Počítačová fyzika I

Numerická stabilita jednoduchých iteračných schém

Úvod

Peter Papp, Ján Ďurian

(Peter Markoš)

Katedra experimentálnej fyziky F2-81

Obsah

- Úvodné komentáre
  1. Zaokrúhľovanie
  2. Mitášov zákon
- Voľba algoritmu: Iterácie (zlatý rez)
- Iteračné schémy a ich vlastnosti
  1. pôvod numerickej nestability
  2. vlastné hodnoty a vlastné vektory
- Úloha

Zaokrúhľovanie

Najjednoduchší program vo Fortrane:
Rozdiel dvoch čísiel, ktoré sa líšia na 10. desatinnom mieste:

```
      real*8 x1,x2
      x1=0.123456789789652
      x2=0.123456789443517
      write(*,*) x2-x1
!
      stop
      end
```

výsledok: x2-x1=0
Počítač je absolútne neinteligentný (Ľ. Fischer).
Je dobre vedieť, ako on interpretuje vstupy.

Zaokrúhľovanie

Rozdiel dvoch čísiel, ktoré sa líšia na 8. desatinnom mieste:

```
                  real*8 x1,x2
                  x1=0.123456789789652
                  x2=0.123456779443517
                  write(*,*) x2-x1
!
                  stop
                  end
```

Výsledok (I7 6-jadrový): x2-x1= - 1.49011612E-08

Správny výsledok: x2-x1= - 1.0346135E-08
Chyba "výpočtu": 44%
Záver: Ak počítač vidí len 8 platných číslic, niektoré elementárne výpočty môžu havarovať.

Zaokrúhlovanie

Hodnotu každej premennej deklarujem ako `double precision` koncovkou d0

```
            real*8 x1,x2
            x1=0.123456789789652d0
            x2=0.123456779443517d0
            write(*,*) x2-x1
!
            stop
            end
```

Výsledok (I7 6-jadrový): x2-x1= - 1.0346134990402156E-008
Správny výsledok: x2-x1= - 1.0346135000000000E-08

Počítač síce uvádza 16 desatinných miest, ale tie posledné nie sú „dôveryhodné".
Prečo ?

Zaokrúhlovanie

Skúsim doplniť nuly:

```
      real*8 x1,x2
      x1=0.1234567897896520000000d0
      x2=0.1234567794435170000000d0
      write(*,*) x2-x1
!
      stop
      end
```

Výsledok (I7 6-jadrový): x2-x1= -1.0346134990402156E-008
Správny výsledok: x2-x1= -1.0346135000000000E-08

Nič nepomôže, lebo každá premenná je známa len na 16 miest.

Zaokrúhlovanie - deklarácia

Deklarujme premenné s presnosťou na 32 desatinných miest:

```
                real*16 x1,x2
                x1=0.123456789789652q0
                x2=0.12345677944351 7q0
                write(*,*) x2-x1
!
                stop
                end
```

Výsledok (I7 6-jadrový):

x2-x1= -1.03461350000000000000000000093185046E-008

Podstatne lepší výsledok, "hausnumerá" sa objavia od 32. miesta.

