

RA

DUPLICAAT

STICHTING
MATHEMATISCH CENTRUM
2e BOERHAAVESTRAAT 49
AMSTERDAM
REKENAFDELING

Cursus Wetenschappelijk Rekenaar A

Numerieke Wiskunde
deel I

door

T.J. Dekker



1965

RA

BIBLIOTHEEK MATHEMATISCH CENTRUM
AMSTERDAM

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam,
The Netherlands.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications; it is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O) and the Central Organization for Applied Scientific Research in the Netherlands (T.N.O), by the Municipality of Amsterdam, by the University of Amsterdam, by the Free University at Amsterdam, and by industries.

Hoofdstuk 1 Algol 60

1.0 Literatuur omtrent ALGOL 60

J.W. Backus, e.a., Revised Report on the Algorithmic Language ALGOL 60 (1962), kortweg genoemd: het Algol-rapport.

E.W. Dijkstra, Cursus programmeren in ALGOL 60 (Amsterdam 1962).

A. van der Sluys, Cursus ALGOL 60 (Utrecht 1963).

D.D. Mc Cracken, A guide to ALGOL programming (New York 1962).

1.1 Inleiding

ALGOL 60 (een samentrekking van 'Algorithmic Language 1960') is een internationaal aanvaarde taal, die ten doel heeft rekenprocessen (algorithmen) nauwkeurig en overzichtelijk te beschrijven.

Overzichtelijkheid wordt bereikt door zich nauw aan te sluiten bij de bekende wiskundige notaties, nauwkeurigheid door aan deze notatie enige nieuwe elementen toe te voegen, die ons in staat stellen de typische dynamiek van rekenprocessen weer te geven.

De taal is geschikt als communicatie-middel, enerzijds tussen mensen onderling, anderzijds tussen mens en machine.

Een in Algol 60 beschreven volledige berekening (programma) kan door een computer automatisch worden verwerkt.

Ter inleiding geven we het volgende voorbeeld:

Zij gevraagd te berekenen de wortels van de vierkantsvergelijking $ax^2 + bx + c = 0$.

We moeten verschillende gevallen onderscheiden:

Als $a \neq 0$, zijn er 2 wortels

$$x_{1,2} = \left(-b \pm \sqrt{b^2 - 4ac} \right) / 2a ,$$

die toegevoegd complex of reëel zijn, al naar gelang de discriminant $b^2 - 4ac$ negatief is of niet.

In het geval $a = 0$, en $b \neq 0$, is er slechts een (reële) wortel en als $a = b = 0$, is de vergelijking strijdig of identiek.

Voor een beschrijving van de gevraagde berekening hebben we nodig naast de gegeven coëfficiënten a , b en c , enige variabelen voor reëel en imaginair deel van de wortels en nog een geheel getal, dat het aantal wortels moet aangeven.

Een en ander wordt in het volgende ALGOL 60 programma beschreven.

```

begin real a,b,c,x1 real, x1 imag, x2 real, x2 imag; integer number
      of roots; input (a); input (b); input (c); NLCR;
      output (a); output (b); output (c);
  begin real disc, s;
    if a = 0 then
      begin if b = 0 then begin number of roots: = 0;
        x1 real: = 0 end else
          begin number of roots: = 1; x1 real: = - c/b end;
          x2 real: = 0; goto zeros
        end;
      number of roots: = 2; disc: = b2 - 4xaxc;
      if disc < 0 then
        begin x1 real: = x2 real: = - b / (2xa);
          x1 imag: = sqrt (- disc)/(2xa); x2 imag: = - x1 imag;
        end else
          begin s: = sqrt (disc);
            x1 real: = (- b + s)/(2xa);
            x2 real: = (- b - s)/(2xa);
          zeros: x1 imag: = x2 imag: = 0
          end
        end;
      if number of roots ≠ 0 then
        begin output (x1 real);
          if number of roots = 2 then
            begin output (x1 imag); output (x2 real); output (x2 imag)
              end
          end
        end
      end
    end
  end

```

Het kan nog mooier. Zo kan men in het geval van 2 reële wortels maatregelen nemen tegen het wegvallen van cijfers.

Opgave 0. Ga dit na.

1.2 Enige principes

- 1.2.1 Alle symbolen staan op een lijn achter elkaar. M.a.w. een Algol-programma is een rij opeenvolgende symbolen (zoals in gesproken taal). Terwille van de bladspiegel mag men overal spaties en overgang op een nieuwe regel inlassen. Deze "symbolen" hebben geen betekenis voor de Algol-tekst.
- 1.2.2 Als namen van variabelen, functies, enz. gebruikt men in Algol identifiers, die per definitie bestaan uit een of meer letters of cijfers en beginnen met een letter (Zie Algol rapport 2.4).
- 1.2.3 Voor getallen in Algol (zie Algol rapport 2.5) is als nieuwigheid ingevoerd het lage tientje gevolgd door een geheel getal. Bijvoorbeeld het getal

$9.34_{10} - 7$ heeft de waarde $9.34 \times 10^{(-7)}$.

- 1.2.4 In Algol-expressies (3.3) heeft elke operatie zijn symbool, dat nooit mag worden weggelaten. De bekende regel "Mijnheer Van Dalen Wacht Op Antwoord" geldt niet.

In Algol heeft machtsverheffing (\uparrow) voorrang op de multiplicatieve operaties (\times , $/$, \div), die onderling gelijkwaardig zijn; deze hebben weer voorrang op de additieve operaties ($+$, $-$). In schema:

$$\begin{array}{c} \uparrow \\ \times / \div \\ + - \end{array}$$

Opgaven

1. Welke van de volgende getallen zijn niet toegestaan in Algol en hoeveel verschillende waarden hebben de correcte Algol getallen?
 16.9; + 16,9; $169 \cdot 10^{-1}$; $+ .169 \cdot 10^2$;
 -.0007; $- .7 \cdot 10^{-3}$; $7 \cdot 10^{-4}$; $7 \cdot 10^{-4}$; $7 \cdot 0 \cdot 10^{-4}$.

2. Welke zijn identifiers? Welke niet-identifiers zijn expressies in Algol 60?

X	i12g	Kat	$x + 3$	$x^{\dagger} - 3$
42 Y	delta	A/M	Hoge Hoed	T1.4
(plein)	IA	X12	1 X 2	arctan
begin	2a	158	Eerste Bessel functie	
T1 punt 4	$2 \cdot 7 \cdot 10^{-k}$	B_7	B_{10}	$a_{10} - 3$
$2.54 \times 10^{(-n)}$				

3. Schrijf arithmetische expressies in Algol 60, die equivalent zijn met de volgende formules:

a. $x + y^3$	b. $(x + y)^3$	c. $x^{1.667}$
d. $A + \frac{B}{C}$	e. $\frac{A + B}{C}$	f. $A + \frac{B}{C + D}$
g. $\frac{A + B}{C + D}$	h. $\frac{A + B}{C + D} + X_1^2$	i. $\frac{A + B}{C + \frac{D}{F + G}}$
j. $1 + x + \frac{x^2}{2!} + \frac{x^3}{3!}$	k. $\left(\frac{p_1}{p_2}\right)^{g-1}$	l. $\frac{x}{1 + \frac{x^2}{3 + \frac{(2x)^2}{5 + \frac{(3x)^2}{7 + (4x)^2}}}}$
m. $a \cdot b + c^d - (2^x)^2$	n. $(x_1^3 + x_2^3 + x_3^3)$	
o. AREA = $2 \cdot P \cdot R \cdot \sin(\pi/P)$	p. CHORD = $2R \sin \frac{A}{2}$	
q. ARC = $2\sqrt{Y^2 + (4X^2/3)}$	r. S = $-\frac{\cos^4 X}{4}$	

4. Schrijf een assignment statement, die aan "pol" de waarde geeft van het polynoom $a_0 x^2 + a_1 x + a_2$, waarbij $a_0 = 3$, $a_1 = 7$, $a_2 = 4$.
5. Schrijf assignments statements, die aan de variabele f_0 de waarde 1 geven, aan f_1 de waarde $1/(x^2 + 1)$, aan f_2 de waarde $1/(4x^2 + 1)$ en aan INT de waarde $\frac{1}{2}(f_0 + 2f_1 + f_2)$.

1.3 Het type van arithmetische expressies

Arithmetische expressies kunnen zijn van type integer of van type real.

Hiervoor gelden de volgende regels:

- 1) constanten zijn van type integer als er geen $.$ of 10 in voorkomt, anders van type real.
- 2) het type van een variabele en een functie wordt bepaald door de betreffende declaratie.
- 3) som, verschil en product van twee expressies van type integer is eveneens van type integer; de berekening is in dit geval volgens Algol 60 exact. Is echter minstens een der expressies van type real, dan krijgen we een resultaat van type real.
- 4) de operatie $"/$ geeft altijd een resultaat van type real.
- 5) de operatie $"\div"$ mag alleen worden toegepast op twee expressies van type integer en geeft een resultaat van type integer, nl. het quotiënt van de deling met rest 0 of een rest zo dicht mogelijk bij 0, maar met hetzelfde teken als het deeltal.
- 6) voor machtsverheffing, zie het Algol rapport 3.3.4.3.

1.4 Machine-voorstelling van getallen

- 1) een geheel getal (de waarde van een expressie van type integer) wordt in een machine 10-tallig of 2-tallig voorgesteld. Welk talstelsel wordt gebruikt, is voor de programmeur van weinig belang; het enige waar het op aan komt is, dat integers exact worden voorgesteld en dat daarop de operaties exact worden uitgevoerd. In de praktijk is er meestal een beperkte capaciteit voor de integers, d.w.z. er is een getal M , zodat alleen integers in absolute waarde $\leq M$ toelaatbaar zijn. Ontstaan in een programma integers buiten deze range, dan kan het effect ongedefinieerd worden, dit in afwijking van Algol 60.

2) de waarde van een expressie van type real wordt in een machine meestal voorgesteld met zgn. drijvende komma (floating point representation), d.w.z. een getal x wordt voorgesteld door een breukdeel a en een exponent b , waarbij $x = a \times g^b$, het grondtal g zijnde 10 of 2.

De exponent heeft een beperkte range, die meestal wel royaal is. Het breukdeel voldoet aan $|a| \leq 1$ en is vaak genormeerd zodanig, dat $|a| \geq 1/g$ of $a = 0$.

Voor a staan een beperkt aantal g -tallige cijfers ter beschikking, zodat $a \times g^t$ een geheel getal is, waarbij $t =$ het aantal cijfers van a . Wij zeggen dan, dat de drijvende komma voorstelling een relatieve precisie van t cijfers heeft.

Als x een drijvend getal $\neq 0$ is, en x' een van zijn burens, dan geldt dat $\left| \frac{x' - x}{x} \right|$ ongeveer gelijk is aan $10^{-(t)}$.

Het resultaat van een arithmetische operatie moet, indien het niet exact voorstelbaar is, naar een naburig drijvend getal worden afgerond. Dit levert een relatieve fout in het resultaat van ca. $10^{-(t)}$.

1.5 Meta-taal

Om een taal als Algol 60 te definiëren, moet men zich bedienen van een andere taal, een zgn. "meta-taal". Als men ook de meta-taal netjes wil definiëren, heeft men een meta-meta-taal nodig, enz. Dit is niet aantrekkelijk en daarom gebruiken wij als meta-taal een bekende taal, nl. Nederlands (of in het rapport Engels), verrijkt met de volgende conventies.

De Algol-begrippen, waarover we willen praten, zetten we tussen speciale meta-haken, b.v.: <letter >.

