Noțiuni introductive

Erori. Condiționare numerică. Stabilitatea algoritmilor. Complexitatea algoritmilor.

Metodele numerice reprezintă tehnici prin care problemele matematice sunt reformulate astfel încât să fie rezolvate numai prin operații aritmetice. Prin trecerea de la infinit la finit, diferențial la algebric, neliniar la liniar problemele complicate sunt înlocuite de probleme mai simple care au aceeaşi sau "aproape" aceeaşi soluție. Astfel soluțiile obținute prin aplicarea metodelor numerice reprezintă doar aproximații ale soluțiilor problemelor originale, şi deci implică erori.

Sursele erorilor şi clasificarea lor

Se pot distinge trei tipuri de erori în cazul aplicării de metode numerice pentru rezolvarea unei probleme:

- Erori provenite din simplificarea modelului fizic, pentru a fi descris într-un model matematic; erori din măsurătorile inițiale sau erori din calcule anterioare. Aceste tipuri de erori se numesc ***erori inerente***.
- Erori datorate metodei utilizate-de exemplu, trunchierea unei serii infinite (mai precis aproximarea sumei unei serii printr-o sumă parțială), sau considerarea unui termen cu un rang "suficient" de mare pentru a aproxima limita unui şir. Aceste erori sunt numite ***erori de metodă*** sau ***erori de trunchiere***.
- Erori datorate reprezentării datelor şi efectuării calculelor într-o aritmetică a virgulei mobile. Aceste erori se numesc ***erori de rotunjire***.

Erorile inerente sunt anterioare aplicării metodei numerice, iar erorile de trunchiere şi de rotunjire apar în timpul calculului numeric.

Erori absolute şi erori relative

Eroarea absolută = valoare aproximativă - valoare exactă

$$\textit{Eroarea relativă} = \frac{\text{eroare absolută}}{\text{valoare exactă}}$$

Din aceste definiții se obține:

Valoare aproximativă = (valoare exactă)(1 + eroare relativă)

Eroarea absolută nu ține seama de ordinul de mărime al valorilor comparate. De exemplu, o eroare în centimetri este mai importantă dacă lungimea calculată este de 100 cm, decât dacă este de 100 km. De aceea, eroarea relativă se raportează la valoarea reală. Adesea eroarea relativă se exprimă în procente:

$$\frac{\text{eroare absolută}}{\text{valoare exactă}} \cdot 100\ \%$$

De obicei valoarea exactă nu este cunoscută. De aceea nici eroarea (absolută sau relativă) nu poate fi calculată, şi doar se estimează valorile limită ale acesteia. Se utilizează majoranți pentru modulul erorii (sau norma erorii, dacă se lucrează într-un spațiu normat).

Tot datorită necunoașterii valorii exacte, frecvent eroarea relativă se raportează la valoarea aproximativă.

Erori ale datelor şi erori de calcul

Considerăm următoarea problemă tipică: calculul valorii unei funcții $f : \mathbf{R} \to \mathbf{R}$ pentru un argument dat. Fie:

x = valoarea de intrare exactă
x* = valoare de intrare aproximativă
f(x) = rezultatul dorit
f* = funcția aproximativă de calcul
Eroarea totală este dată de:

$$f^*(x^*) - f(x) = (f^*(x^*) - f(x^*)) - (f(x^*) - f(x))$$

Deci

Eroare totală = eroare de calcul + eroare propagată a datelor,

unde,

Eroare de calcul = $f^*(x^*) - f(x^*)$
Eroare a datelor = $x^* - x$.

Algoritmul nu are nici un efect asupra erorii propagate a datelor.

Erori de trunchiere şi erori de rotunjire

Eroare de trunchiere = diferența dintre rezultatul exact (pentru datele de intrare curente) şi rezultatul furnizat de un algoritm dat utilizând aritmetica exactă.

