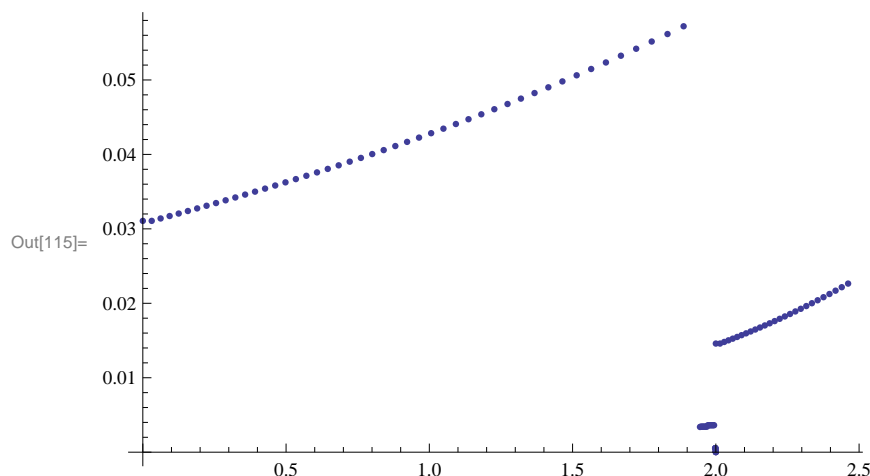
In[107]:= plotErr = ListPlot[Table[{t[[i + 1]], Abs[y[[i + 1]] - exactSolution[t[[i + 1]]]}],
{i, 0, Length[y] - 1}, PlotRange → All]

```



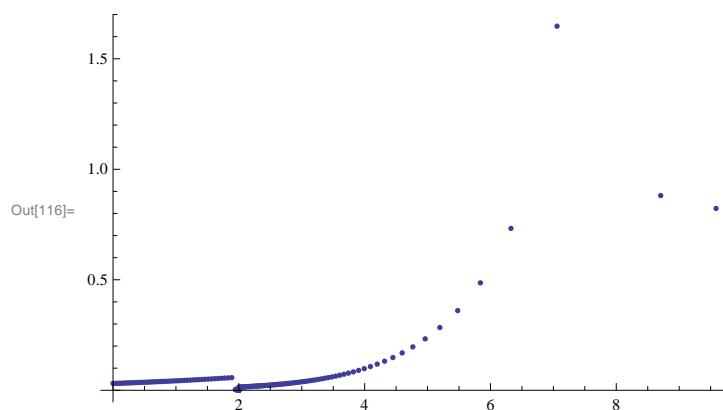
Нека видим как се е изменяла стъпката. На следващата графика изобразяваме стъпката в интервала $0 \leq t \leq 2.5$.

```
In[115]:= ListPlot[Table[{t[[i]], t[[i+1]] - t[[i]]}, {i, 1, (Length[y] - 1) / 1.5}],
, PlotRange -> All]
```



Виждаме, че около особеността на решението стъпката е намалена значително. След тази особеност методът увеличава стъпката, тъй като решението става близо до линейна функция:

```
ListPlot[Table[{t[[i]], t[[i+1]] - t[[i]]}, {i, 1, (Length[y] - 1)}],
PlotRange -> All]
```



2.5 Методи на Адамс

Вторият основен клас методи за решаване на ОДУ от първи ред, освен методите на Рунге–Кута, са т.нар. многостъпкови методи.

При едностъпковите методи, например методите на Рунге–Кута и Ойлер, пресмятаме стойността на приближеното решение y_{i+1} , използвайки само стойността y_i . Например по явния метод на Ойлер имаме

$$y_{i+1} = y_i + hf(t_i, y_i).$$

Идеята на многостъпковите методи е да използваме информацията, която сме получили за решението в няколко предходни точки, като по този начин целим да получим по-висока точност.

Един възможен подход за получаване на такива методи е следният. Както

казахме, диференциалното уравнение

$$\frac{du}{dt} = f(t, u(t))$$

може да се сведе до еквивалентно на него интегрално уравнение. За тази цел интегрираме двете страни в граници от t_i до t_{i+1} ¹ и получаваме

$$u_{i+1} - u_i = \int_{t_i}^{t_{i+1}} f(t, u(t)) dt$$

или, което е същото,

$$u_{i+1} = u_i + \int_{t_i}^{t_{i+1}} f(t, u(t)) dt. \quad (2.20)$$

За да получим диференчно уравнение, апроксимиращо горното интегрално уравнение и следователно оригиналното диференциално уравнение, ще апроксимираме интеграла в дясната страна.

За тази цел ще използваме интерполационна квадратурна формула. Нека означим $\bar{f}(t) = f(t, u(t))$. Да припомним, че идеята на интерполационните квадратурни формули е да апроксимираме подинтегралната функция с интерполационния ѝ полином на Лагранж (вж. Приложението), $\bar{f} \approx L_n(\bar{f}, t)$ и тогава получаваме приближение на интеграла, като интегрираме L_n , вместо \bar{f} . Така получаваме

$$\int_{t_i}^{t_{i+1}} f(t, u(t)) dt = \int_{t_i}^{t_{i+1}} [L_n(\bar{f}; x) + O(h^{n+1})] dt.$$

N.B. 18

Тогава е ясно, че пренебрегвайки осатъчния член, формулата, която ще се получи при апроксимиране на \bar{f} с L_n , ще има ЛГА $O(h^{n+1})$ (Проверете!)

Интерполационния полином на Лагранж построяваме, като за интерполационни възли вземем точките $t_i, t_{i-1}, t_{i-2}, \dots, t_{i-n}$.

Естествено е, разбира се, да използваме именно точките от мрежата, тъй като за построяване на интерполационния полином ни е необходимо да знаем стойността на интерполираната функция, а това е изпълнено именно в точките от мрежата. В зависимост от това дали в редицата от интерполационни възли включваме t_{i+1} , или не, получаваме неявни методи (методи на Адамс–Мултон, наричани още интерполационни или неявни методи на Адамс) и явни методи (методи на Адамс–Башфорт, наричани още екстраполационни или явни методи на Адамс).

2.5.1 Методи на Адамс–Башфорт

Използвайки възела t_i и предходните възли, получаваме **явен метод**, тъй като не използваме стойност на решението, която все още не е намерена.

¹По-общо, можем да интегрираме в произволен интервал $[t_{i-k}, t_i]$, но най-често използваните методи се получават при избор на интервала $[t_{i-1}, t_i]$

Ще изведем няколко конкретни метода, получени по този начин. За удобство правим смяната $t = t_i - \xi h$, която трансформира възлите $t_i, t_{i-1}, t_{i-2}, \dots$ съответно в $0, 1, 2, \dots$. Тогава (2.20) се свежда до

$$u_{i+1} = u_i + h \int_0^1 \bar{f}(t_i - \xi h) d\xi. \quad (2.21)$$

Едностъпков метод. При $n = 0$, единственият интерполационен възел е t_i . Търсим полинома от нулева степен, който интерполира функцията $\bar{f}(t)$ (т.е. съвпада с нея) в точката t_i . Очевидно това е константата $\bar{f}(t_i) = f(t_i, u(t_i)) =: f_i$. Имаме

$$\bar{f}(t) \approx f_i \implies h \int_0^1 \bar{f}(t_i - \xi h) d\xi \approx h f_i.$$

Замествайки апроксимацията на интеграла в (2.21), получаваме

$$y_{i+1} = y_i + h f_i,$$

т.е. отново получихме явния метод на Ойлер с ЛГА $O(h)$.

Двустъпков метод. При $n = 1$ търсим полинома от първа степен, който интерполира функцията $\bar{f}(t)$ в точките t_i, t_{i-1} . С други думи, искаме да са изпълнени интерполационните условия

ξ	0	1
стойност	f_i	f_{i-1}

Използваме интерполационната формула на Лагранж², за да намерим интерполационния полином от първа степен, който ги удовлетворява.

Базисните полиноми на Лагранж са

$$l_0(\xi) = \frac{(\xi - 1)}{(0 - 1)} = 1 - \xi,$$

$$l_1(\xi) = \frac{(\xi - 0)}{(1 - 0)} = \xi.$$

Тогава за интерполационния полином получаваме

$$\bar{f}(t_i - \xi h) \approx f_i(1 - \xi) + f_{i-1}\xi$$

Замествайки апроксимацията в дясната страна на (2.21) и пренебрегвайки остатъчния член, получаваме метода

$$y_{i+1} = y_i + \frac{h}{2}(3f_i - f_{i-1}),$$

който има ЛГА $O(h^2)$.

Тристъпков метод. При $n = 2$ търсим полинома от втора степен, който интерполира функцията $\bar{f}(t)$ в точките t_i, t_{i-1}, t_{i-2} . С други думи, искаме да са изпълнени интерполационните условия

²Информация за формулата на Лагранж виж в Приложението. Вземайки предвид, че възлите са равноотдалечени, по-удачно би било да използваме формулата на Нютон с крайни разлики, както е направено в параграф 5 на [3]. Ние обаче се спираме на формулата на Лагранж, тъй като нейната идея ще бъде много важна за нас по-късно в курса.

ξ	0	1	2
стойност	f_i	f_{i-1}	f_{i-2}

Използваме интерполационната формула на Лагранж, за да намерим интерполационния полином от първа степен, който ги удовлетворява.

Базисните полиноми на Лагранж са

$$l_0(\xi) = \frac{(\xi - 1)(\xi - 2)}{(0 - 1)(0 - 2)} = \frac{1}{2}(\xi^2 - 3\xi + 2),$$

$$l_1(\xi) = \frac{(\xi - 0)(\xi - 2)}{(1 - 0)(1 - 2)} = -\xi(\xi - 2),$$

$$l_2(\xi) = \frac{(\xi - 0)(\xi - 1)}{(2 - 0)(2 - 1)} = \frac{1}{2}(\xi^2 - \xi).$$

Тогава за интерполационния полином получаваме

$$\begin{aligned} \bar{f}(t_i - \xi h) &\approx f_i l_0(\xi) + f_{i-1} l_1(\xi) + f_{i-2} l_2(\xi) \\ &= \frac{f_i - 2f_{i-1} + f_{i-2}}{2} \xi^2 + \frac{-3f_i + 4f_{i-1} - f_{i-2}}{2} \xi + f_i \end{aligned}$$

Замествайки апроксимацията на интеграла в (2.21) и пренебрегвайки остатъчния член, получаваме метода

$$y_{i+1} = y_i + \frac{h}{12}(23f_i - 16f_{i-1} + 5f_{i-2}),$$

който има ЛГА $O(h^3)$.

Алгоритъм 5: Методи на Адамс–Башфорт

Общият вид на m -стъпковия метод на Адамс–Башфорт, който има ЛГА $O(h^m)$, е

$$y_{i+1} = y_i + h(C_1 f_i + C_2 f_{i-1} + \dots + C_m f_{i-m+1}).$$

Ние изведохме методите за $m = 1, 2, 3$. Методи от по-висок ред могат да бъдат получени по същия начин. Коефициентите C_i могат да бъдат намерени лесно и в литературата.

N.B. 19

Методите на Адамс–Башфорт са по-бързи от методите на Рунге–Кута със същия ред на точност, но обикновено методите на Рунге–Кута имат по-добро поведение от гледна точка на устойчивостта на метода. Последното е важно за задачи, при които условието за устойчивост може да означава, че методът на Адамс трябва да се използва с твърде малка стъпка, което да го направи неприложим на практика. **Изборът на метод за решаването на всяка конкретна задача зависи от това какво се изисква от него – бързодействие, висока точност, устойчивост и т.н.**

2.5.2 Методи на Адамс–Мултон

Аналогично могат да се получат **неявни методи**, като при построяването на интерполационния полином, апроксимиращ подинтегралната функция в (2.21), използваме възли $t_{i+1}, t_i, \dots, t_{i+1-n}$. Тези методи се наричат **методи на Адамс–Мултон** или още неявни (интерполационни) методи на Адамс. Аналогично на явните формули, за удобство правим смяната $t = t_{i+1} - \xi h$ и възлите от горната редица се трансформират в $0, 1, \dots, n$.

$n = 0$. При $n = 0$, единственият интерполационен възел е t_{i+1} . Търсим полинома от нулева степен, който интерполира функцията $\bar{f}(t)$ в точката t_{i+1} . Очевидно това е константата $\bar{f}(t) =: f_{i+1}$. Имаме

$$\bar{f}(t_i - \xi h) \approx f_i \implies h \int_0^1 \bar{f}(t_{i+1} - \xi h) d\xi \approx h f_{i+1}.$$

Замествайки апроксимацията на интеграла в (2.20), получаваме

$$y_{i+1} = y_i + h f_{i+1},$$

т.е. получихме неявния метод на Ойлер с ЛГА $O(h)$.

$n = 1$. Базисните полиноми на Лагранж са $l_0 = 1 - \xi$ и $l_1 = \xi$ и

$$L_1(f; \xi) = (f_i - f_{i+1})\xi + f_{i+1}.$$

Тогава получаваме

$$u_{i+1} - u_i \approx h \int_0^1 [(f_i - f_{i+1})\xi + f_{i+1}] d\xi.$$

Пренебрегвайки остатъчния член, получаваме числената схема

$$y_{i+1} = y_i + \frac{h}{2}(f_i + f_{i+1}),$$

т.е. получихме подобрения метод на Ойлер с ЛГА $O(h^2)$.

$n = 2$. Базисните полиноми на Лагранж са $l_0 = \frac{1}{2}(\xi^2 - 3\xi + 2)$, $l_1 = -\xi(\xi - 2)$ и $l_2 = \frac{1}{2}(\xi^2 - \xi)$. Тогава

$$L_2(f; \xi) = \frac{f_{i+1} - 2f_i + f_{i-1}}{2}\xi^2 + \frac{-3f_{i+1} + 4f_i - f_{i-1}}{2}\xi + f_{i+1}.$$

Тогава получаваме

$$u_{i+1} - u_i \approx h \int_0^1 \left[\frac{f_{i+1} - 2f_i + f_{i-1}}{2}\xi^2 + \frac{-3f_{i+1} + 4f_i - f_{i-1}}{2}\xi + f_{i+1} \right] d\xi.$$

Пренебрегвайки остатъчния член, получаваме числената схема

$$y_{i+1} = y_i + \frac{h}{2}(5f_{i+1} + 8f_i - f_{i-1}),$$

която има ЛГА $O(h^3)$.

Алгоритъм 6: Методи на Адамс–Мултон

Общият вид на m -стъпковия метод на Адамс–Мултон, който има ЛГА $O(h^m)$ е

$$y_{i+1} = y_i + h(C_0 f_{i+1} + C_1 f_i + \dots + C_m f_{i+1-m}).$$

Ние изведохме методите за $m = 0, 1, 2$. Методи от по-висок ред могат да бъдат получени по същия начин. Коефициентите C_i могат да бъдат намерени лесно и в литературата.

2.5.3 Практически въпроси, свързани с реализацията на методите на Адамс

Нека отбележим следните два важни въпроса, свързани с използването на методите на Адамс.

N.B. 20

За да могат да се правят изчисления с помощта на даден m -стъпков метод, е необходимо да са известни стойностите на приближеното решение в първите m точки от мрежата. Тези стойности трябва да бъдат намерени с едностъпков метод, **който има поне същия ред на точност като използвания метод на Адамс.**

N.B. 21

Важно е да отбележим, че, **за да бъде имплементацията на метода ефективна, е необходимо да пазим не само стойностите на приближеното решение, но и на дясната страна в точките от мрежата**, за да избегнем многократното пресмятане на десните страни в едни и същи точки. **Би било сериозна грешка да не използваме този факт, тъй като в противен случай методът ще изпълнява много по-голям брой операции** (сравними с метода на Рунге–Кута).

2.5.4 Предикторно-коректорни методи

Тук прочети параграф 5.3 на стр. 34–35 от [3].

2.5.5 Числени експерименти (Практикум)

Ще предложим функция, имплементираща метод на Адамс–Башфорт от четвърти ред:

$$y_{i+1} = y_i + \frac{h}{24} (55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3})$$

Нека отбележим, че за пресмятането на приближеното решение в $i + 1$ -вата точка е необходимо решението в предходните 4 точки. Следователно стойностите в първите четири точки от мрежата не можем да намерим директно чрез горната формула. За тази цел се използва едностъпков метод, имащ същия ред

на апроксимация. Ние ще използваме метода на Рунге–Кута от четвърти ред, който разгледахме по-рано.

```

AB4[f_, u0_, h_, t0_, T_] := (
  n = Ceiling[(T - t0) / h];
  t = Table[i h, {i, 0, n}];
  y = Table[0, {n + 1}];
  fi = Table[0, {n + 1}]; (*list of values of the right-hand side f*)
  y[[1]] = u0;
  (*Compute the first approximate values, using the RK4 method*)
  For[i = 1, i ≤ 3, i++,
    k1 = h f[t[[i]], y[[i]]];
    k2 = h f[t[[i]] + h / 2, y[[i]] + k1 / 2];
    k3 = h f[t[[i]] + h / 2, y[[i]] + k2 / 2];
    k4 = h f[t[[i]] + h, y[[i]] + k3];
    y[[i + 1]] = y[[i]] +  $\frac{1}{6}$  (k1 + 2 k2 + 2 k3 + k4);
    fi[[i + 1]] = f[t[[i + 1]], y[[i + 1]]];
  ];
  (*Compute the remaining values, using the AB4 formula*)
  For[i = 4, i < n + 1, i++,
    y[[i + 1]] =
      y[[i]] +  $\frac{h}{24}$  (55 fi[[i]] - 59 fi[[i - 1]] + 37 fi[[i - 2]] - 9 fi[[i - 3]]);
    fi[[i + 1]] = f[t[[i + 1]], y[[i + 1]]];
  ];
  y
)

```

Обърнете внимание, че в горната реализация на всяка стъпка се запазва не само стойността на y_{i+1} , но и на $f(t_{i+1}, y_{i+1})$.

Тук и в следващите секции ще разгледаме примери, на базата на които ще илюстрираме приложението на разглежданите методи на практика.

Анализ на формата на осево-симетрична капка.

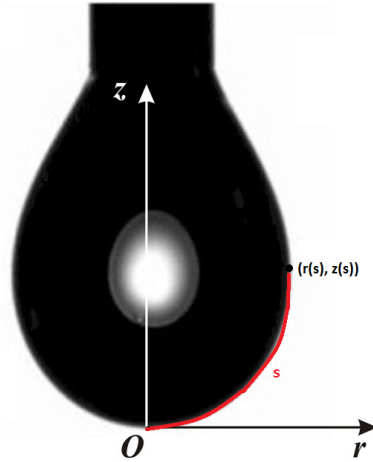
Методът на Адамс–Башфорт датира от края на 19-ти век, като първо той е бил приложен за определяне профила на капка от даден флуид. Ето защо ние ще го илюстрираме именно върху тази задача.

Разглеждаме системата

$$\begin{aligned} \frac{dr}{ds} &= \cos \varphi, \\ \frac{dz}{ds} &= \sin \varphi, \\ \frac{d\varphi}{ds} &= 2b + cz - \frac{\sin \varphi}{r}, \end{aligned}$$

която описва профила на капка, пусната от даден капиляр, под действието на гравитацията. Профилът се описва като параметрична крива $l = (r(s), z(s))$,

която е параметризирана по дължината на дъгата, s , измерена от върха на капката където φ е ъгълът, който допирателната в съответната точка сключва с абсцисната ос. Параметърът b описва кривината във върха на капката, а c е параметър, който характеризира веществото.



Началните условия са $r(0) = z(0) = \varphi(0) = 0$. Да отбележим, че системата има особеност при $s = 0$, тъй като дясната страна на последното уравнение не е дефинирана. Може да се покаже обаче, че дясната граница съществува и

$$\left. \frac{d\varphi}{ds} \right|_{0+} = b.$$

Тогава е естествено да приемем последното като дефиниция на дясната страна за $s = 0$.

За да можем да използваме вече написаната в Mathematica функция *AB4*, първо трябва да запишем системата като едно векторно уравнение. Полагайки $\mathbf{u} = (r, z, \varphi)^T$ и $\mathbf{f}(\mathbf{u}) = (f_1, f_2, f_3)^T$, където

$$\begin{aligned} f_1(\mathbf{u}) &= \cos \varphi, \\ f_2(\mathbf{u}) &= \sin \varphi, \\ f_3(\mathbf{u}) &= \begin{cases} 2b + cz - \frac{\sin \varphi}{r}, & s > 0 \\ b, & s = 0 \end{cases}. \end{aligned}$$

получаваме системата

$$\begin{aligned} \frac{d\mathbf{u}}{dt} &= \mathbf{f}(\mathbf{u}), \quad s > 0, \\ \mathbf{u}(0) &= (0, 0, 0)^T. \end{aligned}$$

Както казахме, ще използваме функцията *AB4* за нейното решаване, като малко ще я модифицираме. Това се налага, предвид факта, че обикновено задачата е скалирана така (т.е. са избрани такива мерни единици), че радиусът на капиляра да бъде единица. Това означава, че трябва да интегрираме, докато r стане 1.

Вземайки предвид, че капката е осево-симетрична, то е достатъчно да вземем само половината от нея.

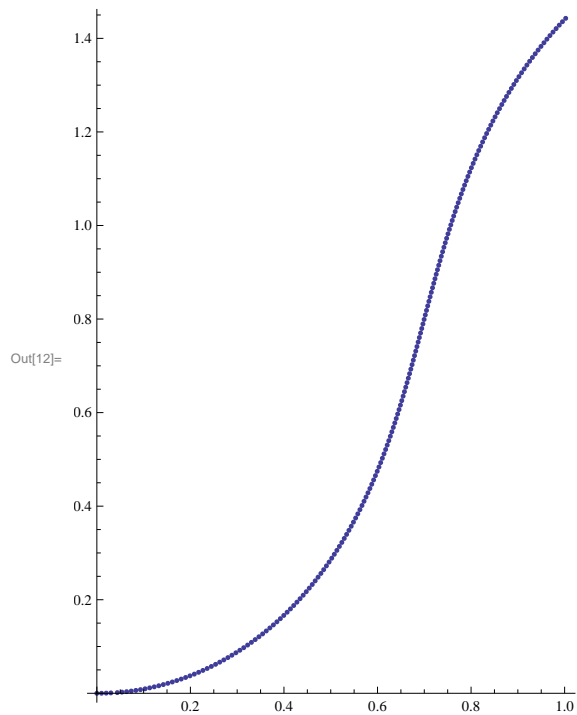
Прилагаме по-долу модифицирания код, заедно с приложението му за определяне на профила на капката при стойности на параметрите $b = 1.84366$ и $c = -2.9$.

```

In[7]:= AB4[f_, u0_, h_] := (
  y = Table[0, {4}];
  fi = Table[0, {4}]; (*list of values of the right-hand side f*)
  y[[1]] = u0;
  (*Compute the first approximate values, using the RK4 method*)
  For[i = 1, i ≤ 3, i++,
    k1 = h f[y[[i]]];
    k2 = h f[y[[i]] + k1 / 2];
    k3 = h f[y[[i]] + k2 / 2];
    k4 = h f[y[[i]] + k3];
    y[[i + 1]] = y[[i]] +  $\frac{1}{6}$  (k1 + 2 k2 + 2 k3 + k4);
    fi[[i + 1]] = f[y[[i + 1]]];
  ];
  i = 4;
  (*Compute the remaining values, using the AB4 formula*)
  While[y[[i, 1]] ≤ 1,
    y = Append[y,
      y[[i]] +  $\frac{h}{24}$  (55 fi[[i]] - 59 fi[[i - 1]] + 37 fi[[i - 2]] - 9 fi[[i - 3]])];
    fi = Append[fi, f[y[[i + 1]]]];
    i++;
  ];
  y
)
In[8]:= b = 1 / 0.5424
c = -2.9;
f[u_] := {
  Cos[u[[3]]],
  Sin[u[[3]]],
  If[u[[1]] ≠ 0, 2 b + c u[[2]] -  $\frac{\text{Sin}[u[[3]]]}{u[[1]]}$ , b]
}
res = AB4[f, {0, 0, 0}, 0.01];
ListPlot[Table[{res[[i, 1]], res[[i, 2]]}, {i, 1, Length[res]}],
  AspectRatio → Automatic]

```

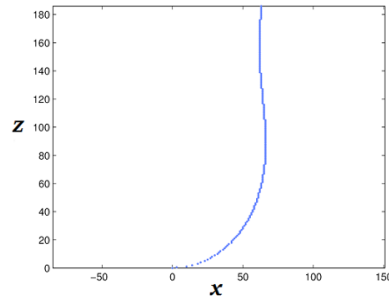
В резултат от изпълнението на горния код, получаваме следния профил на капка.



Решаването на горната система ОДУ е част от т.нар. Axisymmetric Drop Shape Analysis. Това е метод, на базата на който работят много уреди за определяне на повърхностно напрежение на течности (тензиометри).



Повърхностното напрежение е характеристика на течностите, която ги кара да възприемат форма, при която повърхността им е минимална. Направата на измервателни уреди за измерване на повърхностно напрежение е голяма индустрия. За тази цел капка от дадено вещество се пуска от капиляр. На капката се прави снимка, която се дигитализира и така профилът се описва с множество от точки (заради симетрията разглеждаме само половината).



От друга страна, отчитайки силите, действащи върху капката, както видяхме, може да се определи профилът при зададени параметри. Софтуерът, управляващ уредите, определя стойността на параметрите (в частност параметърът c , в който участва повърхностното напрежение) така, че експерименталният и теоретичният профил да съвпадат.

2.6 Сходимост на методите за решаване на ОДУ. Теорема на Лакс. Метод на Рунге за практическа оценка на грешката и реда на сходимост.

Тук прочети стр. 27–30 от [3].

2.7 Сравнение на методите за решаване на задача на Коши за ОДУ

2.7.1 Основни предимства и недостатъци на методите

В настоящия параграф ще обобщим казаното дотук, като ще направим едно кратко сравнение на база на няколко признака между методите за числено решаване на ОДУ, без да претендираме за каквато и да е изчерпателност.

1. Явни методи

(а) Метод на Ойлер

- + Бързи и лесни за имплементиране
- Бавно сходящи.

(б) Методи на Рунге-Кута

- + Съществуват методи с висок ред на сходимост
- + По-добри свойства от гледна точка на устойчивост спрямо методите на Адамс-Башфорт
- + Удобни са за използване с адативен избор на стъпката
- По-бавни от методите на Адамс-Башфорт

(в) Методи на Адамс-Башфорт

- + Съществуват методи с висок ред на сходимост
- + По-бързи спрямо методите на Рунге-Кута
- + Често се комбинират с неявните методи на Адамс-Мултон като предикторно-коректорен метод
- По-лоши свойства от гледна точка на устойчивост спрямо методите на Рунге-Кута

2. Явни срещу неявни методи

- Явните методи са значително по-бързи.
- За твърди задачи явните методи са практически непримени.
- Често неявните методи се използват заедно с явен метод в двойка предиктор-коректор.

3. Адаптивен избор на стъпката срещу равномерна мрежа

- Равномерната мрежа позволява да се контролира времето за изпълнение на алгоритъма, тъй като броят операции е предварително известен.
- Адаптивният избор на стъпката позволява да се контролира автоматично точността на приближеното решение.
- В задачи, при които решението се изменя бързо в малък интервал от дефиниционната област и бавно – в останалата част – методите с адаптивен избор на стъпката имат явно предимство. В противен случай може да се окаже, че използването на равномерна стъпка ще е по-удачно от гледна точка на бързодействие, тъй като се избягват проверките за точност и пресмятането на нова стъпка на всяка итерация.

На база на горното сравнение още веднъж ще подчертаем, че познаването на различни методи ни позволява да използваме метод, който е подходящ, от гледна точка на изискванията на конкретната приложна задача, която решаваме.

2.7.2 Числени експерименти (Практикум)

Определяне траекторията на ракета.

Отчитайки силите, действащи върху тяло, изстреляно под ъгъл θ_0 с начална скорост v_0 , неговата траектория се описва със системата ОДУ

$$\begin{aligned}\frac{dx}{dt} &= v \cos \theta, \\ \frac{dy}{dt} &= v \sin \theta, \\ \frac{dv}{dt} &= -\frac{1}{2m}c\rho sv^2 - g \sin \theta, \\ \frac{d\theta}{dt} &= -\frac{g}{v} \cos \theta,\end{aligned}$$

където фазовите променливи са координатите $x(t)$ и $y(t)$ на тялото в момента от време t , $v(t)$ е скоростта му, а $\theta(t)$ е ъгълът, който допирателната към траекторията сключва с абсцисната ос. Параметрите в модела са следните – m е масата на тялото, ρ е плътността на въздуха, c е коефициент на съпротивление, s е лицето на напречното сечение на тялото.

За начални условия вземаме $x(0) = y(0) = 0$, $v(0) = v_0$, $\theta(0) = \theta_0$, където v_0 и θ_0 са съответно началната скорост и ъгъла, под който е изстреляно тялото.

Нека пресметнем траекторията на тяло с маса 15 kg и плътност 1.29 kg/m^3 , изстреляно с начална скорост $v_0 = 50\text{ m/s}$ под ъгъл $\theta_0 = 0.6\text{ rad}$ и $\theta_0 = 1.2\text{ rad}$. Стойностите на останалите параметри в модела са $c = 0.2$, $g = 9.81\text{ m/s}^2$

Траекторията често трябва да бъде определена с висока точност. Нека например поискаме относителната грешка да бъде не по-голяма от 0.01% . Подходящ метод би бил например методът на Рунге–Кута с адаптивен избор на стъпката, тъй като при него имаме автоматична проверка на точността.

Прилагаме модификация на функцията $RK23$, която предложихме, отчитаща факта, че работим с векторно уравнение, т.е. при оценката на грешката трябва да използваме нормата на съответните вектори. Също така, използваме оценка на относителна грешка, т.е. имаме $err := \frac{\|\mathbf{y}^3 - \mathbf{y}^2\|}{\|\mathbf{y}^2\|}$, където \mathbf{y}^2 и \mathbf{y}^3 са съответно стойностите на приближеното решение при използването на методите от втори и трети ред.