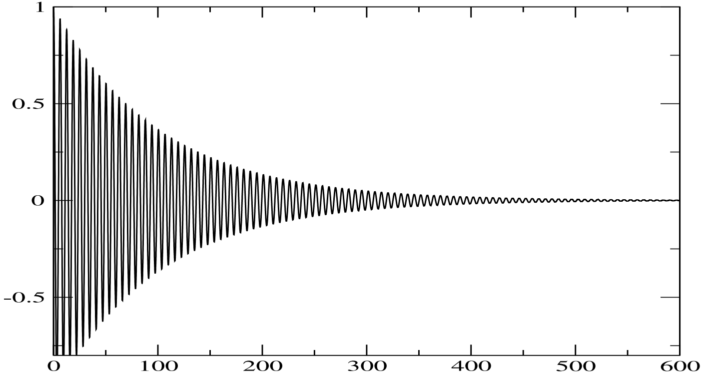
Poučenie

- Záleží na deklarácii čísiel
- Odčítavanie malých čísiel je nebezpečné

Mitášov zákon

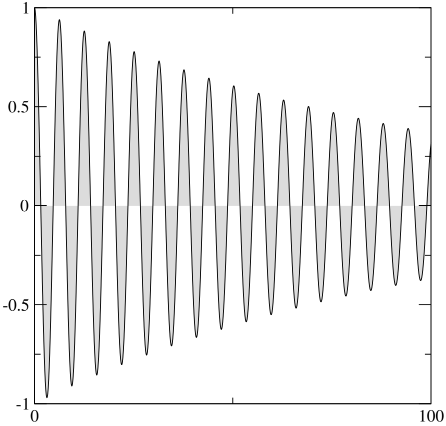
Počítajme numericky integrál

$$I = \int_0^\infty \mathrm{d}x\; e^{-0.01x} \cos(x)$$



Integrovanie „naslepo" nemôže viesť k dobrým výsledkom.
Jediná možnosť:

Mitášov zákon



...integrovať funkciu cez každú pol periódu zvlášť a počítať sumu konvergentného radu so striedavými znamienkami:

$$I = \sum I_n$$

Mitášov zákon

Ak integrujeme numericky správne, dostaneme

$$I = 0.0099990000999900009999\ldots \approx \frac{0.01}{1.0001}$$

Jednoduchosť výsledku naznačuje, že integrál sa dá nájsť aj analyticky. Naozaj, ide o tabuľkový integrál:

$$I = \int_0^\infty \mathrm{d}x \; e^{-ax} \cos(bx) = \frac{a}{a^2 + b^2}$$

a teda numerický výpočet nebol potrebný. Ak je nuemrický vypočet dostatočne presný, dokážeme sa z jeho výsledkov poučiť – interpetovať získaný výsledok.

Dobrý program je ten, ktorý sa po ukončení výpočtu ukáže byť zbytočný.

Riešenie bolo možné nájsť aj bez neho, keby človek trošku porozmýšľal.

Zlatý rez - ukážka zle fungujúceho algoritmu

Zlatý rez: delenie intervalu $L = 1$ na dve časti x, y tak, aby

$$\frac{x}{1} = \frac{y}{x}$$

Keďže $x + y = 1$, dostanem kvadratickú rovnicu pre x:

$$x^2 + x - 1 = 0$$

ktorá má riešenia

$$x_1 = \frac{1}{2}\left[\sqrt{5} - 1\right] \approx 0.61803398$$

$$x_2 = \frac{1}{2}\left[-\sqrt{5} - 1\right] \approx -1.61803398$$

to druhé je, samozrejme, zlé, lebo $x_2 < 0$.

Zlatý rez

Nápad: x_1 splĺňa rovnicu

$$x_1^2 = -x_1 + 1$$

a teda splĺňa aj rovnice

$$x_1^{n+1} = -x_1^n + x_1^{n-1} \qquad n = 1, 2, \dots$$

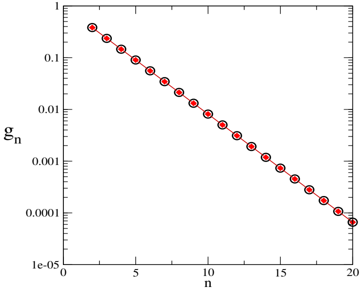
Počítajme teda n-tú mocninu $g_n = x_1^n$ iteračne z rovnice

$$g_{n+1} = g_{n-1} - g_n$$

s počiatočnými podmienkami: $g_0 = 1$, $g_1 = x_1$.

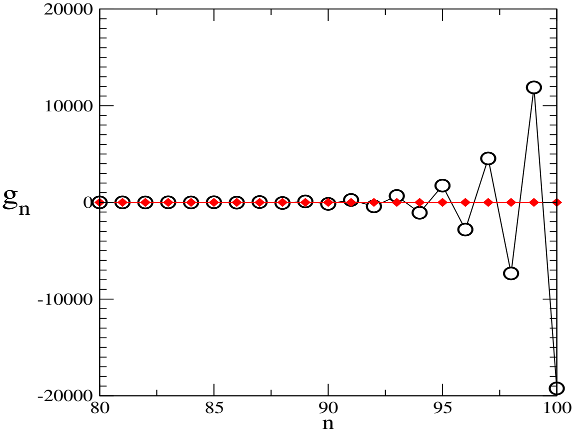
Zlatý rez

Porovnanie numericky získaných hodnôt g_n s presnými hodnotami x_1^n.



Všetko výborne funguje, a šetríme CPU, lebo len sčitujeme, nenásobíme.
Ale ...