Het meta-symbool " ::= " betekent: "heeft de volgende vorm" en het meta-symbool "| " betekent "of".

Voorbeeld: <identifiser> ::= <letter> | <identifiser><letter> | <identifiser><digit> betekent:

Een identifier heeft de volgende vorm: een letter, of een identifier gevolgd door een letter of een identifier gevolgd door een digit. De meta-symbolen zijn zelf geen Algol-symbolen en komen dus niet in een Algol-programma voor.

In afwijking van het Algol rapport zullen we ook weleens losjes het symbool " ... " gebruiken. B.v.:

<identifier> ::= <letter> <letter or digit> ...<letter or digit>
 waarbij <letter or digit> ::= <letter> | <digit> .

Deze conventie zullen we af en toe terwille van de duidelijkheid gebruiken.

1.6 Statements

Elementaire statements, d.w.z. statements die niet worden opgebouwd uit andere statements, heten "unlabelled basic statements".

De belangrijkste hieronder is de assignment statement (toekennings statement), die de volgende vorm heeft:

<toekennings statement> ::= <variabele> := <expressie> |

<variabele> := ... := <variabele> := <expressie>

Het effect is, dat de variabele een nieuwe waarde krijgt, die ter beschikking blijft, totdat een nieuwe toekenning aan deze variabele wordt uitgevoerd. (Zie Dijkstra pag.4,5).

Staan er meerdere variabelen links, dan moeten ze hetzelfde type hebben. Als het type integer is en de expressie is van type real, dan wordt er een afronding naar een dichtstbijzijnde gehele waarde ingelast.

(Zie ook Algol rapport 4.2 en Dijkstra pag.15-17).

Statements worden uitgevoerd in de volgorde, waarin ze zijn neergeschreven, tenzij anders vermeld. Deze andere vermelding geschiedt in

de eerste plaats door de goto statement (bestemming statement), meestal van de vorm go to <label>

Een label is een identifier, die met een ":" voor een statement wordt geplaatst, wat we als volgt kunnen formuleren:

```
<labelled statement> ::= <label>:<statement>
```

De statements kunnen we verder als volgt indelen (hierbij vereenvoudigen we enigszins de in het Algol rapport gegeven indeling):

```
<statement> ::= <unlabelled basic statement> | <labelled statement> |
<samengestelde statement> | <blok> | <conditionele statement> | <for
statement>
```

Deze indeling is wel volledig, maar niet disjunct. Zo moet een gelabelde conditionele statement ook als conditionele statement beschouwd worden.

```
<ongelabelde samengestelde statement> ::= begin <statement>; ...;
<statement> end
<ongelabeld blok> ::= begin <declaratie>; ...;<declaratie>;
<statement>; ...;<statement> end
<program> ::= <blok> | <samengestelde statement>
```

Bij dit laatste moeten we evenwel bedenken, dat niet elk blok of samengestelde statement een programma is. Voor een programma gelden enige extra eisen, zoals volledige declaratie van alle gebruikte identifiers, behalve labels, standaardfuncties, e.d.

Opgaven

- 6) Schrijf statements, die het volgende doen:
- als a groter dan b , maak x gelijk aan 16.9, maar als a kleiner of gelijk aan b is, maak y gelijk aan 23.1;
 - als $\rho + \theta$ kleiner is dan 10^{-6} , spring naar de statement met label "klaar", anders naar de statement met label "nogeens";
 - spring naar de label "eerste" als $i = 1$, naar de label "tussen" als $1 < i < N$, naar de label "laatste" als $i = N$.

- 7) Schrijf een Algol-programma, dat de waarde van $n!$ ("faculteit") berekent en uittipt voor $n = 1, 2, \dots, 20$. Typ op elke regel de waarde van n en de bijbehorende waarde van $n!$ door middel van een procedure statement van de vorm:
- ```
print2(<expressie> , <expressie>).
```

- 8) Het stelsel lineaire vergelijkingen

$$ax + by = c$$

$$dx + ey = f$$

heeft nul, een of oneindig veel oplossingen.

Schrijf een Algol-programma, dat

- de coëfficiënten  $a, b, \dots, f$  inleest door middel van een procedure statement van de vorm `read1(<variabele>);`
  - nagaat hoeveel oplossingen er zijn en de variabele "aantal oplossingen" gelijk maakt aan 0, 1 of 2 (2 in geval er oneindig veel oplossingen zijn) en dit nummer ook uittipt;
  - als het aantal oplossingen  $\geq 1$ , een oplossing typt.
- Gebruik voor het typen een procedure statement van de vorm
- ```
print({expressie}).
```

- 9) Welke waarde hebben x en y aan het eind van de volgende Algol-programma's:

```
begin comment programma nr.1;
  integer a, x; real b, y; a: = 2; b: = 3;
  x: = a↑2 + 2×a×b; y: = sign (x-2);
  x: = x - y; y: = y/a
end
```

```
begin comment programma nr.2;
  integer a,x; real b, y;
  a: = 2; b: = 3.510 - 3;
  b: = a↑2 / 5 × b - 10 - 3 + 1;
  x: = b;
  if a < b then y: = a else y: = (a × b + x) / 3
end
```

- 10) Op een band staan achtereenvolgens een positief geheel getal n en n reële getallen a_1, \dots, a_n .
Schrijf een programma dat het gemiddelde

$$m = \frac{1}{n} \sum_{j=1}^n a_j$$

en de standaardafwijking

$$s = \sqrt{\frac{1}{n} \sum_{j=1}^n a_j^2 - m^2}$$

van a_1, \dots, a_n berekent en uitprint.

Zuinig geheugengebruik wordt op prijs gesteld. (Opgave 10 is uit examen voor het Diploma A van Wetenschappelijk Rekenen dd. 3-9-1964).

1.7 Conditionele statements

Deze hebben de volgende vorm:

```
<condit.stat.> ::= <if clause><uncond.stat.>|
                <if clause> <uncond.stat.> else <statement>|
<if clause> <for statement>|<label>:<condit.stat.>
```

Hierbij geldt: <if clause> ::= if <Boolean expressie> then

De conditionele statement komt dus voor in korte vorm of in lange vorm met else gevolgd door een statement.

De korte vorm wordt gebruikt als er in het "anders" geval niets hoeft te gebeuren. (M.a.w. men mag else gevolgd door een dummy statement weglaten).

Achter de if clause, dus achter then mag geen conditionele statement volgen, omdat het anders te ingewikkeld wordt en niet steeds duidelijk is, welke then bij welke else hoort.

Heeft men na "then" iets conditioneels te doen, dan kapselt men dit in tussen de statement-haken "begin" en "end". Hierdoor gaat een conditionele statement over in een samengestelde statement, die tot de inconditionele statements wordt gerekend.

De for-statement neemt hierbij een bijzondere plaats in.

Achter een if clause gevolgd door een for-statement mag (helaas) geen else komen.

1.8 For statements

Deze hebben de volgende vorm:

```
<for statement> ::= <for clause> <statement>|<label>:<for statement>
waarbij <for clause> ::= for <variabele> := <for list> do
en <for list> ::= <for list element>, ..., <for list element>.
```

Er zijn 3 soorten for list elementen, nl.

1) een arithmetische expressie (<ae>).

Voor een dergelijk for list element wordt de statement achter "do" precies één keer uitgevoerd.

2) een step-until-element van de vorm: <ae> step <ae> until <ae>.

Voor dit element wordt de statement achter "do" nul of meer malen uitgevoerd, volgens het schema vermeld in het Algol-rapport 4.6.4.2.

Waarschuwing. Als de lopende variabele en de arithmetische expressies in een step-until-element niet alle van type integer zijn, houde men rekening met mogelijke inexactheid van de berekening.

B.v. bij de for clause: "for x := 0 step 0.1 until 1 do"

is het niet zeker of de waarde 1 wordt meegenomen.

In zo'n geval geve men de bovengrens een veilig toegiftje:

for x := 0 step 0.1 until 1.05 do

Of men voert een andere lopende variabele in van type integer:

for tienmaalx := 0 step 1 until 10 do begin x := tienmaal x/10; ...

(Zie ook Cursus van der Sluys 1.18).

3) een while-element van de vorm: <ae> while <Boolean expressie>.

Dit element heeft tot gevolg, dat de statement achter do nul

of meer malen wordt uitgevoerd zolang de Boolean expressie true blijft

(zie Algol-rapport 4.6.4.3).

Opmerking. Na voltooiing van een for-statement is de waarde van de lopende variabele (helaas) ongedefinieerd.

Opgaven

- 11) Welk getal wordt er getypt (door de procedure "print") in het volgende Algol-programma:

```
" begin integer i, s; s := 0;
   for i := 0 step 2 until 8 do
     begin i := i + 1; s := s + 1 end;
   print (s)
end"
```

- 12) Welke getallen typt het volgende Algol-programma:

```
"begin comment Test for statement. Vgl. voorbeeld van Dijkstra p.26;
   integer n;
   for n := -1, 0, 2, 3 do
     begin integer k, m;
       m := 3; print (n + 100);
       for k := 0 step m until n do
         begin print (k); m := m - 1 end
       end
     end
   end"
```

- 13) Bereken $\sqrt{x_1^2 + \dots + x_n^2}$, aangenomen dat de elementen x_i ($i = 1, \dots, n$) zijn gegeven in een één-dimensionaal array.

- 14) Zij gegeven een matrix M van de orde n in een twee-dimensionaal array, waarbij beide indices lopen van 1 tot en met n.
- Schrijf een declaratie voor het array, waarin M wordt gegeven.
 - Schrijf een stuk programma in de vorm van een blok, dat het spoor van M uitrekent.
 - Schrijf een blok, dat de norm N van de matrix M uitrekent, gedefinieerd door:

$$N = \max_{1 \leq i \leq n} \sum_{j=1}^n |M_{ij}|$$

- 15) Schrijf een Algol-programma, dat de binomiaal-coëfficiënten $\binom{n}{k}$, $k = 0, \dots, n$, $n = 0, \dots, 20$ uittypt, zodanig dat voor elke waarde van n de bijbehorende binomiaalcoëfficiënten op een regel verschijnen en voor de volgende n op de volgende regel.
- Gebruik voor overgang op een nieuwe regel de procedure "NLCR" en voor het typen van de getallen een procedure statement van de vorm: "outinteger (<expressie van type integer>)"

Opmerking. Zie ook cursus van der Sluys opgaven: 7, 8, 14, 15, 16, 19-24, 26.

Verschil tussen declaraties en specificaties

Ofschoon sommige declaraties dezelfde vorm hebben als sommige specificaties, zijn er toch belangrijke verschillen.

1) Declaraties staan aan het begin van een blok (dus achter begin). Specificaties staan in een procedure heading (dus vóór begin, als de body met begin begint).

2) Declaraties introduceren nieuwe grootheden (quantities) in het betreffende blok. Specificaties vermelden alleen iets omtrent soort en type, waaraan de corresponderende actuele parameters bij aanroep moeten voldoen.

3) Declaraties zijn voor elke gebruikte grootheid verplicht, behalve voor labels, die als het ware gedeclareerd worden door ze met een ":" voor een statement te plaatsen (zie A.R. 4.1.3) en behalve eventueel voor standaardfuncties en standaard procedures, die in Algol implementaties zonder declaratie beschikbaar kunnen zijn.

Specificaties van non-value-parameters zijn niet verplicht.

Sommige implementaties eisen ze evenwel en het is daarom goede gewoonte alle specificaties maar te geven.