```

ln[1]:= RK23[f_, u0_, t0_, T_, h0_, tol_] := (
    h = h0;
    t = t0;
    y = {0};
    y[[1]] = u0;
    i = 1;
    t = {0};
    While[y[[i]][[2]] ≥ 0,
        k1 = h f[t[[i]], y[[i]]];
        k2 = h f[t[[i]] + h, y[[i]] + k1];
        k3 = h f[t[[i]] +  $\frac{h}{2}$ , y[[i]] +  $\frac{1}{4}$  k1 +  $\frac{1}{4}$  k2];
        While[Norm[ $(\frac{1}{3} (k1 + k2 - 2 k3))$ , Infinity] / Norm[y[[i]], Infinity] > tol,
            h = h  $\left( \frac{tol}{\text{Norm}[\frac{1}{3} (k1 + k2 - 2 k3), \text{Infinity}]} \right)^{1/3}$ ;
            k1 = h f[t[[i]], y[[i]]];
            k2 = h f[t[[i]] + h, y[[i]] + k1];
            k3 = h f[t[[i]] +  $\frac{h}{2}$ , y[[i]] +  $\frac{1}{4}$  k1 +  $\frac{1}{4}$  k2];
        ];
        y = Append[y, y[[i]] +  $\frac{1}{6}$  k1 +  $\frac{1}{6}$  k2 +  $\frac{2}{3}$  k3];
        t = Append[t, t[[i]] + h];
        h = h  $\left( \frac{tol}{\text{Norm}[\frac{1}{3} (k1 + k2 - 2 k3), \text{Infinity}]} \right)^{1/3}$ ;
        i++;
    ];
    y
)

```

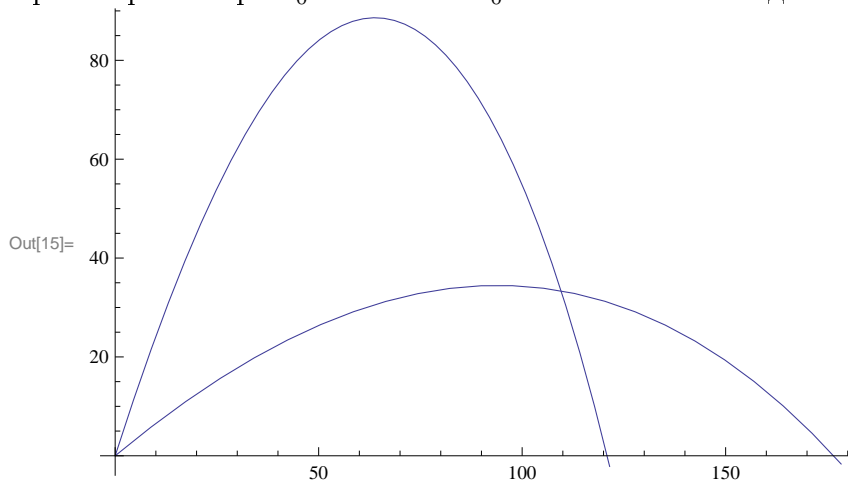
Прилагаме горната функция за решаване на поставената задача:

```

In[2]:=
m = 15;
c = 0.2; ρ = 1.29; s = 0.25; g = 9.81; v0 = 50; θ0 = 0.6; T = 0;
u0 = {0, 0, v0, θ0};
f[t_, y_] := {y[[3]] Cos[y[[4]]], y[[3]] Sin[y[[4]]],
  1/m (T - 1/2 c ρ s y[[3]]2) - g Sin[y[[4]]], -g/y[[3]] Cos[y[[4]]]}
res = RK23[f, u0, 0, 10, 0.0001, 10-2];
points = Table[{res[[i]][[1]], res[[i]][[2]]}, {i, 1, Length[res]}];
plotAppr1 = ListLinePlot[points];
θ0 = 1.2;
u0 = {0, 0, v0, θ0};
f[t_, y_] := {y[[3]] Cos[y[[4]]], y[[3]] Sin[y[[4]]],
  1/m (T - 1/2 c ρ s y[[3]]2) - g Sin[y[[4]]], -g/y[[3]] Cos[y[[4]]]}
res = RK23[f, u0, 0, 10, 0.0001, 10-2];
points = Table[{res[[i]][[1]], res[[i]][[2]]}, {i, 1, Length[res]}];
plotAppr2 = ListLinePlot[points];
Show[plotAppr1, plotAppr2, PlotRange → All]

```

Траекториите при $\theta_0 = 0.6rad$ и $\theta_0 = 1.2rad$ имат следния вид:



Решаване на твърди системи.

Нека разгледаме задачата на Коши

$$\frac{du}{dt} = \lambda(-u + \sin t), \quad 0 < t \leq 10,$$

$$u(0) = 0,$$

имаща точно решение

$$u(t) = \frac{\lambda}{1 + \lambda^2} e^{-\lambda t} + \frac{\lambda^2}{1 + \lambda^2} \sin t - \frac{\lambda}{1 + \lambda^2} \cos t.$$

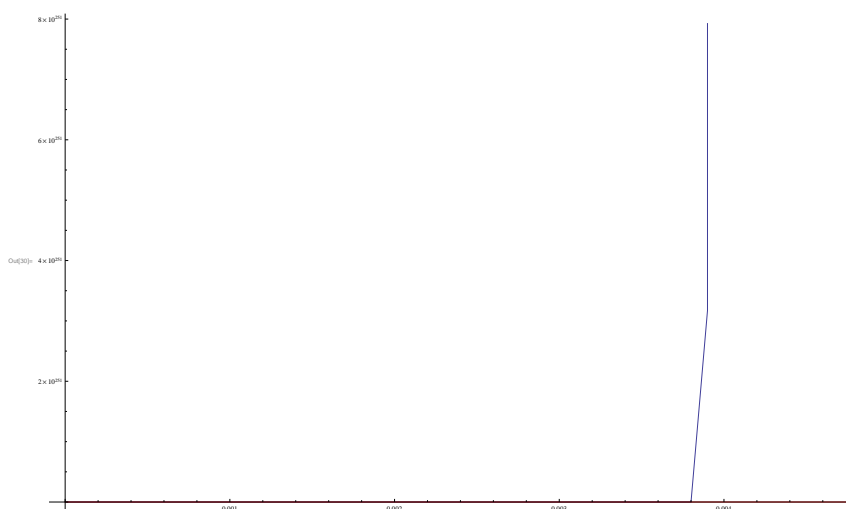
Да решим задачата при $\lambda = 1000000$. Първата ни идея може да бъде да използваме явен метод, поради относителната простота и бързодействие спрямо неявните. Нека приложим например метода на Рунге-Кута от четвърти ред.

```

In[24]:= λ = 1 000 000;
f[t_, y_] := λ (-y + Sin[t]);
y = RK4[f, 0, 0.0001, 0, 10];
plotAppr = ListLinePlot[Table[{t[[i + 1]], y[[i + 1]]}, {i, 0, n}]];
c =  $\frac{\lambda}{1 + \lambda^2}$ ;
plotExact = Plot[ $c E^{-\lambda t} + \frac{\lambda^2}{1 + \lambda^2} \text{Sin}[t] - \frac{\lambda}{1 + \lambda^2} \text{Cos}[t]$ , {t, 0, 10}, PlotStyle → Red];
Show[plotAppr, plotExact]

```

Да отбележим, че сме използвали доста малка стъпка -10^{-4} . Резултатът от изпълнението на горния код е следният:



Ясно е, че причината за получения резултат е условната устойчивост на метода. За конкретната задача условието за устойчивост означава, че стъпката трябва да е твърде малка, за да бъде методът устойчив, което го прави на практика неприложим.

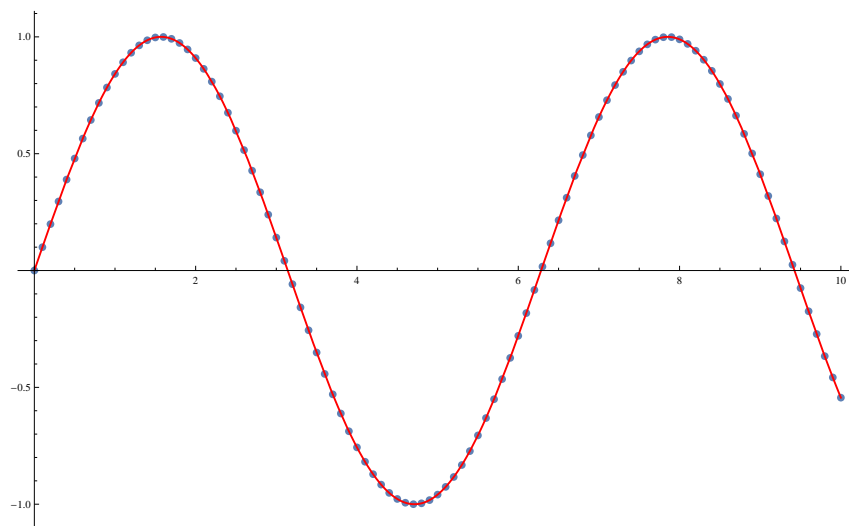
Това е пример за т.нар. **твърди задачи**. Естествен подход тогава е да опитаме да решим задачата с неявен метод. Ще изпробваме неявния метод на Ойлер за целта.

