Consideraţii privind aproximarea calculelor în informatica numerică

Asist. drd. Diana Sophia Codaț
Universitatea "Tibiscus" din Timişoara

ABSTRACT: Le calcul numérique a pris naissance bien avant l'apparition des ordinateurs. Les scientifiques se sont trouvés amenés depuis toujours à essayer d'évaluer, de chiffrer des effets, à tester les modèles qu'ils ou elles avaient élaborés pour rendre compte de phénomènes observés.
Aussi, un grand nombre de méthodes d'évaluation de fonctions, d'intégrales, de résolution d'équations, ... ont été mises au point il y a longtemps. Le développement des ordinateurs a provoqué une véritable explosion dans leur emploi en même temps qu'ont vu le jour des algorithmes de plus en plus performants.

1 Generalităţi

Calculul numeric a luat naştere cu mult înaintea apariţiei calculatoarelor. Cercetătorii au încercat dintotdeauna să evalueze modelele pe care le-au elaborat ţinând cont de fenomenele observate.

Astfel, un mare număr de metode de evaluare a funcţiilor, de integrale, de rezolvări de ecuaţii, ... au fost dezvoltate acum mult timp. Dezvoltarea calculatoarelor a dus la elaborarea de algoritmi din ce în ce mai performanţi.

Fenomenele observate în ştiinţe sunt, în cele mai multe cazuri, considerate apriori ca şi continue: depind de timp, sau de spaţiu şi formalizarea lor se bazează pe dezvoltările analizei matematice. De fapt, rezultatele experimentale sunt ele însăşi deseori înregistrate într-o manieră discontinuă. Dacă eşantionăm, de exemplu, un semnal sonor într-un număr de ori pe secundă, va apare o funcţie continuă în timp şi spaţiu.

La nivelul calculului numeric lucrul se întâmpla în sens invers. Plecăm de exemplu de la ecuația care descrie evoluția în timp a unei populații, ecuație dată sub forma analitică, şi încercăm plecând de la condițiile inițiale date, să estimăm ce va fi această populație la un moment ulterior.

O mare parte a analizei numerice se fondează pe ideea că toate funcțiile (care prezintă proprietățile matematice: continue, derivabile...) pot fi reprezentate printr-o dezvoltare în serie limitată, alegerea limitei, număr de termeni ai dezvoltării, în funcție de precizia care o aşteptăm reprezentând funcția.

Astfel, dacă dorim să evaluăm funcția sin (x), putem să ne bazăm, pe dezvoltarea în serie $x - x^3/3! + x^5/5! - \ldots$
Dar, în acest caz vedem că vor apărea erori, pe de o parte legate de faptul că seria este trunchiată după un număr de termeni, iar pe de altă parte o eroare legată de faptul că reprezentarea numerelor utilizate şi deci rezultatele operațiilor este limitat la un număr de cifre semnificative.

2 Aproximații

Cazul evaluării unei funcții sau a unei integrale este interesant pentru a studia problemele de trunchiere.

a. Evaluarea unei integrale
Considerăm integrarea unei funcții date f(x)

$$\int_a^b f(x)\,dx$$

i. Simpson

1. ordinul 0

Cel mai simplu este să ne limităm la o sumă de termeni:

$$\int_a^b f(x)\,dx \approx \sum_a^b f(x_i)\,\Delta x_i$$

Aceasta revine în a înlocui în fiecare punct x_i, funcția prin primul termen al dezvoltării în serie.

Dacă funcția este evaluată la un pas constant Δx, integrala (sau mai bine zis aproximarea) este egală cu:

Δx [f(a) + f(a+ Δx) + f(a + 2Δx) +...]

Putem evident ameliora calitatea aproximării mărind numărul de termeni ai sumei (şi deci diminuând valoarea Δx)

Algoritmul de evaluare al integralei este următorul:

calculază $\Delta x = (b - a) / n$

!unde n este numărul de termeni ales

evaluează funcția în a, a+ Δx, ..., a+ n Δx

!cu $a + n \Delta x = b$!

adaugă toți termenii

multiplică cu Δx

Pentru a estima calitatea rezultatului obținut, putem multiplica numărul de termeni cu 2, de exemplu, şi reîncepe întregul proces, până când noul rezultat nu diferă de precedentul decât printr-o limită pe care am fixat-o (şi pentru care rezultatul converge spre o anumită valoare!). Această aproximare a integralei de ordinul 0 este rezultatul exact, în cazul în care integrăm o constantă.

2. ordinul 2

La ordinul următor dezvoltarea în serie se scrie în felul următor:

$$\int_{x_i}^{x_{i+1}} f(x)\,dx \approx 1/2\,\Delta x\,[f(x_i) + f(x_{i+1})]$$

Recunoaştem aici regula trapezului.

Generalizând la fiecare interval de integrare, vom obține:

$$\int_a^b f(x)\,dx \approx 1/2\,\Delta x[f(a)+2*f(a+\Delta x)+2*f(a+2\Delta x)+\ldots+f(b]$$

Acest rezultat este exact, în cazul unei funcții lineare de x.

Această aproximare este, pentru un număr dat de puncte, sensibil superioară precedentei (pentru $\Delta x/(b-a) \ll 1$); vom arăta că termenii neglijați în dezvoltare sunt de această dată de ordinul Δx^3 multiplicați prin valoarea derivatei secunde a funcției într-un punct în intervalul între x_i şi x_{i+1} pentru fiecare interval.

3. ordinul 3

Putem, evident, continua micul joc luând din ce în ce mai mulți termeni în dezvoltare, ceea ce implică şi din ce în ce mai multe valori ale funcției de evaluat.
Foarte utilizată este cea de ordinul 2, ceea ce revine la a aproxima funcția printr-un polinom în x de gradul 2, de forma $c_1 x^2 + c_2 x + c_3$.

