Nazomercursus Numerieke Wiskunde

Uitgewerkte bordaantekeningen

Kees Vuik

Deze uitwerking volgt de handgeschreven aantekeningen.

2. Interpolatie

2.2 Lineaire interpolatie

Motivatie: interpolatie wordt gebruikt wanneer waarden in een tabel beschikbaar zijn, bij het oplossen van differentiaalvergelijkingen, bij geometrische beschrijvingen en bij functies op de computer. De centrale vraag is hoe groot de fout is.

Herhaal Taylor: eerste orde plus een restterm:

$$f(x) = f(x_0) + (x-x_0)f'(x_0) + \cdots + ((x-x_0)^{n+1}/(n+1)!)\, f^{(n+1)}(\xi)$$

Lineaire interpolatie geeft een gemiddelde schatting tussen twee punten x_0 en x_1. De interpolant L_1 is een rechte lijn met $L_1(x_0)=f(x_0)$ en $L_1(x_1)=f(x_1)$:

$$L_1(x) = L_{01}(x)\, f(x_0) + L_{11}(x)\, f(x_1)$$

$$L_{01}(x) = (x-x_1)/(x_0-x_1), \quad L_{11}(x) = (x-x_0)/(x_1-x_0)$$

Dit zijn de Lagrange-basisfuncties: $L_{01}(x_0)=1$, $L_{01}(x_1)=0$, enzovoort.

Voorbeeld: benader sin(37°) uit tabelwaarden bij 36° en 38°. We gebruiken hiervoor de tabelwaarden

x	sin x
36°	0.58778525
38°	0.61566148

$$\sin(37°) = 0.60181502 \qquad L_1(37°) = 0.601723 \qquad \text{fout} \simeq 0.9 \times 10^{-4}$$

Omdat 37° precies tussen 36° en 38° ligt, is dit een bijzonder transparant voorbeeld van lineaire interpolatie. De uitkomst ligt zeer dicht bij de werkelijke waarde; de fout is van orde 10^{-4}.

Vervolgens gebruiken we de foutformule uit stelling 2.2.1 en maken we onderscheid tussen interpolatie, extrapolatie en voorspelling.

De fout van lineaire interpolatie kan worden geschreven als:

$$f(x) - L_1(x) = \tfrac{1}{2}\,(x-x_0)(x-x_1)\, f''(\xi)$$

Interpolatie versus extrapolatie. Binnen $[x_0,x_1]$ is de fout beheersbaar via de interpolatiefout. Buiten het interval kunnen de gewichten groter worden en kan een kleine fout in de tabelwaarden worden versterkt.

$$|f(x) - L_1(x)| \le \tfrac{1}{8}\,(x_1-x_0)^2 \max_{\xi\in(x_0,x_1)}|f''(\xi)|$$

Bij interpolatie leidt dit tot een foutbijdrage van hoogstens ε; bij extrapolatie kunnen de gewichten de fout versterken.

$$|L_1(x)-\tilde{L}_1(x)| \le |(x-x_1)/(x_0-x_1)|\varepsilon + |(x-x_0)/(x_1-x_0)|\varepsilon.$$

$$\tilde{L}_1(x)=((x-x_1)/(x_0-x_1))\tilde{f}(x_0)+((x-x_0)/(x_1-x_0))\tilde{f}(x_1)$$

Voor de afrondfout nemen we $|f(x)-\tilde{f}(x)|<\varepsilon$ en schrijven we de interpolant met de afgeronde waarden uit:

Als de beschikbare functiewaarden slechts tot op ε nauwkeurig zijn, dan volgt voor de lineaire interpolant een foutbijdrage die wordt begrensd door de absolute waarden van de twee Lagrange-gewichten. Bij interpolatie is hun som effectief 1; bij extrapolatie kan de versterking aanzienlijk groter zijn.

Hogere-orde interpolatie

Een kleinere stap h lost niet zonder meer alle numerieke problemen op; bovendien maakt een hogere polynoomgraad de benadering complexer.

Met meer steunpunten ontstaat een interpolatiepolynoom van hogere orde. Voor drie punten:

$$L_2(x) = f(x_0)L_{02}(x) + f(x_1)L_{12}(x) + f(x_2)L_{22}(x)$$

$$L_{02}(x) = ((x-x_1)(x-x_2))/((x_0-x_1)(x_0-x_2))$$

Algemeen is de interpolatiefout:

$$f(x) - L_n(x) = ((x-x_0)\cdots(x-x_n)/(n+1)!)\, f^{(n+1)}(\xi)$$

De afrondfout blijft echter gekoppeld aan de interpolatiegewichten. Hogere orde betekent dus niet automatisch een betere numerieke benadering.