```

λ = 1 000 000;
f[t_, y_] := λ (-y + Sin[t]);
y = implicitEuler[f, 0, 0.1, 0, 10];
plotAppr = ListPlot[Table[{t[[i + 1]], y[[i + 1]]}, {i, 0, n}],
  PlotRange → All];
c =  $\frac{\lambda}{1 + \lambda^2}$ ;
plotExact = Plot[ $c E^{-\lambda t} + \frac{\lambda^2}{1 + \lambda^2} \text{Sin}[t] - \frac{\lambda}{1 + \lambda^2} \text{Cos}[t]$ , {t, 0, 10}, PlotStyle → Red];
Show[plotAppr, plotExact, PlotRange → All]

```

Резултатът (при използвана стъпка $h = 0.1$) е следният:



Виждаме, че дори неявният метод на Ойлер, който има първи ред на сходимост, се справя добре с решаването на задачата.

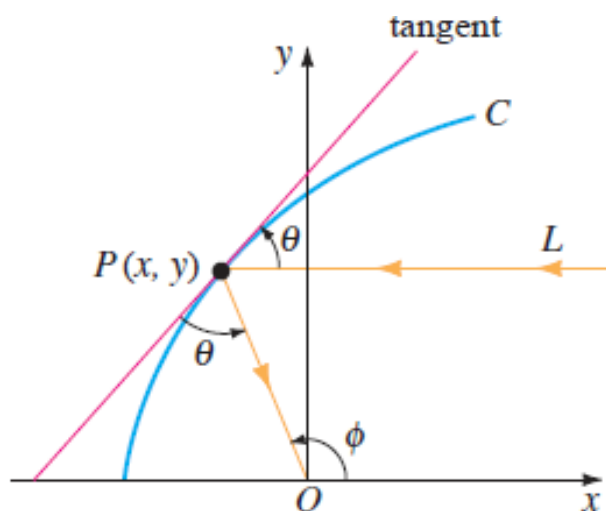
2.8 Допълнителни задачи

Задача 1. Диференциалното уравнение

$$\frac{dy}{dx} = \frac{-x + \sqrt{x^2 + 4y^2}}{2y}$$

описва формата на равнинна крива, която отразява всички лъчи, успоредни на абсцисата, в една и съща точка (началото на координатната система). Такива модели намират приложение при огледалата в телескопите, сателитните антени, соларните колектори и др.

- Диференциалното уравнение може да се изведе на базата на геометрични съображения, като се вземе предвид, че на долната фигура $\phi = 2\theta$ (Защо?). Изведете го.



- Съставете явна числена схема с втори ред на сходимост, която решава даденото ОДУ по подадено начално условие y_0 и интервал на интегриране.



Напишете функция в Mathematica, която реализира предложената числена схема. Използвайте я, за да визуализирате примерна форма на сателитна антена, отговаряща на даденото ОДУ.

Задача 2. Разглеждаме числения метод с тегло θ

$$y_{i+1} = y_i + h\theta f_{i+1} + h(1 - \theta)f_i.$$

- За кои стойности на θ методът е неявен?
- Изведете условие за A-устойчивост за произволно θ .
- За кои стойности на θ методът е A-устойчив (за всяко $h > 0$)?
- За кои стойности на θ методът е монотонен (за всяко $h > 0$)?
- Приложете метода за задачата на Коши:

$$\begin{aligned} \frac{du}{dt} &= -10u + 5, \quad t \in (0, 1], \\ u(0) &= 4. \end{aligned}$$

За целта изведете връзка между y_{i+1} и y_i . Тъй като зададената задача е линейна, в случая може да се намери явен вид на y_{i+1} . Направете го.

Задача 3. Разглеждаме уравнението

$$u'(t) = -300t^2y^3, \quad 0 \leq t \leq 3$$

с начално условие $u(0) = 1$.

- Покажете, че точното решение на горната задача на Коши е

$$u(t) = \frac{1}{\sqrt{200t^3 + 1}}.$$



Приложете метода с адаптивен избор на стъпката RK23, разглеждан на упражнения, с толеранс 0.0001.



Намерете допуснатата грешка във всяка точка и сравнете с толеранса.



Покажете как се е изменяла стъпката при решаването на задачата. Къде стъпката е малка? Защо?



Каква равномерна стъпка очаквате да е необходима за метод на Рунге-Кута от трети ред, за да бъде постигната същата точност?



Приложете метод на Рунге-Кута от трети ред с равномерна мрежа, така че да получите същата точност. Сравнете времената за изпълнение на програмата при равномерна и адаптивна стъпка, като използвате вградената функция TimeUsed



Направете същото, като в предишната подточка, за явния и неявния метод на Ойлер и за методи на Рунге-Кута от четвърти и пети ред (за съответните таблици на Butcher вижте в интернет).



Потърсете информация в интернет за метода на Runge-Kutta-Fehlberg (RK45), който е най-често използваният на практика метод с адаптивен избор на стъпката. Решете задачата, като го използвате. Сравнете времето за изпълнение на програмата с това при RK23.

Задача 4. Даден е следният метод на Рунге-Кута от втори ред:

$$\begin{array}{c|cc} 0 & & \\ 2/3 & 2/3 & \\ \hline & 1/4 & 3/4 \end{array}$$

- Проверете, че това действително е метод на Рунге-Кута, като пресметнете ЛГА.
- Изследвайте метода за А-устойчивост и монотонност.
- Намерете y_1 – стойността на приближеното решение в точката t_1 при дискретизация с равномерна мрежа със стъпка $h = 0.1$ за задачата

$$\begin{aligned} \frac{du}{dt} &= -10u + 5, \quad t \in (0, 1], \\ u(0) &= 4. \end{aligned}$$

- Каква стъпка бихте избрали, за да осигурите устойчивост и монотонност на метода?

Задача 5. На първото занятие от практикума беше разгледана следната диференциална задача, моделираща движението на система от пружина и маса, като съпротивлението на въздуха се пренебрегне:

$$\begin{aligned} \frac{d^2x}{dt^2} &= g - \frac{kx}{m}, \\ x(0) &= x_0, \quad x'(0) = 0, \end{aligned}$$

където g е земното ускорение, а k е еластичната константа на пружината.

Ако вземем предвид и съпротивлението на въздуха, моделът добива вида

$$\begin{aligned} \frac{d^2x}{dt^2} &= g - \frac{kx}{m} - \frac{\mu}{m} \frac{dx}{dt}, \\ x(0) &= x_0, \quad x'(0) = 0, \end{aligned}$$

където μ е коефициент на триене.

За така получения модел:

- Сведете задачата до задача на Коши за система ОДУ от първи ред.
- Изведете метод на Адамс-Башфорт от четвърти ред.

- Изведете метод на Адамс-Мултон от четвърти ред.
- Съставете алгоритми, използващ изведените методи, за численото решаване на задачата.
- Съставете предикторно-коректорен метод от четвърти ред.



Имплементирайте всеки от методите. Направете числени експерименти при различни стойности на параметрите (вкл. $\mu = 0$). Направете анимация на движението на махалото във всеки от случаите. Какъв е ефектът на съпротивлението на въздуха? Какъв е ефектът на масата на тялото? Какъв е ефектът на еластичността на пружината? Изглеждат ли ви резултатите физически смислени? Обосновайте се.



Сравнете бързодействието на реализираните от вас методи с това на RK4.

Задача 6. За редица биотехнологични процеси се използват биореактори. Биореакторът представлява съд, в който се отглеждат микроорганизми с цел производството на биогаз, антибиотици, алкохолни напитки и др.



Ако в биореактора се влива хранителен разтвор със скорост D , в който концентрацията на хранителното вещество е скалирана на 1, и от биореактора се извежда разтвор със същата скорост, процесът може да се опише със следната система ОДУ (Модел на Моно):

$$\begin{aligned}\frac{ds}{dt} &= (1 - s)D - \frac{as}{b + s}x, \\ \frac{dx}{dt} &= \left(\frac{as}{b + s} - D \right)x.\end{aligned}$$

В системата $s(t)$ и $x(t)$ са концентрациите на субстрата (веществото, с което се хранят микроорганизмите) и на микроорганизмите в момента от време t . Параметрите a и b са положителни константи, които описват скоростта, с която конкретните микроорганизми консумират субстрата.

- Съставете алгоритъм, базиран на неявния метод на Ойлер, за решаване на задачата.



Имплементирайте метода и решете задачата за различни стойности на параметъра D . За всяка избрана от вас стойност на D , визуализирайте параметричните криви $(s(t), x(t))$, съответстващи на няколко начални условия (последното се нарича фазов портрет). Какви изводи можете да направите от числените експерименти за възможното дългосрочно поведение на биотехнологичния процес? Каква скорост на отмиване е допустима, за да оцелее микробналната популация?

Глава 3

Диференчни методи за решаване на ЧДУ

3.1 Въведение в задачите, които ЧДУ описват. Уравнение на непрекъснатостта.

Досега разглеждахме ОДУ, т.е. търсените функции зависеха само от една независима променлива. Най-често обаче изучаваните процеси зависят както от времето, така и от пространството и затова е естествено техните математически модели да бъдат частни диференциални уравнения (ЧДУ).

N.B. 22

В зависимост от това дали участва времевата променлива, или не, говорим за нестационарни и стационарни задачи.

- Стационарните задачи (в които не участва времето t) биват 2-мерни (по пространството) (неизвестната функция е $u(x, y)$) и 3-мерни $u(x, y, z)$. Те описват равновесното състояние на даден процес.
- Нестационарните задачи описват времевата еволюция на даден процес. Те могат да бъдат 1-, 2- или 3-мерни. Съответно неизвестната функция е $u(x, t)$, $u(x, y, t)$, $u(x, y, z, t)$.

Голяма част от математическите модели на реални процеси са отражение на някакъв закон за запазване.

Ще изведем уравнението, отразяващо даден закон за запазване в най-общия вид, като за определеност ще мислим за концентрацията на дадено вещество, но с аналогични разсъждения можем да изведем същото уравнение, интерпретирано от гледна точка разпространението на топлина и др.

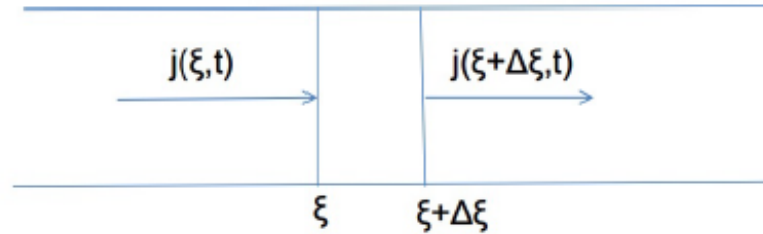
За простота ще изведем уравнението в 1D, но с аналогични разсъждения то може да се обобщи и за повече пространствени измерения.

И така, разглеждаме дълга и тънка тръба, в която приемаме, че разпределението на вещество в едно напречно сечение е хомогенно, т.е. концентрацията на веществото (нека я бележим с u) зависи само от една пространствена променлива x , т.е. $u = u(x, t)$.

Нека с $j(x, t)$ означим потока на веществото в точката x в момента от време

t (т.е. това е количеството вещество, минаващо в точката за единица време). Нека приемем, че тръбата е добре уплътнена, т.е. през околната ѝ повърхнина не преминава вещество.

Нека разгледаме един достатъчно малък интервал $[\xi, \xi + \Delta\xi]$:



Изменението на концентрацията в дадения интервал е равно на разликата на входящия и изходящия поток, т.е.

$$\text{изменението на концентрацията} = j(\xi, t) - j(\xi + \Delta\xi, t).$$

Изменението за единица време в дадена точка от интервала е $\frac{\partial u}{\partial t}$. За да получим общото изменение в интервала, трябва да сумираме измененията във всяка точка и получаваме

$$\int_{\xi}^{\xi + \Delta\xi} \frac{\partial u}{\partial t} dx = j(\xi, t) - j(\xi + \Delta\xi, t).$$

Това е **интегралната форма на закона за запазване**. Обикновено, тя не се използва в този вид, а се свежда до диференциална. Един възможен подход е следният.

Пресмятаме интеграла в лявата страна по формулата на централните правоъгълници и получаваме

$$\frac{\partial u}{\partial t} \left(\xi + \frac{\Delta\xi}{2}, t \right) \Delta\xi + O(\Delta\xi^3) = j(\xi, t) - j(\xi + \Delta\xi, t).$$

Делим двете страни на $\Delta\xi$ и получаваме

$$\frac{\partial u}{\partial t} \left(\xi + \frac{\Delta\xi}{2}, t \right) + O(\Delta\xi^2) = \frac{j(\xi, t) - j(\xi + \Delta\xi, t)}{\Delta\xi}.$$

И сега, пускайки $\Delta\xi$ да клони към 0, получаваме **диференциалната форма на закона за запазване**:

$$\frac{\partial u}{\partial t} = -\frac{\partial j}{\partial x}(\xi, t).$$

По-общо, ние ще разглеждаме **уравнението на непрекъснатостта**

$$\frac{\partial u}{\partial t} = -\frac{\partial j}{\partial x} + f, \tag{3.1}$$

където f е функция, която може да описва източници/поглъщатели (на топлина, вещество и т.н.), реакции, вследствие на които се увеличава или намалява концентрацията (температурата, енергията и т.н.) и др.

Това е общият вид на много от моделите на реални процеси. В зависимост от причината за потока (дифузия, конвекция и др.) и видът на f се получават конкретни модели, които описват различни неща и могат да имат различна сложност, от гледна точка на численото им решаване.

N.B. 23

Уравнението на непрекъснатостта, което отразява даден закон за запазване, задава общия вид на много от математическите модели, срещани в практиката, тъй като много от тях се основават именно на закони за запазване.

N.B. 24

За разлика от ОДУ, за ЧДУ няма как да се построи “обща” числена схема, която (поне формално) да може да се приложи за всички уравнения. При ЧДУ за всяка задача (клас задачи) е необходимо да се подхожда конкретно, като се изхожда от общите принципи. Ние ще ги изложим върху трите основни класа ЧДУ:

- параболични (ще разгледаме линейното 1D уравнение на топлопроводността);
- хиперболични (ще разгледаме линейните 1D уравнения на преноса/адвекцията и струната);
- елиптични (ще разгледаме 2D уравнението на Поасон).

3.2 Диференчни методи за параболични ЧДУ. Линейно 1D нестационарно уравнение на дифузията/топлопроводността.

Дифузията е единият основен процес, който обуславя движението в природата. При него веществата се разпространяват от места с по-голяма концентрация (температура, енергия,...) към места с по-малка концентрация (температура, енергия,...). Ще изведем уравнението на дифузията (топлопроводността) в едномерния случай (т.е. когато процесът зависи само от една пространствена променлива).

Експериментално установено е, че дифузионният поток е пропорционален на разликата в концентрациите (по-точно на градиента), т.е.

$$j_{diff}(x, t) = -D \frac{\partial u}{\partial x}. \quad (3.2)$$

Последното уравнение носи името **Закон на Фик** (в топлопроводността – **Закон на Фурие**). Замествайки го в диференциалната форма на Закона за запазване (уравнението на непрекъснатостта) (3.1), получаваме окончателно уравнението на дифузията (още, уравнение на топлопроводността)

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2} + f.$$

В последното сме направили предположението, че дифузионният коефициент $D > 0$ е константа. Това означава, че материалът, в който се случва дифузията, е хомогенен, т.е. неговата проводимост е една и съща във всяка точка и не се изменя във времето.

Още веднъж ще подчертаем, че това уравнение описва много различни процеси – разпространение на топлина, дифузия на вещество, разпространение на заразни заболявания и др. Това е едно от големите достойнства на математическите модели – те са абстрактни задачи, които могат да се интерпретират по много различни начини – един и същ математически обект може да описва различни реални процеси.

3.2.1 Явна двуслойна диференчна схема при гранични условия на Дирихле

И така, ще изложим основните идеи на диференчните методи за решаване на ЧДУ върху линейното едномерно нестационарно уравнение на топлопроводността

$$\frac{\partial u}{\partial t} - D \frac{\partial^2 u}{\partial x^2} = f. \quad (3.3)$$

Имайки предвид, че в уравнението присъстват производна от втори ред по пространството и от първи ред по времето, е ясно, че за да затворим диференциалната задача, е необходимо да наложим две (гранични) условия по пространството и едно (начално) по времето:

$$u(x, 0) = u_0(x), \quad (3.4)$$

$$u(0, t) = u_L(t), \quad u(l, t) = u_R(t). \quad (3.5)$$

Наложените гранични условия се наричат **гранични условия от първи род (гранични условия на Дирихле)**. По-нататък ще разгледаме и по-обща гранични условия.

За самото построяване на диференчната схема подходим по подобен начин на това, което направихме при методите за ОДУ от първи ред. Ще дискретизираме диференциалната задача, като въведем мрежа от възли, в които ще търсим стойностите на приближеното решение, след което ще апроксимираме производните чрез известните ни формули за числено диференциране.

Областта, в която разглеждаме задачата, е правоъгълникът

$$\mathfrak{R} := \{(x, t) : 0 \leq x \leq l, 0 \leq t \leq T\}.$$

Простата геометрия на дефиниционната област на $u(x, t)$ (която е напълно естествена за разглежданата едномерна по пространството нестационарна задача) ни позволява да въведем **равномерна мрежа**¹ със стъпки по пространс-

¹Да отбележим, че е възможно използването на неравномерни и адаптивни мрежи, които да бъдат подходящи (в някакъв смисъл) за конкретната решавана задача, но това излиза съществено извън рамките на настоящия уводен курс.

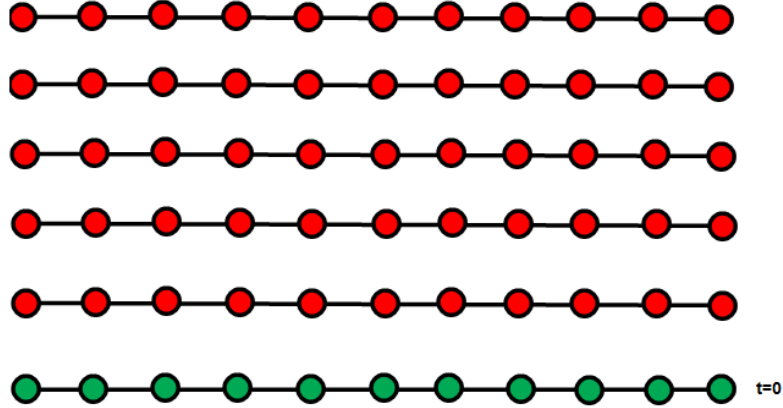
твото и по времето, съответно h и τ , по следния начин:

$$\bar{\omega}_{h,\tau} := \{(x_i, t_j) : x_i = ih, t_j = j\tau, i = \overline{0, n}, j = \overline{0, m}, n = l/h, m = T/\tau\}.$$

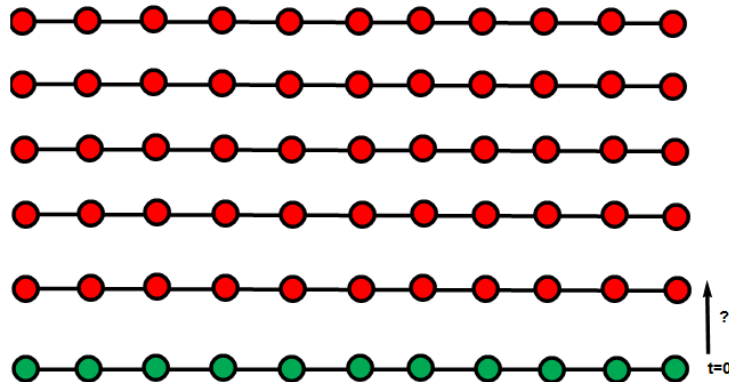
Стойността на приближеното решение в точката (x_i, t_j) ще бележим с y_i^j . Множеството от възли, за които $t = t_j$ ще наричаме j -ти **слой по времето**.

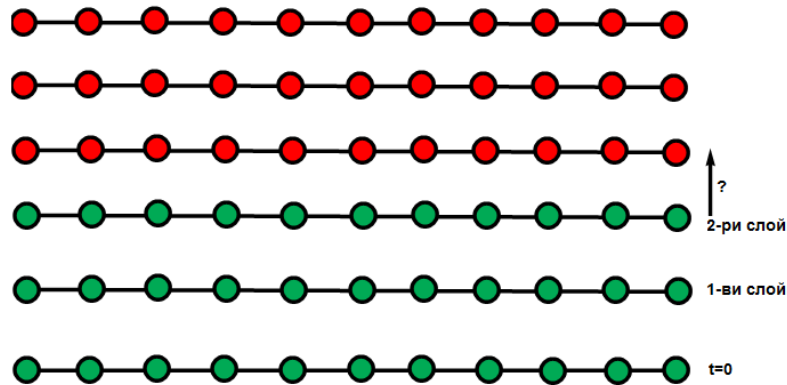
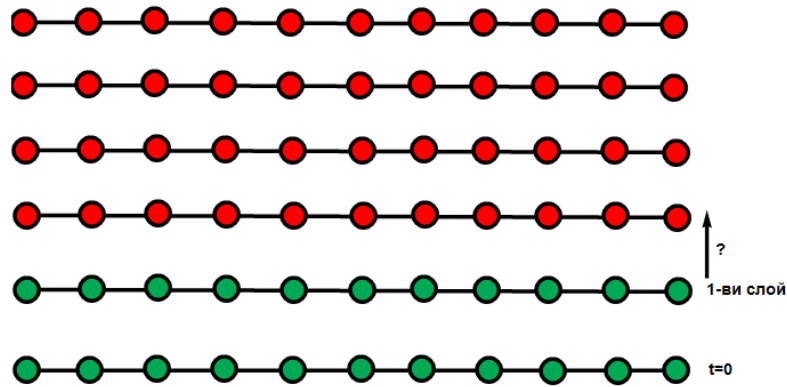
От началното условие ни е известна стойността на решението на 0-вия слой по времето

$$y_i^0 = u_0(x_i), \quad i = \overline{0, n}. \quad (3.6)$$



Тогава, ако на базата на информацията за 0-вия слой, можем да пресметнем стойностите на приближеното решение на 1-вия, оттам на 2-рия и т.н., или изобщо, ако, знаейки стойностите на приближеното решение на j -тия слой, можем да намерим тези на $j + 1$ -вия ($j = 0, \dots, m - 1$), ще можем да намерим всички търсени стойности.

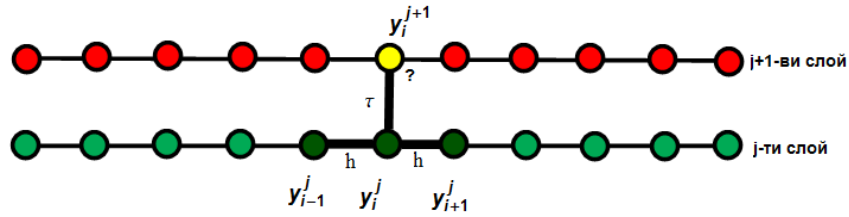




И така, нека приемем, че са ни известни стойностите на j -тия слой по времето. Ще търсим тези на $j+1$ -вия. Връзката между стойностите на приближеното решение, разбира се, ще получим като апроксимираме диференциалното уравнение (3.3) в точката (x_i, t_j) . Апроксимирайки производната по времето с формулата с разлика напред и производната по пространството с формула с централна разлика от втори ред, получаваме диференчното уравнение с ЛГА $O(h^2 + \tau)$

$$\frac{y_i^{j+1} - y_i^j}{\tau} - D \frac{y_{i-1}^j - 2y_i^j + y_{i+1}^j}{h^2} = f_i^j, \quad i = \overline{1, n-1}, \quad j = \overline{0, m-1}. \quad (3.7)$$

В него участват стойностите на приближеното решение в точките с индекси $(i-1, j)$, (i, j) , $(i+1, j)$ и $(i, j+1)$, като само последната стойност е неизвестна. Това означава, че **получената схема е явна**, тъй като от горното уравнение можем директно да намерим стойността y_i^{j+1} .



Тези четири точки образуват т.нар. **шаблон** за точката (x_i, t_j) , в която апроксимираме оригиналната диференциална задача.

Шаблонът можем да поставим така, че да намерим всички вътрешни точки на $j + 1$ -вия слой по времето.

За граничните точки ще използваме граничните условия:

$$y_0^{j+1} = u_L(t_{j+1}), \quad y_n^{j+1} = u_R(t_{j+1}), \quad j = \overline{0, m-1}. \quad (3.8)$$

И така, уравненията (3.6), (3.7), (3.8) апроксимират оригиналната диференциална задача с диференчна такава. Решавайки уравнение (3.7) по отношение на единственото неизвестно y_i^{j+1} , получаваме т.нар. **каноничен вид на диференчната схема**:

$$\begin{aligned} y_i^{j+1} &= \left(1 - 2\frac{D\tau}{h^2}\right) y_i^j + \frac{D\tau}{h^2}(y_{i-1}^j + y_{i+1}^j) + \tau f_i^j, \quad i = \overline{1, n-1}, \quad j = \overline{0, m-1}, \\ y_i^0 &= u_0(x_i), \quad i = \overline{0, n}, \\ y_0^{j+1} &= u_L(t_{j+1}), \quad y_n^{j+1} = u_R(t_{j+1}), \quad j = \overline{0, m-1}. \end{aligned} \quad (3.9)$$

N.B. 25

ЛГА на схемата (3.9) е $O(h^2 + \tau)$.

Алгоритъм 7: Реализиране на явни двуслойни схеми

Имплементирането на една явна двуслойна диференчна трябва да следва следния алгоритъм:

1. Използвайки началното условие, намираме стойностите на първия слой по времето.
2. Итериране по $j = \overline{0, m-1}$. На всяка итерация, от стойностите на j -тия слой намираме тези на $j + 1$ -вия.
 - От апроксимацията на основното диференциално уравнение намираме стойностите във вътрешните точки от мрежата.
 - От граничните условия намираме стойностите на приближеното решение в граничните точки.

За да можем да използваме на практика разглежданите схеми, е необходимо да знаем нещо за тяхната устойчивост и монотонност. В конкретния случай изследването за устойчивост е свързано с т.нар. условие за положителност на

коэффициентите. Засега ще дадем формулировка, която не е пълна, но е достатъчна към настоящия момент, а в последваща лекция ще разгледаме въпроса много по-точно и по-подробно.

Н.В. 26

Ако коефициентите пред “ y ”-ците в каноничния вид (3.9) са положителни числа и, при това, разликата между коефициента в лявата и тези в дясната страна е неотрицателна, то методът е монотонен.

И така, ясно е, че за да бъде диференчната схема (3.9) монотонна (и устойчива), е достатъчно да поискаме

$$\tau < \frac{h^2}{2D}. \quad (3.10)$$

Пример 25.

3.2.2 Апроксимиране на гранични условия на Нойман и Робин

Освен гранични условия на Дирихле е естествено да се разглеждат и условия за потока. Например, ако лявата граница е непропусклива, можем да наложим условие за нулев поток:

$$j(0) = 0.$$

Вземайки предвид (3.2), за уравнението на дифузията последното е еквивалентно на

$$\frac{\partial u}{\partial x}(0, t) = 0.$$

Гранични условия за производната на неизвестната функция се наричат гранични условия на Нойман (гранични условия от втори род).

Дефиниция 9: Гранични условия на Дирихле, Нойман и Робин

- Гранични условия от вида

$$u(0, t) = u_L(t),$$

се наричат **гранични условия от първи род (гранични условия на Дирихле)**.

- Гранични условия от вида

$$\frac{\partial u}{\partial x}(0, t) = u_L(t)$$

се наричат **гранични условия от втори род (гранични условия на Нойман)**.

- Гранични условия от вида

$$\frac{\partial u}{\partial x}(0, t) + \alpha u(0, t) = u_L(t)$$

се наричат **гранични условия от трети род (смесени гранични условия, гранични условия на Робин)**.

Разбира се, аналогични условия могат да се наложат и на дясната граница.

От гледна точка на съставянето на числената схема е важно, че в условията на Нойман и Робин стойността на търсената функция на границата не е известна, а в условието участва първата производна. Независимо дали условието е на Нойман, или на Робин, се подхожда по аналогичен начин и затова ще илюстрираме идеята върху задача с условие на Нойман. Ще го наложим на лявата граница.

И така, разглеждаме диференциалната задача

$$\begin{aligned} \frac{\partial u}{\partial t} - D \frac{\partial^2 u}{\partial x^2} &= f, \quad 0 < x < l, \quad 0 < t \leq T, \\ u(x, 0) &= u_0(x), \quad 0 \leq x \leq l, \\ \frac{\partial u}{\partial x}(0, t) &= u_L(t), \quad u(l, t) = u_R(t), \quad 0 \leq t \leq T. \end{aligned} \tag{3.11}$$

Единствената разлика с диференциалната схема (3.9) ще бъде лявото гранично условие. Бихме искали да използваме за него формулата за числено диференциране с централна разлика, тъй като тя има втори ред на точност. Това обаче, разбира се, е невъзможно, тъй като нямаме точка от мрежата вляво от границата. За да го апроксимираме, бихме могли да използваме единствено формулата за числено диференциране с разлика напред. Получаваме диференциалното уравнение

$$\frac{y_1^j - y_0^j}{h} = u_L(t_j),$$

което има ЛГА $O(h)$. Ако в граничните точки обаче въвеждаме грешка, която

е от по-нисък ред, то е безполезен фактът, че в останалите точки грешката ни е от втори ред. **Сега ще видим как можем да повишим реда на апроксимация за граничното условие на Нойман.** Бихме могли да запишем следното равенство

$$\frac{u_1^j - u_0^j}{h} = u_L(t_j) + O(h).$$

Бихме могли обаче да развием остатъчния член и да получим за него вида $\diamond + O(h^2)$ (с \diamond сме означили члена от първи ред, който трябва да определим). Тогава последното уравнение добива вида

$$\frac{u_1^j - u_0^j}{h} = u_L(t_j) + \diamond + O(h^2)$$

или

$$\frac{u_1^j - u_0^j}{h} - \diamond = u_L(t_j) + O(h^2)$$

и следователно диференчното уравнение

$$\frac{y_1^j - y_0^j}{h} - \diamond = u_L(t_j) \quad (3.12)$$

ще апроксимира лявото гранично условие с ЛГА $O(h^2)$.

Остатъчният член $O(h)$, който искаме да представим във вида $\diamond + O(h^2)$, е всъщност ЛГА. Нека я разпишем, като развием в ред на Тейлър около $x = 0$:

$$\begin{aligned} \psi_{h,\tau} &= \frac{1}{h} \left(u_0^j + \left. \frac{\partial u}{\partial x} \right|_0^j h + \left. \frac{\partial^2 u}{\partial x^2} \right|_0^j \frac{h^2}{2} + O(h^3) - u_0^j \right) - u_L(t_j) \\ &= \frac{h}{2} \left. \frac{\partial^2 u}{\partial x^2} \right|_0^j + O(h^2). \end{aligned}$$

В последното използвахме, че $\left. \frac{\partial u}{\partial x} \right|_0 = u_L(t)$. И така, получихме представяне на ЛГА в желания вид. Тогава, ако заместим \diamond с $\frac{h}{2} \left. \frac{\partial^2 u}{\partial x^2} \right|_0^j$ (по-точно, апроксимация на последното) в лявата страна на (3.12), ще получим апроксимация с втори ред на точност. Проблемът обаче е, че по абсолютно същата причина, поради която не можахме да използваме формулата с централна разлика за апроксимация на производната от първи ред, не можем да апроксимираме и втората производна. **За да решим този проблем правим едно съществено допускане, а именно – че основното диференциално уравнение (3.11) е изпълнено и при $x = 0$.** Това допускане е обосновано, тъй като физическите процеси имат достатъчно гладко поведение и следователно фактът, че (3.11) е изпълнено върху отворения интервал $x \in (0, 1)$, ни дава основание да приемем, че в граничните точки също е изпълнено с достатъчно малка грешка.

Тогава можем да изразим втората производна по пространството чрез производната по времето. Получаваме

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{D} \left(\frac{\partial u}{\partial t} - f \right).$$

Тогава, като използваме последното, заместваем \diamond в (3.12) и получаваме окончателно апроксимация на граничното условие с грешка $O(h^2 + \tau)$:

$$\frac{y_1^j - y_0^j}{h} - \frac{h}{2D} \left(\frac{y_0^{j+1} - y_0^j}{\tau} - f_0^j \right) = 0.$$

Записано в каноничен вид, диференчното уравнение добива вида

$$y_0^{j+1} = \left(1 - \frac{2\tau D}{h^2}\right) y_0^j + \frac{2D\tau}{h^2} y_1^j + \tau f_0^j, \quad j = \overline{0, m-1}. \quad (3.13)$$

И така, за приближеното решаване на задачата (3.11) получихме явна диференчна схема, като в (3.9) заменихме апроксимацията на лявото гранично условие с (3.13).

Н.В. 27

За да повишим реда на апроксимация, която е от първи ред, на гранично условие на Нойман или Робин, можем да извадим от лявата ѝ страна апроксимация на члена от първи ред в ЛГА.

3.2.3 Неявни схеми. Чисто неявна схема. Схема на Кранк–Никълсън.

Чисто неявна схема

Условието за устойчивост (3.10) при явната схема означава, че стъпката по времето, τ , трябва да е от порядъка на h^2 . Това може да доведе до необходимостта от работата с много малка стъпка по времето.

Опитът ни от решаването на ОДУ подсказва, че този проблем би могъл да се реши, ако използваме неявни методи за решаването на задачата. Ще изложим идеята отново върху задачата (3.3)–(3.5). За удобство, нека тук отново приложим основното диференциално уравнение:

$$\frac{\partial u}{\partial t} - D \frac{\partial^2 u}{\partial x^2} = f, \quad 0 < x < l, \quad 0 < t \leq T \quad (3.14)$$

При извеждането на явната схема, апроксимирахме основното уравнение (3.14) на j -тия слой по времето (вж. (3.7)). Получихме

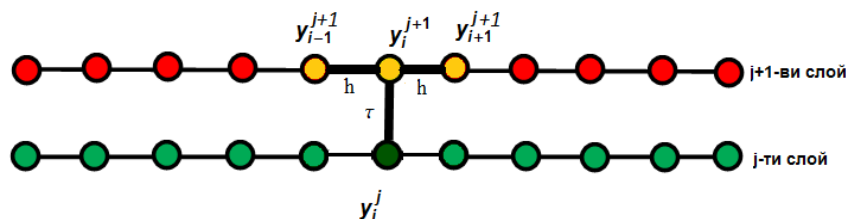
$$\frac{y_i^{j+1} - y_i^j}{\tau} - D \frac{y_{i-1}^j - 2y_i^j + y_{i+1}^j}{h^2} = f_i^j, \quad i = \overline{1, n-1}, \quad j = \overline{0, m-1}, \quad (3.15)$$

където единственото неизвестно е y_i^{j+1} .

Можем обаче да подходим по аналогичен начин, като направим апроксимацията на $j+1$ -вия слой, като за производната по времето използваме формулата с разлика назад. Получаваме

$$\frac{y_i^{j+1} - y_i^j}{\tau} - D \frac{y_{i-1}^{j+1} - 2y_i^{j+1} + y_{i+1}^{j+1}}{h^2} = f_i^{j+1}, \quad i = \overline{1, n-1}, \quad j = \overline{0, m-1}. \quad (3.16)$$

Тук шаблонът изглежда по следния начин:



Очевидно получената апроксимация е неявна, тъй като в уравнението участват 3 неизвестни – $y_{i-1}^{j+1}, y_i^{j+1}, y_{i+1}^{j+1}$. За да намерим стойностите на приближеното решение на $j + 1$ -вия слой по времето, трябва да решаваме система от $n + 1$ уравнения и $n + 1$ неизвестни (включвайки и уравненията $y_0^{j+1} = u_L(t_{j+1})$ и $y_n^{j+1} = u_R(t_{j+1})$, получени от граничните условия).

Записана във векторно-матрична форма, линейната алгебрична система има вида

$$\begin{bmatrix} 1 & 0 & & & \\ -\frac{D}{h^2} & \frac{1}{\tau} + \frac{2D}{h^2} & -\frac{D}{h^2} & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{D}{2h^2} & \frac{1}{\tau} + \frac{D}{h^2} & -\frac{D}{2h^2} \\ & & & 0 & 1 \end{bmatrix} \begin{bmatrix} y_0^{j+1} \\ y_1^{j+1} \\ \vdots \\ y_{n-1}^{j+1} \\ y_n^{j+1} \end{bmatrix} = \begin{bmatrix} u_L(t_{j+1}) \\ \frac{y_1^j}{\tau} + f_1^{j+1} \\ \vdots \\ \frac{y_{n-1}^j}{\tau} + f_{n-1}^{j+1} \\ u_R(t_{j+1}) \end{bmatrix}. \quad (3.17)$$

Забележка. Ако граничните условия са други, в последната система трябва да се променят по подходящ начин първото и последното уравнение.

Матрицата на (3.17) има ненулеви елементи само по главния диагонал и диагоналите под и над него. Такива матрици се наричат тридиагонални. Съществуват специални методи за решаването на системи с тридиагонална матрица, които са значително по-бързи от класическите методи за решаване на линейни системи, приложими в общия случай. Класическият метод за целта е методът на прогонката, който има сложност $O(n)$ операции, където n е размерността на системата. За сравнение методът на Гаус има сложност $O(n^3)$ операции. За метода на прогонката можете да прочетете в §3 на стр. 63–66 в [3].

Алгоритъм 8: Реализиране на неявни двуслойни схеми

Имплементирането на една неявна диференчна схема от горния вид трябва да следва следния алгоритъм:

1. Използвайки началното условие, намираме стойностите на първия слой по времето.
2. Итерираме по $j = \overline{0, m-1}$. На всяка итерация от стойностите на j -тия слой намираме тези на $j + 1$ -вия, като решаваме системата (3.17).

Разбира се, фактът, че на всеки слой по времето трябва да решаваме алгебрична система означава, че реализирането на една неявна схема е по-сложно и при това, ако стъпките при явната схема и при неявната са еднакви, неявната ще е по-бавна. Оказва се обаче, че чисто неявната схема е монотонна за всеки избор на τ , т.е. при твърди задачи неявните схеми на теория ще имат предимство.

Действително, нека запишем (3.16) в каноничен вид. Получаваме

$$\left(1 + \frac{2\tau D}{h^2}\right) y_i^{j+1} = \frac{\tau D}{h^2} (y_{i-1}^{j+1} + y_{i+1}^{j+1}) + y_i^j + \tau f_i^{j+1}.$$

Ясно е, че последното изпълнява изискванията за положителност на коефициентите винаги и следователно **чисто неявната схема е монотонна за всяко τ** .

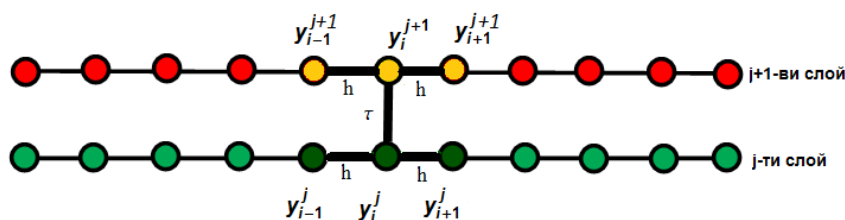
Схема с тегло. Схема на Кранк–Никълсън.

От (3.15) и (3.16) можем да получим клас от схеми с тегло. Умножавайки двете страни на (3.15) със σ , а тези на (3.7) с $1 - \sigma$ и събирайки почленно, получаваме

$$\frac{y_i^{j+1} - y_i^j}{\tau} - \sigma D \frac{y_{i-1}^j - 2y_i^j + y_{i+1}^j}{h^2} - (1 - \sigma) D \frac{y_{i-1}^{j+1} - 2y_i^{j+1} + y_{i+1}^{j+1}}{h^2} = \sigma f_i^j + (1 - \sigma) f_i^{j+1},$$

$$i = \overline{1, n-1}, \quad j = \overline{0, m-1}. \quad (3.18)$$

В общия случай (при $\sigma \neq 0, \sigma \neq 1$) шаблонът е 6-точков:



Може да се покаже (вж. [3]), че най-добрият избор на σ , от гледна точка на точност на метода, е $\sigma = 1/2$. Получената схема се нарича схема на Кранк–Никълсън и има ЛГА $O(h^2 + \tau^2)$.

За да покажем последното, нека мислим, че сме апроксимирали диференциалното уравнение в междинната точка $(x_i, t_j + \tau/2)$. Съответните стойности ще означаваме с индекси $(i, j + 1/2)$. Тогава апроксимацията на производната по времето може да се разглежда като формула с централна разлика относно въведената междинна точка. Получаваме последователно

$$\begin{aligned} (\psi_{h,\tau})_i^{j+1/2} &= \frac{y_i^{j+1} - y_i^j}{\tau} - \frac{1}{2} D \frac{y_{i-1}^j - 2y_i^j + y_{i+1}^j}{h^2} - \frac{1}{2} D \frac{y_{i-1}^{j+1} - 2y_i^{j+1} + y_{i+1}^{j+1}}{h^2} - \frac{1}{2} (f_i^j + f_i^{j+1}) \\ &= \frac{\partial u}{\partial t} \Big|_i^{j+1/2} + O(\tau^2) - \frac{D}{2} \left(\frac{\partial^2 u}{\partial x^2} \Big|_i^j + \frac{\partial^2 u}{\partial x^2} \Big|_i^{j+1} \right) - \frac{1}{2} (f_i^j + f_i^{j+1}) \\ &= \cancel{\frac{\partial u}{\partial t} \Big|_i^{j+1/2}} + O(\tau^2) - \cancel{D \frac{\partial^2 u}{\partial x^2} \Big|_i^{j+1/2}} - \cancel{f_i^{j+1/2}} + O(h^2) \\ &= O(h^2 + \tau^2). \end{aligned}$$

В последното използвахме Твърдение 6 от приложението.

Да обобщим основното, което изведохме за схемите за решаване на уравнението на дифузията/топлопроводността.

Диференчното уравнение (3.18), допълнено с подходящи апроксимации на началното и граничните условия, дава клас диференчни схеми за приближеното решаване на уравнението на дифузията/топлопроводността.

- При $\sigma = 1$ схемата е явна, има ЛГА $O(h^2 + \tau)$ и е условно монотонна при $\tau < \frac{h^2}{2D}$;
- При $\sigma \neq 1$ схемата е неявна, има ЛГА $O(h^2 + \tau)$ и на всеки слой по времето трябва да се решава система с тридиагонална матрица. В частност:
 - Ако $\sigma = 0$ се получава чисто неявна схема, която е монотонна за всеки избор на τ ;
 - Ако $\sigma = 1/2$ се получава схема на Кранк–Никълсън, която има ЛГА $O(h^2 + \tau^2)$ и е монотонна при $\tau < \frac{h^2}{D}$. В следващите параграфи обаче ще докажем, че тя е устойчива за всяко τ .

3.2.4 Числени експерименти (Практикум)

Пример 26. Ще предложим функция в Mathematica, която имплементира явната диференчна схема за хомогенното уравнение на дифузията (когато $f \equiv 0$).