Eroare de rotunjire = diferența dintre rezultatul produs de un algoritm dat utilizând aritmetica exactă şi rezultatul produs de acelaşi algoritm utilizând o aritmetică cu precizie limitată (de exemplu aritmetica virgulei mobile).

Eroarea de calcul este suma dintre eroarea de trunchiere şi eroarea de rotunjire, dar de obicei una dintre acestea predomină.

Erori apriori şi erori aposteriori

Să presupunem că dorim să calculăm $y = f(x)$, unde $f : \mathbf{R} \to \mathbf{R}$, dar obținem o valoare aproximativă y^*.

Eroare apriori = $\Delta x = x^* - x$

Eroare aposteriori = $\Delta y = y^* - y$, unde $f(x^*) = y^*$.

De exemplu, dacă aproximăm $\sqrt{y}$ prin $y^* = 1.4$, eroarea aposteriori absolută este

$$| \Delta y | = | y^* - y | = | 1.4 - 1.41421\ldots | \approx 0.0142$$

iar eroarea aposteriori relativă este aproximativ 1%.

Pentru a calcula eroarea apriori, observăm că $\sqrt{1.96} = 1.4$. Eroarea apriori absolută este

$$| \Delta x | = | x^* - x | = | 1.96 - 2 | \approx 0.04,$$

iar eroarea apriori relativă este aproximativ 2%.

Analiza erorilor apriori

Ideea analizei erorilor din punct de vedere a erorilor apriori este următoarea: soluția aproximativă este soluția exactă a unei probleme modificate.

Soluția aproximativă se consideră "bună" dacă este soluție exactă pentru o problemă cu datele "uşor" perturbate.

Deseori eroarea apriori este mai uşor de estimat decât eroarea aposteriori.

Condiționarea numerică. Factor de condiționare.

Problema se numeşte ***bine condiționată*** dacă variațiile relative ale soluției au acelaşi ordin de mărime cu variațiile relative ale datelor de intrare ce le cauzează.

Problema este ***rău condiționată*** dacă modificările relative care au loc în soluției pot fi mult mai mari decât cele ale datelor de intrare.

Factorul de condiționare se defineşte prin:

$$\text{cond} = \frac{\left|\text{variatia relativă a solutiei}\right|}{\left|\text{variatia relativă a datelor de intrare}\right|}$$

Să revenim la calculul y = f(x), unde f : $\mathbf{R} \to \mathbf{R}$. Să presupunem că se obține valoarea aproximativă y*. Fie x* cu proprietatea că f(x*) = y*.

Avem $\text{cond} = \dfrac{\left|\dfrac{f(x^*)-f(x)}{f(x)}\right|}{\left|\dfrac{x^*-x}{x}\right|} = \dfrac{\left|\dfrac{\Delta y}{y}\right|}{\left|\dfrac{\Delta x}{x}\right|}$.

Problema este rău condiționată, dacă factorul de condiționare

$$\text{cond} >> 1.$$

Factorul de condiționare acționează ca un "factor de amplificare" legând eroarea aposteriori de eroarea apriori:

| eroarea relativă aposteriori | = cond × | eroarea relativă apriori |

De obicei factorul de condiționare nu este cunoscut exact şi poate varia în funcție de datele de intrare. De aceea se utilizează o estimație margine superioară pentru cond. Deci

$$\text{| eroarea relativă aposteriori |} \lessapprox \text{cond} \times \text{| eroarea relativă apriori |}.$$

Considerăm un exemplu de estimare pentru factorul de condiționare. Să presupunem că se evaluează funcția f pentru data de intrare aproximativă

$$x^* = x + \Delta x$$

în locul datei exacte de intrare x. Eroarea absolută aposteriori este

$$f(x + \Delta x) - f(x) \approx f'(x)\,\Delta x$$

iar eroarea relativă aposteriori este

$$\frac{f(x+\Delta x)-f(x)}{f(x)} \approx \frac{f'(x)\Delta x}{f(x)}$$

Factorul de condiționare este

$$\text{cond} \approx \left|\frac{\dfrac{f'(x)\Delta x}{f(x)}}{\dfrac{\Delta x}{x}}\right| = \left|\frac{x f'(x)}{f(x)}\right|.$$

Stabilitatea algoritmilor

Noțiunea referitoare la algoritmi analoagă condiționării numerice a problemelor este ***stabilitatea***. Un algoritm de rezolvare a unei probleme este ***stabil*** dacă rezultatul produs este soluția exactă a aceleaşi probleme cu datele "uşor" perturbate.