Voor value-parameters zijn de specificaties wel verplicht. In dit geval heeft de specificatie een tweeslachtig karakter, nl. het vermelden van restricties voor de actuele parameters (specificerend karakter) en het vastleggen van het type van de lokale variabelen of arrays, die door het value-mechanisme worden geïntroduceerd (declarerend karakter).

Vanwege het declarerend karakter zijn de specificaties van value-parameters verplicht.

4) Declaraties en specificaties hebben vaak een verschillende vorm.

Voorbeelden van declaraties

real x, y, z

array A [1 : 10]

real procedure tan(x); value x; real x;
tan: = sin (x)/cos(x)

Voorbeelden van specificaties

real x, y, z

array A

real procedure tan

Opmerking: Formele non-value-parameters zijn geen grootheden (quantities), maar zijn bestemd om te worden vervangen door grootheden, nl. de corresponderende actuele parameters van de aanroep van de procedure of functie-procedure. Formele value-parameters zijn grootheden, lokaal t.o.v. de procedure, die vóór de inserering en uitvoering van de procedure-body de waarde van de corresponderende actuele parameters toegerekend krijgen. Daarom hebben voor value-parameters de specificaties tevens een declarerend karakter en zijn ze verplicht.

Voorbeeld: Berekening van polynomen

Zij gevraagd te berekenen een polynoom P van de graad n, m.a.w.

$$P(x) = a_0 + a_1x + \dots + a_nx^n .$$

Dit kan op twee manieren:

- 1) de machten van x worden berekend (bv. door herhaald vermenigvuldigen met x) en daarna worden deze met hun coëfficiënt vermenigvuldigd en bij elkaar opgeteld. Het aantal operaties is dan n-1 machtsverheffingen (of vermenigvuldigingen) plus n vermenigvuldigingen en optellingen.
- 2) P(x) wordt berekend volgens de formule:

$$a_0 + x(a_1 + x(a_2 + \dots + x(a_{n-1} + xa_n) \dots)) .$$

Het aantal operaties is nu n vermenigvuldigingen en optellingen. De tweede manier gaat dus sneller en daarom zal een rekenaar deze methode kiezen.

Voor Algol procedures, die een polynoom uitrekenen, zie cursus van der Sluys hoofdstuk III: pol, pol 1, ..., pol 5.

Hier volgt nog een andere declaratie. Deze is geschreven als declaratie voor een functie procedure, die voor gegeven argument x de waarde van een polynoom berekent van willekeurige graad (als graad < 0 de waarde 0

leverend). Er wordt verondersteld, dat de coëfficiënten staan in array A [0 : graad] in volgorde van opklimmende machten van x.

```

real procedure polynoom (x, graad, A);
value x; real x; integer graad; array A;
begin integer i; real r; r = 0;
      for i := graad step -1 until 0 do r := r*x + A[i];
      polynoom := r
end polynoom;

```

Een mogelijke aanroep van "polynoom" zou kunnen luiden:
 polynoom (i/10, 5, c) .

Opgaven (Zie ook cursus van der Sluys, opgaven 30 t/m 36, 39 en 40).

16) Een procedure, die een matrix maal vector operatie moet uitvoeren, zou de volgende heading kunnen hebben:

```
"procedure MAVEC (n, A, x, y); integer n; array A, x, y;"
```

Het 2-dimensionale array A bevat de matrix, de 1-dimensionale arrays x en y zijn resp. voor de aangeboden vector en voor de resulterende vector Ax. Alle indices lopen van 1 tot en met n.

a) Schrijf een geschikte body voor de procedure MAVEC.

b) Schrijf een programma, dat, met behulp van MAVEC, de matrix H, gedefinieerd door

$$H_{ij} = 1 / (i + j - 1) \quad (i, j = 1, \dots, 7),$$

vermenigvuldigt met de vector v, gedefinieerd door:

$$v_i = i \quad (i = 1, \dots, 7)$$

en de vector Hv uittipt.

17) Schrijf een declaratie voor een functie procedure van de gedaante "absmax (<expression>, <expression>)", die het maximum van de absolute waarden van de twee gegeven parameters berekent.

18) Schrijf een declaratie voor een functie procedure ter berekening van

$$x^2 + \sqrt{1 + 2x + 3x^2}.$$

Schrijf vervolgens toekenningsstatements, die deze functie gebruiken en aan alfa, beta en gamma respectievelijk toekennen:

$$\frac{6.9 + y}{y^2 + \sqrt{1 + 2y + 3y^2}},$$

$$\frac{2.1z + z^4}{4z^2 + \sqrt{1 + 4z + 12z^2}},$$

$$\sin^2 y + \sqrt{1 + 2 \sin y + 3 \sin^2 y}.$$

19) Schrijf een declaratie voor een functie procedure Y(x), gedefinieerd door:

$$\begin{aligned} Y(x) &= 1 + \sqrt{1 + x^2} && \text{als } x < 0 \\ &= 0 && \text{als } x = 0 \\ &= 1 - \sqrt{1 + x^2} && \text{als } x > 0. \end{aligned}$$

Schrijf vervolgens een blok, dat deze functie gebruikt en de waarde van Y(x) berekent voor x = -1 met 0.2 oplopend tot en met 0 en van 0 af oplopend met 0.1 tot en met 1.0.

Laat de resultaten achter in een non-locaal array Y tabel [-10 : +10] .

20) In C. Hastings, Approximations for digital computers (1955) p. 141, staat de volgende benadering voor de functie 10^X , geldig voor het interval $0 \leq x \leq 1$:

$$"(10^X)^* = [1 + a_1X + a_2X^2 + a_3X^3 + a_4X^4]^2$$

$$a_1 = 1.1499196 \quad a_3 = .2080030$$

$$a_2 = .6774323 \quad a_4 = .1268089"$$

a) Schrijf een declaratie voor een functie-procedure, die voor willekeurige X in het bovengenoemde interval de bijbehorende waarde van $(10^X)^*$ berekent.

b) Schrijf een blok, dat deze (non-locale) real procedure gebruikt en het verschil $(10^X)^* - 10^X$ berekent voor X = 0.0 met 0.1 oplopend tot en met 1.0 en het resultaat aflevert in een non-locaal array D [0 : 10] .

Quiz

- 1) Geef de syntactische definitie van <identifiser> volgens het Algol-rapport (Backus' notatie).
- 2) Welke verschillende soorten declaraties kent U?
Geef van elk een voorbeeld.
- 3) Geef, zonder gebruikmaking van de for-statement, een equivalente beschrijving van een for-statement van de gedaante:
"for V: = E step F until G do statement S".
- 4) Schrijf de volgende formule in de vorm van een equivalente Algol-expressie:

$$\left(\frac{n(n-1)}{(2n-1)} \right)^3 / 4ab + \frac{-y + \sqrt{y^2 - 4xz}}{2x} .$$

Hoofdstuk 2. Interpolatie0. Literatuur

1. National Physical Laboratories, Modern Computing Methods (London 1962)
2. F.B. Hildebrand, Introduction to Numerical Analysis
3. Nautical Almanac Office, Interpolation and allied Tables (London 1956)
4. Fletcher, Miller & Rosenhead, An Index of Mathematical Tables (London, 1e ed. 1946, 2e ed. 1962)
5. L.J. Comrie, Chambers' six-figure Mathematical Tables
6. Nat. Bur-Standards, Tables of Lagrangian interpolation coefficients
7. R.A. Buckingham, Numerical Methods
8. C. Gram, Selected numerical methods
9. R.W. Hamming, Computing for scientists and engineers
10. D.R. Hartree, Numerical Analysis
11. C. Hastings, Approximations for digital computers
12. A.S. Honscholder, Principles of N.A.
13. Z. Kopal, Numerical Analysis
14. K.S. Kunz, Numerical Analysis
15. J. Kuntzmann, Méthodes Numériques
16. G.N. Lance, Numerical methods for high speed computers
17. N. Macon, Numerical Analysis
18. W.E. Milne, Numerical Calculus
19. K.L. Nielsen, Methods in N.A.
20. Ralson and Wilf, Mathematical methods for digital computers
21. Salvadori and Bacon, Numerical methods in engineering

22. J. Todd, Survey of N.A.

23. Zurmühl, Praktische Mathematik

Opmerking: Zie ook de uitstekende Bibliografie met commentaar in [1].

1. Inleiding

1.1) Het doel van de numerieke analyse is het vinden van een benaderde oplossing, waar het bepalen van een preciese oplossing onmogelijk of ondoenlijk is. Het gaat hierbij dus om praktische uitvoerbaarheid van een rekenproces. De gekozen methode is vaak een compromis tussen de graad van nauwkeurigheid, die men wenst, en de middelen, waarover men beschikt. Numerieke processen zijn dus per se eindig, d.w.z. bestaan uit een eindig aantal stappen, dat liefst zo gering mogelijk moet zijn. Een numerieke methode moet dus worden gewaardeerd naar 1) betrouwbaarheid 2) bereikbare precisie 3) snelheid 4) compactheid en overzichtelijkheid 5) algemene bruikbaarheid.

1.2) De objecten van de numerieke analyse, d.w.z. de grootheden die we willen berekenen, zijn voornamelijk getallen en functies. Onder getallen verstaan we hier gehele, rationale, reële en complexe getallen. Wij noemen de verzameling der gehele getallen I, die der reële getallen R en die der complexe getallen C. Grootheden, die essentieel een gehele waarde hebben, wil men meestal wel exact berekenen. Daarom is de integer-arithmetiek in ALGOL 60 exact. Voor rationale en reële getallen gebruikt men gewoonlijk een beperkte verzameling van zgn. representeerbare getallen. Een benadering wordt dan verkregen, door een getal af te ronden naar een dichtbij gelegen representeerbaar getal. Complexe getallen worden gehanteerd als paren reële getallen.

Onder "functies" zullen we steeds verstaan "eenduidige functies".

De verzameling van argumenten, waarvoor een functie gedefinieerd is heet de definitie-verzameling of het domein van de functie en de verzameling van

functie-waarden heet de waarden-verzameling van de functie. Een functie f met domein A en waarden-verzameling B , is dus een voorschrift, dat aan ieder element x van A een eenduidig bepaald element $f(x)$ van B toevoegt.

Wij beschouwen in de eerste plaats functies met reële argumenten en waarden, dus domein en waarden-verzameling beide gelegen in \mathbb{R} . De verzameling van dit soort functies duiden we aan met $\mathbb{R} \rightarrow \mathbb{R}$.

Een bijzonder geval vormen de functies met geheel argument en reële waarde (verzameling $\mathbb{I} \rightarrow \mathbb{R}$). Is het domein een eindige rij van opeenvolgende integers, dan hebben we te doen met een vector. B.v. een vector x voegt aan de integers 1 t/m n toe de reële getallen x_1, \dots, x_n , wat wij noteren als $x = (x_1, \dots, x_n)$. Een functie kan worden benaderd door bij elk in het domein gegeven argument de exact berekende functie-waarde af te ronden naar een naburige representeerbare waarde. Dit is echter alleen doenlijk voor zeer eenvoudige functies (bv: de functies c plus en c maal gedefinieerd door: c plus $(x) = c + x$ en c maal $(x) = c \times x$, waarbij c een constante voorstelt). Voor ingewikkelder functies moet meestal ook de berekening van de waarde door een benaderend proces vervangen worden. Een machtig middel voor het verkrijgen van bruikbare benaderingen van functies is de interpolatie.

Opgaven

- 16) Bereken benaderde waarden van
- $\sin(x)$
- met de formule

$$\sin(x) \approx x - x^3/6$$

voor $x = 0.0174532925$ (dat is 1 graad) en

voor $x = 0.523598775$ (dat is 30 graden).