```

In[48]:= heatEquation[l_, T_, h_, u0_, uL_, uR_, d_] := (
    τ = h2 / (3 d);
    n = Ceiling[l / h];
    m = Ceiling[T / τ];
    t = Table[(j - 1) τ, {j, 1, m + 1}];
    x = Table[(i - 1) h, {i, 1, n + 1}];
    y = Table[0, {n + 1}, {m + 1}];
    (*Initial condition*)
    For[i = 1, i ≤ n + 1, i++,
        y[[i, 1]] = u0[x[[i]]]
    ];
    (*Iterate over the time nodes*)
    For[j = 1, j ≤ m, j++,
        (*Compute the internal values from the main PDE*)
        For[i = 2, i ≤ n, i++,
            y[[i, j + 1]] = (1 - 2  $\frac{d \tau}{h^2}$ ) y[[i, j]] +  $\frac{d \tau}{h^2}$  (y[[i - 1, j]] + y[[i + 1, j]])
        ];
        (*Compute the boundary values from the boundary conditions*)
        y[[1, j + 1]] = uL[t[[j + 1]]];
        y[[n + 1, j + 1]] = uR[t[[j + 1]]]
    ];
    {x, t, y}
)

```

Когато изследваме един диференчен метод, е добре да можем да го тестваме в частен случай на задачата, за който е известно точно решение (когато това е възможно). За тази цел ще приложим имплементираната диференчна схема за следната задача:

$$\begin{aligned}
 \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} &= 0, \quad 0 < x < 1, \quad 0 < t \leq 0.1 \\
 u(x, 0) &= \sin(2\pi x), \\
 u(0, t) = u(1, t) &= 0,
 \end{aligned}
 \tag{3.19}$$

която има точно решение

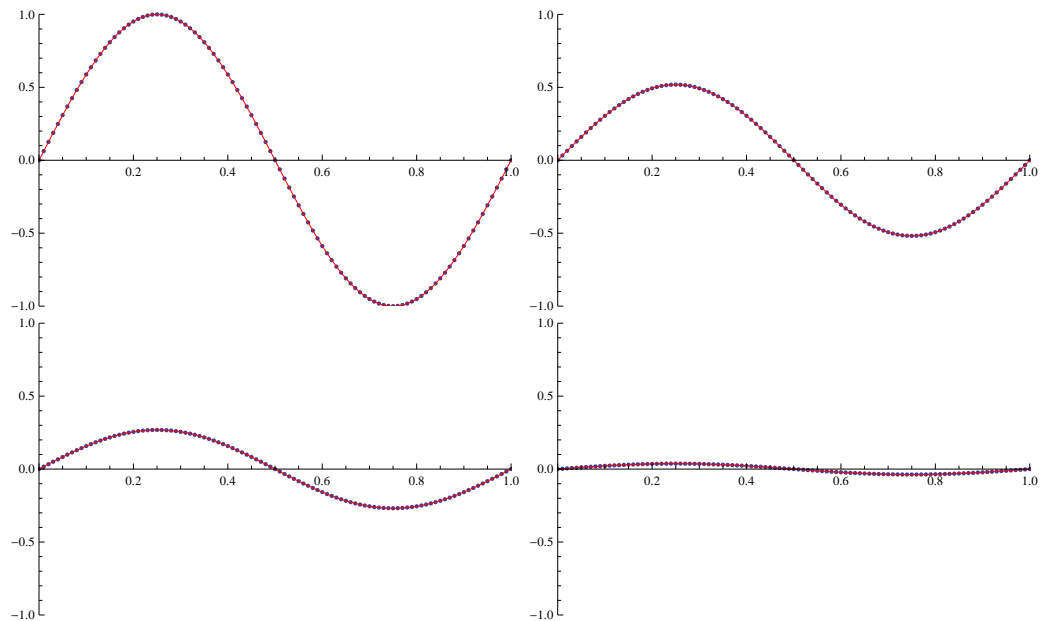
$$u(x, t) = e^{-4\pi^2 t} \sin(2\pi x).
 \tag{3.20}$$


```

u0[x_] := Sin[2 Pi x]
uL[t_] := 0
uR[t_] := 0
resX = heatEquation[1, 0.5, 0.01, u0, uL, uR, 1][[1]];
resT = heatEquation[1, 0.5, 0.01, u0, uL, uR, 1][[2]];
resY = heatEquation[1, 0.5, 0.01, u0, uL, uR, 1][[3]];
Manipulate[
  Show[
    ListPlot[
      Table[{resX[[i]], resY[[i, j]]}, {i, 1, Length[resX]}], PlotRange -> All],
    Plot[E^-4 Pi^2 resT[[j]] Sin[2 Pi x], {x, 0, 1}, PlotStyle -> Red]
  ],
  {j, 1, Length[resT], 1}
]

```

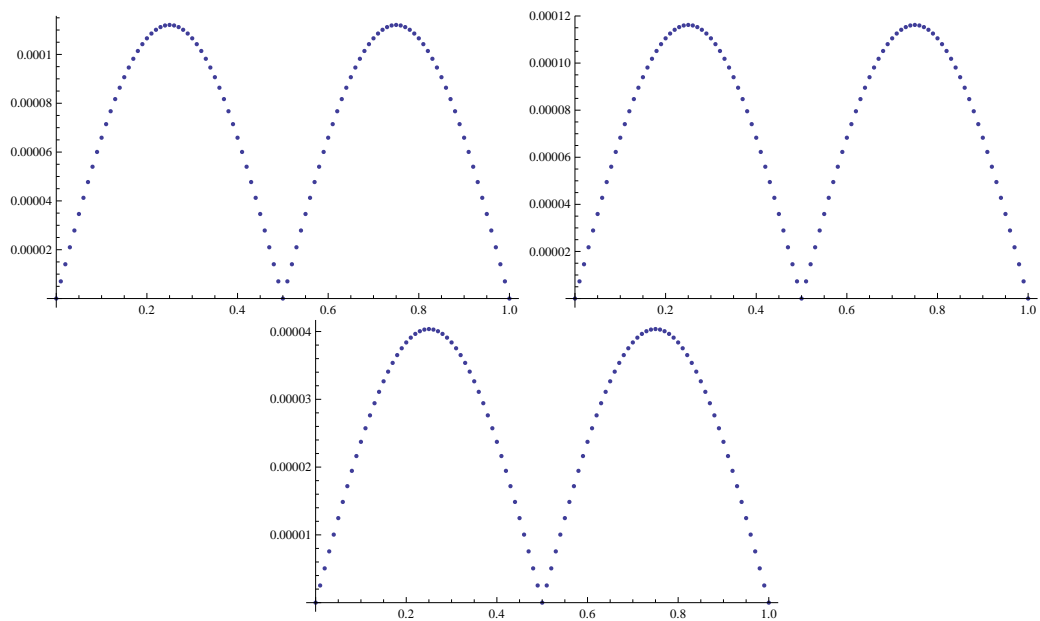
Графиките на приближеното и точното решение (съответно със сини точки и червена линия) за $t = 0, 0.015, 0.03, 0.08$ са приложени по-долу:



Визуално, приближеното и точното решение съвпадат.

В началния момент от време температурата в първата половина от интервала е положителна, а във втората – отрицателна. Граничните условия означават, че в двата края на областта има поглъщател на топлина, който поддържа температурата 0 (нека за определеност мислим, че температурата е в $^{\circ}C$). С течение на времето, вследствие на дифузията, температурата в лявата половина намалява, а в дясната страна расте, като окончателно се установява около равновесно състояние $0^{\circ}C$ в цялата дефиниционна област.

Съответните стойности на грешката по абсолютна стойност в точките от мрежата (пропускаме $t = 0$, тъй като там грешката очевидно е 0) са:

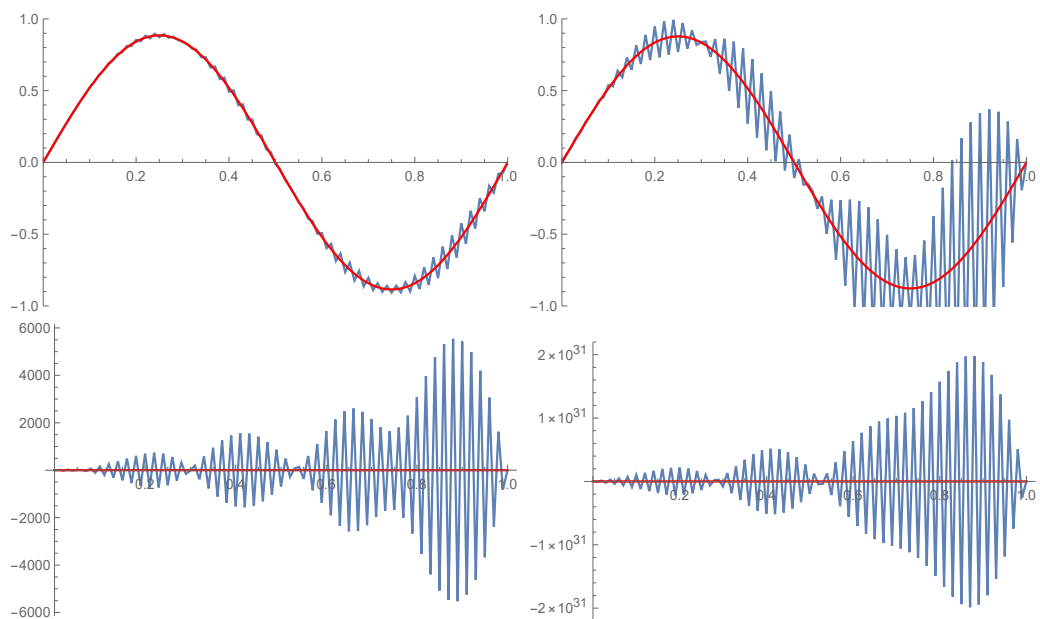


За разгледания времеви интервал максимумът на грешката намалява с цял порядък – от 10^{-4} до 10^{-5} . Това е свързано със свойството на параболичния диференциален оператор, който има изглаждащ ефект. Засега ще се ограничим с този лаконичен коментар. По-подробно ще коментираме въпроса, когато разглеждаме хиперболичните уравнения.

Нека видим и какво се случва, ако нарушим условието за монотонност и изберем стъпката по времето да бъде например

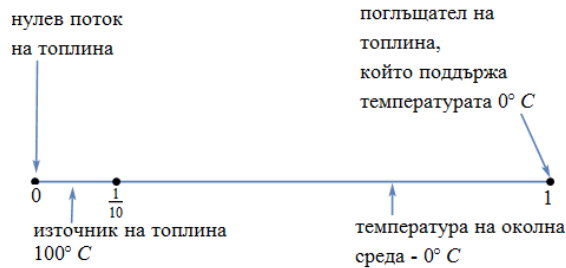
$$\tau = h^2/D.$$

Графики на приближеното и точното решение на слоевете по времето с индекси 30, 32, 40, 100 са приведени по-долу:



Както виждаме само за няколко стъпки по времето в приближеното решение започват да се наблюдават осцилации (немонотонност), които бързо се увеличават и решението става абсолютно неизползваемо.

Пример 27. Нека сега разгледаме задача за едномерно тяло с начална температура 0°C , която схематично е представена по-долу:



Освен дифузията, в този случай ще разгледаме топлообмен с външната среда. Тогава

изменението на температурата = дифузионен член + топлообмен с външната среда.

Топлообменът с външната среда е пропорционален на разликата в температурата на тялото и външната среда, т.е. на $v(x) - u(x, t)$, където

$$v(x) := \begin{cases} 100, & x \leq 1/10 \\ 0, & x > 1/10 \end{cases}.$$

Нека означим коефициента на пропорционалност с \varkappa . В числените експерименти ще приемем, че $\varkappa = 1$.

Тогава диференциалното уравнение, описващо процеса, е

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2} + \varkappa(v(x) - u).$$

Дясното гранично условие е от първи род – $u(1, t) = 0$.

За да опишем нулев поток на топлина през лявата граница, ще наложим условието на Нойман

$$\partial u / \partial x(0, t) = 0.$$

И така, разглеждаме диференциалната задача

$$\begin{aligned} \frac{\partial u}{\partial t} - D \frac{\partial^2 u}{\partial x^2} + \varkappa u &= v(x)\varkappa, & 0 < x < 1, & 0 < t \leq 0.1 & (3.21) \\ u(x, 0) &= 0, \\ \frac{\partial u}{\partial x}(0, t) &= 0, \\ u(1, t) &= 0. \end{aligned}$$

За апроксимацията на диференциалното уравнение, началното и дясното гранични условия подходдаме, както и в предишния пример.

Получаваме

$$\begin{aligned} \frac{y_i^{j+1} - y_i^j}{\tau} - D \frac{y_{i+1}^j - 2y_i^j + y_{i-1}^j}{h^2} + \varkappa y_i^j &= v(x_i) \varkappa, \quad i = \overline{1, n-1}, \quad j = \overline{1, m-1} \\ y_i^0 &= 0, \quad i = \overline{0, n}, \\ y_n^j &= 0, \quad j = \overline{1, m}. \end{aligned} \tag{3.22}$$

Записано в каноничен вид, първото уравнение е

$$y_i^{j+1} = \frac{D\tau}{h^2} (y_{i+1}^j + y_{i-1}^j) + \left(1 - \frac{2D\tau}{h^2} - \tau\varkappa\right) y_i^j + v(x_i)\tau\varkappa.$$

Нека отбележим, че неговата ЛГА е $O(h^2 + \tau)$.

От условието за положителност на коефициентите получаваме

$$1 - \frac{2D\tau}{h^2} - \tau\varkappa \leq 0 \implies \tau \leq \frac{h^2}{2D + \varkappa h^2}.$$

За лявото гранично условие подхождаме по начина, който описахме в параграф 3.2.2. Започваме, като апроксимираме производната с разлика напред:

$$\frac{y_1^j - y_0^j}{h} = 0.$$

Сега пресмятаме ЛГА и членът от първи ред в нея вадим от лявата страна на последното уравнение. Развивайки в ред на Тейлър около $x = 0$, получаваме:

$$\begin{aligned} \psi_{h,\tau} &= \frac{1}{h} \left(u_0^j + \frac{\partial u}{\partial x} \Big|_0^j h + \frac{\partial^2 u}{\partial x^2} \Big|_0^j \frac{h^2}{2} + O(h^3) - u_0^j \right) \\ &= \frac{h}{2} \frac{\partial^2 u}{\partial x^2} \Big|_0^j + O(h^2). \end{aligned}$$

В последното използвахме, че $\frac{\partial u}{\partial x}(0, t) = 0$. Като допускаме, че основното уравнение е изпълнено и на границата с достатъчно висока точност, получаваме.

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{D} \left(\frac{\partial u}{\partial t} - \varkappa(v(0) - u) \right).$$

Тогава, като използваме последното, получаваме окончателно апроксимация на граничното условие с грешка $O(h^2 + \tau)$:

$$\frac{y_1^j - y_0^j}{h} - \frac{h}{2D} \left(\frac{y_0^{j+1} - y_0^j}{\tau} - \varkappa(v(0) - y_0^j) \right) = 0.$$

Записано в каноничен вид, диференчното уравнение добива вида

$$y_0^{j+1} = \left(1 - \frac{2\tau D}{h^2} - \varkappa\tau\right) y_0^j + \frac{2D\tau}{h^2} y_1^j + v(0)\varkappa\tau. \tag{3.23}$$

И така, получихме диференчната схема (3.22), (3.23).

Прилагаме нейна примерна имплементация в Mathematica:

```

In[1]:= heatEquation[l_, T_, h_, u0_, d_, κ_] := (
    τ = h^2 / (3 d + κ h^2);
    n = Ceiling[l / h];
    m = Ceiling[T / τ];
    t = Table[(j - 1) τ, {j, 1, m + 1}];
    x = Table[(i - 1) h, {i, 1, n + 1}];
    y = Table[0, {n + 1}, {m + 1}];
    (*Initial condition*)
    For[i = 1, i ≤ n + 1, i++,
        y[[i, 1]] = u0[x[[i]]]
    ];
    (*Iterate over the time nodes*)
    For[j = 1, j ≤ m, j++,
        (*Compute the internal values from the main PDE*)
        For[i = 2, i ≤ n, i++,
            y[[i, j + 1]] =
                (1 - 2  $\frac{d \tau}{h^2}$  - τ κ) y[[i, j]] +  $\frac{d \tau}{h^2}$  (y[[i - 1, j]] + y[[i + 1, j]]) + v[x[[i]]] τ κ
        ];
        (*Compute the boundary values from the boundary conditions*)
        y[[1, j + 1]] = (1 - 2  $\frac{d \tau}{h^2}$  - τ κ) y[[1, j]] +  $\frac{2 d \tau}{h^2}$  y[[2, j]] + 100 τ κ;
        y[[n + 1, j + 1]] = 0
    ];
    {x, t, y}
)

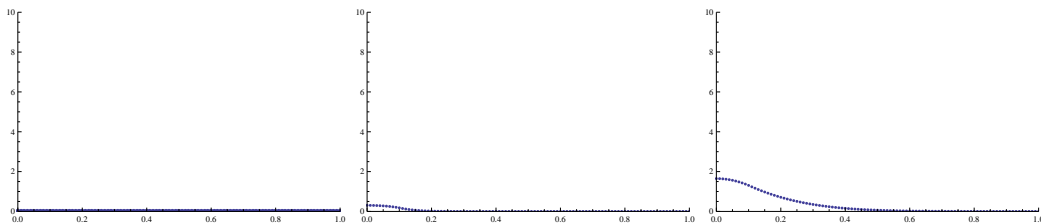
```

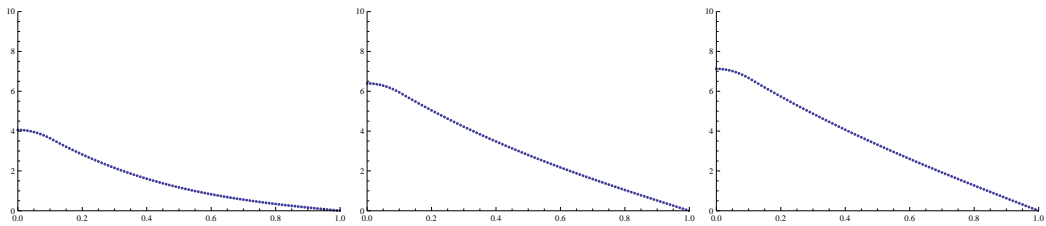
Нека разгледаме приближеното решение за $t = 0, 0.003, 0.03, 0.17, 0.5, 0.83$:

```

In[2]:= u0[x_] := 0
v[x_] := If[x ≤ 1 / 10, 100, 0]
resX = heatEquation[1, 1, 0.01, u0, 1, 1][[1]];
resT = heatEquation[1, 1, 0.01, u0, 1, 1][[2]];
resY = heatEquation[1, 1, 0.01, u0, 1, 1][[3]];
Manipulate[
    ListPlot[Table[{resX[[i]], resY[[i, j]]}, {i, 1, Length[resX]}],
        PlotRange → {{0, 1}, {0, 10}},
        {j, 1, Length[resT], 1}
]

```





Вследствие на източника на топлина, температурата се повишава в началото на интервала, като дифузията я разпространява в целия интервал. Поради поглъщателя на топлина на дясната граница, температурният профил достига равновесното си състояние, изобразено на последната графика.

Важно свойство на параболичния оператор е, че той има заглаждащ ефект. Наличието на особености в началните условия – липса на гладкост или дори непрекъснатост – не води до съществени проблеми при численото решаване на параболични уравнения. Профилът на решението с течение на времето “се заглажда”.

Пример 28. За схемата на Кранк–Никълсън линейната система, която трябва да се решава на всеки слой по времето за зад. (3.19), има вида (проверете!):

$$\begin{bmatrix} 1 & 0 & & & & \\ -\frac{1}{2h^2} & \frac{1}{\tau} + \frac{1}{h^2} & -\frac{1}{2h^2} & & & \\ & \ddots & \ddots & \ddots & & \\ & & -\frac{1}{2h^2} & \frac{1}{\tau} + \frac{1}{h^2} & -\frac{1}{2h^2} & \\ & & & 0 & 1 & \\ & & & & & 1 \end{bmatrix} \begin{bmatrix} y_0^{j+1} \\ y_1^{j+1} \\ \vdots \\ y_{n-1}^{j+1} \\ y_n^{j+1} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{y_1^j}{\tau} + \frac{y_0^j - 2y_1^j + y_2^j}{2h^2} \\ \vdots \\ \frac{y_{n-1}^j}{\tau} + \frac{y_{n-2}^j - 2y_{n-1}^j + y_n^j}{2h^2} \\ 0 \end{bmatrix} \quad (3.24)$$

Прилагаме примерна имплементация:

```

In[1]:= h = 0.01;
        τ = 0.01;
        l = 1;
        T = 0.1;
        n = Ceiling[l / h];
        m = Ceiling[T / τ];
        t = Table[(j - 1) τ, {j, 1, m + 1}];
        x = Table[(i - 1) h, {i, 1, n + 1}];
        y = Table[0, {n + 1}, {m + 1}];

(*Initial condition*)
For[i = 1, i ≤ n + 1, i++,
    y[[i, 1]] = Sin[2 Pi (i - 1) h]
];

(*Generate the matrix of the linear
system. This can be done outside the for-loop, since the
matrix is constant*)
A = Table[0, {n + 1}, {n + 1}];
A[[1, 1]] = 1;
A[[n + 1, n + 1]] = 1;
For[i = 2, i < n + 1, i++,
    A[[i, i - 1]] = -1 / (2 h^2);
    A[[i, i]] = 1 / τ + 1 / h^2;
    A[[i, i + 1]] = -1 / (2 h^2);
];