2.3 Splines

Splines combineren een goede lokale benadering met een gladde aansluiting tussen intervallen. Het idee is om op ieder deelinterval een laaggradig polynoom te gebruiken, in plaats van één hooggradig polynoom over het hele gebied.

- Lineaire splines: continu, maar met knikken.
- Kwadratische splines: meer gladheid.
- Kubische splines: de standaardkeuze voor een gladde aansluiting.

Splines kunnen worden voorgesteld als dunne latjes die door interpolatiepunten worden gelegd: we zoeken een gladde aansluiting met minimale spanning. We onderscheiden lineaire, kwadratische en kubische splines.

Voor een natuurlijke kubische spline op knopen $x_0,\ldots,x_n$ wordt op $[x_k,x_{k+1}]$ geschreven:

$$s_k(x) = a_k(x-x_k)^3 + b_k(x-x_k)^2 + c_k(x-x_k) + d_k$$

$$h_k = x_{k+1} - x_k$$

De voorwaarden zijn: interpolatie in de knopen, continuïteit van de spline en van de eerste en tweede afgeleide, plus natuurlijke randvoorwaarden.

$$s(x_k)=f(x_k), \quad s' \text{ en } s'' \text{ continu}, \quad s''(x_0)=s''(x_n)=0$$

Voor ieder deelinterval schrijven we $s_k(x)=a_k(x-x_k)^3+b_k(x-x_k)^2+c_k(x-x_k)+d_k$ en $h_k=x_{k+1}-x_k$. Uit $s_k(x_k)=f_k$ volgt direct $d_k=f_k$.

Uit de voorwaarden volgen onder meer:

$$a_k = (b_{k+1}-b_k)/(3h_k)$$

$$c_k = (f_{k+1}-f_k)/h_k - h_k(2b_k+b_{k+1})/3$$

Voor de onbekenden $b_1,\ldots,b_{n-1}$ ontstaat een tridiagonaal lineair stelsel. Voor de natuurlijke spline gelden $b_o=b_n=0$.

4. Niet-lineaire vergelijkingen

Motivatie: niet-lineaire vergelijkingen ontstaan onder meer bij impliciete methoden voor beginwaardeproblemen en bij niet-lineaire randwaardeproblemen. Gegeven f, zoek p zodat

$$f(p)=0$$

Drie basisvragen:

- Bestaat er een oplossing p?
- Is de oplossing uniek?
- Hoe bepalen we de oplossing numeriek?

We formuleren drie vragen: bestaat de oplossing (existentie), is zij uniek (uniciteit), en hoe bepalen we haar (methode)? Vervolgens kiezen we als uitgangspunt een iteratieve methode.

Numerieke methoden leveren doorgaans een rij $p_0,p_1,p_2,\ldots$ die naar p convergeert.

Convergentieorde

De orde van convergentie α met asymptotische constante λ wordt beschreven door

$$\lim_{n\to\infty} |p_{n+1}-p| \,/\, |p_n-p|^{\alpha} = \lambda$$

α=1 geeft lineaire convergentie; α=2 kwadratische convergentie.

Bisectiemethode

Als f continu is op [a,b] en f(a)f(b)<0, dan ligt er volgens de tussenwaardestelling minstens één nulpunt in het interval. Halveer het interval en behoud telkens de helft waarin het teken verandert.

$$p_0 = (a_0+b_0)/2$$

De iteratie verloopt als volgt: als $f(p_0)=0$ zijn we klaar; anders wordt, afhankelijk van het teken van $f(a_0)f(p_0)$, $[a_0,p_0]$ of $[p_0,b_0]$ het nieuwe interval.

Stop bijvoorbeeld wanneer $|f(p_n)|<\varepsilon$ of wanneer de interval-/stapfout klein genoeg is. De methode is globaal convergent.

In het voorbeeld geldt $a_0=1$, $b_0=2$ en $p_0=1.5$, met $|p_0-\sqrt{2}|\approx 0.0858$. Daarna volgen onder meer $p_1=1.25$ en 1.375 als opeenvolgende middelpunten.