Idem met de formule

$$\sin(x) \approx x - x^3/6 + x^5/120.$$

- 17) Bereken de waarde van
- $(10^X)^{**}$
- met de volgende formule uit [11]:

$$(10^X)^{**} = \left[1 + 1.1499196 X + .6774323 X^2 + .2080030 X^3 + .1268089 X^4 \right]^2$$

voor $X = 0.0$ (0.2) 1.0 en

voor $X = .70710678$.

- 18) Bereken de vectoren
- $y = Ax/\lambda$
- en
- $z = Ax - \lambda x$
- , waarbij
- $\lambda = 7.605551275$
- en de matrix A en de vector x zijn:

$$A = \begin{pmatrix} 6 & 3 \\ 3 & 2 \end{pmatrix}, \quad x = \begin{pmatrix} .8816745988 \\ .4718579255 \end{pmatrix}.$$

Idem voor $\lambda = .6300899976_{10}3$,

$$A = \begin{pmatrix} 420 & 210 & 140 & 105 \\ 210 & 140 & 105 & 84 \\ 140 & 105 & 84 & 70 \\ 105 & 84 & 70 & 60 \end{pmatrix}, \quad x = \begin{pmatrix} .7926082912 \\ .4519231209 \\ .3224163986 \\ .2521611697 \end{pmatrix}.$$

2. Interpolatie door middel van polynomen

Wij willen een gegeven functie f uit de verzameling $R \rightarrow R$ benaderen door een eenvoudiger te berekenen benaderende functie f^{**} . Wij spreken van interpolatie als f^{**} enerzijds wordt gekozen uit een gegeven klasse van functies, anderzijds wordt vastgelegd door de eis, dat f^{**} in een aantal gegeven punten met de functie f overeenstemt. De gegeven punten heten basispunten. Wij beschouwen nu interpolatie door middel van polynomen; de benaderende functie f^{**} is een polynoom van graad kleiner dan een gegeven getal n .

Voorbeeld: $n = 2$, d.w.z. f^{**} is een lineaire functie, m.a.w.

$$f^{**}(x) = a_0 + a_1x .$$

Wij kunnen f^{**} in 2 punten met de gegeven functie f laten overeenstemmen. Als x_0 en x_1 de basispunten zijn hebben we de volgende vergelijkingen voor a_0 en a_1 :

$$a_0 + a_1x_0 = f(x_0) ,$$

$$a_0 + a_1x_1 = f(x_1) .$$

En we vinden

$$f^{**}(x) = \frac{x_1 f(x_0) - x_0 f(x_1)}{x_1 - x_0} + \frac{f(x_1) - f(x_0)}{x_1 - x_0} x \quad (\text{Grünert})$$

$$= f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0} (x - x_0) \quad (\text{Newton})$$

$$= \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1) . \quad (\text{Lagrange}).$$

3. Intermezzo over lineaire vergelijkingen en determinanten

Een stelsel lineaire vergelijkingen kan worden opgelost door eliminatie. In een eliminatie-stap wordt een veelvoud van een vergelijking bij een andere vergelijking opgeteld, en wel zodanig, dat een bepaalde onbekende niet meer in de resulterende vergelijking voorkomt. Een belangrijke invariante grootte hierbij is de determinant van de matrix van het stelsel.

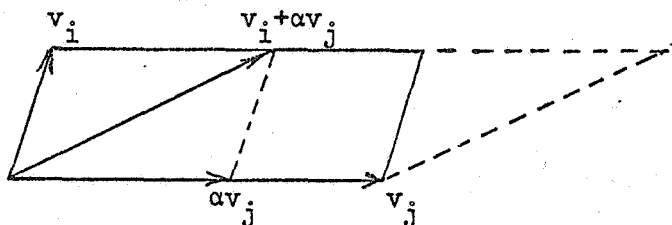
Definitie: Aan iedere vierkante (reële of complexe) matrix is toegevoegd een (reëel of complex) getal, de determinant van de matrix genaamd. Deze toevoeging is volledig gekarakteriseerd door de volgende twee eigenschappen:

- 1) de determinant is invariant bij (rij-)eliminatie, d.w.z. de determinant van een matrix verandert niet, als een veelvoud van een rij van de matrix bij een andere rij wordt opgeteld.
- 2) de determinant van een driehoeksmatrix is gelijk aan het produkt van de elementen van de hoofddiagonaal.

Notatie: De determinant van matrix A wordt genoteerd als $\det(A)$ of als $|A|$.

Meetkundig beeld: Beschouw de n-dimensionale parallelotoop (dat is generalisatie van parallelogram en parallelopipedum) opgespannen door de n rij-vectoren van de gegeven $n \times n$ -matrix. Dan is de determinant van deze matrix de inhoud van deze parallelotoop, voorzien van een teken samenhangend met de oriëntatie van de opspannende vectoren.

Een eliminiatiestap komt neer op een oppervlakte-trouwe "verandering" in het vlak, opgespannen door de twee betrokken vectoren. Het paar vectoren v_i, v_j wordt vervangen door het paar $v_i + \alpha v_j, v_j$.



Het parallellogram opgespannen door v_i en v_j wordt veranderd in een parallellogram van dezelfde oppervlakte en men kan bewijzen, dat dan ook de inhoud van de hele parallelotoop en de oriëntatie, dus ook de determinant, invariant blijven.

Consequenties: Uit de eigenschappen (1) en (2) van de determinant kunnen we de volgende belangrijke eigenschappen afleiden:

3) Homogeniteit: wordt een rij van een matrix met een getal α vermenigvuldigd, dan ook de determinant.

Bewijs: beschouw een eliminatie, die de gegeven matrix overvoert in een driehoeksmatrix (zo'n eliminatie is er altijd). Vermenigvuldigen wij een rij met α , dan is er een volkomen analoge eliminatie, die leidt tot een driehoek, die uit de vorige driehoek ontstaat, door de betreffende rij met α te vermenigvuldigen. De determinant wordt dus volgens eigenschap (2) ook met α vermenigvuldigd.

3a) In het bijzonder voor $\alpha = 0$:

als in een rij van een matrix alle elementen 0 zijn, dan is ook de determinant gelijk aan 0.

In dit geval leidt eliminatie tot een driehoeksvorm, waarin een 0-rij voorkomt.

3b) Een matrix met twee gelijke of evenredige rijen heeft een determinant 0. Deze matrix kan immers door een eliminatiestap worden overgevoerd in een matrix met een 0-rij.

4) Verwisseling van twee rijen van een matrix voert de determinant over in zijn tegengestelde.

Bewijs: beschouw twee verschillende rijen v_i, v_j . Door eliminatie kunnen we achtereenvolgens krijgen:

$$\begin{array}{cccc} v_i & v_i+v_j & v_i+v_j & v_j \\ v_j & v_j & -v_i & -v_i \end{array}$$

Vervangen we tenslotte $-v_i$ door v_i dan zijn v_i en v_j verwisseld en de determinant krijgt wegens (1) en (3) zijn tegengestelde waarde.

5) De determinant van een matrix kan worden berekend door ontwikkeling naar een kolom, in formule:

$$\det(A) = \sum_{i=1}^n A_{ij} A_{ij}^* \quad \text{voor elke } j \text{ tussen } 1 \text{ en } n.$$

Hierbij is A_{ij}^* de i,j -de minor van A , gedefinieerd als $A_{ij}^* = (-1)^{i+j}$ maal de i,j -de onderdeterminant, dat is de determinant van de matrix, die uit A ontstaat door daarin de i -de rij en de j -de kolom te schrappen.

Bewijsschets: Voor een driehoeksmatrix is de formule gemakkelijk af te leiden. De enige bijdrage komt van de term $A_{jj} A_{jj}^*$, de andere termen voor $i \neq j$ zijn 0 omdat dan hetzij A_{ij} , hetzij A_{ij}^* gelijk aan 0 is. Verder blijkt de formule invariant te zijn voor rij-eliminatie. De formule voldoet dus aan de boven gegeven karakterisering.

6) De determinant van matrix A is gelijk aan de determinant van de getransponeerde matrix A^T .

Een belangrijk gevolg hiervan is, dat in alle bovengenoemde eigenschappen de woorden "rij" en "kolom" mogen worden verwisseld. Determinanten zijn dus invariant bij kolom-eliminatie; vermenigvuldiging van een kolom met α levert een α maal zo grote determinant; verwisseling van 2 kolommen keert de determinant van teken en de determinant kan worden verkregen door ontwikkeling naar een rij.

Beschouwen wij nu een stelsel van n vergelijkingen, lineair, in de n onbekenden x_1, \dots, x_n . De algemene gedaante is

$$\sum_{j=1}^n A_{ij} x_j = b_j, \quad i=1(1)n.$$

De matrix A der coëfficiënten A_{ij} heet de matrix van het stelsel, de kolom-vector b heet rechterlid-vector en de kolom-vector x de vector der onbekenden.

Dan kunnen wij dit stelsel ook schrijven in de vorm $Ax = b$.

Of kortweg: $\sum_{j=0}^{n-1} a_j x_i^j = f(x_i)$, $i=0(1)n-1$.

Dit is een stelsel van n vergelijkingen, lineair in de n onbekenden a_0, a_1, \dots, a_{n-1} .

De matrix van dit stelsel is de Van der Monde-matrix van de orde n:

$$V(x_0, \dots, x_{n-1}) = \begin{pmatrix} 1 & x_0 & \dots & x_0^{n-1} \\ 1 & x_1 & \dots & x_1^{n-1} \\ \dots & \dots & \dots & \dots \\ 1 & x_{n-1} & \dots & x_{n-1}^{n-1} \end{pmatrix}.$$

Duiden we deze matrix kortweg aan met V, zij verder \bar{f} de rechterlid-vector, dan kan het stelsel worden geschreven in de vorm:

$$Va = \bar{f}.$$

De determinant van het stelsel, dus $\det(V)$, berekenen we door kolom-eliminatie als volgt:

Verminder de kolommen, beginnend met de laatste en eindigend met de voorste op een na, elk met x_{n-1} maal de vorige kolom. Omdat eliminatie de determinant invariant laat, krijgen wij dan

$$\det(V(x_0, \dots, x_{n-1})) =$$

$$= \begin{vmatrix} 1 & x_0 - x_{n-1} & x_0(x_0 - x_{n-1}) & \dots & x_0^{n-2}(x_0 - x_{n-1}) \\ 1 & x_1 - x_{n-1} & x_1(x_1 - x_{n-1}) & \dots & x_1^{n-2}(x_1 - x_{n-1}) \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_{n-2} - x_{n-1} & x_{n-2}(x_{n-2} - x_{n-1}) & \dots & x_{n-2}^{n-2}(x_{n-2} - x_{n-1}) \\ 1 & 0 & 0 & \dots & 0 \end{vmatrix}$$

$$= (-1)^{n-1} (x_0 - x_{n-1}) \dots (x_{n-2} - x_{n-1}) \det(V(x_0, \dots, x_{n-2})) =$$

$$= \prod_{k=0}^{n-2} (x_{n-1} - x_k) \det(V(x_0, \dots, x_{n-2})).$$

Dit proces herhalende komen wij tenslotte uit bij $\det(V(x_0))$, die natuurlijk de waarde 1 heeft en wij vinden dus voor de determinant van Van der Monde

$$\det(V) = \det(V(x_0, \dots, x_{n-1})) = \prod_{i=1}^{n-1} \prod_{k=0}^{i-1} (x_i - x_k) .$$

(Hierbij houden wij ons aan de afspraak, dat een leeg produkt (d.w.z. een produkt van 0 factoren) de waarde 1 heeft, zodat wij voor $n=1$ netjes krijgen $\det(V(x_0)) = 1$.)