(*Iterate over the time nodes*)
For[j = 1, j ≤ m, j++,
    (*Generate the right-hand side vector of the linear system*)
    b = Table[0, {n + 1}];
    For[k = 2, k ≤ n, k++,
        b[[k]] =  $\frac{y[[k, j]]}{\tau} + \frac{y[[k - 1, j]] - 2 y[[k, j]] + y[[k + 1, j]]}{2 h^2}$ 
    ];
    yNew = LinearSolve[A, b];
    y[[All, j + 1]] = yNew
];

```

3.3 Устойчивост на диференчните схеми за решаване на ЧДУ

Нека е дадена диференциалната задача

$$\begin{aligned}
 \mathcal{L}u &= f, & 0 < x < X, & 0 < t \leq T, \\
 \ell_{init}u &= u_0, & t &= 0 \\
 \ell_{bound}u &= \mu, & x &= 0, x = X.
 \end{aligned}
 \tag{3.25}$$

В последното \mathcal{L} е линеен диференциален оператор, който описва лявата страна на основното уравнение в дадена диференциална задача (например за уравнението на дифузията (3.3) имаме $\mathcal{L} \equiv \frac{\partial}{\partial t} - D \frac{\partial^2}{\partial x^2}$). Операторът l_{init} е линеен оператор описващ началното условие (например в случая на началното условие (3.4) l_{init} е просто операторът идентитет). Операторът l_{bound} съответно описва граничните условия в задачата.

Зададохме на практика произволна линейна диференциална задача, за да може анализът, който ще направим по-долу, да е в сила за всички разглеждани в курса задачи.

Решаваме задачата (3.25) приближено, като построяваме диференчна схема върху мрежата

$$\bar{\omega}_{h,\tau} := \{(x_i, t_j) : x_i = ih, t_j = j\tau, i = \overline{0, n}, j = \overline{0, m}, n = l/h, m = T/\tau\}.$$

Нека означим $y_h(x_i, t_j) := y_i^j$ за всяка точка от мрежата. С други думи y_h е **мрежовата функция** (т.е. функция, дефинирана само върху точките от мрежата), отговаряща на приближеното решение, получено по диференчната схема. С помощта на операторния запис, който използвахме по-горе, можем да запишем в общ вид и разглежданите от нас диференчни схеми:

$$\begin{aligned} \mathcal{L}_h y_h &= f, & 0 < x_i < X, & \quad 0 < t_j \leq T, \\ l_{init,h} y_h &= u_0, & t_j &= 0 \\ l_{bound,h} y_h &= \mu, & x_i &= 0, x_i = X. \end{aligned} \tag{3.26}$$

В последното $\mathcal{L}_h, l_{init,h}, l_{bound,h}$ са **мрежови/диференчни оператори** (т.е. оператори, които на мрежова функция съпоставят мрежова функция). Те описват съответно апроксимациите на левите страни в основното уравнение, началното условие, граничното условие. Например за явната схема за уравнението на дифузията имаме

$$\mathcal{L}_h y_h(x_i, t_j) := \frac{y_h(x_i, t_{j+1}) - y_h(x_i, t_j)}{\tau} - D \frac{y_h(x_{i+1}, t_j) - 2y_h(x_i, t_j) + y_h(x_{i-1}, t_j))}{h^2}.$$

Сега ще дефинираме точно и ще разгледаме въпроса за устойчивостта на диференчните схеми, който коментирахме на интуитивно ниво в предишната секция.

Дефиниция 10: Устойчивост на диференчната схема в $\|\cdot\|_h$ -норма

Нека в пространството от мрежови функции y_h е въведена норма $\|\cdot\|_h$. Казваме, че диференчната схема (3.26) е устойчива в $\|\cdot\|_h$, ако приближеното решение y_h изпълнява оценката

$$\|y_h\|_h \leq C_1 \|u_0\|_h + C_2 \|\mu\|_h + C_3 \|f\|_h$$

за някои константи C_1, C_2, C_3 .

Нека коментираме по-подробно последната дефиниция. Интуитивната ни представа за понятието устойчивост е, че малки изменения във входните данни (например грешки от закръглявания, апроксимация и т.н.) трябва да водят

до малки изменения в численото решение на задачата. Тази идея не е съвсем очевидна в последната дефиниция. Можем да се уверим в това обаче, ако разгледаме задачата с малко променени дясна страна, начално условие и гранично условие – нека ги означим съответно с \tilde{f} , \tilde{u}_0 , $\tilde{\mu}$. Тогава решението \tilde{y}_h на задачата с променените данни трябва да изпълнява

$$\begin{aligned}\mathcal{L}_h \tilde{y}_h &= \tilde{f}, \quad 0 < x < X, \quad 0 < t \leq T, \\ \ell_{init,h} \tilde{y}_h &= \tilde{u}_0, \quad t = 0, \\ \ell_{bound,h} \tilde{y}_h &= \tilde{\mu}, \quad x = 0, x = X.\end{aligned}\tag{3.27}$$

Но тогава, ако извадим почленно (3.27) от (3.26), ще се уверим, че грешката удовлетворява

$$\begin{aligned}\mathcal{L}_h(y_h - \tilde{y}_h) &= f - \tilde{f}, \quad 0 < x < X, \quad 0 < t \leq T, \\ \ell_{init,h}(y_h - \tilde{y}_h) &= u_0 - \tilde{u}_0, \quad t = 0, \\ \ell_{bound,h}(y_h - \tilde{y}_h) &= \mu - \tilde{\mu}, \quad x = 0, x = X,\end{aligned}$$

т.е. **удовлетворява същата задача** с дясна страна $f - \tilde{f}$, начално условие $u_0 - \tilde{u}_0$ и гранично условие $\mu - \tilde{\mu}$. Следователно, ако диференчната схема е устойчива (съгласно приведената по-горе дефиниция), то ще бъде в сила оценката

$$\|y_h - \tilde{y}_h\|_h \leq C_1 \|u_0 - \tilde{u}_0\|_h + C_2 \|\mu - \tilde{\mu}\|_h + C_3 \|f - \tilde{f}\|_h,$$

т.е. ако изменението на дясната страна, началните условия и граничните условия е малко, то и грешката ще бъде малка.

Да обърнем внимание още на следното – в дясната страна на дефиницията за устойчивост има три члена – нормите на началното условие u_0 , граничното условие μ и дясната страна f . Оказва се, че ние можем да изследваме поотделно влиянието на грешки във всяка от входните данни, тъй като можем да запишем $y_h = y_{h,1} + y_{h,2} + y_{h,3}$, където

$$\begin{aligned}\mathcal{L}_h y_{h,1} &= 0, & \mathcal{L}_h y_{h,2} &= 0, & \mathcal{L}_h y_{h,3} &= f, \\ \ell_{init,h} y_{h,1} &= u_0, & \ell_{init,h} y_{h,2} &= 0, & \ell_{init,h} y_{h,3} &= 0, \\ \ell_{bound,h} y_{h,1} &= 0, & \ell_{bound,h} y_{h,2} &= \mu, & \ell_{bound,h} y_{h,3} &= 0,\end{aligned}\tag{3.28} \tag{3.29} \tag{3.30}$$

Следователно ние можем да изследваме устойчивостта на дадена диференчна схема, като поотделно изследваме как влияят различните входни данни в задачата, т.е. началните условия, граничните условия и дясната страна.

Дефиниция 11: Устойчивост по начални данни, по гранични условия, по дясна страна

Казваме, че диференчната схема (3.26) е:

- **устойчива по начални данни**, ако решението на (3.28) изпълнява

$$\|y_{h,1}\| \leq C_1 \|u_0\|_h;$$

- **устойчива по гранични условия**, ако решението на (3.29) изпълнява

$$\|y_{h,2}\| \leq C_2 \|\mu\|_h;$$

- **устойчива по дясна страна**, ако решението на (3.30) изпълнява

$$\|y_{h,3}\| \leq C_3 \|f\|_h.$$

Следователно една диференчна схема е устойчива точно тогава, когато е устойчива по начални данни, по гранични условия и по дясна страна.

Н.В. 29

Да отбележим, че при нестационарните задачи най-важна е устойчивостта по начални условия. Ако схемата е устойчива по начални данни, тя обикновено на практика е устойчива. Ние няма да се спираме по-подробно на този въпрос в курса, но ще насочим вниманието си именно към изследването на устойчивост по начални данни. Всички числени експерименти, които направихме в предишната секция и които ще направим в следващите, ще свидетелстват, че когато изпълним условието за устойчивост по начални данни, всичко с приближеното решение е наред.

Въведените дефиниции в настоящата тема са общи, за произволна норма. На практика най-често се използват следните две норми, с които мерим “колко голяма” е грешката:

- мрежова максимум-норма:

$$\|y_h\|_{h,\infty} := \max_{(x_i,t_j) \in \omega_{h,\tau}} |y_h(x_i,t_j)|; \quad (3.31)$$

- мрежова ℓ_2 -норма:

$$\|y_h\|_{h,2} := \sqrt{\frac{1}{N} \sum_{(x_i,t_j) \in \omega_{h,\tau}} |y_h(x_i,t_j)|^2}, \quad (3.32)$$

където N е броят на точките в мрежата.

3.3.1 Принцип за максимума. Условие за положителност на коефициентите. Устойчивост в $\|\cdot\|_{h,\infty}$ -норма

Както отбелязахме в предишния параграф, ние няма да разглеждаме въпроса за устойчивост по дясна страна и ще считаме, че разглежданите от нас схеми

са устойчиви по дясна страна (това за много схеми може да се провери съвсем непосредствено, вж. например [3]). Затова в следващите редове ние ще приемаме, че в задачата $f \equiv 0$, т.е. няма източников член (напр. източник на топлина в областта).

Нека разгледаме отново Пример 26. Както можем да се уверим от приложените графики на решението в различни моменти от време, то достига максималната си положителна и минималната си отрицателна стойности в началния момент от време. Това, разбира се, е напълно естествено от физична гледна точка. С течение на времето дифузията кара топлината да се разпространява от места, където температурата е висока, към места, където температурата е ниска. Следователно там, където стойността на u е голяма, тя ще намалява и обратното.

Другата възможност е максималната положителна/минималната отрицателна стойност да се получават на границата на областта (при $x = 0$, $x = X$) – вж. пример 27. Ако ние например загряваме тялото на лявата или дясната граница, е естествено температурата там да достига своята максимална стойност.

Не е възможно обаче максимумът да се достига другаде – ако нито в началния момент температурата е надминавала например 10° , нито сме загрявали тялото на двете граници с повече, то няма откъде да дойде топлинна енергия, която да накара температурата да стане например 20° .

Казваме, че уравнението на дифузията изпълнява **принципа за максимума**.

Дефиниция 12: Принцип за максимума за диференциални задачи

Казваме, че дадена диференциална задача изпълнява принципа за максимума, ако решението на хомогенната задача (при $f \equiv 0$) може да достига своята максимална положителна/минимална отрицателна стойност или в началния момент от време, или на границата на областта по пространството, в която решаваме задачата. Да обърнем внимание, че това е свойство на решението на диференциалната задача (не на числената схема), което е в сила например за уравнението на дифузията, уравнението на преноса, уравнението на струната и др.

Да разгледаме сега графиките в края на Пример 26, които илюстрират какво се случва, ако числената схема е неустойчива. Както виждаме, това води до появата на осцилации и приближеното решение не изпълнява принципа за максимума, нещо повече – то не е ограничено, а “бяга” към безкрайност.

Тези наблюдения ни навеждат на мисълта, че ако “накараме” приближеното решение също да изпълнява принципа за максимума, то няма как да има такова лошо поведение.

N.B. 30

Оказва се, че именно условието за положителност на коефициентите, което формално приложихме, когато разглеждахме диференчни схеми за уравнението на дифузията, гарантира, че приближеното решение ще изпълнява принципа за максимума. С други думи то гарантира, че е в сила оценката

$$\|y_h\|_{h,\infty} \leq \|u_0\|_{h,\infty} + \|\mu\|_{h,\infty}.$$

Тогава, ако е изпълнено условието за положителност на коефициентите и следователно принципът за максимума е в сила и за приближеното решение, то диференчната схема е устойчива (по начални данни и гранични условия) в $\|\cdot\|_{h,\infty}$ -норма.

Можем да се убедим лесно, че последното наистина е изпълнено. Нека разгледаме първо следния пример, който съответства на явна схема. Нека y_i^{j+1} се изразява явно чрез $y_{i-1}^j, y_i^j, y_{i+1}^j$ (нека за определеност си мислим, че те са положителни) по следния начин:

$$y_i^{j+1} = Ay_{i-1}^j + By_i^j + Cy_{i+1}^j,$$

където коефициентите A, B, C удовлетворяват условията за положителност на коефициентите, т.е.

$$A > 0, B > 0, C > 0, A + B + C \leq 1.$$

Тогава е ясно, че $y_i^{j+1} \leq \max\{y_i^j, y_{i-1}^j, y_{i+1}^j\}$, като равенство се достига само ако $y_i^j = y_{i-1}^j = y_{i+1}^j$ и $A + B + C = 1$. С други думи, приближеното решение на $j + 1$ -вия слой не може да има по-голяма положителна стойност отколкото максимума на j -тия (да забележим, че на j -тия слой максимумът може да се достига в граничните точки!). Разсъждавайки индуктивно за всички предишни слоеве, получаваме, че максималната положителна стойност се достига или на 0-вия слой по времето, или по границата на областта.

Тази идея се обобщава и за произволни схеми, не само за явни. Действително, да допуснем, че условията за положителност на коефициентите са в сила, но максимумът се достига във вътрешна точка y_i^{j+1} . Тогава от каноничния вид

$$y_i^{j+1} = \sum_{(x,t) \in S} A_{(x,t)} y_h(x,t),$$

където S е шаблонът, върху който е построена схемата, без точката (x_i, t_{j+1}) , от условието за положителност на коефициентите и факта, че y_i^{j+1} е максимална стойност следва, че стойностите върху всички точки на шаблона са равни. Поставяйки шаблона последователно върху всяка точка от мрежата, ще получим, че решението е константа, т.е. максимумът (който е равен на тази константа) се достига навсякъде, в частност по границата.

N.B. 31

Устойчивостта в $\|\cdot\|_{h,\infty}$ -норма е най-доброто нещо, на което можем да се надяваме. Тя означава, че **максимумът на грешката е ограничен**, т.е. във всяка точка от мрежата решението следва да бъде “правилно”.

3.3.2 Метод на хармониките за изследване на устойчивост в мрежова $\|\cdot\|_{h,2}$ -норма

Както отбелязахме в предходния параграф, ако условието за положителност на коефициентите е удовлетворено, т.е. е в сила принципът за максимума, то е налице устойчивост в $\|\cdot\|_{h,\infty}$ -норма, което е най-силният резултат, на който можем да се надяваме. **Той обаче не е в сила за всяка диференчна схема.** За схемите, за които не може да се гарантира това, се разглежда по-слабо изискване за устойчивост, а именно устойчивост в мрежова l_2 -норма.

N.B. 32

Ако се върнем към дефиницията (3.32), е ясно, че мрежовата l_2 -норма е показател за “средната стойност” на y_h в точките от мрежата и следователно устойчивост в тази норма ще означава, че “средната грешка” върху точките от мрежата ще е “малка”. Това, разбира се, е по-слаб резултат от устойчивостта в максимум-норма, тъй като е възможно да има точки, в които грешката има по-големи стойности, но все пак резултатът, получен чрез диференчната схема следва да бъде “смислен”.

Класическият подход за изследване на устойчивост в мрежова l_2 -норма е **методът на хармониките (още, метод на фон Нойман)**. Той позволява да се изведат условия за устойчивост по начални данни и се основава на следната идея. Говорейки за устойчивост по начални данни разглеждаме задачата с гранични условия $u(0, t) = u(X, t) = 0$ и следователно решението ѝ е периодично. Тогава е естествено да разглеждаме как диференчната схема се държи ако началните данни (или решението на някой произволен слой) може да се развие в ред на Фурие:

$$\sum_{\varphi=-\infty}^{+\infty} C_{\varphi} e^{ix\varphi},$$

където $i = \sqrt{-1}$.

Ще изложим идеята върху няколко примера. Нека се върнем на разгледащите схеми за уравнението на дифузията.

- Чисто неявната схема е безусловно устойчива в $\|\cdot\|_{h,\infty}$ -норма и следователно няма защо да изследваме устойчивост в по-слабата $\|\cdot\|_{h,2}$ -норма.
- Ще разгледаме как се прилага методът на хармониките първо върху явната схема, която за удобство прилагаме отново тук (б.о.о. нека $D = 1$ и нека обърнем внимание, че разглеждаме $f \equiv 0$, тъй като изследваме устойчивост по начални данни):

$$\frac{y_k^{j+1} - y_k^j}{\tau} - \frac{y_{k+1}^j - 2y_k^j + y_{k-1}^j}{h^2} = 0.$$

Нека видим как схемата се държи за един произволен член от реда на Фурие. Нека на произволен j -ти слой по времето имаме $y_k^j = e^{ikh\varphi}$, където φ е произволно реално число. Вземайки предвид, че

$$\|y_h(x, t_j)\|_{h,2} = \sqrt{\frac{1}{n+1} \sum_{k=0}^n |e^{ikh\varphi}|^2}$$

и формулата на Ойлер, според която

$$e^{ikh\varphi} = \cos(kh\varphi) + i \sin(kh\varphi),$$

получаваме

$$\|y_h(x, t_j)\|_{h,2} = 1. \quad (3.33)$$

Нека сега

$$y_k^{j+1} = qy_k^j.$$

Частното q , което ни показва какво е отношението между стойността в k -тата точка на $j+1$ -вия и j -тия слой, разбира се, зависи от пресмятанията, направени по диференчната схема. Нека го изразим. Имаме

$$\frac{y_k^{j+1} - y_k^j}{\tau} = \frac{y_{k+1}^j - 2y_k^j + y_{k-1}^j}{h^2}$$

и следователно

$$\frac{qe^{ikh\varphi} - e^{ikh\varphi}}{\tau} = \frac{e^{i(k+1)h\varphi} - 2e^{ikh\varphi} + e^{i(k-1)h\varphi}}{h^2}.$$

Като разделим двете страни на $e^{ikh\varphi}$, получаваме

$$\frac{q - 1}{\tau} = \frac{e^{ih\varphi} - 2 + e^{-ih\varphi}}{h^2}.$$

Прилагаме формулата на Ойлер и получаваме

$$\frac{q - 1}{\tau} = \frac{\cos(h\varphi) + i \sin(h\varphi) - 2 + \cos(h\varphi) - i \sin(h\varphi)}{h^2}.$$

Сега, вземайки предвид тригонометричното твърждение

$$1 - \cos \alpha = 2 \sin^2 \frac{\alpha}{2}, \quad (3.34)$$

получаваме

$$\frac{q - 1}{\tau} = -4 \frac{\sin^2 \frac{h\varphi}{2}}{h^2},$$

т.е.

$$q = 1 - \frac{4\tau}{h^2} \sin^2 \frac{h\varphi}{2}.$$

Нека забележим, че

$$\|y_h(x, t_{j+1})\|_{h,2} = \sqrt{\frac{1}{n+1} \sum_{k=0}^n |qe^{ikh\varphi}|^2} = |q|.$$

Тогава от последното и (3.33) следва, че за да бъде схемата устойчива в мрежова ℓ_2 норма е необходимо $|q| < 1$, като последното трябва да е изпълнено за произволно φ (тъй като ние взехме произволен член от реда на Фурие). И така, условието за устойчивост на диференчната схема в ℓ_2 -норма е

$$\left| 1 - \frac{4\tau}{h^2} \sin^2 \frac{h\varphi}{2} \right| < 1 \iff -1 < 1 - \frac{4\tau}{h^2} \sin^2 \frac{h\varphi}{2} < 1, \forall \varphi \in \mathbb{R}$$

Лесно се проверява, че последното е изпълнено точно когато $\tau < \frac{h^2}{2}$. С други думи, условието за стъпката по времето (3.10), което получихме за устойчивост в максимум-норма, не може да бъде отслабено, дори и да изискваме само устойчивост в ℓ_2 -норма.

Да отбележим, че направените тук разсъждения означават, че ако $y_k^0 = e^{ikh\varphi}$, то $y_k^j = q^j e^{ikh\varphi}$ и $\|y_h(x, t_j)\|_{h,2} = |q|^j$ и при $|q| < 1$ е в сила $\|y_h(x, t_j)\|_{h,2} < \|y_h(x, t_0)\|_{h,2}$, т.е. схемата е устойчива по начални данни в $\|\cdot\|_{h,2}$ -норма.

- За схемата на Кранк–Никълсън ние показахме, че условието за положителност на коефициентите е изпълнено при $\tau < h^2/D$, т.е. схемата е условно устойчива в $\|\cdot\|_{h,\infty}$ -норма. Оказва се обаче, че тя е безусловно устойчива в $\|\cdot\|_{h,2}$ -норма. Това отново се проверява, като се тества схемата върху специалните частни решения от вида $y_k^j = q^j e^{ikh\varphi}$. Замествайки в схемата на Кранк–Никълсън, получаваме

$$\frac{q^{j+1}e^{ikh\varphi} - q^j e^{ikh\varphi}}{\tau} = \frac{q^j(e^{i(k+1)h\varphi} - 2e^{ikh\varphi} + e^{i(k-1)h\varphi})}{2h^2} + \frac{q^{j+1}(e^{i(k+1)h\varphi} - 2e^{ikh\varphi} + e^{i(k-1)h\varphi})}{2h^2}$$

или, което е същото,

$$\frac{qe^{ikh\varphi} - e^{ikh\varphi}}{\tau} = \frac{e^{i(k+1)h\varphi} - 2e^{ikh\varphi} + e^{i(k-1)h\varphi}}{2h^2} + \frac{q(e^{i(k+1)h\varphi} - 2e^{ikh\varphi} + e^{i(k-1)h\varphi})}{2h^2}.$$

Делим двете страни на $e^{ikh\varphi}$ и получаваме

$$\frac{q-1}{\tau} = \frac{e^{ih\varphi} - 2 + e^{-ih\varphi}}{2h^2} + \frac{q(e^{ih\varphi} - 2 + e^{-ih\varphi})}{2h^2}.$$

Прилагайки формулата на Ойлер и тригонометричното тъждество (3.34), последното се свежда до

$$\frac{q-1}{\tau} = -\frac{2\sin^2 \frac{\varphi h}{2}}{h^2} - \frac{2q\sin^2 \frac{\varphi h}{2}}{h^2}$$

и следователно

$$q = \frac{1 - \frac{2\tau\sin^2 \frac{\varphi h}{2}}{h^2}}{1 + \frac{2\tau\sin^2 \frac{\varphi h}{2}}{h^2}}.$$

Както видяхме, условието за устойчивост е $|q| < 1, \forall \varphi \in \mathbb{R}$. Лесно се проверява, че последното е изпълнено винаги.

N.B. 33

Методът на хармониките се състои в тестване на схемата за специални нейни частни решения от вида $y_k^j = q^j e^{ikh\varphi}$. Ако, прилагайки схемата, е в сила $|q| < 1$ за произволно реално число φ , то казваме, че схемата е устойчива по фон Нойман. Последното е свидетелство за устойчивост в $\|\cdot\|_{h,2}$ -норма.

3.4 Диференчни методи за хиперболични уравнения

Наред с дифузията, другият основен процес, който обуславя движението на вещество, температура и др., е конвекцията или простият пренос. Като най-прост пример за такъв тип движение ще дадем концентрацията на вещество, пуснато в течаща вода (например река). Причината за преноса на веществото (ако пренебрегнем дифузията) е самото движение на водата, т.е. тя ще го пренася със скорост c , с която се движи реката. За да опишем този процес, ще използваме уравнението на непрекъснатостта (3.1), което изведохме по-рано. Тъй като водата носи веществото с константна скорост c , то

$$j_{adv} = cu.$$

Като заместим в (3.1), получаваме следното хиперболичното частно диференциално уравнение от първи ред

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = f(x, t). \quad (3.35)$$

Последното се нарича **уравнение на преноса** или **уравнение на конвекцията**.

Имайки предвид, че в диференциалното уравнение участват производна от първи ред по пространството и от първи ред по времето, е ясно, че за да затворим диференциалната задача, е необходимо да наложим едно гранично условие по пространството и едно начално условие по времето, т.е. решаваме задачата:

$$\begin{aligned} \frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} &= f(x, t), \quad 0 < x \leq l, \quad 0 < t \leq T, \\ u(x, 0) &= u_0(x), \quad 0 \leq x \leq l, \\ u(0, t) &= u_L(t), \quad 0 \leq t \leq T. \end{aligned}$$

Понеже решението при $t = 0$ има вида $u(x, 0) = u_0(x)$, то тогава може да се провери, че $u(x, t) = u_0(x - ct)$ удовлетворява уравнението на преноса, което означава, че профилът на решението се запазва с течение на времето, като се пренася надясно със скорост c . Това е пример за т.нар. решения от тип “бягаща вълна”.

3.4.1 Двуслойни монотонни схеми с първи ред на точност за уравнението на преноса

Ще построим явна двуслойна схема за решаване на горната задача. За тази цел нека разгледаме два произволни слоя по времето $j, j + 1$ от мрежата

$$\bar{\omega}_{h,\tau} := \{(x_i, t_j) : x_i = ih, t_j = j\tau, i = \overline{0, n}, j = \overline{0, m}, n = l/h, m = T/\tau\}.$$

Първата ни цел ще бъде да апроксимираме основното диференциално уравнение (3.35) с **устойчиво** диференчно уравнение, като апроксимацията ще направим в точката (x_i, t_j) , т.е. като начало ще построим **явна схема**.

Предвид факта, че разглеждаме двуслойна диференчна схема, е ясно, че за апроксимацията на производната по времето ще използваме формула за числено диференциране с разлика напред:

$$\frac{\partial u}{\partial t} = \frac{u_i^{j+1} - u_i^j}{\tau} + O(\tau).$$

За апроксимацията на производната по пространството обаче имаме три варианта – формулите за числено диференциране с централна разлика, с разлика напред и разлика назад. Да разгледаме какви апроксимации можем да получим, използвайки трите формули.

1. **Формула с централна разлика.** Имайки предвид, че тази формула има най-добър ред на апроксимация, бихме искали да започнем с нея. Тогава апроксимацията на (3.35) има вида

$$\frac{y_i^{j+1} - y_i^j}{\tau} + c \frac{y_{i+1}^j - y_{i-1}^j}{2h} = f_i^j,$$

което, записано в каноничен вид, е

$$y_i^{j+1} = y_i^j + \frac{c\tau}{2h}(y_{i-1}^j - y_{i+1}^j) + \tau f_i^j.$$

Очевидно е, че не могат да се изберат стойности на стъпките τ и h така, че условието за положителност на коефициентите да е изпълнено.

2. **Формула с разлика напред.** В този случай получаваме диференчното уравнение

$$\frac{y_i^{j+1} - y_i^j}{\tau} + c \frac{y_{i+1}^j - y_i^j}{h} = f_i^j,$$

което, записано в каноничен вид, е

$$y_i^{j+1} = \left(1 + \frac{c\tau}{h}\right) y_i^j - \frac{c\tau}{h} y_{i+1}^j + \tau f_i^j.$$

И в този случай се оказва, че не съществуват стойности на стъпките τ и h такива, че условието за положителност на коефициентите да е изпълнено.

3. **Формула с разлика назад.** В този случай получаваме диференчното уравнение

$$\frac{y_i^{j+1} - y_i^j}{\tau} + c \frac{y_i^j - y_{i-1}^j}{h} = f_i^j,$$

каноничният вид, на което е

$$y_i^{j+1} = \left(1 - \frac{c\tau}{h}\right) y_i^j + \frac{c\tau}{h} y_{i-1}^j + \tau f_i^j. \quad (3.36)$$

Сега вече е възможно да изберем стъпки, които да удовлетворяват условието за положителност на коефициентите. За тази цел е необходимо да бъде изпълнено неравенството

$$1 - \frac{c\tau}{h} \geq 0,$$

т.е. $\tau \leq h/c$. Последното условие е известно като условие на Courant–Friedrichs–Lewy, като по-често в литературата се среща абривиатурата CFL, а числото $c\tau/h$ е известно като число на Courant.

И така, за да получим устойчива диференчна схема, ще използваме апроксимацията (3.36), която има локална грешка $O(h + \tau)$.

N.B. 34

Може да се докаже, че **не съществуват монотонни (и устойчиви в $\|\cdot\|_{h,\infty}$ -норма) за уравнението на преноса, които имат втори ред на точност.**

N.B. 35

От друга страна, ако работим със схема от първи ред, за да получим голяма точност, трябва да работим с много малки стъпки, което често на практика е невъзможно. Това води до редица проблеми. При числено решаване на хиперболични уравнения се наблюдава силен ефект на числена дифузия (т.е. приближеното решение бързо се заглажда и съответно губи правилната си форма), както и загуба на енергия в системата (което не е физически правилно). Тези въпроси са илюстрирани в секция „Числени експерименти – практикум“.

Аналогично може да се изследват схеми с тегло (тук прочети 4.2 от [3] стр.127–131).

3.4.2 Схеми с втори ред на точност за уравнението на преноса, устойчиви в мрежова l_2 -норма

Както отбелязахме в предната секция, при използването на схемите от първи ред възникват редица проблеми (ефект на числена дифузия и загуба на енергията в системата). За да намалим влиянието на тези фактори, бихме искали да използваме схеми с по-висок ред на апроксимация.

Както казахме в предишната секция, за уравнението (3.35) не съществуват схеми, устойчиви в $\|\cdot\|_{h,\infty}$ -норма. Интуитивно това означава, че грешка, допусната на j -тия слой по времето, може да се натрупва и да стане по-голяма на следващия, т.е. $\|\varepsilon^{j+1}\|_{h,\infty} > \|\varepsilon^j\|_{h,\infty}$. Това, че максимумът на грешката може да расте обаче, не означава, че задължително това нарастване ще бъде драстично и методът ще бъде неизползваем. Да припомним, че по-слабо изискване е устойчивостта в мрежова l_2 норма. В някакъв смисъл това означава устойчивост на средната грешка. Максимумът може и да расте, но средната грешка остава контролирана.

Една от най-известните числени схеми за решаване на уравнението на преноса е схемата на Lax-Wendroff. Тя може да бъде изведена и като се използва подход, аналогичен на това, което сме разгледали досега в курса, но тук ще приложим едно различно извеждане на схемата върху хомогенното уравнение на преноса, т.е.

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0. \quad (3.37)$$

Развиваме в ред на Тейлър $u(x_i, t_{j+1})$:

$$u_i^{j+1} = u_i^j + \left. \frac{\partial u}{\partial t} \right|_i^j \tau + \left. \frac{\partial^2 u}{\partial t^2} \right|_i^j \frac{\tau^2}{2} + O(\tau^3). \quad (3.38)$$

От (3.37) получаваме

$$\frac{\partial u}{\partial t} = -c \frac{\partial u}{\partial x}.$$

Диференцирайки двете страни на последното равенство относно t , достигаме до следната връзка:

$$\frac{\partial^2 u}{\partial t^2} = -c \frac{\partial}{\partial t} \left(\frac{\partial u}{\partial x} \right) = -c \frac{\partial}{\partial x} \left(\frac{\partial u}{\partial t} \right) = c^2 \frac{\partial^2 u}{\partial x^2}.$$

Замествайки в (3.38), получаваме

$$u_i^{j+1} = u_i^j - c \left. \frac{\partial u}{\partial x} \right|_i^j \tau + \frac{c^2 \tau^2}{2} \left. \frac{\partial^2 u}{\partial x^2} \right|_i^j + O(\tau^3).$$

Пренебрегвайки остатъчния член и минавайки към приближеното решение, получаваме следното диференчно уравнение:

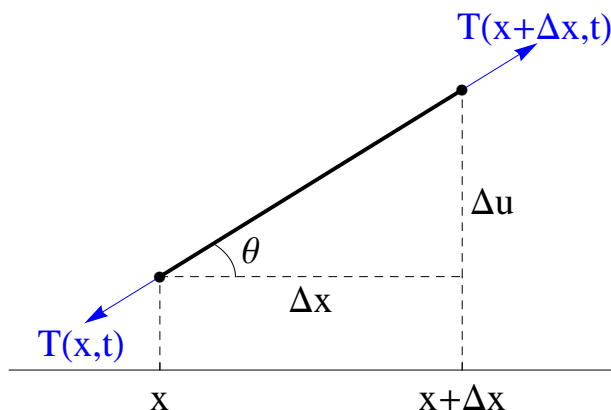
$$y_i^{j+1} = y_i^j - \frac{c\tau}{2h}(y_{i+1}^j - y_{i-1}^j) + \frac{c^2\tau^2}{2h^2}(y_{i+1}^j - 2y_i^j + y_{i-1}^j).$$

Последното има ЛГА $O(h^2 + \tau^2)$. Може да се покаже, че то е устойчиво в ℓ_2 норма, ако за числото на Courant е изпълнено $CourantNumber := \frac{c\tau}{h} \leq 1$ (тук прочети [3], втората половина от стр.131). За друга числена схема, Lax–Friedrichs, виж задача 8.

3.4.3 Диференчни схеми за уравнението на струната

Нека разгледаме еластична струна с дължина l , която вибрира под действието на напречна на струната сила. Ако дебелината на струната е пренебрежимо малка в сравнение с дължината l и вибрациите са само във вертикално направление, тогава положението на струната може да се моделира само чрез напречното преместване на струната $u(x, t)$ в позиция x и момент t .

Нека разгледаме малка част от струна, на която действат силата на опън $T(x, t)$ и вертикална сила $F(x, t)$.



От втория закон на Нютон следват следните уравнения

$$\begin{aligned} \cos \theta(x + \Delta x, t)T(x + \Delta x, t) - \cos \theta(x, t)T(x, t) &= 0, \\ \sin \theta(x + \Delta x, t)T(x + \Delta x, t) - \sin \theta(x, t)T(x, t) + \Delta x F(x, t) &= ma. \end{aligned} \quad (3.39)$$

Масата на струната може да се представи чрез плътността ρ и дължината l на струната:

$$m = \rho l = \rho \sqrt{(\Delta x)^2 + (\Delta u)^2}.$$

Замествайки последния израз в (3.39), получаваме

$$\cos \theta(x + \Delta x, t)T(x + \Delta x, t) - \cos \theta(x, t)T(x, t) = 0,$$

$$\sin \theta(x + \Delta x, t)T(x + \Delta x, t) - \sin \theta(x, t)T(x, t) + \Delta x F(x, t) = \rho \frac{\partial^2 u}{\partial t^2} \sqrt{(\Delta x)^2 + (\Delta u)^2}.$$

Делим двете страни на последните уравнения на Δx , пускаме Δx да клони към 0 и получаваме следното:

$$\begin{aligned} \frac{\partial}{\partial x} (T \cos \theta) &= 0, \\ \frac{\partial}{\partial x} (T \sin \theta) + F(x, t) &= \rho \frac{\partial^2 u}{\partial t^2} \sqrt{1 + \left(\frac{\partial u}{\partial x}\right)^2}. \end{aligned}$$

От първото уравнение следва, че $T \cos \theta = T_0(t)$. Заместваме получения резултат във второто уравнение:

$$\frac{\partial}{\partial x} (T_0(t) \operatorname{tg} \theta) + F(x, t) = \rho \frac{\partial^2 u}{\partial t^2} \sqrt{1 + \left(\frac{\partial u}{\partial x}\right)^2}.$$

Тъй като $\operatorname{tg} \theta = \frac{\partial u}{\partial x}$, то

$$T_0(t) \frac{\partial^2 u}{\partial x^2} + F(x, t) = \rho \frac{\partial^2 u}{\partial t^2} \sqrt{1 + \left(\frac{\partial u}{\partial x}\right)^2}.$$

Това уравнение е известно под името **нелинейно уравнение на струната**. То се опростява значително, когато ъгълът θ е близък до нула. Тогава е в сила $\frac{\partial u}{\partial x} = \operatorname{tg} \theta \approx 0$ и $T_0(t) = \operatorname{const}$, т.е.

$$T_0 \frac{\partial^2 u}{\partial x^2} + F(x, t) = \rho \frac{\partial^2 u}{\partial t^2}.$$

Като разделим двете страни на ρ и въведем нови променливи $c^2 = T_0/\rho$ и $F(x, t) = f(x, t)/\rho$, получаваме линейното уравнение на струната:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} + f(x, t),$$

За малки напречни премествания $u(x, t)$ ще е в сила **линеаризираното уравнение на струната**:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} + f(x, t), \quad (3.40)$$

където c е скоростта на разпространение на трептенията, а $f(x, t)$ е големината на силата, приложена в напречно направление.

Хиперболичното уравнение (3.40) е известно и под името **едномерно вълново уравнение**, тъй като то се използва за моделиране на разпространението на вълни. Терминът „вълна“ означава процесът на разпространение на трептения в дадена среда. Примери за вълни са водните и звуковите вълни, които са съответно смущения в нивото на водата и в налягането във флуида. Такива вълни, които представляват вибрации в дадена среда, се наричат механични вълни. Освен това съществуват и електромагнитни вълни, които са колебания в електричното и магнитното поле, например радиовълните, рентгеновите лъчи, ултравиолетовите лъчи и гама лъчите. Друг пример са гравитационните вълни, които представляват смущения в гравитационното поле.

Всички тези типове вълни могат да се опишат с вълновото уравнение (едномерно, двумерно или тримерно по пространството). Поради това вълновото уравнение намира многобройни приложения:

- Комуникационни технологии. Радиовълните се използват не само за излъчване на радио и телевизионни предавания, но също така чрез тях работят мобилните телефони и Wi-fi.
- Медицината. Ултравиолетовите лъчи се използват при стерилизирането на медицински инструменти. Рентгеновите лъчи се използват в рентгена, за да се диагностицират различни заболявания.
- Техника. Звуковите, микровълните и ултрачервените вълни намират приложение съответно при производство на музикални инструменти, микровълнови печки и камери за нощно виждане.
- Моделирането на сеизмичните вълни и на вълни цунами могат да спасят много човешки животи.

За да можем да използваме вълновото уравнение за описването на кой да е от горепосочените процеси, трябва да затворим системата, като наложим две гранични условия и две начални условия. Двете начални условия задават началното положение и началната скорост на вълната/струната, а граничните условия описват положението на краищата на вълната струната. Ние ще разгледаме струна с фиксирани краища и така ще достигнем до следната начално-гранична задача:

$$\begin{aligned} \frac{\partial^2 u}{\partial t^2} &= c^2 \frac{\partial^2 u}{\partial x^2} + f(x, t), \quad 0 < x \leq l, \quad 0 < t \leq T, \\ u(x, 0) &= u_0(x), \quad \frac{\partial u}{\partial t}(x, 0) = v_0(x), \quad 0 \leq x \leq l, \\ u(0, t) &= u_L(t), \quad 0 \leq t \leq T. \end{aligned}$$

Ще разгледаме трислойна диференчна схема върху шаблон “кръст” и ще я изследваме за устойчивост с принципа на максимума и чрез метода на хармониките. **Тук прочети 5.2, 5.3 от [3] на стр. 134–136.**

3.4.4 Числени експерименти - практикум

Пример 29. Разглеждаме задачата:

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0, \quad -10 < x < 10, \quad 0 < t \leq 5,$$

$$u(x, 0) = \begin{cases} 25, & x \leq 0 \\ 25 - 25x, & 0 \leq x \leq 1 \\ 0, & x \geq 1 \end{cases}$$

$$u(0, t) = 25.$$

Ще построим явна устойчива двуслойна схема за решаване на горната задача, като използваме апроксимацията (3.36) с $f_i^j = 0$ (вж. секция 3.4.2). Началното и лявото гранични условия се апроксимират тривиално:

$$\begin{aligned} y_i^0 &= u_0(x_i), \quad i = \overline{0, n}, \\ y_0^j &= 25, \quad j = \overline{0, m}. \end{aligned} \quad (3.41)$$

И така, получихме явна двуслойна диференчна схема с ЛГА $O(h + \tau)$, чийто каноничен вид се състои от диференчните уравнения (3.36) и (3.41) и която е устойчива при $\tau \leq h/c$.

Да разгледаме една примерна имплементация на диференчната схема:

```

Transport[l_, T_, h_, u0_, uL_, c_] := (
  τ = h/(2 c);
  n = Ceiling[l/h];
  m = Ceiling[T/τ];
  x = Table[-10 + (i-1)*h, {i, 1, n+1}];
  t = Table[(j-1)*τ, {j, 1, m+1}];
  y = Table[0, {n+1}, {m+1}];

  (*Initial condition*)
  For[i = 1, i ≤ n+1, i++,
    y[[i, 1]] = u0[x[[i]]];
  ];

  (*Iterate over time*)
  For[j = 1, j ≤ m, j++,
    (*Boundary condition*)
    y[[1, j+1]] = uL[t[[j]]];

    (*Internal values from the main PDE*)
    For[i = 2, i ≤ n, i++,
      y[[i, j+1]] = (1 - c*τ/h) y[[i, j]] + c*τ/h*y[[i-1, j]]
    ];
  ];

  {x, t, y}
)

```

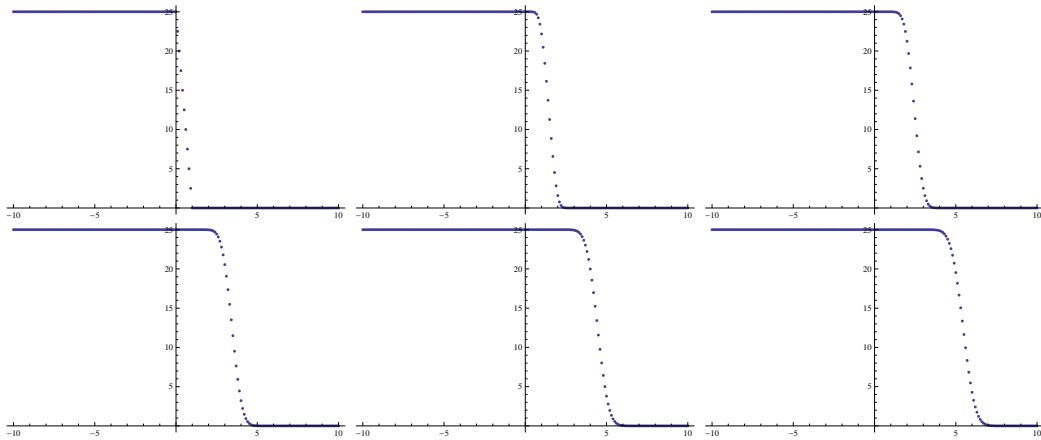
```

u0[x_] := Which[x ≤ 0, 25, x ≤ 1, 25 - 25 * x, True, 0]
uL[x_] := 25
apprSol = Transport[20, 5, 0.01, u0, uL, 1];
apprX = apprSol[[1]];
apprT = apprSol[[2]];
apprY = apprSol[[3]];

Manipulate[
  Show[
    Plot[u0[x - apprT[[j]]], {x, 0, 10}, PlotStyle → Red, PlotRange → {{0, 10}, {0, 25}},
    ListPlot[Transpose[{apprX, apprY[[All, j]]}], PlotRange → {{0, 10}, {0, 1.5}}]
  ],
  {j, 1, Length[apprT], 1}
]

```

Използвайки горния код, ще представим решението за $t = 0, 1, 2, 3, 4, 5$:



Решението запазва формата си и се придвижва надясно. Такива решения се наричат решения от тип “бягаща вълна”. Поведението на решението, разбира се, напълно съответства на физическия смисъл на задачата – топлината (или веществото и др.) се пренася, носено от флуид, с константна скорост c .

Както видяхме, при параболичните уравнения особеностите в началното условие не водят до съществени проблеми при численото решаване на дадена диференциална задача, предвид изглаждащия ефект на параболичния оператор. Нека видим дали това е така и при хиперболичните уравнения. Да разгледаме за целта следната задача:

Пример 30. Решете хиперболичната задача:

$$\begin{aligned}
 \frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} &= 0, \quad -10 < x < 10, \quad 0 < t \leq 5, \\
 u(x, 0) &= \begin{cases} 1, & 1/4 < x < 3/4, \\ 1 - |4x - 6|, & 5/4 < x < 7/4, \\ \cos^2[\pi(2x - 5)], & 9/4 < x < 11/4, \\ 0, & \text{иначе,} \end{cases} \\
 u(0, t) &= 0.
 \end{aligned} \tag{3.42}$$

Преизползваме функцията, която написяхме в Пример. 29:

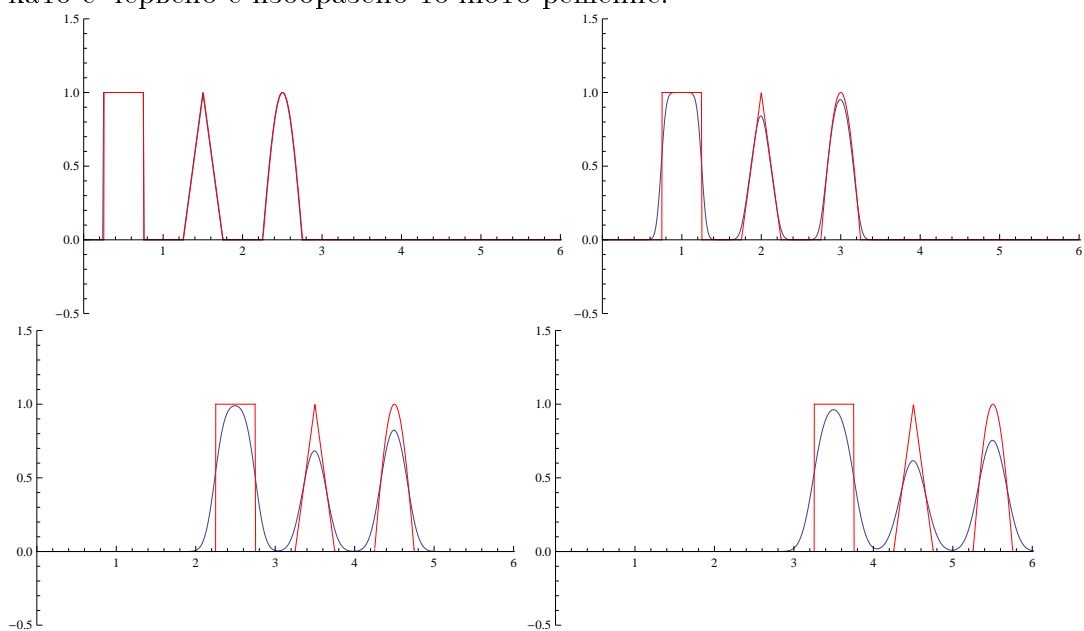
```

u0[x_] := Which[And[x ≥ 1/4, x ≤ 3/4], 1, And[x ≥ 5/4, x ≤ 7/4],
  1 - Abs[4*x - 6], And[x ≥ 9/4, x ≤ 11/4], (Cos[Pi (2 x - 5)])^2, True, 0]
uL[x_] := 0
apprSol = Transport[20, 5, 0.01, u0, uL, 1];
apprX = apprSol[[1]];
apprT = apprSol[[2]];
apprY = apprSol[[3]];

Manipulate[
  Show[
    Plot[u0[x - apprT[[j]]], {x, 0, 10}, PlotStyle → Red, PlotRange → {{0, 10}, {0, 1.5}},
    ListPlot[Transpose[{apprX, apprY[[All, j]]}]]
  ],
  {j, 1, Length[apprT], 1}

```

Резултатите за $j = 0, 100, 400, 600$ са изобразени на следващите графики, като с червено е изобразено точното решение.



Както можем да забележим, приближеното решение с течение на времето губи от първоначалната енергия и намалява, като се отдалечава от точното. Освен това, правоъгълната част от сигнала се заглажда и загубва първоначалната си форма. Този ефект е известен като “числена дифузия”.

Пример 31. За да избегнем съществена загуба на точност, която наблюдавахме в последния пример, бихме искали да използваме схеми с по-висок ред на апроксимация, т.е. $O(\tau^2 + h^2)$. Една такава схема за решаване на уравнението на преноса, която изведохме в секция 3.4.2, е схемата на Lax-Wendroff.

Нека сега решим задачата (3.42), като използваме схемата на Lax-Wendroff.


```

Transport2Order[l_, T_, h_, u0_, uL_, c_] := (
   $\tau = h / (2 c)$ ;
   $n = \text{Ceiling}[l/h]$ ;
   $m = \text{Ceiling}[T/\tau]$ ;
   $x = \text{Table}[-10 + (i - 1) * h, \{i, 1, n + 1\}]$ ;
   $t = \text{Table}[(j - 1) * \tau, \{j, 1, m + 1\}]$ ;
   $y = \text{Table}[0, \{n + 1\}, \{m + 1\}]$ ;

  (*Initial condition*)
  For[i = 1, i  $\leq$  n + 1, i++,
    y[[i, 1]] = u0[x[[i]]];
  ];

  (*Iterate over time*)
  For[j = 1, j  $\leq$  m, j++,
    (*Boundary condition*)
    y[[1, j + 1]] = uL[t[[j]]];

    (*Internal values from the main PDE*)
    For[i = 2, i  $\leq$  n, i++,

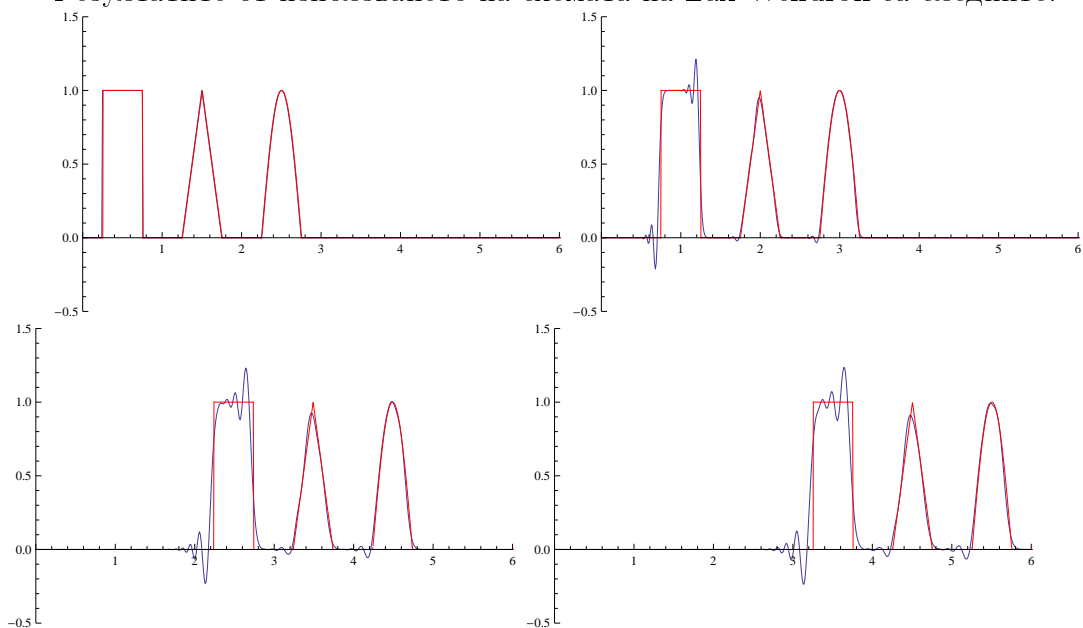
       $y[[i, j + 1]] = c \tau / (2 h) (1 + c * \tau / h) * y[[i - 1, j]] + (1 - c * \tau / h) (1 + c * \tau / h) y[[i, j]] + c \tau / (2 h) (c * \tau / h - 1) * y[[i + 1, j]]$ ;

    ];