Rezultatul se scrie:

$$\int_a^b f(x)\,dx \approx \frac{1}{3} \Delta x \, [\, f(a) + 4 * f(a + \Delta x) + 2 * f(a + 2\Delta x) + 4 * f(a + 3\Delta x) + \ldots + f(b) \,]$$

Observăm alternanța factorilor 2 şi 4 în serie.

Dacă funcția a fost evaluată în a, a+Δx, etc., pentru a da f(1), f(1+1),..., f(n), algoritmul care furnizează integrala se rezumă la:

s <- 0 *! s=0 este valoarea inițială a integralei !*
pentru i **de la** 1 **la** n-2 **pas** 2 **fă**
  s <- s + f(i) + 4 * f(i+1) + f(i+2)
sfpentru
s <- s * Δx /3

Remarcăm că numărul n trebuie să fie impar.[1]

ii. Gauss

Putem, de asemenea, să înlocuim funcția printr-o sumă de polinoame uşor de evaluat, eventual multiplicată printr-o funcție x. Trebuie pentru aceasta să dispunem de o bază completă de polinoame ortogonale. Utilizăm frecvent polinoamele lui Cebâşev.

[1] Atenție la implementarea în Pascal, buclele sunt limitate la valori de pas 1 şi -1

În acest caz va trebui să găsim funcțiile x în care trebuie să multiplicăm diferite polinoame pentru a obține o aproximare dată.
Pe această schemă se bazează şi metoda lui Gauss.

Vom arăta că, prin media evaluării funcției în puncte bine specificate (nu echidistante) şi utilizarea de coeficienți numerici adecvați, obținem un ordin superior celui din formulele lui Simpson pentru un număr dat de evaluări de f(x).

Vom expune mai jos algoritmul, în forma schematică, pentru N=10:

```
procedure integrala(a, b, s)
        ! f,  funcția este definită precedent!
        ! a şi b sunt borne de integrare !
        ! s este rezultatul de integrare !
        variabile
                x, w: vector (1..5) real
                xm, xr, dx: real
                ! x sunt abscise unde trebuie să fie evaluată funcția
                (simetric în raport cu punctul din mijloc!
                !w este greutatea fiecăruia dintre puncte !
corp
        x<-
(.1488743389,.4333953941,.6794095682,.8650633666,.9739065285)
        w<-
(.2955242247,.2692667193,.2190863625,.1494513491,.0666713443)
        xm <- 0.5*(b+a)
        xr <- 0.5*(b-a)
        s <- 0.0
        pentru j de la1 la 5 fă
                dx <- xr * x(j)
                s <- s + w(j) * (f(xm+dx) + f(xm-dx))
        sfpentru
        s <- xr * s
sfproc
```

b. Integrarea ecuațiilor de mişcare

Nu avem nevoie, întotdedauna, să recurgem la metode foarte sofisticate pentru a obține rezultate interesante. Luând, de exemplu, cazul soluției ecuației de mişcare a unei planete în jurul Soarelui. Ea se scrie:

$$m \frac{d^2}{dt^2} \mathbf{r} = - G \frac{m\ m'}{r^2} \mathbf{1_r}$$

Exprimând în coordonate carteziene aceeastă ecuaţie şi introducând viteza $(\mathbf{v} = \frac{d}{dt}\mathbf{r})$ şi constanta C (= G m m') ajungem la :

$$\frac{dv_x}{dt} = - C \frac{x}{r^3} \ ; \ \frac{dv_y}{dt} = - C \frac{y}{r^3} \ ; \ r = \sqrt{x^2 + y^2}$$

Soluţia numerică cea mai simplă constă în a scrie:

$$x_1 (\Delta t) = x_0 + v_x (0) * \Delta t \qquad y_1 (\Delta t) = y_0 + v_y (0) * \Delta t$$

şi

$$v_x (\Delta t) = v_x (0) + a_x * \Delta t \qquad v_y (\Delta t) = v_y (0) + a_y * \Delta t$$

sau a_x şi a_y valori respectiv $–c\ x/r^3$ şi $–c\ y/r^3$

vom da x_0 , y_0 şi vitezele iniţiale $v_x (0)$ şi $v_y (0)$. În plus, alegem arbitrar un interval de timp Δt suficient de mic.

Obţinând noua poziţie după un timp Δt calculăm acceleraţiile:

$$a_x = -c\ x/r^3 \text{ şi } a_y = -c\ y/r^3$$

plecând de la noile valori ale lui x, y, z, ceea ce ne permite a calcula noile viteze. Obţinem, deci, din aproape în aproape, prin iteraţii succesive, poziţiile şi vitezele în timp.

Un rezultat mult mai bun vom obţine dacă vom lua o viteză din mijlocul intervalului şi nu de la început, cum am luat în exemplul nostru.

Începem prin a evalua viteza la un interval de timp $\Delta t/2$. Apoi, pentru a calcula poziţia în $x(\Delta t)$, adăugăm la $x(0)$ termenul $\Delta t * v(\Delta t/2)$.

Restul procedurii este identic cu ceea ce am expus mai sus unde înlocuim viteza de debut al intervalului cu o viteză la mijlocul intervalului de timp plus produsul lui Δt cu acceleraţia la timpul t.

Bibliografie

[Esc88] **Esclozas,** *Calcul matricel*, Langages et Informatique, Toulouse, 1988.

[LGT95] **P. Laurent-Gengoux and D. Trystram,** *Comprendre l'informatique numerique,* Langages et Informatique, Toulouse, 1995.

[IIJ89] **C. Ignat, C. Ilioi and T. Jucan**, *Elemente de informatica si calcul numeric*, (vol.1-2), Edit. Univ. Iasi, 1989.