În cazul algoritmilor stabili efectul erorii de calcul nu este mai puternic decât efectul erorii (mici) a datelor de intrare.

Un algoritm instabil poate amplifica mult perturbațiile date de erorile de calcul.

Acuratețea metodelor

Acuratețea se referă la apropierea soluției calculate de soluția exactă a problemei. Stabilitatea algoritmului nu garantează acuratețea. Aceasta depinde în egală măsură de buna condiționare a problemei şi de stabilitatea algoritmului. Inacuratețea poate rezulta din aplicarea unui algoritm stabil unei probleme rău condiționate, ca şi din aplicarea unui algoritm instabil unei probleme bine condiționate.

Aplicarea (cu ajutorul calculatorului) unui algoritm stabil unei probleme bine condiționată garantează obținerea soluției cu o precizie bună, în sensul că eroarea relativă a soluției calculate față de soluția exactă este de ordinul de mărime al erorilor de reprezentare a datelor în calculator.

Complexitatea algoritmilor

În ***evaluarea complexității unui algoritm*** se ține cont de două aspecte

- ***timpul*** necesar execuției algoritmului (dat de numărul de operații elementare)
- ***spațiul de memorie*** necesitat de algoritm

În general nu este posibil să obținem simultan un timp de execuție mai scurt precum şi un necesar de spațiu de memorare mai mic. Progresele tehnologice din ultima vreme impun drept criteriu primordial criteriul timp.

Fie n numărul de date de intrare pentru un anumit algoritm (eventual considerăm n egal cu numărul de locații de memorie necesare pentru memorarea datelor inițiale). Fie $T_S(n)$ timpul cerut de algoritm pentru un anumit set de n date de intrare S. Vom nota $\tau(n)$ ***timpul cerut de algoritm în cazul cel mai defavorabil***, i.e.:

$$\tau(n) = \sup \{T_S(n) : S \text{ este un set de date de intrare de dimensiune } n \}$$

În general nu este posibil să determinăm o formulă pentru $\tau(n)$. În acelaşi timp ne interesează comportarea lui $\tau(n)$ pentru valori mari ale lui n. În acest sens introducem următoarele notații:

- $\tau(n) = O(f(n))$ dacă $\exists\ C > 0$, $n_0 \in \mathbf{N}$ cu $\tau(n) \le C\ f(n)\ \forall\ n \ge n_0$

Interpretare: τ ***are o creştere mai lentă decât f***

- $\tau(n) = o(f(n))$ dacă $\lim_{n\to\infty} \frac{\tau(n)}{f(n)} = 0$

Interpretare: τ ***are o creştere mai strict mai lentă decât f***

- $\tau(n) = \theta\ (f(n))$ dacă $\exists\ C_1, C_2 > 0$, $n_0 \in \mathbf{N}$ cu $C_1 f(n) \le \tau(n) \le C_2 f(n)\ \forall n \ge n_0$

Interpretare: τ ***are o creştere la fel de lentă ca f***

- $\tau(n) = \Omega(f(n))$ dacă $f(n) = O(\tau(n)$

Interpretare: f are o creştere mai lentă decât τ

O şi o se folosesc pentru a stabili marginile superioare ale timpului de execuție, iar Ω pentru limita inferioară a timpului de execuție.

Un algoritm se numeşte ***algoritm polinomial*** dacă $\tau(n) = O(P(n))$, unde τ este timpul cerut de algoritm, iar P este un polinom.