    {x, t, y}
  )
)

```

Резултатите от използването на схемата на Lax-Wendroff са следните:



Както виждаме, при нея приближеното решение не се отдалечава значително от точното за разглеждания интервал от време, т.е. този метод запазва много по-добре енергията в системата. Отново се наблюдава числена дифузия, особено в точките на прекъсване, но този ефект е значително по-малък. От друга страна обаче, схемата губи монотонност и се наблюдават нефизични осцилации около особеностите на решението. Причината за това е, че тя вече не е устойчива в мрежова $\|\cdot\|_{h,\infty}$ -норма, т.е. няма контрол върху максимума на

грешката, а само върху “общото” ѝ поведение. Съществуват различни подходи за намаляването на този ефект, но този въпрос излиза извън рамките на настоящия курс.

N.B. 36

И така, при решаването на хиперболични уравнения трябва да се имат предвид съответните особености и трудности. Това е тема, която и днес продължава да бъде активна област на изследване.

Схемите от първи ред могат да бъдат устойчиви в $\|\cdot\|_{h,\infty}$ -норма и монотонни, но не запазват добре енергията в системата и се наблюдава силна числена дифузия, която води до разрушаване на структурата на решението. Схемите от втори ред могат да бъдат устойчиви само в $\|\cdot\|_{h,2}$ -норма. Те запазват много по-добре енергията и структурата на решението, но при тях се наблюдават нефизични осцилации в околност на особености в решението, за които трябва да се полагат допълнителни грижи.

3.5 Диференчни схеми за решаване на нестационарни задачи – обобщение

3.6 Диференчни схеми за решаване на стационарни задачи

Тук прочети параграф 2.3 на стр. 55-57 и параграфи 2.1, 2.2 на стр. 108–110 от [3].

3.7 Допълнителни задачи

Задача 7. Свойствата на параболичния оператор правят параболичните уравнения подходящи в случаите, когато трябва да се заглади дадена функция. Те намират приложение например в обезшумяването на изображения, сигнали и др.

Нека имаме сигнал от даден сензор, който се описва като $u_0(x) = \sin x + 0.125 \cos(50x)$. Ще използваме следната параболична задача, за да го обезшумим:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad 0 < x < 2\pi, \quad 0 < t \leq 0.03,$$

$$u(x, 0) = \sin x + 0.125 \cos(50x)$$

$$\left. \frac{\partial u}{\partial x} \right|_{x=0} + u(0, t) = 2,$$

$$\left. \frac{\partial u}{\partial x} \right|_{x=2\pi} = 0.$$

За целта:

- Съставете диференчна схема с ЛГА $O(h^2 + \tau)$.



Анимирайте поведението на сигнала.

Забележка. Разбира се, на практика сигналът не се представя с аналитична функция, подобна на дадената, но нашата цел тук е да илюстрираме идеята. Тя може да бъде приложена по абсолютно същия начин за реални сензорни данни/изображения и т.н.

Задача 8. Разглеждаме уравнението на преноса при $f \equiv 0$. За него е построена следната схема (схема на Lax–Friedrichs):

$$y_i^{j+1} = \frac{1}{2} \left(1 - \frac{c\tau}{h} \right) y_{i+1}^j + \frac{1}{2} y_{i-1}^j.$$

- Какъв е шаблонът, върху който е построена схемата? Явна или неявна е тази схема?
- Изведете условия за устойчивост в $\| \cdot \|_{h,\infty}$ и $\| \cdot \|_{h,2}$ норми.

Задача 9. Разглеждаме схемата на Dufort–Frankel за решаване на уравнението на дифузията (при $f \equiv 0$):

$$\frac{y_{i+1}^j - y_i^{j+1} - y_i^{j-1} + y_{i-1}^j}{h^2} = \frac{y_i^{j+1} - y_i^{j-1}}{2\tau}.$$

- Какъв е шаблонът, върху който е построена схемата? Какви са границите за i, j ?
- Явна или неявна е схемата?
- Покажете, че ЛГА е $O(h^2 + \tau^2 + (\tau/h)^2)$.
- Изследвайте за устойчивост в $\| \cdot \|_{h,\infty}$ и $\| \cdot \|_{h,2}$ норми.

Задача 10. Дадена е диференциалната задача

$$\begin{aligned} \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} &= \sin(xt), \quad 0 < x < 1, \quad 0 < t \leq 0.1, \\ u(x, 0) &= x^2 - x, \\ \frac{\partial u}{\partial x}(0, t) + u(0, t) &= -1 \\ \frac{\partial u}{\partial x}(1, t) &= 1. \end{aligned}$$

- Да се построи устойчива явна диференчна схема с ЛГА $O(h^2 + \tau)$;
- Да се построи диференчна схема с тегло с ЛГА $O(h^2 + \tau)$;



Да се реализират така построените схеми.

Задача 11. Дадена е струна с дължина 1, която е закрепена неподвижно в двата си края. Началното положение на струната се описва от функцията $x(x-1)$, а началната скорост на струната е 0. Тези условия задават следната диференциална задача:

$$\begin{aligned}\frac{\partial^2 u}{\partial t^2} &= \frac{\partial^2 u}{\partial x^2}, \quad 0 < x \leq 1, \quad 0 < t \leq 10, \\ u(x, 0) &= u_0(x) = x(x-1) \\ \left. \frac{\partial u}{\partial t} \right|_{t=0} &= 0 \\ u(0, t) &= 0 \\ u(1, t) &= 0.\end{aligned}$$

- Съставете диференчна схема.



Анимирайте поведението на струната.

Глава 4

Метод на крайните елементи

4.1 По части полиноми на една променлива. Интерполация, L_2 -проекция, априорни оценки на грешката.

В МКЕ търсим най-доброто приближение от определен вид (по части полином) на решението на дадена диференциална задача. Преди да преминем към този въпрос, нека разгледаме въпроса за апроксимацията на дадена функция (нека си мислим засега, че тя е известна). Идеите, които ще разгледаме, са в основата на МКЕ и ще ги срещнем многократно в курса в различни техни модификации.

Съществуват две основни идеи, които са разглеждани и в курса “Числени методи на анализа” – **интерполация** и търсене на **най-добро приближение по отношение на дадена норма**. Ще ги разгледаме последователно, тъй като и двете имат отношение към по-нататъшното изложение.

Функциите, които се използват за приближаване в настоящия курс, ще бъдат по части полиноми. От една страна, те са достатъчно прости, за да работим лесно с тях, а от друга страна, дават достатъчна гъвкавост, за да можем да приближаваме с тях сложни функции.

Да формулираме следната, поставена възможно най-общо, задача:

Дадена е функция $u \in V$. Търсим функция $u_h \in V_h$, по части полином, която е “близо” до u .

Нека засега си мислим, че функцията u е достатъчно гладка (например безкрайно гладка, т.е. $V \equiv C^\infty[a, b]$). По-късно ще уточним този въпрос. Формулираната задача е основна в теорията на апроксимациите. Искаме да приближим дадена (в някакъв смисъл сложна) функция u от безкрайномерното пространство V с нещо по-просто, в подходящо избрано крайномерно подпространство $V_h \subset V$. Предимството на това да работим в крайномерното пространство е, че знаем вида на всички функции в него. Те могат да се представят като $\sum_{i=0}^n a_i \varphi_i(x)$, където $\{\varphi_0(x), \dots, \varphi_n(x)\}$ е даден базис на пространството. С други думи, всяка функция може да бъде определена чрез избора на $n + 1$ числа a_0, \dots, a_n и въпросът за намирането на конкретна функция се свежда до намирането на $(a_0, \dots, a_n) \in \mathbb{R}^{n+1}$. Вземайки предвид последното, първо трябва да си изясним структурата на V_h и да изберем подходящ базис, с който да работим.

Ще изложим идеите върху пространството на функциите, които са по части линейни полиноми. Всички идеи могат непосредствено да се обобщят за полиноми от по-висока степен. Работата с такива има редица предимства при прилагане на МКЕ, но с цел опростяване на изложението засега ще се концентрираме единствено върху по части линейни функции, а по-нататък ще разгледаме различните възможности в това отношение.

4.1.1 Пространство на линейните полиноми P_1 . Интерполация.

За да разгледаме въпроса за работата с по части линейни функции, нека първо разгледаме пространството на линейните полиноми

$$P_1 := \{p(x) = a_0 + a_1x : (a_0, a_1) \in \mathbb{R}^2\}.$$

Най-простият базис на P_1 е $\{1, x\}$. Той обаче често не е най-удобният за работа. Както знаем, всяка права може да бъде определена от произволни две точки. Нека изберем например точките (x_0, q_0) и (x_1, q_1) . Полиномът, който минава през тези две точки, удовлетворява системата

$$\begin{aligned} p(x_0) &= a_0 + a_1x_0 = q_0, \\ p(x_1) &= a_0 + a_1x_1 = q_1 \end{aligned}$$

или, записано във векторно-матрична форма,

$$\begin{bmatrix} 1 & x_0 \\ 1 & x_1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \end{bmatrix}.$$

Последната система има матрица на Вандермонд, за която е известно, че няма добри свойства при числени пресмятания, особено при по-големи размерности.

Значително по-удобен за работа ще бъде базисът $\{\varphi_0(x), \varphi_1(x)\}$ такъв, че матрицата на системата

$$\begin{bmatrix} \varphi_0(x_0) & \varphi_1(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \end{bmatrix}.$$

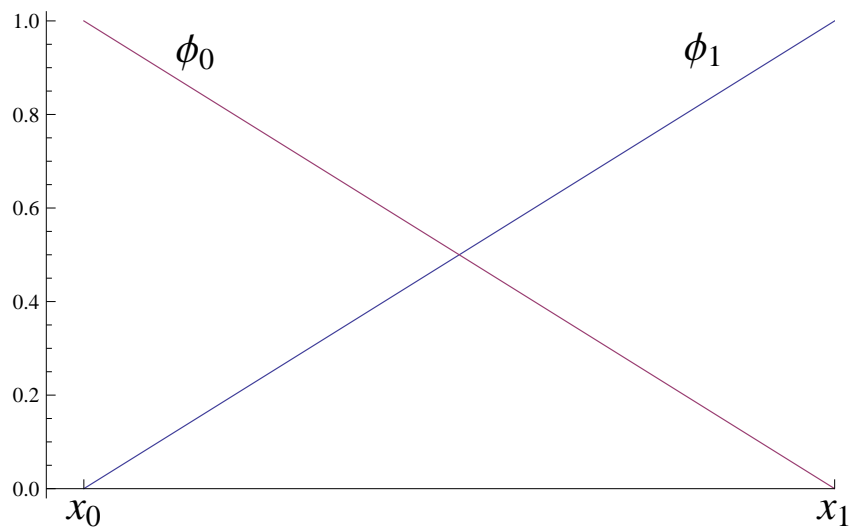
да е диагонална. Това ни дава основание да изберем $\varphi_i(x)$, $i = 0, 1$ така, че

$$\varphi_i(x_j) = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

Такъв базис ще наричаме **интерполационен** или **възлов базис**. В случая възловият базис се задава от базисните полиноми на Лагранж:

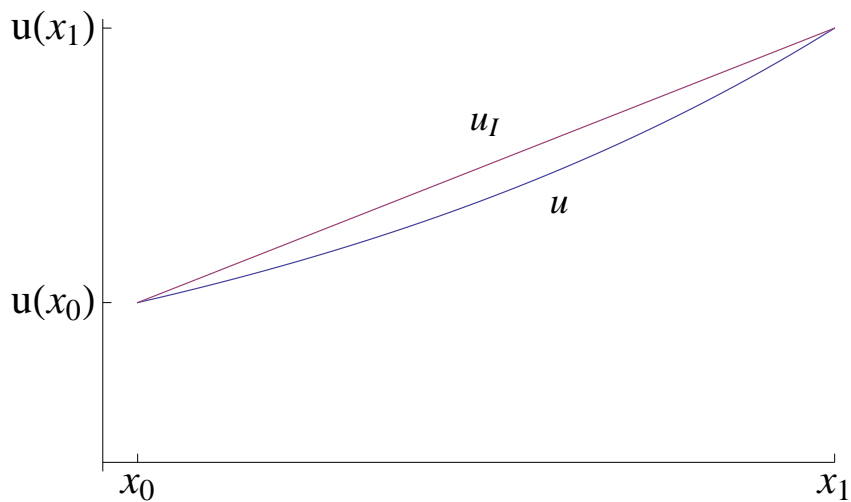
$$\varphi_0(x) = \frac{x - x_1}{x_0 - x_1}, \quad \varphi_1(x) = \frac{x - x_0}{x_1 - x_0}.$$

Графиките на двете базисни функции изглеждат по следния начин:



Тогава, ако има дадена функцията $u(x)$, намираме

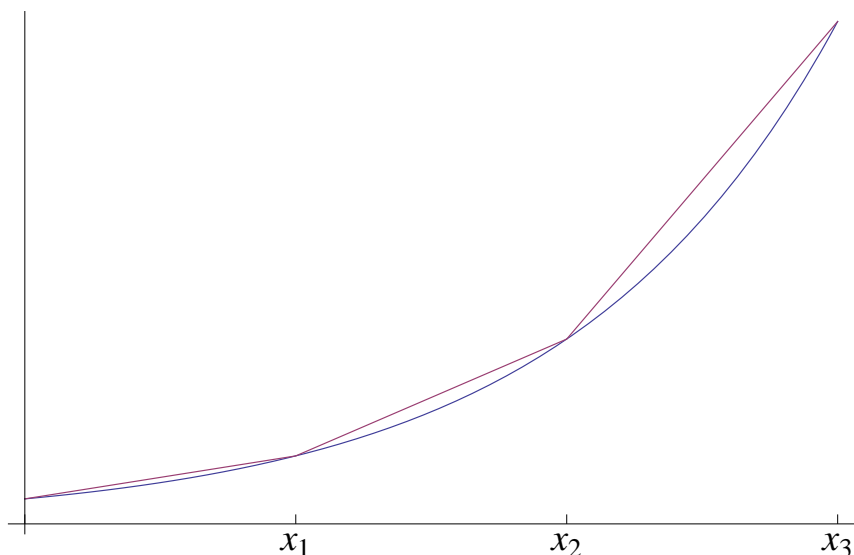
$$u_I(x) = u(x_0)\varphi_0(x) + u(x_1)\varphi_1(x).$$



4.1.2 Пространство на по части линейните полиноми V_h . Интерполация.

Разбира се, ако искаме да приближим добре функция със сложно поведение в голям интервал, това няма как да стане с помощта на линейна функция.

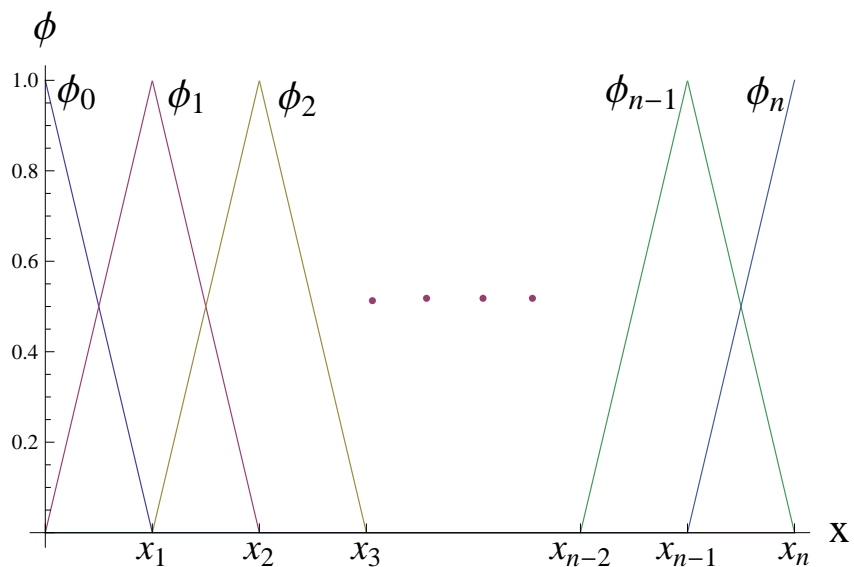
Стандартна идея в числените методи е да разделим интервала на подинтервали и да приближаваме във всеки подинтервал поотделно, например:



Това ни дава основание да разглеждаме пространството на функциите, които са по части полиноми (в случая линейни). Нека сме избрали възлите $x_0 < x_1 < \dots < x_n$ и нека $h_i = x_i - x_{i-1}$, $I_i := [x_{i-1}, x_i]$, $i = \overline{1, n}$. Подинтервалите I_i ще наричаме **елементи**. Дефинираме

$$V_h := \{p(x) \in C[a, b] : p(x) \in P_1 \text{ за } x \in I_i, i \in \overline{1, n}\}.$$

Възловият базис за това пространство трябва да изглежда по следния начин:



Това са т.нар. функции-“колибки”. Можем да ги дефинираме аналитично посредством

$$\varphi_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}}, & x \in I_i, \\ \frac{x_{i+1} - x}{x_{i+1} - x_i}, & x \in I_{i+1}, \\ 0, & \text{otherwise.} \end{cases}$$

Очевидно дефиницията трябва да се модифицира тривиално за първата и последната базисни функции.

Важно свойство на функциите $\varphi_i(x)$ е, че те имат краен носител, т.е. стойността им е различна от нула само върху два подинтервала:

$$\text{supp } \varphi_i(x) = [x_{i-1}, x_i] \cup [x_i, x_{i+1}], \quad i = \overline{1, n-1},$$

с изключение на първата и последната базисна функция, чийто носител е съответно първият и последният интервал.

Така, интерполантата на функцията $u(x)$ във възлите x_0, \dots, x_n е

$$u_I(x) = \sum_{i=0}^n u(x_i) \varphi_i(x).$$

4.1.3 L_2 -проекция

Теорията на интерполацията е фундаментална за МКЕ от теоретична гледна точка. Идеята на самия метод обаче е много повече свързана с другия основен подход при приближаването на функции – търсенето на най-добро приближение по отношение на дадена норма. В частност в настоящия параграф ще се интересуваме от най-добри приближения по отношение на L_2 -нормата, тъй като те могат да бъдат намерени алгоритмично. Следователно дефинираме следната задача:

По дадена функция $u \in L_2(I)$ да се намери $u_h \in V_h$ така, че

$$\|u - u_h\|_{L_2(I)} \rightarrow \min_{u_h \in V_h}.$$

Да припомним, че пространството $L_2(I)$ е дефинирано като

$$L_2(I) := \left\{ u : \int_I u^2 dx < \infty \right\}$$

и е снабдено с нормата

$$\|u\|_{L_2(I)}^2 := \int_I u^2 dx.$$

Съществен факт за пространството $L_2(I)$ е, че L_2 -нормата може да бъде дефинирана посредством скаларното произведение

$$(u, v) := \int_I uv dx,$$

т.е. $\|u\|_{L_2(I)}^2 := (u, u)$.

Дефиниция 13

Казваме, че едно пълно линейно пространство е Хилбертово, ако в него е въведено скаларно произведение.

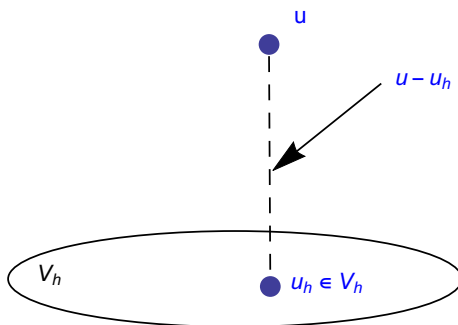
Предимството на работата в Хилбертови пространства е, че скаларното произведение въвежда геометрия в пространството. В частност имаме понятия за ъгли, ортогоналност и проекция. Като използваме този факт, можем да подходим към формулираната задача по следния начин.

Естественият “кандидат” за най-добро приближение на u от V_h е ортогоналната проекция на u , т.е. функцията $u_h \in V_h$, за която

$$u - u_h \perp v, \quad \forall v \in V_h,$$

или, което е същото,

$$(u - u_h, v) = 0, \quad \forall v \in V_h. \quad (4.1)$$



Може да се докаже, че ортогоналната проекция действително е най-доброто приближение (този резултат е доказван в курса “Числени методи на анализа”).

Така, вече сме готови да изведем алгоритъм за намирането на u_h . Задачата (4.1) е еквивалентна на задачата за намирането на $u_h \in V_h$, което

$$(u_h, v) = (u, v), \quad \forall v \in V_h.$$

Използвайки факта, че работим в крайномерно пространство, последното ще е изпълнено, ако

$$(u_h, \varphi_i) = (u, \varphi_i), \quad i = \overline{0, n}.$$

Сега, като използваме това, че знаем вида на $u_h = \sum_{j=0}^n q_j \varphi_j(x)$, където $\varphi_j(x)$ са функциите-“колибка” и линейността на скаларното произведение, окончателно получаваме линейна алгебрична система за неизвестните коефициенти:

$$\sum_{j=0}^n q_j (\varphi_j, \varphi_i) = (u, \varphi_i), \quad i = \overline{0, n}$$

или, записано във векторно-матрична форма

$$\begin{bmatrix} (\varphi_0, \varphi_0) & (\varphi_0, \varphi_1) & \cdots & (\varphi_0, \varphi_n) \\ \vdots & \vdots & \ddots & \vdots \\ (\varphi_0, \varphi_n) & (\varphi_1, \varphi_n) & \cdots & (\varphi_n, \varphi_n) \end{bmatrix} \begin{bmatrix} q_0 \\ \vdots \\ q_n \end{bmatrix} = \begin{bmatrix} (u, \varphi_0) \\ \vdots \\ (u, \varphi_n) \end{bmatrix}. \quad (4.2)$$

Това впрочем е най-общият вид на системата, която се получава, когато търсим ортогоналната проекция на дадена функция u в подпространството, породено от $\varphi_0, \dots, \varphi_n$, по отношение на скалярно произведение, въведено в пространството. В разглеждания от нас случай, т.е. когато търсим L_2 -проекция във V_h , получаваме

$$\begin{bmatrix} \int_I \varphi_0^2 dx & \int_I \varphi_1 \varphi_0 dx & \cdots & \int_I \varphi_n \varphi_0 dx \\ \vdots & \vdots & \ddots & \vdots \\ \int_I \varphi_0 \varphi_n dx & \int_I \varphi_1 \varphi_n dx & \cdots & \int_I \varphi_n^2 dx \end{bmatrix} \begin{bmatrix} q_0 \\ \vdots \\ q_n \end{bmatrix} = \begin{bmatrix} \int_I u \varphi_0 dx \\ \vdots \\ \int_I u \varphi_n dx \end{bmatrix}.$$

Накратко можем да запишем системата като

$$M\mathbf{q} = \mathbf{b},$$

където M е т.нар. глобална матрица на масата, а \mathbf{b} е глобалният вектор на натоварванията.

Следователно, за да намерим неизвестните коефициенти, е необходимо да асемблираме матрицата M и векторът \mathbf{b} и да решим системата. Ще използваме факта, че базисните функции на V_h имат краен носител, за да се пресметнат M и \mathbf{b} ефективно. Тъй като φ_i има ненулеви стойности само върху I_i и I_{i+1} , е смислено да запишем M като сума на матрици, съдържащи интеграли върху всеки от елементите поотделно

$$\begin{aligned} M &= \sum_{i=1}^n \begin{bmatrix} \int_{I_i} \varphi_0^2 dx & \int_{I_i} \varphi_1 \varphi_0 dx & \cdots & \int_{I_i} \varphi_n \varphi_0 dx \\ \vdots & \vdots & \ddots & \vdots \\ \int_{I_i} \varphi_0 \varphi_n dx & \int_{I_i} \varphi_1 \varphi_n dx & \cdots & \int_{I_i} \varphi_n^2 dx \end{bmatrix} \\ &= \begin{bmatrix} \int_{I_0} \varphi_0^2 dx & \int_{I_0} \varphi_1 \varphi_0 dx & 0 & \cdots & 0 & 0 \\ \int_{I_0} \varphi_0 \varphi_1 dx & \int_{I_0} \varphi_1^2 dx & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \int_{I_1} \varphi_1^2 dx & \int_{I_1} \varphi_1 \varphi_2 dx & \cdots & 0 & 0 \\ 0 & \int_{I_1} \varphi_2 \varphi_1 dx & \int_{I_1} \varphi_2^2 dx & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \\ &+ \cdots + \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \int_{I_n} \varphi_{n-1}^2 dx & \int_{I_n} \varphi_{n-1} \varphi_n dx \\ 0 & 0 & 0 & \cdots & \int_{I_n} \varphi_n \varphi_{n-1} dx & \int_{I_n} \varphi_n^2 dx \end{bmatrix}. \end{aligned}$$

Разбира се, на практика би било абсолютно непрактично да запазим в паметта на компютъра всички тези матрици, след като повечето им елементи са нули. Тогава можем да пресметнем само ненулевите блокове с размер 2×2 и да ги

поставим “на правилните места” в M . Този процес се нарича **асемблиране на глобалната матрица на масата**.

Елементните 2×2 матрици на масата могат да бъдат дефинирани, както следва:

$$m_i = \begin{bmatrix} \int_{I_i} \varphi_{i-1}^2 dx & \int_{I_i} \varphi_{i-1} \varphi_i dx \\ \int_{I_i} \varphi_i \varphi_{i-1} dx & \int_{I_i} \varphi_i^2 dx \end{bmatrix} = \frac{h_i}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}.$$

Последното равенство може да бъде лесно проверено и го оставяме като упражнение.

Можем да подходим аналогично за вектора на натоварванията \mathbf{b} , като въведем

$$\mathbf{b}_i = \begin{bmatrix} \int_{I_i} \varphi_{i-1} u dx \\ \int_{I_i} \varphi_i u dx \end{bmatrix}.$$

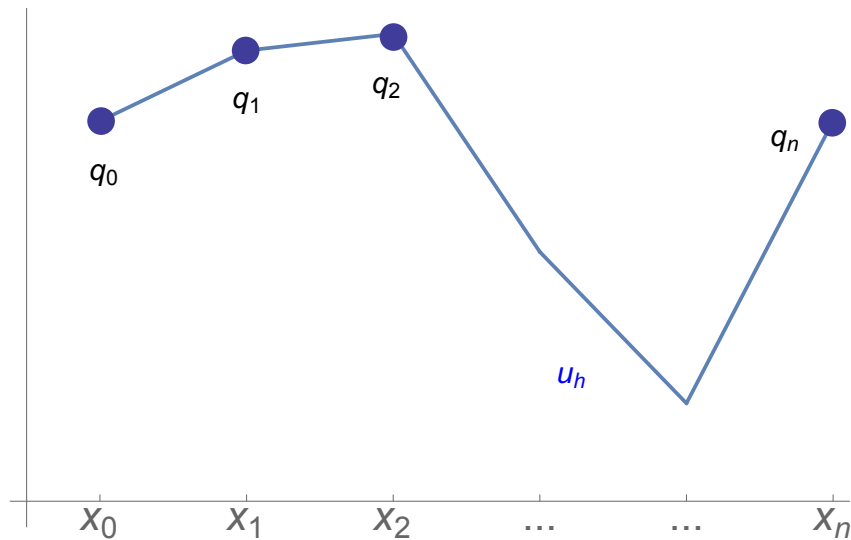
И така, дефинираме следния алгоритъм за получаването на най-доброто приближение във вид на по части полином на дадена функция $u \in L_2(I)$ по отношение на L_2 -нормата.

Алгоритъм 9: Намиране на L_2 -проекция

1. Дискретизираме интервала I , като въвеждаме мрежата $x_0 < x_1 < \dots < x_n$, $I_i := [x_{i-1}, x_i]$, $h_i := x_i - x_{i-1}$.
2. Дефинираме интерполационния базис $\varphi_i(x)$, $i = \overline{0, n}$, функции “колибка”.
3. Имплементираме функции за пресмятането на елементните матрици на масата M_i и елементните вектори на натоварванията \mathbf{b}_i .
4. Асемблираме глобалната матрица на масата M и глобалния вектор на натоварванията \mathbf{b} .
5. Решаваме линейната алгебрична система $M\mathbf{q} = \mathbf{b}$.
6. Използваме решението \mathbf{q} , за да получим

$$u_h = \sum_{i=0}^n q_i \varphi_i(x).$$

Да отбележим отново, че $q_i = u_h(x_i)$ са **точно стойностите на u_h във възлите от мрежата**.



4.2 МКЕ за 1D задачи с хомогенни гранични условия на Дирихле. Вариационна формулировка. Задача на Ритц–Гальоркин. Дискретизация и извеждане на линейна алгебрична система. Прилагане на метода при по-общи гранични условия.

4.2.1 Обща идея на метода

Вече сме готови да приложим идеите, разгледани в предишната секция, за решаването на ДУ. Ще изложим идеята върху съвсем прост пример. Разглеждаме диференциалната задача

$$\begin{aligned} -u''(x) &= f, \quad x \in (0, L), \\ u(0) &= u(L) = 0, \end{aligned} \tag{D}$$

където $f \in L_2(I)$ е дадена функция, $I := [0, L]$. Казано иначе, търсим функцията u от пространството

$$\mathcal{D} = \{v \in C^2(0, L) \cap C[0, L] : v(0) = v(L) = 0\},$$

която удовлетворява диференциалното уравнение. Ще наричаме такава функция **класическо решение** на диференциалната задача (D).

Нашият подход ще бъде да получим интегрална задача, която има същото решение, тъй като работата с интеграли има съществени предимства при работата със сложни геометрии, дискретизирани с мрежа от произволни фигури (напр. триъгълна мрежа). В тези случаи класическите диференчни методи имат съществени трудности. От друга страна, няма разлика дали ще пресмятаме интеграли върху правоъгълници (от правоъгълна мрежа, каквато изключително използвахме досега) или върху други области.

Ще реализираме последната цел, като използваме идеята, която имахме за намиране на L_2 -проекцията. С други думи, ще поискаме грешката (т.е. разликата между лявата и дясната страна в диференциалното уравнение $-u'' - f$) да бъде ортогонална “на всяко v ”. Реализираме тази идея по следния начин.

Вземаме произволна функция v , която ще наричаме **тестова функция** и умножаваме скалярно двете страни на уравнението с v . Така получаваме

$$(-u'', v) = (f, v)$$

или, което е същото

$$\int_I -u'' v dx = \int_I f v dx.$$

Да разгледаме сега интеграла в лявата страна на последното уравнение и да интегрираме по части, с което целим следните две неща:

- да направим лявата страна “по-симетрична”;
- да намалим реда на производните в уравнението и така да намалим изискванията за гладкост върху допустимите функции u .

Така, за лявата страна получаваме

$$\int_I -u'' v dx = \int_I u' v' dx - u'(1)v(1) + u'(0)v(0).$$

За да опростим израза допълнително, ще изискаме тестовите функции също да удовлетворяват граничните условия, т.е. $v(0) = v(1) = 0$. Това води до удобно опростяване, но е и съвсем естествено да посикаме тестовата функция v и неизвестната функция u да бъдат от едно и също пространство (и следователно да изпълняват еднакви гранични условия¹).

Така получаваме следната интегрална задача, която ще наричаме **вариационна формулировка** или **слаба форма** (на Галъоркин):

Да се намери $u \in V$ така, че

$$\int_I u' v' dx = \int_I f v dx, \quad \forall v \in V, \tag{V}$$

¹Всъщност това е доста “тънък” момент, но в настоящия курс ще се ограничим с този коментар

Ще дефинираме пространството V да бъде най-голямото функционално пространство, за което задачата има смисъл, т.е.

$$V = \left\{ v : v' \text{ съществува (в слаб смисъл)}, \int_I v^2 < \infty, \int_I v'^2 < \infty, v(0) = v(L) = 0 \right\} =: H_0^1.$$

Забележка. Преди да продължим, е необходимо да направим няколко коментара:

- Пространството H_0^1 е пример за Соболево пространство. Това са пространства, в които функциите имат (слаби) производни до определен ред, които са интегрируеми. Ние няма да използваме тук понятието слаба (обобщена) производна и затова няма да го конкретизираме. За целите на курса е достатъчно да си мислим, че функциите могат да бъдат еднократно диференцирани навсякъде, където това е необходимо.
- Очевидно пространството $V \equiv H_0^1$ е “по-голямо” от \mathcal{D} . От една страна, това е добре – ако (D) има решение, то задължително се съдържа в H_0^1 и следователно е решение на (V). От друга страна обаче, в общия случай задачата (V) не е еквивалентна на (D). Може да се докаже обаче, че при определени условия (V) има единствено решение. Тогава, ако (D) има решение, двете задачи действително са еквивалентни. В противен случай решението на (V) се нарича **слабо решение**.

За съжаление, в общия случай задачата (V) не може да бъде решена. Следователно се налага да търсим приближено решение в крайномерно подпространство на H_0^1 (задача, която може да бъде решена). В частност ще търсим по части линейна функция, която се нулира по границите, т.е. ще решим т.нар. задача на Ритц–Галъоркин (Ritz–Galerkin):

Търсим $u_h \in V_{h,0} \subset H_0^1$ така, че

$$a(u_h, v) = F(v), \quad \forall v \in V_{h,0}, \quad (\text{R.-G.})$$

където