Zlatý rez



Katastrofa. Pre veľké hodnoty n začne $|g_n|$ exponenciálne rásť, hoci x_1^n exponenciálne klesá.

Zlatý rez

Dôvod: riešenie rovnice $g_{n+1} = g_{n-1} - g_n$ má všeobecný tvar

$$g_n = Ax_1^n + Bx_2^n$$

kde $x_1 = 0.61803398, \qquad x_2 = -1.61803398 = -1 - x_1.$
Z počiatočných podmienok $g_0 = 1,\ g_1 = x_1$ ľahko nájdeme

$$\begin{aligned} A + B &= 1 \\ Ax_1 + Bx_2 &= x_1 \end{aligned}$$

Riešením týchto rovníc dostaneme

$$A = 1 \qquad \text{a} \qquad B = 0$$

a teda predpokladáme, že správne riešenie je $g_n = x_1^n$.
Počítač ale pozná g_0 a g_1 len s nejakou presnosťou, a preto nevidí $B \equiv 0$, ale nahradí ho nejakým malým číslom (pod prahom svojej presnosti).

Zlatý rez

Namiesto

$$B = 0.000\,000\,000\,000\,000\,00$$

počítač môže vidieť napr.

$$B' = 0.000\,000\,000\,000\,000\,01$$

alebo

$$B'' = -0.000\,000\,000\,000\,000\,07$$

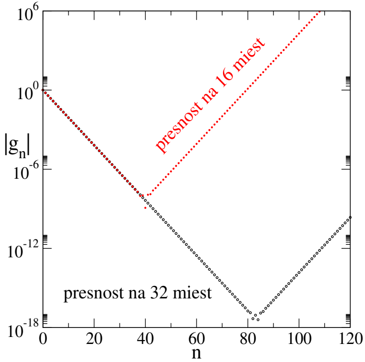
pritom ale

$$B' x_2^{90} = 644 \quad \gg \quad A x_1^{90} = 1.55 \times 10^{-19}$$

Preto exponenciálne rastúce riešenie x_2^n vždy „vyhrá".

Je to naozaj tak?

Skúsme v programe deklarovať presnosť na 32 desatinných miest a porovnajme výsledok s výpočtom na 16 desatinných miest. Do obrázku kreslime absolútnu hodnotu $|g_n|$, aby sme mohli vertikálnu os voliť logaritmickú:



Riešenie väčšou presnosťou (čierne body na obrázku) je presnejšie, numerická nestabilita sa prejaví až po $n > 80$, pretože iterácie prebiehajú s

$$|B| = 0,000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 01$$

Prečo o tom hovoriť?

Exponenciálne rastúce riešenie "prebije" vždy naše správne riešenie a

výsledky sú bezcenné.

- Veľká časť numerických algoritmov je založená na iteračných procedúrach.
  - časový vývoj nelineárnych systémov
  - diferenciálne rovnice
  - parciálne diferenciálne rovnice (rovnica difúzie, Schrödingerova rovnica ...)
  - výpočet špeciálnych funkcií (napr. Besselove funkcie)
- Nesprávne zvolená iteračná schéma spôsobí numerické nestability a skôr či neskôr vedie k numerickým nestabilitám
- Základný cieľ pri tvorbe programov:
  - rozpoznať, že v probléme môžu byť numerické nestability
  - nájsť algoritmy, ktoré dokážu numerické nestability eliminovať

Triviálna iteračná schéma

$$x_{n+1} = ax_n$$

má riešenie

$$x_n = a^n x_0$$

Odlišujeme tri základné typy riešenia:

- $|a| > 1 \ldots$ exponenciálny nárast
- $|a| < 1 \ldots$ exponenciálny pokles
- $|a| \equiv 1 \ldots$ oscilácie:

$$a = e^{i\phi}$$

$$x_n = x_0 e^{in\phi}$$

Zodpovedajú typickým výsledkom numerických simulácií: exponenciálny nárast/pokles alebo oscilácie.

Pozn. všimnime si, že potrebujeme len jednu „štartovaciu“ hodnotu x_0.
Pozn. táto triviálna úloha nevedie k numerickým nestabilitám, lebo iteračná schéma má len jedno možné riešenie.