Voor $f(x)=x^2-2$ met $p=\sqrt{2}$ en startinterval [1,2] geeft de methode achtereenvolgens middelpunten zoals 1.5, 1.25, 1.375, … .

$$|p-p_n| \le (b-a)/2^{n+1}$$

Voor b−a=1 en een gewenste fout kleiner dan 10^{-3} is bijvoorbeeld $2^{-n} \le 0.001$ nodig; dus n is ongeveer 10.

Onbetrouwbaarheidsinterval bij afrondfouten

We noemen dit het ‘onbetrouwbaarheidsinterval’: door meet- of afrondfouten kan zelfs het waargenomen teken van f nabij het nulpunt onzeker worden.

Wanneer alleen een afgeronde functiewaarde beschikbaar is, bijvoorbeeld met $|\tilde{f}(x)-f(x)|<\varepsilon$, kan een tekenwisseling door de afronding onbetrouwbaar worden. Linearisatie rond een nulpunt p geeft:

$$f(x) \approx f(p) + (x-p)f'(p) = (x-p)f'(p)$$

$$|x-p| \approx \varepsilon/|f'(p)|$$

Een bijbehorend onzekerheidsinterval rond p heeft dus ruwweg de vorm

$$[\, p - \varepsilon/|f'(p)|\,,\; p + \varepsilon/|f'(p)|\,]$$

De grootte van f′(p) is cruciaal: bij een kleine afgeleide kan een kleine functiefout tot een relatief grote fout in het nulpunt leiden.

Newton–Raphson

Na bisectie en vaste-puntiteratie introduceren we Newton–Raphson via de raaklijn ofwel de eerste-orde Taylor-benadering.

Newton gebruikt de eerste-orde Taylor-benadering in p_n en kiest het nulpunt van de raaklijn:

$$\ell(x) = f(p_n) + (x-p_n)f'(p_n)$$

$$p_{n+1} = p_n - f(p_n)/f'(p_n)$$

Voor $f(x)=x^2-2$ en $p_0=1$ volgen bijvoorbeeld $p_1=1.5$, $p_2\approx1.4167$ en $p_3\approx1.4142$. Dicht bij een eenvoudig nulpunt is Newton veel sneller dan bisectie.

Newton is lokaal convergent: een geschikte startwaarde is belangrijk. Voor de iteratiefunctie $g(x)=x-f(x)/f'(x)$ geldt bij een eenvoudig nulpunt $g'(p)=0$, wat de snelle lokale convergentie verklaart.

Voor de vaste-puntvorm $g(x)=x-f(x)/f'(x)$ geldt bij het nulpunt $g'(p)=0$; dit is een verklaring voor de snelle lokale convergentie.

Voor $f(x)=x^2-2$ en $p_0=1$ vinden we $p_1=1.5$, $p_2\approx1.4167$ en $p_3\approx1.4142$. Hierbij is het lokale karakter van Newton belangrijk: een goede startwaarde is van belang.

Waarom Newton kwadratisch convergeert

Uit Taylor rond p_n, met $f(p)=0$, en de Newton-relatie volgt:

$$0 = f(p_n) + (p-p_n)f'(p_n) + \tfrac{1}{2}(p-p_n)^2 f''(\xi_n)$$

$$0 = f(p_n) + (p_{n+1}-p_n)f'(p_n)$$

Aftrekken geeft:

$$|p_{n+1}-p| = \tfrac{1}{2}\,|p-p_n|^2 \cdot |f''(\xi_n)/f'(p_n)|$$

$|p_{n+1}-p| = \tfrac{1}{2}\,|p-p_n|^2\,|f''(\xi_n)/f'(p_n)|$.

Onder gebruikelijke regulariteitsvoorwaarden en zolang f' niet te klein wordt, is de fout in de volgende stap dus evenredig met het kwadraat van de huidige fout: kwadratische convergentie.

Vergelijking van methoden

Methode	Sterk punt	Aandachtspunt
Bisectie	Robuust; globale convergentie bij tekenwisseling	Lineair en dus relatief langzaam
Picard / vaste-puntiteratie	Eenvoudige iteratieve vorm	Convergentie hangt af van de iteratiefunctie
Newton–Raphson	Zeer snel nabij een nulpunt	Lokale methode; afgeleide en startwaarde zijn belangrijk