$$V_{h,0} := \{v \in V_h : v(0) = v(L) = 0\} = \text{span}(\varphi_1, \dots, \varphi_{n-1}).$$

Вземайки предвид, че вече работим в крайномерното пространство $V_{h,0}$, е достатъчно да бъде удовлетворено

$$a(u_h, \varphi_j) = F(\varphi_j), \quad j = \overline{1, n-1}.$$

След това, като използваме общия вид на приближеното решение, $u_h(x) = \sum_{i=1}^{n-1} q_i \varphi_i(x)$, получаваме следната линейна алгебрична система за коефициентите q_i :

$$\begin{bmatrix} a(\varphi_1, \varphi_1) & a(\varphi_1, \varphi_2) & \cdots & a(\varphi_1, \varphi_{n-1}) \\ \vdots & \vdots & \ddots & \vdots \\ a(\varphi_1, \varphi_{n-1}) & a(\varphi_2, \varphi_{n-1}) & \cdots & a(\varphi_{n-1}, \varphi_{n-1}) \end{bmatrix} \begin{bmatrix} q_1 \\ \vdots \\ q_{n-1} \end{bmatrix} = \begin{bmatrix} (u, \varphi_1) \\ \vdots \\ (u, \varphi_{n-1}) \end{bmatrix}. \quad (4.3)$$

Ще запишем последната система компактно като $M\mathbf{q} = \mathbf{b}$ където M е т.нар. **матрица на коравина**

$$M = \begin{bmatrix} \int_I \varphi_1'^2 dx & \cdots & \int_I \varphi_1' \varphi_{n-1}' dx \\ \vdots & \ddots & \vdots \\ \int_I \varphi_1' \varphi_{n-1}' dx & \cdots & \int_I \varphi_{n-1}'^2 dx \end{bmatrix}.$$

Можем да асемблираме матрицата на коравина по абсолютно същия начин, по който асемблирахме матрицата на масата – като представим интегралите като сума от интеграли върху всеки от елементите.

Решавайки системата по отношение на \mathbf{q} , получаваме желаното решение.

4.3 Прилагане на МКЕ при нехомогенни условия на Дирихле и условия на Нойман

4.4 Идея на МКЕ в 2D

Достойнствата на МКЕ се проявяват най-вече при 2D и 3D задачи в области със сложна геометрия. Да разгледаме за пример следната дискретизация на Черно море.



За разлика от диференчните методи, при които е необходима работата върху правоъгълна мрежа, МКЕ може да се приложи върху произволни мрежи, тъй като, както видяхме, за асемблиране на линейната алгебрична система е необходимо пресмятането на интеграли върху всеки от елементите. Впрочем, както ще видим след малко, идеята се пренася тривиално и в 2D и 3D.

За да може да стане това, е необходимо първо да се опише мрежата. За тази цел на всеки елемент (т.е. всеки триъгълник) и всеки възел (т.е. всеки връх на триъгълник) се съпоставя номер, както е показано по-долу:

може да бъде запазена в паметта по следния начин:

$$nodes = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 2 & 0 \\ 0 & 1 \\ 1 & 1 \\ 2 & 1 \\ 0 & 2 \\ 1 & 2 \end{bmatrix}, \quad elements = \begin{bmatrix} 1 & 2 & 4 \\ 2 & 5 & 4 \\ 2 & 3 & 5 \\ 3 & 6 & 5 \\ 4 & 5 & 7 \\ 5 & 8 & 7 \end{bmatrix}.$$

Като използваме въведената номерация на възлите и елементите, всеки ред от матриците отговоря на съответен възел (неговите координати) или елемент (индексите на възлите, които определят съответния елемент).

Върху така въведената дискретизация, можем непосредствено да обобщим идеята, която изложихме в предходната секция. Да разгледаме задачата

$$\begin{aligned} -\Delta u &= f, \text{ в } \Omega, \\ u &= 0, \text{ върху } \partial\Omega, \end{aligned} \quad (D)$$

където $\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$ е операторът на Лаплас. Т.е. разглеждаме уравнение на Поасон. За да получим вариационната формулировка, умножаваме двете страни скаларно с v и получаваме

$$(-\Delta u, v) = (f, v).$$

Интегрираме по части лявата страна и получаваме последователно

$$\begin{aligned} (-\Delta u, v) &= \iint_{\Omega} -\Delta u v d\Omega \\ &= \iint_{\Omega} \nabla u \cdot \nabla v d\Omega - \int_{\partial\Omega} (\nabla u \cdot \mathbf{n}) v ds. \end{aligned} \quad \text{0, предполагайки } v \in H_0^1(\Omega)$$

Така получихме следната вариационна задача:

Да се намери $u \in H_0^1(\Omega)$, така че

$$\iint_{\Omega} \nabla u \cdot \nabla v d\Omega = \iint_{\Omega} f v d\Omega, \quad \forall v \in H_0^1(\Omega). \quad (V)$$

Дискретизираме областта Ω , като въвеждаме триангулацията \mathcal{K} и формулираме съответната приближена задача:

Да се намери u_h в пространството на по части полиномите така, че

$$\iint_{\Omega} \nabla u_h \cdot \nabla v d\Omega = \iint_{\Omega} f v d\Omega, \quad \forall v \in V_{h,0}(\mathcal{K}). \quad (R.-G.)$$

Като използваме, че $V_{h,0} = span\{\varphi_1, \dots, \varphi_{N_{inter}}\}$, където $\varphi_1, \dots, \varphi_{N_{inter}}$ са 2D функции-“колибка”, съответстващи на вътрешните възли от мрежата, и $u_h(x) = \sum_{j=1}^{N_{inter}} q_j \varphi_j(x)$, получаваме еквивалентната задача за решаване на линейната алгебрична система $M\mathbf{q} = \mathbf{b}$:

$$\begin{bmatrix} \iint_{\Omega} \nabla \varphi_1 \cdot \nabla \varphi_1 d\Omega & \cdots & \iint_{\Omega} \nabla \varphi_1 \cdot \nabla \varphi_{N_{inter}} d\Omega \\ \vdots & \ddots & \vdots \\ \iint_{\Omega} \nabla \varphi_{N_{inter}} \cdot \nabla \varphi_1 d\Omega & \cdots & \iint_{\Omega} \nabla \varphi_{N_{inter}} \cdot \nabla \varphi_{N_{inter}} d\Omega \end{bmatrix} \begin{bmatrix} q_1 \\ \vdots \\ q_{N_{inter}} \end{bmatrix} = \begin{bmatrix} \iint_{\Omega} f \varphi_1 d\Omega \\ \vdots \\ \iint_{\Omega} f \varphi_{N_{inter}} d\Omega \end{bmatrix}.$$

Тук $M = M^1$ е матрицата на коравина. Тя отново, както и в едномерния случай, може да се асемблира, като се представи като сума от интеграли върху всеки от елементите. Съответно в елементните матрици ще има блокове от 3×3 ненулеви елемента.

N.B. 38

Общият алгоритъм на МКЕ може да се приложи практически непроменен, независимо от задачата, нейната размерност и вида на дискретизацията.

Приложение

Някои сведения от ДИС

Твърдение 1: Линеаризация на функция на много променливи

Линеаризацията на скаларната функция $f(x_1, \dots, x_n)$ около точката $(\bar{x}_1, \dots, \bar{x}_n)$ се задава с

$$f(\bar{x}_1, \dots, \bar{x}_n) + \left. \frac{\partial f}{\partial x_1} \right|_{(\bar{x}_1, \dots, \bar{x}_n)} (x_1 - \bar{x}_1) + \dots + \left. \frac{\partial f}{\partial x_n} \right|_{(\bar{x}_1, \dots, \bar{x}_n)} (x_n - \bar{x}_n).$$

Доказателство. Ще докажем за функция на две променливи. Обобщението е непосредствено.

Да разгледаме изменението

$$\Delta f := f(x, y) - f(x_0, y_0).$$

Нека означим още $\Delta x := x - x_0$, $\Delta y := y - y_0$. Тогава

$$\begin{aligned} \Delta f &= f(x_0 + \Delta x, y_0 + \Delta y) - f(x_0, y_0) \\ &= [f(x_0 + \Delta x, y_0) - f(x_0, y_0)] + [f(x_0 + \Delta x, y_0 + \Delta y) - f(x_0 + \Delta x, y_0)]. \end{aligned}$$

С други думи описваме изменението, като първо се движим по x , а след това по y . Последното е еквивалентно на

$$\begin{aligned} \Delta f &= \frac{f(x_0 + \Delta x, y_0) - f(x_0, y_0)}{\Delta x} \Delta x + \frac{f(x_0 + \Delta x, y_0 + \Delta y) - f(x_0 + \Delta x, y_0)}{\Delta y} \Delta y \\ &\approx \frac{\partial f}{\partial x} \Delta x + \frac{\partial f}{\partial y} \Delta y. \end{aligned}$$

В последното приближение използвахме факта, че Δx и Δy са малки числа. \square

Така полученият резултат ни позволява лесно да изведем и правилото за пресмятане на производна на сложна функция (англ. *Chain rule*).

Твърдение 2: Chain rule (Производна на сложна функция)

Нека имаме функцията $f(x(t), y(t))$. Тогава

$$\frac{df}{dt} = \frac{\partial f}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial t}.$$

Доказателство. Имаме

$$\Delta f = \frac{\partial f}{\partial x} \Delta x + \frac{\partial f}{\partial y} \Delta y + O(|\Delta|).$$

Делим двете страни на Δt и го пускаме да клони към нула, за да получим търсения резултат. \square

Интерполация

Идея на интерполацията

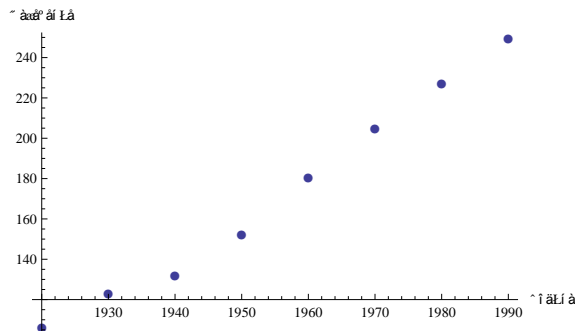
Изучавайки света около нас, ние искаме да намерим зависимости между различни величини. За да се изследва дадено явление или даден процес, е необходимо да се направят експерименти, които да дадат информация за него. След това данните от тези експерименти се използват, за да се създаде математически модел, който ги описва. Нека разгледаме следния пример.

Пример 32. В таблицата са дадени данни за населението на САЩ в млн. в периода 1920-1990.

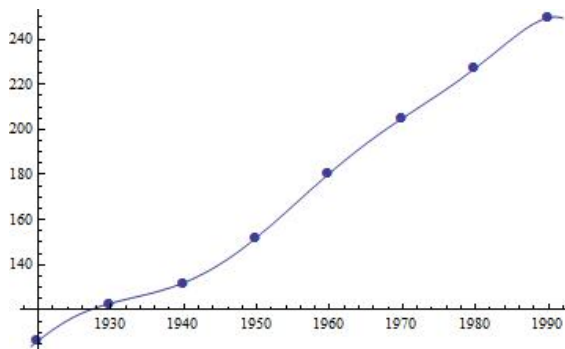
Година	1920	1930	1940	1950	1960	1970	1980	1990
Население	106.46	123.08	132.12	152.27	180.67	205.05	227.23	249.46

Търсим функция, описваща изменението на населението през този период.

В случая търсим зависимост между две величини, а резултатите от измерванията можем да интерпретираме геометрично като точки в равнината.



Тогава един възможен начин да опишем това явление е да намерим функция, чиято графика минава през дадените точки.

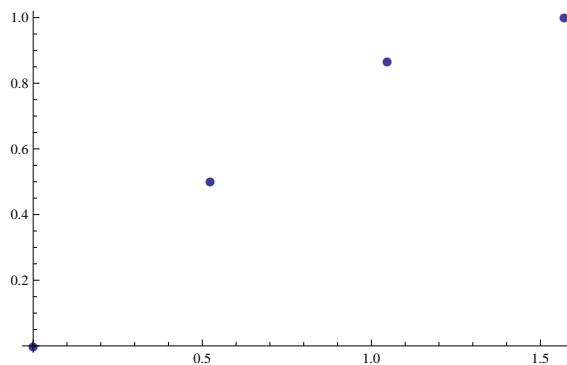


Намирането на функция, чиято графика минава през дадени точки, се нарича **интерполация**.

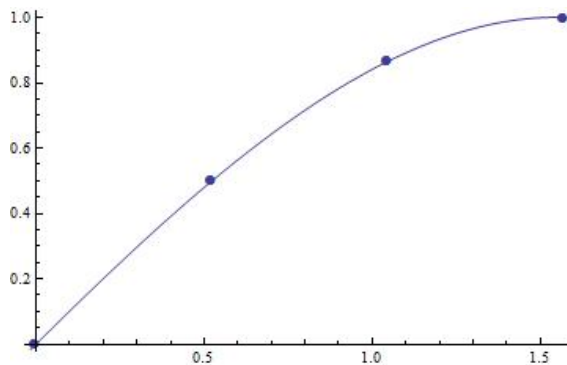
Нека сега разгледаме още една ситуация, в която ще използваме интерполация. На практика често се налага да се намират стойностите в дадена точка на функции като $\sin x$, $\cos x$, e^x , $\ln x$ и др. Както сами можем да се убедим, в общия случай това не изглежда тривиално. Един възможен подход ще илюстрираме със следващия пример.

Пример 33. Търсим приближение на стойността на функцията $f(x) = \sin x$ за $x = \pi/5$.

Оказва се, че можем да сведем тази задача до аналогична на предходната. Стойността на функцията $f(x) = \sin x$ ни е известна например за $x_0 = 0$, $x_1 = \frac{\pi}{6}$, $x_2 = \frac{\pi}{3}$, $x_3 = \frac{\pi}{2}$ (съответните стойности на $f(x)$ в тези точки са $0, 1/2, \sqrt{3}/2, 1$). Геометрично тази информация е представена на следващата фигура:



Тогава, ако намерим някаква функция $g(x)$, чиято графика да минава през тези точки (и чиято стойност в дадена точка може да бъде лесно пресметната!), ние ще имаме приближение на стойността на $f(x)$.



Така ние ще можем да намерим приблизително $\sin \pi/5$, като пресметнем $g(\pi/5)$. Тук възниква много важният въпрос колко точно ще бъде нашето приближение, т.е. колко ще се отличава неговата стойност от стойността на оригиналната функция. В настоящата глава ще коментираме и него.

Изобщо казано, интерполацията ни позволява да намерим приближение на дадена функция, използвайки стойностите ѝ в дадени точки.² И така, в настоящата глава ние ще търсим отговора на следните въпроси:

²Както ще видим по-нататък, можем да наложим условия и върху стойностите на нейните производни, но засега ще разглеждаме ситуацията, когато сме наложили условия само върху стойностите на функцията.

- Как да намерим функция, чиято графика минава през дадени точки?
- Как да оценим точността на приближението, което сме получили?

Интерполационна задача на Лагранж

Алгебричните полиноми са функции, чиято стойност в дадена точка може да бъде пресметната лесно (един бърз алгоритъм за целта е например схемата на Хорнер). Ето защо те се явяват добър избор за решаване на задачата, която си поставихме. И така, ние ще търсим алгебричен полином, който минава през дадени точки. Нека сега формулираме точно поставената вече задача.

Постановка на интерполационната задача на Лагранж.

Нека x_0, x_1, \dots, x_n са дадени различни точки от реалната права (възли) и y_0, y_1, \dots, y_n са дадени реални числа (стойности). Искаме да построим полином $P(x) \in \pi_n$ (π_n – класът от всички алгебрични полиноми от степен, ненадминаваща n) такъв, че

$$\begin{cases} P(x_0) = y_0 \\ P(x_1) = y_1 \\ \dots\dots\dots \\ P(x_n) = y_n \end{cases} \quad (4.4)$$

Винаги, когато формулираме една математическа задача, много съществен е въпросът за съществуване и единственост на решението. От гледна точка на програмното реализиране на алгоритми за нейното решение например, е важно да знаем дали задачата винаги е решима и, ако не е, да можем да обработваме съответните изключения. В случая на интерполационната задача на Лагранж е в сила следното твърдение.

Твърдение 3

Съществува, при това единствен полином $P(x) \in \pi_n$, удовлетворяващ интерполационната задача на Лагранж за произволни възли и стойности.

Да отбележим още веднъж, че геометричната интерпретация на тази задача е следната – дадени са $n + 1$ точки в равнината и търсим алгебричен полином от степен, ненадминаваща n , чиято графика минава през тези точки.

Интерполационна формула на Лагранж

Твърдение 4: Интерполационна формула на Лагранж

Полиномът, удовлетворяващ условията (4.4), се представя по формулата

$$P(x) = \sum_{k=0}^n l_k(x)y_k,$$

където $l_0(x), l_1(x), \dots, l_n(x)$ са базисните полиноми на Лагранж. Те изпълняват условията

$$l_k(x_i) = \begin{cases} 0, & \text{ако } k \neq i \\ 1, & \text{ако } k = i \end{cases}$$

и се задават с формулата

$$l_k(x) = \prod_{i=0, i \neq k}^n \frac{x - x_i}{x_k - x_i}.$$

Ще обясним смисъла на формулата чрез пример.

Пример 34. Като използваме интерполационната формула на Лагранж, ще намерим полином $P(x) \in \pi_3$, удовлетворяващ условията

$$P(1) = 2; P(2) = 9; P(4) = 41; P(6) = 97$$

Нека означим

$$x_0 = 1, x_1 = 2, x_2 = 4, x_3 = 6;$$

$$y_0 = 2, y_1 = 9, y_2 = 41, y_3 = 97.$$

Първо ще построим базисните полиноми на Лагранж. За полинома $l_0(x)$ искаме да се нулира във всички възли освен в x_0 . В x_0 стойността му трябва да бъде 1. Тогава имаме

$$l_0(x) = \frac{(x - x_1)(x - x_2)(x - x_3)}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)}.$$

Действително, във възлите x_1, x_2, x_3 съответно първият, вторият и третият множител в числителя става 0 и цялата дроб е 0. Във възела x_0 получаваме

$$l_0(x_0) = \frac{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)} = 1,$$

т.е. така дефинираният полином $l_0(x)$ изпълнява поставените му условия. Като заместим x_0, x_1, x_2, x_3 с техните равни, получаваме окончателно за $l_0(x)$

$$l_0(x) = \frac{(x - 2)(x - 4)(x - 6)}{(1 - 2)(1 - 4)(1 - 6)} = -\frac{(x - 2)(x - 4)(x - 6)}{15}.$$

Аналогично имаме

$$l_1(x) = \frac{(x-1)(x-4)(x-6)}{(2-1)(2-4)(2-6)} = \frac{(x-1)(x-4)(x-6)}{8};$$

$$l_2(x) = \frac{(x-1)(x-2)(x-6)}{(4-1)(4-2)(4-6)} = -\frac{(x-1)(x-2)(x-6)}{12};$$

$$l_3(x) = \frac{(x-1)(x-2)(x-4)}{(6-1)(6-2)(6-4)} = \frac{(x-1)(x-2)(x-4)}{40}.$$

Тогава интерполационният полином, удовлетворяващ задачата, може да бъде представен във вида

$$P(x) = l_0(x).y_0 + l_1(x).y_1 + l_2(x).y_2 + l_3(x).y_3.$$

За да се убедим в това, нека проверим какво се случва например в точката $x = x_1$. Имаме $l_0(x_1) = l_2(x_1) = l_3(x_1) = 0$ и $l_1(x_1) = 1$. Тогава

$$P(x_1) = 0.y_0 + 1.y_1 + 0.y_2 + 0.y_3 = y_1.$$

Аналогично се вижда, че този полином удовлетворява интерполационните условия и в другите възли.

И така, получихме

$$P(x) = l_0(x).2 + l_1(x).9 + l_2(x).41 + l_3(x).97.$$

След заместване и опростяване, получаваме окончателно

$$P(x) = 3x^2 - 2x + 1.$$

Оттук нататък с $L_n(f; x)$ ще бележим интерполационния полином от ред n за функцията f , а с $\omega(x)$ ще бележим $(x - x_0)(x - x_1)\dots(x - x_n)$, където x_0, x_1, \dots, x_n са дадени различни точки.

Дотук показахме как можем да построим полином, чиято графика минава през дадени точки. Следващото твърдение дава отговор на въпроса как да оценим каква е точността на приближението, което се получава, когато заменим дадена непрекъсната функция с нейния интерполационен полином.

Твърдение 5

Нека $[a, b]$ е даден краен интервал и x_0, \dots, x_n са различни точки в него. Нека функцията $f(x)$ има непрекъсната $(n + 1)$ -ва производна в този интервал. Тогава за всяко $x \in [a, b]$ съществува точка $\xi \in [a, b]$ такава, че

$$f(x) - L_n(f; x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega(x),$$

където $\omega(x) := (x - x_0) \dots (x - x_n)$.

В частност, ако интерполационните възли са равноотдалечени точки, дискретизиращи интервала $[a, b]$ с равномерна стъпка h , получаваме

$$f(x) - L_n(f; x) = O(h^{n+1}).$$

Вече сме готови да решим задачата, която си поставихме в пример 33 и да видим как можем да оценим приближено стойността на дадена функция в някоя точка, използвайки интерполация.

Пример 35. Да се намери приближено стойността на $\sin \frac{\pi}{5}$ и да се даде оценка на грешката при апроксимация.

Решение. Очевидно оценяването на функцията $f(x) = \sin x$ в дадена точка не е никак проста работа. Ето защо, вместо да работим с тази функция, ние ще намерим нейно приближение и ще работим с него. Да изберем първо възли, в които да интерполираме. Точки, в които стойността на $\sin x$ ни е известна, са например $x_0 = 0$, $x_1 = \frac{\pi}{6}$, $x_2 = \frac{\pi}{3}$, $x_3 = \frac{\pi}{2}$ (съответните стойности на $f(x)$ в тези точки са $0, 1/2, \sqrt{3}/2, 1$). Ще намерим полинома $L_3(f; x)$, който интерполира функцията $f(x)$ в тези точки. За целта първо намираме базисните полиноми на Лагранж:

$$\begin{aligned} l_0(x) &= \frac{(x - \frac{\pi}{6})(x - \frac{\pi}{3})(x - \frac{\pi}{2})}{(0 - \frac{\pi}{6})(0 - \frac{\pi}{3})(0 - \frac{\pi}{2})}, \\ l_1(x) &= \frac{(x - 0)(x - \frac{\pi}{3})(x - \frac{\pi}{2})}{(\frac{\pi}{6} - 0)(\frac{\pi}{6} - \frac{\pi}{3})(\frac{\pi}{6} - \frac{\pi}{2})}, \\ l_2(x) &= \frac{(x - 0)(x - \frac{\pi}{6})(x - \frac{\pi}{2})}{(\frac{\pi}{3} - 0)(\frac{\pi}{3} - \frac{\pi}{6})(\frac{\pi}{3} - \frac{\pi}{2})}, \\ l_3(x) &= \frac{(x - 0)(x - \frac{\pi}{6})(x - \frac{\pi}{3})}{(\frac{\pi}{2} - 0)(\frac{\pi}{2} - \frac{\pi}{6})(\frac{\pi}{2} - \frac{\pi}{3})}. \end{aligned}$$

Тогава получаваме

$$L_3(f; x) = l_0(x) \cdot 0 + l_1(x) \cdot \frac{1}{2} + l_2(x) \cdot \frac{\sqrt{3}}{2} + l_3(x) \cdot 1.$$

След кратки преобразования получаваме

$$L_3(x) \approx 1.02043x - 0.0654708x^2 - 0.113872x^3.$$

Сега вече можем да намерим стойността на полинома $L_3(x)$ при $x = \frac{\pi}{5}$, тъй като тя ще бъде „близо“ до истинската стойност на $\sin \frac{\pi}{5}$. Получаваме $L_3(\frac{\pi}{5}) \approx 0.587061$.

Остана да дадем оценка за това колко „близо“ всъщност е стойността, която ние сме намерили, до точната стойност. С други думи, искаме да дадем оценка за грешката

$$R\left(\frac{\pi}{5}\right) = \left| f\left(\frac{\pi}{5}\right) - L_3\left(f; \frac{\pi}{5}\right) \right|.$$

От Твърдение 4.4 непосредствено следва, че

$$R\left(\frac{\pi}{5}\right) = \frac{|f^{(4)}(\xi)|}{4!} \left| \omega\left(\frac{\pi}{5}\right) \right|,$$

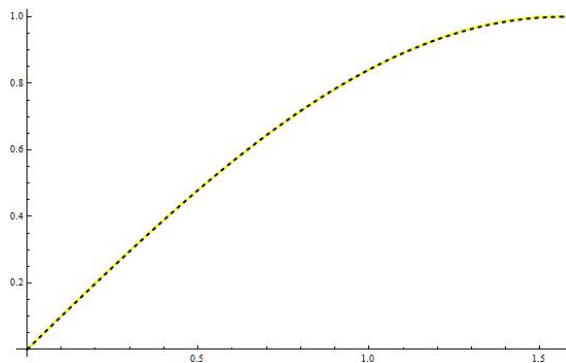
където ξ е число от интервала $[0, \frac{\pi}{2}]$. Имаме $f^{(4)}(\xi) = \sin \xi$. В интервала $[0, \frac{\pi}{2}]$ функцията $\sin x$ приема стойности между 0 и 1, т.е. $|f^{(4)}(\xi)| \leq 1$. Тогава

$$R\left(\frac{\pi}{5}\right) \leq \frac{1}{24} \left| \omega\left(\frac{\pi}{5}\right) \right| \approx 0.00108232.$$

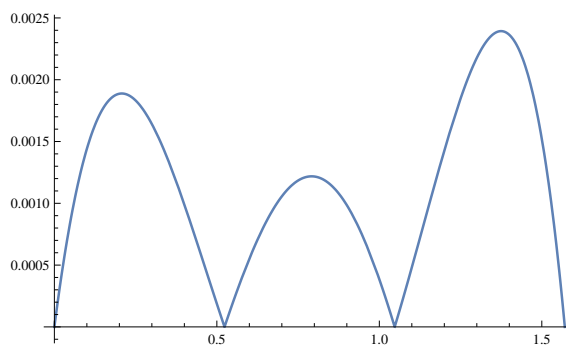
Окончателно получихме, че грешката по абсолютна стойност не надминава 0.0011. Ако сравним стойността, която ние получихме (0.587061) със стойността, която Mathematica връща за $\sin \frac{\pi}{5}$ (0.587785), ще се убедим, че това действително е така. Обърнете внимание, че това е оценка отгоре за грешката, т.е. тя може и да е значително по-малка.

Да обърнем внимание и че можехме да намерим по-добро приближение на $\sin \frac{\pi}{5}$, ако бяхме подбрали интерполационните възли по по-подходящ начин или бяхме взели повече възли. \square

Нека коментираме още няколко неща, свързани с предходната задача. Да илюстрираме първо нейното решение графично – заместваме функцията $\sin x$ (която на фигурата е с черната пунктирана линия) с интерполационния полином от степен 3, $L_3(f; x)$ (жълтата непрекъсната линия). Както виждаме, двете графики почти съвпадат в интервала на интерполация $[0, \pi/2]$, което обосновава приближаването на $\sin \frac{\pi}{5}$ с $L_3(f; \frac{\pi}{5})$.

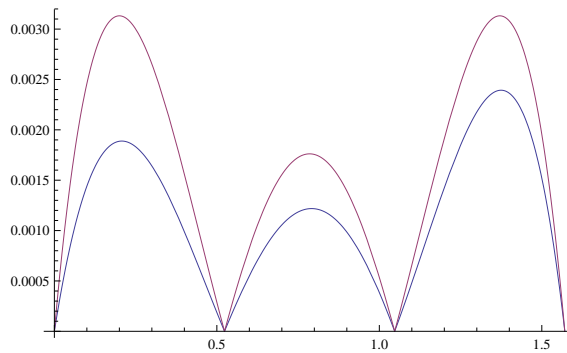


Привеждаме и графиката на абсолютната грешка (по модул) при приближаването на $\sin x$ с $L_3(f; x)$ в разглеждания интервал:

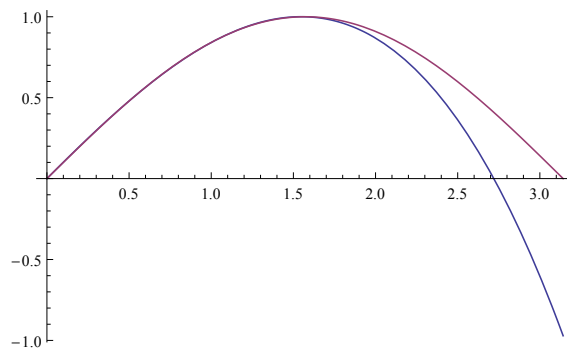


Допуснатата грешка, както можем да очакваме, във всички точки е не по-голяма от изведената оценка

$$R(x) \leq \frac{1}{24} |\omega(x)| :$$



Нека сега разгледаме графиките на $\sin x$ и $L_3(f; x)$ в интервала $[0, \pi]$:



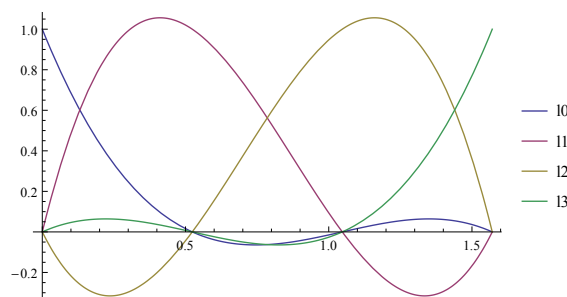
Както виждаме, извън границите на интерполация двете графики сериозно се разминават.

Дефиниция 14

Когато използваме интерполационния полином за приближаване на стойност в интервала, определен от интерполационните възли, говорим за **интерполация**. В противен случай говорим за **екстраполация**.

При екстраполация не можем да разчитаме на това, че ще получим добро приближение. Полиномът няма точки, за които да се „хване“ и ето защо няма как да очакваме неговото поведение да следва това на приближаваната функция.

Нека сега разгледаме графиките на базисните полиноми на Лагранж, които намерихме в предходната задача:



Графиката добре илюстрира условието, което наложихме на базисните полиноми на Лагранж при тяхното дефиниране – всеки от тях има стойност 1 във възела, за който “отговаря” и 0 във всички останали възли.

Дефиниция 15

Нека са дадени точките $x_0 < \dots < x_n$. Казваме, че функциите $\varphi_0(x), \dots, \varphi_n(x)$ образуват интерполационен базис, ако

$$\varphi_i(x_j) = \begin{cases} 0 & \text{ако } i \neq j, \\ 1 & \text{ако } i = j. \end{cases}$$

Ясно е, че ако $\varphi_0(x), \dots, \varphi_n(x)$ образуват интерполационен базис и във формулата на Лагранж заместим базисните полиноми $l_i(x)$ с $\varphi_i(x)$, то получената функция $\bar{\varphi}(x)$ ще изпълнява условията $\bar{\varphi}(x_i) = y_i$.

Други

Твърдение 6

Нека е дадена достатъчно гладка функция u , определена в $[a, b]$. Средно-аритметичното на $u(a)$ и $u(b)$ апроксимира $u\left(\frac{a+b}{2}\right)$ с грешка $O((b-a)^2)$.

Доказателство. Пресмятаме грешката на апроксимация. Нека означим $h := b - a$. Тогава

$$\begin{aligned} & u\left(\frac{a+b}{2}\right) - \frac{1}{2}(u(a) + u(b)) \\ &= u\left(\frac{a+b}{2}\right) - \frac{1}{2}\left(u\left(\frac{a+b}{2}\right) - u'\left(\frac{a+b}{2}\right)\frac{h}{2} + O(h^2)\right) + u\left(\frac{a+b}{2}\right) + u'\left(\frac{a+b}{2}\right)\frac{h}{2} + O(h^2) \\ &= O(h^2). \end{aligned}$$

□

Библиография

- [1] J. Butcher, Numerical Methods for Ordinary Differential Equations. Wiley, 2nd edition, 2009.
- [2] E. Coddington, An Introduction to Ordinary Differential Equations. Dover Publications, Unabridged edition, 1989.
- [3] С. Димова, Т. Черногорова, А. Йотова, Числени методи за диференциални уравнения. Университетско издателство “Св. Климент Охридски”, София, 2010.
- [4] J. Hale, Ordinary Differential Equations. Dover Publications, 2009.
- [5] M. Hirsh, S. Smale, R. Devaney, Differential Equations, Dynamical Systems, and an Introduction to Chaos. Academic Press, 3rd edition, 2012.
- [6] M.H. Holmes, Introduction to Numerical Methods in Differential Equations. Springer, 2007.
- [7] J. Murray, Mathematical Biology I. An Introduction. Springer, 3rd edition, 2002.
- [8] M. Tenenbaum, H. Pollard, Ordinary Differential Equations. Dover Publications, Revised ed., 1985.