Keren wij nu terug tot het stelsel lineaire vergelijkingen (4.1) voor de coëfficiënten van f^* .

Omdat de basispunten onderling verschillend zijn is de determinant van het stelsel, $\det(V)$, niet nul.

Volgens de theorie der lineaire vergelijkingen is er dus een unieke oplossing voor de onbekenden a_0, \dots, a_{n-1} . Deze oplossing kan worden verkregen door successieve eliminatie.

Opgaven:

- 24) Bereken de coëfficiënten van f^{**} voor polynoominterpolatie van de orde $n=3$.
- 25) Bereken de volgende determinanten (door eliminatie):

$$\begin{vmatrix} 420 & 210 & 140 & 105 \\ 210 & 140 & 105 & 84 \\ 140 & 105 & 84 & 70 \\ 105 & 84 & 70 & 60 \end{vmatrix} \quad \begin{vmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \\ 4 & 5 & 6 & 0 \end{vmatrix} \quad \begin{vmatrix} 9 & 1 & 3 & 1 & 9 \\ 4 & 1 & 2 & 1 & 4 \\ -2 & -1 & 0 & 1 & 2 \\ -4 & -1 & 0 & 1 & 4 \\ -9 & -1 & 0 & 1 & 9 \end{vmatrix} .$$

- 26)a) Zij gegeven de volgende tabel van een functie f :

x	-2	-1	1	2
f(x)	.1353	.3679	2.718	7.389

Bereken de coëfficiënten van het interpolerend polynoom f_4^* van graad < 4 , dat voor de gegeven waarden van x met f overeenstemt. Wat is de waarde van f_4^* voor $x = \frac{3}{2}$?

- b) Bereken eveneens de coëfficiënten van het polynoom f_2^* van graad < 2 , dat voor $x=1$ en voor $x=2$ met f overeenstemt. Wat is de waarde van $f_2^*\left(\frac{3}{2}\right)$?

Erratum: pag. 24. De nummers van de opgaven moeten zijn 21, 22 en 23.

Oplissing opgave 24

Voor de orde $n = 3$ vinden we de coëfficiënten van $f^*(x) = a_0 + a_1x + a_2x^2$ als de oplossing van het stelsel:

$$\begin{aligned} a_0 + x_0 a_1 + x_0^2 a_2 &= f_0 \\ a_0 + x_1 a_1 + x_1^2 a_2 &= f_1 \\ a_0 + x_2 a_1 + x_2^2 a_2 &= f_2. \end{aligned}$$

Eliminatie van a_0 geeft:

$$\begin{aligned} a_1 + (x_0 + x_1)a_2 &= (f_1 - f_0)/(x_1 - x_0) \\ a_1 + (x_1 + x_2)a_2 &= (f_2 - f_1)/(x_2 - x_1). \end{aligned}$$

Vervolgens a_1 eliminerend vinden we (vgl. formule 10.3 en het bewijs van stelling 10.1 pag. 51):

$$a_2 = f[x_0, x_1, x_2] = \frac{f_0}{(x_0-x_1)(x_0-x_2)} + \frac{f_1}{(x_1-x_0)(x_1-x_2)} + \frac{f_2}{(x_2-x_0)(x_2-x_1)}.$$

$$\text{Dus } a_1 = (f_2 - f_1)/(x_2 - x_1) - (x_1 + x_2)a_2 =$$

$$= - \frac{(x_1+x_2)f_0}{(x_0-x_1)(x_0-x_2)} - \frac{(x_0+x_2)f_1}{(x_1-x_0)(x_1-x_2)} - \frac{(x_0+x_1)f_2}{(x_2-x_0)(x_2-x_1)}.$$

$$\text{En } a_0 = f_0 - x_0^2 a_2 - x_0 a_1 =$$

$$= \frac{x_1x_2 f_0}{(x_0-x_1)(x_0-x_2)} + \frac{x_0x_2 f_1}{(x_1-x_0)(x_1-x_2)} + \frac{x_0x_1 f_2}{(x_2-x_0)(x_2-x_1)}.$$

(Dit is gelijk aan $\sum_{j=0}^2 \mathcal{L}_j^3(0) f_j$, zie sectie 6 p. 39-40).

4.2. De interpolatie-formule van Grünert

Voor de liefhebbers volgt hier de oplossing van het stelsel (4.1) voor willekeurige orde n . De uitwerking van opgave 24 suggereert de coëfficiënten a_j te schrijven als een lineaire combinatie van de functie-waarden $f_i = f(x_i)$, dus:

$$(4.2.1) \quad a_j = \sum_{i=0}^{n-1} f_i C_{ij}, \quad j = 0(1)n - 1$$

waarbij de C_{ij} noch van f noch van x afhangen.

Dan hebben we dus:

$$(4.2.2) \quad f^{**}(x) = \sum_{j=0}^{n-1} a_j x^j = \sum_{j=0}^{n-1} \left(\sum_{i=0}^{n-1} f_i C_{ij} \right) x^j = \\ = \sum_{i=0}^{n-1} f_i \left(\sum_{j=0}^{n-1} C_{ij} x^j \right) = \sum_{i=0}^{n-1} f_i \ell_i^n(x).$$

De verwisseling der beide sommaties leidt kennelijk tot de Lagrange-vorm (zie sectie 6). De C_{ij} zijn dus de coëfficiënten der Lagrange-polynomen $\ell_i^n(x)$, dus de symmetrische functies der basispunten x_k ($k \neq i$) gedeeld door de leidende coëfficiënt $\prod_{\substack{k=0 \\ k \neq i}}^{n-1} (x_i - x_k)$.

$$(4.2.3) \quad \text{Stellen we } N(i) = \prod_{\substack{k=0 \\ k \neq i}}^{n-1} (x_i - x_k) \text{ dan krijgen we dus:}$$

$$C_{i,n-1} = 1/N(i).$$

$$C_{i,n-2} = - \left(\sum_{\substack{k=0 \\ k \neq i}}^{n-1} x_k \right) / N(i).$$

$$(4.2.4) \quad C_{i,j} = (-1)^{n-1-j} \left(\sum_{k_{j+1}, \dots, k_{n-1}} x_{k_{j+1}} \dots x_{k_{n-1}} \right) / N(i),$$

waarbij de sommatie-indices k_{j+1}, \dots, k_{n-1} alle mogelijke waarden $\neq i$ aannemen, die voldoen aan $0 \leq k_{j+1} < k_{j+2} < \dots < k_{n-1} \leq n-1$.

Voor $j = 0$ krijgen we tenslotte:

$$C_{i,0} = (-1)^{n-1} x_0^{**} \dots^{**} x_{i-1}^{**} x_{i+1}^{**} \dots^{**} x_{n-1} / N(i).$$

De formule

$$(4.2.5) \quad f^{**}(x) = \sum_{j=0}^{n-1} a_j x^j, \text{ waarbij } a_j \text{ kunnen worden verkregen uit}$$

(4.2.4) en (4.2.1), heet expliciete polynoom-interpolatie-formule of interpolatie-formule van Grünert van de orde n .

5. Interpolatie door middel van een lineaire familie van functies

De verzameling van polynomen van graad $< n$ is een n -dimensionale lineaire familie van functies.

Onder een n -dimensionale lineaire familie van functies verstaan we een verzameling bestaande uit alle lineaire combinaties van n gegeven basisfuncties G_j , $j = 0(1)n - 1$.

Een interpolerende functie f^* heeft dus, als element van zo'n lineaire functie, de vorm:

$$f^*(x) = \sum_{j=0}^{n-1} a_j G_j(x).$$

Kiezen we in het bijzonder de basisfuncties $G_j(x) = x^j$ ($j = 0(1)n - 1$), dan krijgen we als lineaire familie de verzameling van polynomen van graad $< n$.

De eis, dat f^* in n gegeven punten met de gegeven functie f overeenstemt, leidt nu tot het stelsel (vgl. formule (4.1)):

$$(5.1) \quad \sum_{j=0}^{n-1} a_j G_j(x_i) = f(x_i), \quad i = 0(1)n - 1.$$

Dit is eveneens een stelsel van n vergelijkingen, lineair in de n onbekenden a_0, \dots, a_{n-1} .

De matrix M van dit stelsel is gedefinieerd door

$$(5.2) \quad M_{ij} = G_j(x_i), \quad i, j = 0(1)n - 1.$$

Het stelsel (5.1) krijgt dan de vorm: $Ma = \bar{f}$.

Wij hebben dan:

Stelling: Als de basisfuncties G_j en de basispunten x_i zodanig zijn dat $\det(M) \neq 0$ (waarbij M is gedefinieerd in 5.2), dan is er precies één interpolerende functie in de gegeven familie te vinden, die op de basispunten met de gegeven functie f overeenstemt.

Toepassingen

Wij zullen dit in de eerste plaats toepassen voor het verkrijgen van verschillende formules van polynoom-interpolatie.

$$1) G_j(x) = x^j.$$

Deze keuze leidt tot het stelsel 4.1, waarvan de onbekenden zijn de coëfficiënten van het interpolerende polynoom (Grünert's formule).

$$2) G_j(x) = \prod_{\substack{k=0 \\ k \neq j}}^{n-1} (x - x_k).$$

Dit leidt tot de formule van Lagrange.

Het stelsel voor de coëfficiënten heeft als matrix een diagonaal-matrix.

$$3) G_j(x) = \prod_{k=0}^{j-1} (x - x_k).$$

Dit leidt tot de formule van Newton.

Het stelsel voor de coëfficiënten heeft de driehoeksvorm.

Wij zullen later ook interpolaties beschouwen waarbij de klasse van interpolerende functies geen polynomen zijn. Hier noemen we slechts een paar voorbeelden.

$$4) G_j(x) = e^{\lambda_j x},$$

waarin λ_j gegeven constanten zijn. Dit leidt tot exponentiële interpolatie.

$$5) G_j(x) = \cos \frac{jx}{2} \quad \text{als } j \text{ even} \\ = \sin \frac{(j+1)x}{2} \quad \text{als } j \text{ oneven.}$$

Dit leidt tot Fourier-approximatie, welke speciaal geschikt is voor periodieke functies.

Opgaven

27) Bereken de volgende determinanten:

$$\begin{vmatrix} 125 & -24 \\ 101 & 8 \end{vmatrix}, \begin{vmatrix} 7 & -3 & 5 \\ 6 & 11 & 9 \\ -9 & -7 & 5 \end{vmatrix}, \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & 3 & 9 & 27 \\ 1 & 5 & 25 & 125 \\ 1 & 7 & 49 & 343 \end{vmatrix},$$

$$\begin{vmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 \\ 1 & 3 & 9 & 27 & 81 \\ 1 & 4 & 16 & 64 & 256 \\ 1 & 5 & 25 & 125 & 625 \end{vmatrix}$$

28) Bepaal het lineaire stelsel voor de coëfficiënten van de veelterm f^* van graad < 3 die met de volgende tabel overeenstemt.

argument	functie-waarde
8	.90309
9	.95424
10	1.00000

Bereken vervolgens de coëfficiënten van f^* , de determinant van de kleine matrix en de waarde van $f^*(7)$.29) Bepaal het lineaire stelsel voor de coëfficiënten van het polynoom f^* van graad < 4 overeenstemmend met de tabel

argument	functie-waarde
5	.69897
6	.77815
8	.90309
10	1.00000

Los het stelsel op en bepaal de determinant van de kleine matrix. Bereken tenslotte $f^*(7)$.