Menej triviálna iteračná schéma I

Iterujme rovnicu (vyskytuje sa vo fyzike veľmi často)

$$x_{n+1} = ax_n - x_{n-1}$$

s počiatočnými podmienkami: $x_0 = 0$, $x_1 = 1$ (potrebujeme dve podmienky!)
Úloha: zvoľte si hodnotu a a iterujte.

- $a = 0$
- $a = 1$
- $a = 2$
- $a = -2$
- $a = 3$

Nakreslite graf x_n *vs* n.
Riešenie: pre $|a| > 2$ dostaneme vždy exponenciálne rastúce riešenia !
Hypotéza: iteračná schéma má dve základné riešenia, a jedno z nich exponenciálne rastie.

Prečo

$$x_{n+1} = a x_n - x_{n-1}$$

Obsahuje okrem násobenia aj súčet. Dá sa ale prepísať na schému len s násobením:

$$\begin{pmatrix} x_{n+1} \\ x_n \end{pmatrix} = \begin{pmatrix} a & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x_n \\ x_{n-1} \end{pmatrix}$$

Iteračná schéma nie pre čísla, ale pre vektory:

$$\vec{u}_{n+1} = \mathbf{M} \vec{u}_n$$

Potrebujeme poznať vlastné hodnoty matice

$$\mathbf{M} = \begin{pmatrix} a & -1 \\ 1 & 0 \end{pmatrix}$$

Vlastné hodnoty matice $\mathbf{M}$

Vieme, že:

$$\det \mathbf{M} = 1 = \lambda_1 \lambda_2, \quad \text{preto} \quad \lambda_2 = \lambda_1^{-1}$$

$$\mathrm{Tr}\ \mathbf{M} = \lambda_1 + \lambda_2 = a$$

$$a = \lambda_1 + \lambda_2 = \lambda_1 + \frac{1}{\lambda_1}$$

- ak $|a| < 2$ tak $\lambda_1 = e^{iq}$ a$= 2\cos q$
- ak $|a| > 2$ tak $\lambda_1 = e^{\kappa}$, $a = 2\cosh \kappa$ a teda $|\lambda_1| > 1$

Menej triviálna iteračná schéma II

Iterujeme súčasne viac vzájomne zviazaných premenných:

$$\begin{array}{rcrcl} x_{n+1} & = & ax_n & + & by_n \\ y_{n+1} & = & cx_n & + & dy_n \end{array}$$

$$\vec{u}_{n+1} = \mathbf{M}\vec{u}_n$$

$$\vec{u}_n = \begin{pmatrix} x_n \\ y_n \end{pmatrix}, \qquad \mathbf{M} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Opäť jednoduchá iteračná schéma: $(n+1)$-vá iterácia sa dostane len násobením n-tej

ALE: násobíme matice, nie čísla.

Potrebujeme poznať vlastné hodnoty a vlastné vektory matice $\mathbf{M}$.

Uvažujeme zatiaľ len homogénnu úlohu, v reálnom svete budeme mať v každom iteračnom kroku inú maticu: $\mathbf{M} \rightarrow \mathbf{M}_n$.

Vlastné hodnoty matice

Maticu **M** vyjadrime v tvare

$$\mathbf{M} = \mathbf{Q}^{-1} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \mathbf{Q}$$

λ_1, λ_2 sú vlastné hodnoty matice
Qobsahuje vlastné vektory matice **M**potom

$$\vec{u}_{n+1} = \mathbf{M}\vec{u}_n = \mathbf{Q}^{-1} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \mathbf{Q}\vec{u}_n$$

Iteračná schéma je určená vlastnými hodnotami

Ak aspoň jedna z vlastných hodnôt $|\lambda_k| > 1$ tak je iteračná schéma numericky nestabilná.

Cvičenie: zlatý rez

$$g_{n+1} = -g_n + g_{n-1}$$

Definujem vektor

$$\vec{u}_n = \begin{pmatrix} g_n \\ g_{n-1} \end{pmatrix}$$

Dostal som iteračnú schému

$$\vec{u}_{n+1} = \begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix} \vec{u}_n$$

det $\mathbf{M} = -1$

$$\lambda_1 = x_1 = 0.61803398, \qquad \lambda_2 = x_2 = -1.61803398 = -\lambda_1^{-1}$$

Príklad iteračnej schémy, ktorá má $|\lambda_1| < 1$ ale $|\lambda_2| > 1$.