30) Bereken het product van de volgende twee matrices

$$A = \begin{pmatrix} -210.09 & +210 & +140 & +105 \\ +210 & -490.09 & +105 & +84 \\ +140 & +105 & -546.09 & +70 \\ +105 & +84 & +70 & -570.09 \end{pmatrix}$$

$$B = \begin{pmatrix} +.7926082912 & +.0291933232 \\ +.4519231209 & -.3287120557 \\ +.3224163986 & +.7914111458 \\ +.2521611697 & -.5145527500 \end{pmatrix}$$

Bereken ook de lengte van de kolommen van B en hun scalair product.

6. Interpolatie-formule van Lagrange

Wij kiezen de basisfuncties G_j , $j = 0(1)n - 1$, als volgt:

$$G_j(x) = \prod_{\substack{k=0 \\ k \neq j}}^{n-1} (x - x_k).$$

Als wij stellen

$$\pi_n(x) = \prod_{k=0}^{n-1} (x - x_k) = (x - x_0)(x - x_1) \dots (x - x_{n-1}),$$

kunnen wij G_j ook schrijven als

$$G_j(x) = \pi_n(x) / (x - x_j),$$

waarbij we dan wel moeten afspreken, dat in het punt $x = x_j$ de functie continu bij de omgeving van x_j aansluit.

Wij zoeken nu een interpolerende functie f^* van de gedaante

$$f^*(x) = \sum_{j=0}^{n-1} a_j G_j(x),$$

die in de punten x_i , $i = 0(1)n - 1$, overeenstemt met de gegeven functie f , m.a.w. de coëfficiënten moeten voldoen aan het stelsel (5.1).

De matrix M van het stelsel (vgl. 5.2) voldoet nu aan

$$M_{ij} = G_j(x_i) = 0 \quad \text{als } i \neq j,$$

m.a.w. M is een diagonaal-matrix.

Dit volgt onmiddellijk uit de definitie van de basis functies G_j .

De determinant van M bevat juist alle factoren $x_i - x_k$ voor $i \neq k$ zodat blijkbaar $\det(M) = (\det(V))^2 \neq 0$.

Er is dus een unieke oplossing voor de coëfficiënten a_j , die, wegens het diagonaal zijn van M , heel gemakkelijk te vinden is.

We hebben namelijk:

$$a_j = f(x_j) / G_j(x_j), \quad j = 0(1)n - 1.$$

Dus

$$(6.1) \quad f^*(x) = \sum_{j=0}^{n-1} \frac{G_j(x)}{G_j(x_j)} f(x_j) = \sum_{j=0}^{n-1} \left(\prod_{\substack{k=0 \\ k \neq j}}^{n-1} \frac{x - x_k}{x_j - x_k} \right) f(x_j).$$

Dit is de interpolatie-formule van Lagrange van de orde n , en van de rang n , of, wat men vaak zegt: "n-punts Lagrange"-formule.

De coëfficiënt van $f(x_j)$ schrijven we als $L_j^n(x)$, in formule:

$$(6.2) \quad L_j^n(x) = \frac{G_j(x)}{G_j(x_j)} = \prod_{\substack{k=0 \\ k \neq j}}^{n-1} \frac{x - x_k}{x_j - x_k}, \quad j = 0(1)n - 1.$$

De n-punts Lagrange-formule krijgt dan de vorm:

$$(6.3) \quad f^*(x) = L_0^n(x) f(x_0) + L_1^n(x) f(x_1) \dots + L_{n-1}^n(x) f(x_{n-1}).$$

7. Enige belangrijke eigenschappen van polynomen

Hier volgen enige eigenschappen van polynomen met reële (of complexe) coëfficiënten in de vorm van stellingen.

Stelling 7.1, de zgn. "Hoofdstelling van de Algebra".

Een polynoom van graad > 0 heeft minstens één reëel of complex nulpunt.

Stelling 7.2. Een polynoom van graad n heeft precies n nulpunten, waarvan sommige kunnen samenvallen en dan meervoudig tellen.

Precieser gezegd: Een polynoom van graad n kan worden geschreven in de vorm:

$$a_n(x - w_1)(x - w_2) \dots (x - w_n),$$

waarbij w_i reëel of complex zijn en a_n de coëfficiënt van x^n is, dus $a_n \neq 0$.

Deze stelling volgt gemakkelijk uit stelling 1.

Stelling 7.3. Een polynoom van graad $< n$, die minstens n verschillende nulpunten heeft, is identiek nul. Dit volgt meteen uit stelling 2.

Stelling 7.4. Als twee polynomen van graad $< n$ voor n verschillende argumenten overeenstemmen, dan zijn zij identiek. Dit volgt uit stelling 3.

Stelling 7.5. Als voor n verschillende argumenten x_i , $i = 0(1)n - 1$ bijbehorende functiewaarden f_i gegeven zijn, dan is er precies één polynoom, dat voor deze n argumenten de voorgeschreven waarde heeft.

Bewijs: Dat hoogstens één polynoom voldoet, volgt uit stelling 4, dat er minstens zo een polynoom is, volgt uit de voorafgaande secties 4 en 6.

Opmerking. De stellingen 3, 4, 5 kunnen direct bewezen worden uit het behandelde in sectie 4.

We hebben steeds te maken met een lineair stelsel van de gedaante (4.1), waarbij in stelling 3 als rechterlid de $\underline{0}$ -vector optreedt. Omdat de determinant van het stelsel niet nul is, is er steeds een unieke oplossing.

De stellingen 1 en 2 zijn voor onze theorie dus niet noodzakelijk. Zij zijn hier om hun algemeen belang en wegens de samenhang toegevoegd.

8. Vergelijking Lagrange met expliciete polynoom-interpolatie (Grünert).

In sectie 4 hebben wij behandeld de expliciete polynoom-interpolatie, d.w.z. de coëfficiënten van het polynoom treden expliciet in de formule op. We zullen dit noemen Grünert-interpolatie. Schrijven we het polynoom op zijn meest economisch, dan luidt de formule van Grünert:

$$f_G^*(x) = a_0 + x * (a_1 + x * (a_2 + \dots + x * (a_{n-2} + x * a_{n-1}) \dots)).$$

Hierbij worden de coëfficiënten a_j zo gekozen, dat f_G^* met f overeenstemt in n gegeven basispunten x_i , $i = 0(1)n - 1$. M.a.w. de vector a der coëfficiënten a_j is de oplossingsvector van het lineaire stelsel $Va = \bar{f}$, waarbij V de Van der Monde matrix $V(x_0, x_1, \dots, x_{n-1})$ is en \bar{f} de vector der functiewaarden in x_i .

Wij willen deze formule vergelijken met die van Lagrange (6.3)

$$f_L^{**}(x) = \mathcal{L}_0^n(x) f(x_0) + \mathcal{L}_1^n(x) f(x_1) + \dots + \mathcal{L}_{n-1}^n(x) f(x_{n-1}).$$

Ten eerste hebben we de volgende

Stelling 8.1. De formules van Grünert en Lagrange van dezelfde orde en op dezelfde basispunten zijn mathematisch equivalent.

M.a.w. als f_G^{**} en f_L^{**} overeenstemmen in n verschillende basispunten, dan zijn zij identiek.

Bewijs: De Lagrange-coëfficiënten $\mathcal{L}_j^n(x)$ zijn, blijkens hun definitie (6.2) polynomen van de graad $n - 1$ in x . Dus is $f_L^{**}(x)$ een polynoom van graad $< n$. (De graad kan kleiner dan $n - 1$ zijn, omdat termen van hoogste graad kunnen wegvallen.) Dus $f_G^{**}(x)$ en $f_L^{**}(x)$ zijn beide polynomen van graad $< n$, die in n verschillende punten met f overeenstemmen. Maar dan zijn zij identiek volgens stelling 7.4. (Ook volgt dit uit het feit dat $\det(V) \neq 0$.)

Stelling 8.2. De formule van Lagrange van de orde n is exact voor polynomen van graad $< n$. M.a.w. als de gegeven functie f een polynoom van graad $< n$ is, is het interpolerende Lagrange-polynoom f_L^{**} identiek met f .

Toepassing: Kies voor f een polynoom van de graad 0, dus een constante $c \neq 0$. Voor deze f is Lagrange van de orde $n > 0$ dus exact en we krijgen:

$$c = \sum_{j=0}^{n-1} \mathcal{L}_j^n(x) * c.$$

Dus, na deling door c , hebben we

Stelling 8.3. De som der Lagrange-coëfficiënten van een zekere orde is gelijk aan 1 voor alle waarden van x .

Deze eigenschap is belangrijk bij het controleren van onze berekeningen.

Voor- en nadelen

De formules van Grünert en Lagrange zijn wel mathematisch equivalent, maar numeriek niet. Ten eerste kunnen ten gevolge van afrondingsfouten de resultaten verschillen. Ten tweede verschillen de formules in het aantal benodigde uit te voeren operaties. Wat dit betreft is Grünert het meest economisch, als de (door interpolatie benaderde) waarde wordt gevraagd van een vaste gegeven functie f voor verscheidene argumenten. De coëfficiënten a_j hangen immers wel van f af, maar niet van x . Ze hoeven dus maar eens voor al te worden uitgerekend en voor elk argument wordt een geïnterpoleerde waarde gevonden met n vermenigvuldigingen en optellingen. Een belangrijke toepassing zijn de standaardfuncties (zie Algol-rapport 3.2.4), die in een Algol-systeem vaak door een polynoom-interpolatie in Grünert's vorm worden benaderd.

Aan de andere kant is Lagrange economischer, als voor een vast argument de benaderde waarde wordt gevraagd van verscheidene functies. De coëfficiënten $L_j^n(x)$ hangen immers niet van f af, en kunnen dus eens voor al worden berekend voor de gewenste waarde van x . De geïnterpoleerde waarde wordt verkregen als scalair product van de vector der Lagrange-coëfficiënten in het punt x en de vector der functie-waarden in de overeenkomstige basispunten.

Het aantal operaties is dus wederom n vermenigvuldigingen en optellingen. De formule van Lagrange wordt, zoals we zullen zien, vaak toegepast. Belangrijk is nog de volgende

Stelling 8.4. De Lagrange-coëfficiënten zijn invariant bij lineaire transformatie. D.w.z. als de basispunten x_i en het punt x worden vervangen door

$$x'_i = a + x_i \cdot b \quad \text{en} \quad x' = a + x \cdot b$$

dan geldt $L_i^n(x')$ (op de basispunten x'_i) = $L_i^n(x)$ (op de basispunten x_i).

Bewijs: Dit blijkt onmiddellijk uit de definitie (6.2)

Immers $x' - x'_i = b(x - x_i)$. Dus in (6.2) worden teller en noemer met $n - 1$ factoren b vermenigvuldigd en de Lagrange-coëfficiënten blijven dus onveranderd.

9. Formule van Lagrange op equidistante basispunten

In de praktijk van het handrekenen (met behulp van een tafelrekenmachine) werken we vaak met functies, gegeven in een tabel. De argumenten zijn dan heel vaak equidistant:

$$(9.1) \quad x_i = x_0 + ih, \quad i = k(1)l.$$

Ik laat expres i niet van 0 tot $n - 1$, maar van k tot l lopen (de orde is dus $n = l - k + 1$), om wat meer vrijheid te hebben. In de algemene formule (6.3) mag de volgorde (nummering) der basispunten willekeurig gekozen worden, de punten hoeven niet naar grootte geordend te zijn. In het equidistante geval evenwel wordt de volgorde der basispunten altijd vastgelegd door formule (9.1) en daarom willen we graag wat vrijheid terugkrijgen door als begin-index i.p.v. 0 de kiesbare waarde k te nemen.