Cvičenie: zlatý rez

vlastné vektory matice **M**

$$\mathbf{M} = \begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix}$$

nájdeme riešením systému lineárnych rovníc

$$\mathbf{M}\vec{v}_1 = \lambda_1 \vec{v}_1 \qquad \mathbf{M}\vec{v}_2 = \lambda_2 \vec{v}_2$$

$$\vec{v}_1 = \frac{1}{\sqrt{1+\lambda_1^2}} \begin{pmatrix} \lambda_1 \\ 1 \end{pmatrix} \qquad \vec{v}_2 = \frac{1}{\sqrt{1+\lambda_2^2}} \begin{pmatrix} \lambda_2 \\ 1 \end{pmatrix}$$

Oba vektory sú normalizované, $|\vec{v}_1| = 1, \quad |\vec{v}_2| = 1$
a kolmé na seba

$$\vec{v}_1 \cdot \vec{v}_2 = 0$$

Záver

- Veľmi mnoho numerických algoritmov je založených na iteráciách Pre porozumenie iteráciám musíme vedie niečo o vlastných hodnotách
- Diskretizácia vyžaduje porozumenie fyzikálnej podstaty problému. Je dobré vedieť "typickú" dĺžku a jej prispôsobiť diskretizáciu

Archimedov zákon:
nezáleží na veľkosti, ale na pomere

- Numerické chyby sa akumulujú. Čím dlhšie integrujeme, tým menší krok *a* potrebujeme.
- Numerické chyby sa nedajú úplne vylúčiť, musíme s nimi žiť. Môžeme ich čiastočne eliminovať ak poznáme ich pôvod:
  1. vhodnou voľbou vstupných parametrov
  2. voľbou vhodného algoritmu

Úloha

povinná a nepovinná, ale zaujímavá:

1. Naprogramujte iteračnú schému, ktorá počíta g_n (úloha 1-ZR). Nakreslite obrázok, odhanite, pre aké hodnoty n_z schéma zlyháva. Nájdite interpretáciu tohto výsledku.
   Zmení sa n_z ak budete počítať s inou presnosťou?

2. Presvedčte sa, že schéma $x_{n+1} = ax_n - x_{n-1}$ je stabilná pre $|a| < 2$. Nakreslite obrázok pre tri ľubovoľné hodnoty a, napr. $a = 1.99$, $a = -2.01$, $a = 0.001$.
   Skúste odhadnúť periódu N takú, že $a_{n+N} = a_n$ pre každé dostatočne veľké n. Počiatočné podmienky x_0 a x_1 zvoľte podľa ľubovôle. Presvedčte sa, že získané riešenie môžete písať v tvare $x_n = A\lambda_1^n + B\lambda_2^n$. Vlastné hodnoty λ_1 a λ_2 nájdete z matice $\boldsymbol{M}$, konštanty A a B z počiatočných podmienok.

3. Riešte úlohu 1-ITER s vami zvolenou počiatočnou podmienkou, napr. $x_0 = \sqrt{\pi}$. Nakreslite obrázok. (pozri nasldujúci slide)

Odovzdaná úloha vyžaduje napísané programy, obrázky, diskusiu výsledkov.

Úloha 1-ITER

Nájdite spôsob, ako numericky vypočítať riešenie rovníc

$$x_{n+1} - [4 + 0.1\cos(\pi n)]\, x_n + x_{n-1} = 0 \qquad\qquad x_0 = X$$

ktoré exponenciálne klesá v limite $n \to \infty$. Napíšte program a nakreslite obrázok.

Riešenie: skúsme iterovať v opačnom smere

- zvoľme veľké N a položme $x_N = 0$, $x_{N-1} = 1$
- iterujme rovnicu „naspať": počítajme $x_{N-2}, \dots x_2, x_1$
čísla budú exponenciálne rásť !
- všetky čísla vynásobíme X/x_0
(to môžeme, pretože iteračná schéma je homogénna)

Úlohu musíte zopakovať pre niekoľko hodnôt N, aby sme sa overili presnosť výsledku. Pre malé N (napr. $N = 4$ dostaneme zlý výsledok, lebo skutočná hodota x_4 nemusí byť malá. Pre vľmi veľké hodoty N môže počítač mať problémy dopočítať všetky iterácie, lebo vypočítane'hdonoty môžu byť obrovské.