Op het argument x passen we dezelfde lineaire transformatie toe:

$$(9.2) \quad x = x_0 + ph.$$

Vervangen we in formule (6.2) overal x_i door i en x door p , dan blijven volgens stelling 8.4 de Lagrange-coëfficiënten invariant. Zo hebben we voor het geval van equidistante basispunten (i.p.v. \mathcal{L} schrijven we nu L):

$$\begin{aligned} L_i^n(p) &= \prod_{\substack{j=k \\ j \neq i}}^l \frac{p-j}{i-j} \quad (i = k(1)l) \\ &= \frac{(p-k) \dots (p-i+2)(p-i+1)}{(i-k) \dots * 2 * 1} * \frac{(p-i-1)(p-i-2) \dots (p-l)}{-1 * -2 * \dots (i-l)} \\ &= \binom{p-k}{i-k} * \binom{l-p}{l-i}. \end{aligned}$$

We hebben dus voor het equidistante geval:

$$(9.3) \quad L_i^n(p) = \binom{p-k}{i-k} * \binom{l-p}{l-i}, \quad i = k(1)l, \text{ en}$$

$$(9.4) \quad f_L^*(x_0 + ph) = L_k^n(p) f_k + L_{k+1}^n(p) f_{k+1} + \dots + L_l^n(p) f_l,$$

waarbij $l = k + n - 1$.

Deze formules zijn (wegens stelling 8.4) onafhankelijk van het tabel-interval h en van de keuze van het nulpunt x_0 . We kunnen dus voor x_0 elk getabelleerd argument kiezen.

Toepassing. Zij gegeven een equidistante tabel van een functie f . We willen voor een zeker argument x de functie-waarde $f^{**}(x)$ berekenen met Lagrange-interpolatie van de orde n . (Hoe de orde n gekozen wordt in verband met de gewenste precisie, zal later ter sprake komen.)

Als x_0 kiezen we een van de dichtstbij x gelegen getabelleerde argumenten. De interpolatie fractie p is dan

$$p = (x - x_0) / h,$$

waarbij h het tabel-interval is, dus $h = x_1 - x_0$.

Als x_0 het meest nabije tabel-argument is, geldt dus $|p| \leq 1/2$.

De basispunten kiezen we aan weerskanten van x zo dichtbij mogelijk.

M.a.w. we kiezen beginindex k en eindindex l als volgt:

als het aantal punten n oneven is: $k = -(n-1) / 2$, $l = +(n-1) / 2$;

als n even is : $k = -(n/2-1)$, $l = +n/2$.

In het eerste geval hebben we symmetrie rondom $p = 0$, dus

$$L_i^{2k+1}(p) = L_{-i}^{2k+1}(-p).$$

In het tweede geval hebben we symmetrie rondom $p = 1/2$, dus

$$L_i^{2k}(p) = L_{1-i}^{2k}(1 - p).$$

De Lagrange-coëfficiënten voor het equidistante geval zijn uitvoerig getabelleerd in Nat. Bur. Stand, Tables of Lagrangian interpolation coefficients [6]. Vanwege de bovengenoemde symmetrie-relaties, kan men volstaan met p -waarden voldoende aan $0 \leq p \leq 1/2$.

Opmerking. Denk er om, dat bij afronding van Lagrange-coëfficiënten, men zodanig moet afronden, dat hun som gelijk aan 1 blijft.

Opgaven

31) Bewijs de formule: $\sum_{j=0}^{n-1} x \ell_j^{(n)}(x) = 0$.

32) a) Bereken de Lagrange-coëfficiënten $\ell_i^4(x)$ als de basispunten zijn: -2, -1, 1, 2.

Bereken vervolgens deze coëfficiënten voor $x = 0$ en $x = 3/2$.

b) Bereken het interpolerend polynoom $f_L^*(x)$ van graad < 4 dat in de basispunten respectievelijk de volgende waarden heeft: -5, -1, 1, 11.

33) a) Bereken door middel van 4-punts Lagrange-interpolatie de waarden $f^*(x)$ voor $x = 1.13, 1.15, 1.17, 1.19$ als van f de volgende tabel is gegeven:

x	f(x)
1.1275	.11971
1.1503	.13954
1.1735	.15932
1.1972	.17903

b) Bereken vervolgens, uit de zo verkregen equidistante tabel en met gebruik van een tabel van Lagrange-coëfficiënten de waarden $f^*(x)$ voor $x = 1.16$ (.001) 1.17 .

34) Bereken de Lagrange-coëfficiënten (voor het equidistante geval)

$L_i^n(p)$, waarbij $n = 2, 3, 4, 5$ en $p = 1/2, 1/4, 3/4$.

Kies de grenzen van i zoals boven aangegeven (zie pag. 45).

VIER PUNTS LAGRANGE COEFFICIENTEN

p	i = -1	i = 0	i = 1	i = 2	
.00	- .0000000	+1.0000000	+ .0000000	- .0000000	1.00
.01	- .0032835	+ .9949005	+ .0100495	- .0016665	.99
.02	- .0064680	+ .9896040	+ .0201960	- .0033320	.98
.03	- .0095545	+ .9841135	+ .0304365	- .0049955	.97
.04	- .0125440	+ .9784320	+ .0407680	- .0066560	.96
.05	- .0154375	+ .9725625	+ .0511875	- .0083125	.95
.06	- .0182360	+ .9665080	+ .0616920	- .0099640	.94
.07	- .0209405	+ .9602715	+ .0722785	- .0116095	.93
.08	- .0235520	+ .9538560	+ .0829440	- .0132480	.92
.09	- .0260715	+ .9472645	+ .0936855	- .0148785	.91
.10	- .0285000	+ .9405000	+ .1045000	- .0165000	.90
.11	- .0308385	+ .9335655	+ .1153845	- .0181115	.89
.12	- .0330880	+ .9264640	+ .1263360	- .0197120	.88
.13	- .0352495	+ .9191985	+ .1373515	- .0213005	.87
.14	- .0373240	+ .9117720	+ .1484280	- .0228760	.86
.15	- .0393125	+ .9041875	+ .1595625	- .0244375	.85
.16	- .0412160	+ .8964480	+ .1707520	- .0259840	.84
.17	- .0430355	+ .8885565	+ .1819935	- .0275145	.83
.18	- .0447720	+ .8805160	+ .1932840	- .0290280	.82
.19	- .0464265	+ .8723295	+ .2046205	- .0305235	.81
.20	- .0480000	+ .8640000	+ .2160000	- .0320000	.80
.21	- .0494935	+ .8555305	+ .2274195	- .0334565	.79
.22	- .0509080	+ .8469240	+ .2388760	- .0348920	.78
.23	- .0522445	+ .8381835	+ .2503665	- .0363055	.77
.24	- .0535040	+ .8293120	+ .2618880	- .0376960	.76
.25	- .0546875	+ .8203125	+ .2734375	- .0390625	.75
.26	- .0557960	+ .8111880	+ .2850120	- .0404040	.74
.27	- .0568305	+ .8019415	+ .2966085	- .0417195	.73
.28	- .0577920	+ .7925760	+ .3082240	- .0430080	.72
.29	- .0586815	+ .7830945	+ .3198555	- .0442685	.71
.30	- .0595000	+ .7735000	+ .3315000	- .0455000	.70
.31	- .0602485	+ .7637955	+ .3431545	- .0467015	.69
.32	- .0609280	+ .7539840	+ .3548160	- .0478720	.68
.33	- .0615395	+ .7440685	+ .3664815	- .0490105	.67
.34	- .0620840	+ .7340520	+ .3781480	- .0501160	.66
.35	- .0625625	+ .7239375	+ .3898125	- .0511875	.65
.36	- .0629760	+ .7137280	+ .4014720	- .0522240	.64
.37	- .0633255	+ .7034265	+ .4131235	- .0532245	.63
.38	- .0636120	+ .6930360	+ .4247640	- .0541880	.62
.39	- .0638365	+ .6825595	+ .4363905	- .0551135	.61
.40	- .0640000	+ .6720000	+ .4480000	- .0560000	.60
.41	- .0641035	+ .6613605	+ .4595895	- .0568465	.59
.42	- .0641480	+ .6506440	+ .4711560	- .0576520	.58
.43	- .0641345	+ .6398535	+ .4826965	- .0584155	.57
.44	- .0640640	+ .6289920	+ .4942080	- .0591360	.56
.45	- .0639375	+ .6180625	+ .5056875	- .0598125	.55
.46	- .0637560	+ .6070680	+ .5171320	- .0604440	.54
.47	- .0635205	+ .5960115	+ .5285385	- .0610295	.53
.48	- .0632320	+ .5848960	+ .5399040	- .0615680	.52
.49	- .0628915	+ .5737245	+ .5512255	- .0620585	.51
.50	- .0625000	+ .5625000	+ .5625000	- .0625000	.50

i = 2 i = 1 i = 0 i = -1 p

begin comment Opdracht R 1086, TJD 250165/11858.

Programma ter berekening van 4-punts Lagrange
coëfficiënten voor index $i := -1, 0, 1, 2$ en voor
interpoleer-fractie $p := 0$ step .01 until .50;

real p,lmin1,10,11,12,sum ;

procedure out (k,x) ; value k; integer k; real x;

begin if k = 1 then PUNLCR else

begin PUSPACE(1); PUSPACE(1); PUSPACE(1) end ;

if k \leq 2 then ABSFIXP(1,2,x) else FIXP(1,k,x)

end out ;

procedure outtext (s); string s;

begin PUNLCR; PUNLCR; PUTEXT1 (s); PUNLCR end outtext;

outtext(

⋆

VIER PUNTS LAGRANGE COEFFICIENTEN⋆);

outtext(

⋆

p i = -1 i = 0 i = 1 i = 2⋆);

for p:= 0 step .01 until .505 do

begin out(1,p);

lmin1:= -p × (1-p) × (2-p)/6; out(7,lmin1);

10 := (p+1) × (1-p) × (2-p)/2; out(7,10);

11 := (p+1) × p × (2-p)/2; out(7,11);

12 := (p+1) × p × (p-1)/6; out(7,12);

sum := lmin1 + 10 + 11 + 12 ;

if abs(sum - 1) > 10⁻⁸ then

begin out(12,sum); go to end end ;

out(2,1-p)

end ;

outtext(

⋆

i = 2 i = 1 i = 0 i = -1 p⋆);

end :

end

10. Interpolatie-formule van Newton

We zoeken nu een formule, waarbij we van orde n op orde $n+1$ kunnen overgaan, door toevoeging van een betrekkelijk eenvoudige term. Wij nemen aan dat deze term een veelvoud is van de n -de basisfunctie G_n , zodat we krijgen:

$$f_{n+1}^{**}(x) = f_n^{**}(x) + a_n \cdot G_n(x).$$

Hierin is f_n^{**} een polynoom van graad $< n$ overeenstemmend met f in de punten x_i , $i = 0(1)n - 1$, en f_{n+1}^{**} is een polynoom van graad $\leq n$ die met f overeenstemt in de bovengenoemde x_i en bovendien in een nieuw punt x_n .

Hieruit volgt, dat G_n een polynoom van graad $\leq n$ is en nul is in x_i , $i = 0(1)n - 1$. Dus

$$G_n(x) = c(x - x_0)(x - x_1) \dots (x - x_{n-1}) = c \pi_n(x),$$

waarbij c een constante is. Omdat we ook de constante coëfficiënt a_n hebben, kunnen we rustig stellen $c = 1$.

Zo krijgen we als definitie voor de basisfuncties:

$$(10.1) \quad G_j(x) = \pi_j(x) = \prod_{k=0}^{j-1} (x - x_k), \quad j = 0(1)n - 1,$$

en als interpolerend polynoom:

$$f^{**}(x) = \sum_{j=0}^{n-1} a_j \pi_j(x).$$

Dit moet in de punten x_i met f overeenstemmen:

$$(10.2) \quad \sum_{j=0}^{n-1} a_j \pi_j(x_i) = f(x_i), \quad i = 0(1)n - 1.$$

Omdat $M_{ij} = \pi_j(x_i) = 0$ als $i < j$, heeft dit stelsel de driehoeksvorm en is dus eenvoudig oplosbaar. De determinant van het stelsel is gelijk aan het product van de diagonaal-elementen:

$$\det(M) = \pi_0(x_0) \pi_1(x_1) \dots \pi_{n-1}(x_{n-1}) = \det(V) \neq 0 \text{ (vgl. pag. 31).}$$

We schrijven het stelsel (10.2) wat vollediger uit:

$$\begin{aligned} a_0 &= f(x_0) \\ a_0 + a_1(x_1 - x_0) &= f(x_1) \\ a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) &= f(x_2) \\ \dots & \\ a_0 + a_1(x_{n-1} - x_0) + \dots + a_{n-1}(x_{n-1} - x_0) \dots (x_{n-1} - x_{n-2}) &= f(x_{n-1}) \end{aligned}$$

De 0-de vergelijking levert direct a_0 . We trekken deze vergelijking van de andere af en delen door $x_i - x_0$. De zo ontstaande rechterleden heten eerste gedeelde differenties, waarvoor we de volgende notatie invoeren:

$$f [x_0, x_i] = (f(x_i) - f(x_0)) / (x_i - x_0).$$

Het stelsel vergelijkingen komt er dan als volgt uit te zien:

$$\begin{aligned} a_1 &= f [x_0, x_1] \\ a_1 + a_2(x_2 - x_1) &= f [x_0, x_2] \\ \dots & \\ a_1 + a_2(x_{n-1} - x_1) + \dots + a_{n-1}(x_{n-1} - x_1) \dots (x_{n-1} - x_{n-2}) &= f [x_0, x_{n-1}]. \end{aligned}$$

De eerste vergelijking levert direct a_1 en trekken we weer af van de andere vergelijkingen, waarna we door $x_i - x_1$ delen. De ontstaande rechterleden heten tweede gedeelde differenties en noteren we als volgt:

$$f [x_0, x_1, x_i] = (f [x_0, x_i] - f [x_0, x_1]) / (x_i - x_1).$$

We krijgen dan $a_2 = f [x_0, x_1, x_2]$ en zo voortgaande berekenen we alle coëfficiënten a_j .

Samenvatting

De k-de gedeelde differentie wordt recurrent gedefiniëerd voor de opeenvolgende waarden van k als volgt:

$$(10.3) \text{ Voor } k = 0: f[x_i] = f(x_i) \text{ en voor } k > 0: f[x_0, \dots, x_{k-1}, x_i] = \\ = \frac{f[x_0, \dots, x_{k-2}, x_i] - f[x_0, \dots, x_{k-2}, x_{k-1}]}{x_i - x_{k-1}}.$$

Dan geldt voor de coëfficiënten a_j van het stelsel (10.2):

$$(10.4) \quad a_j = f[x_0, \dots, x_j].$$

Het interpolerend polynoom f^* krijgt dan de vorm:

$$(10.5) \quad f^*(x) = f[x_0] + f[x_0, x_1](x - x_0) + \dots + f[x_0, \dots, x_{n-1}](x - x_0) \dots \\ \dots (x - x_{n-2}).$$

Dit is de interpolatie-formule van Newton van de orde n (en rang n).

De coëfficiënten $f[x_0, \dots, x_j]$ heten Newton-coëfficiënten.

Evenals bij de expliciete polynoom-vorm van Grünert, kunnen we bezuinigen op het aantal bewerkingen, door (10.5) als volgt te schrijven:

$$(10.6) \quad f^*(x) = f[x_0] + (x - x_0)(f[x_0, x_1] + (x - x_1)(f[x_0, x_1, x_2] + \dots \\ \dots (x - x_{n-2})f[x_0, \dots, x_{n-1}]) \dots)).$$

De k -de gedeelde differenties kunnen we definiëren en berekenen op elk $(k+1)$ -tal verschillende basispunten. Hierbij geldt dan de volgende prettige Stelling 10.1: De waarde van een k -de differentie hangt niet af van de volgorde der basispunten, die erin voorkomen. M.a.w. zij (i_0, \dots, i_j) een permutatie van de indices $(0, \dots, j)$, dan geldt:

$$f[x_0, \dots, x_j] = f[x_{i_0}, \dots, x_{i_j}].$$

Bewijs: Beschouw Newton's formule van de orde $j+1$ op de basispunten x_0, \dots, x_j . Hierin is $f[x_0, \dots, x_j]$ de coëfficiënt van x^j . Aangezien het polynoom f^* uniek bepaald is door de basispunten geeft permutatie van de basispunten geen verandering, waaruit meteen de stelling volgt. Hieruit volgt, dat we de Newton-coëfficiënten kunnen verkrijgen, door

successievelijk voor elke k te berekenen de k -de differenties $f[x_i, \dots, x_{i+k}]$. De te berekenen getallen kunnen we als volgt in een schema zetten:

$$\begin{array}{ccccccc}
 x_0 & f[x_0] & & & & & \\
 & & f[x_0, x_1] & & & & \\
 x_1 & f[x_1] & & f[x_0, x_1, x_2] & & & \\
 & & f[x_1, x_2] & & f[x_0, x_1, x_2, x_3] & & \\
 x_2 & f[x_2] & & f[x_1, x_2, x_3] & & & \\
 & & f[x_2, x_3] & & & & \\
 x_3 & f[x_3] & & & & &
 \end{array}$$

Voorbeeld

x	$f(x) = x^3$					
0	0	1				
1	1	4				
3	27	13	1			
6	216	63	16	0		
7	343	127	24	0	0	
11	1331	247	30	0	0	
12	1728	397				

Newton's formule van de orde n is exact voor polynomen van graad $< n$. Dientengevolge zijn de k -de gedeelde differenties van een polynoom van graad k constant en alle hogere gedeelde differenties 0. Men kan deze eigenschap gebruiken, om een polynoom voor verscheidene argumenten te berekenen.

Vergelijking Newton met Grünert en Lagrange

Newton's formule van de orde n levert, evenals Grünert en Lagrange, een polynoom van graad kleiner dan n . Omdat overeenstemming met een gegeven functie op n verschillende basispunten een polynoom van graad kleiner dan n uniek bepaalt, hebben we blijkbaar

Stelling (10.2): De formule van Newton is mathematisch equivalent met die van Grünert en Lagrange, mits van dezelfde orde en genomen op dezelfde basispunten.

Newton's formule munt niet uit in zuinigheid van het aantal bewerkingen. Zij heeft evenwel het voordeel, dat men gemakkelijk een basispunt kan toevoegen en een formule van orde een hoger kan krijgen. Immers, als f_n^* en f_{n+1}^* de interpolerende polynomen zijn van de orde n respectievelijk $n+1$, dan geldt (vgl. pag. 49 en (10.5)):

$$(10.7) \quad f_{n+1}^*(x) = f_n^*(x) + f[x_0, \dots, x_n] (x - x_0) \dots (x - x_{n-1}).$$

De grootte van de toegevoegde term kan een idee geven van de bereikte precisie. Newton's formule zullen we dan ook gebruiken voor de beschouwing van de restterm.

Errata

De pagina's 33 en 34 ontbreken.

Pagina 46) opgave 31 moet luiden:

Bewijs de formule: $\sum_{j=0}^{n-1} x_j \mathcal{L}_j^n(x) = 0$, voor $n > 1$.

Opgaven

35) Doe opgave 26 met Lagrange interpolatie en vergelijk de resultaten.

36) Bereken $f''(7)$ uit de opgaven 28 en 29 met Lagrange's formule en vergelijk de resultaten.

37) Bewijs direct uit definitie (10.3):

$$f[x_0, x_1, x_2] = f[x_2, x_0, x_1].$$

38) Bewijs de formule:

$$L_0^n(x) = 1 + \frac{x - x_0}{x_0 - x_1} + \frac{(x - x_0)(x - x_1)}{(x_0 - x_1)(x_0 - x_2)} + \dots + \frac{(x - x_0)\dots(x - x_{n-2})}{(x_0 - x_1)\dots(x_0 - x_{n-1})}.$$

Aanwijzing: Pas Newton's formule toe op $L_0^n(x)$, zijnde een polynoom van graad $< n$, dat 1 is in x_0 en 0 is in de andere basispunten.

39) Zij gegeven het volgende stuk tabel van de Gamma-functie:

x	$\Gamma(x)$
1.950	0.97988 06513
1.955	0.98180 15524
1.960	0.98374 25404
1.965	0.98570 36664
1.970	0.98768 49838
1.975	0.98968 65462
1.980	0.99170 84087
1.985	0.99375 06274
1.990	0.99581 32598
1.995	0.99789 63643
2.000	1.00000 00000

Bereken benaderde waarden van $\Gamma(x)$ met 4-punts Lagrange voor $x = 1.956, 1.9622, 1.9725$ en 1.980625 .

11. De restterm

We hebben nu drie vormen van polynoom-interpolatie beschouwd, die, zoals we zagen, mathematisch equivalent zijn. Als de functie f een polynoom van graad $< n$ is, zijn de interpolatie-formules van orde n exact. Voor andere functies geldt dat natuurlijk niet en we zullen dus moeten nagaan welke fout wordt gemaakt, als f door een benaderend polynoom f_n^{**} vervangen wordt. De correctie-term, die nodig is om het n -de orde interpolatie-polynoom f_n^{**} tot f te corrigeren, noemen we de restterm $R_n(x)$, dus:

$$(11.1) \quad f(x) = f_n^{**}(x) + R_n(x).$$

Omdat de formules van Grünert, Lagrange en Newton voor dezelfde orde en basispunten mathematisch gesproken equivalent zijn, hebben zij alle drie dezelfde restterm. Voor het onderzoeken van de restterm gaan we uit van Newton's formule, omdat zij voor dit doel het handigst is. In formule (10.7) vervangen we x_n door x . M.a.w. aan de basispunten x_i ($i = 0(1)n - 1$) voegen we toe het basispunt x . Vanwege de exacte overeenstemming in x geldt dan: $f_{n+1}^{**}(x) = f(x)$ en (10.7) gaat over in

$$f(x) = f_n^{**}(x) + f[x_0, \dots, x_{n-1}, x] (x - x_0) \dots (x - x_{n-1}).$$

Vergelijking met (11.1) levert onmiddellijk

$$(11.2) \quad R_n(x) = f[x_0, \dots, x_{n-1}, x] \pi_n(x).$$

In de gedeelde differentie komt $f(x)$ voor, zodat we hiermee niet veel verder komen. We kunnen echter een nuttige bovengrens voor R_n geven, als we weten, dat de functie f voldoende vaak differentiëerbaar is in een interval, dat alle basispunten en het punt x bevat.

We hebben nl. de volgende

Stelling 11.3: Zij a het minimum en b het maximum van x en de basispunten x_i , $i = 0(1)n - 1$. Zij verder de functie f minstens n maal differentiëerbaar in het interval $[a, b]$. Dan geldt voor de restterm van de n -de orde interpolatie op deze basispunten:

$$R_n(x) = \frac{f^{(n)}(\xi)}{n!} \pi_n(x),$$

waarbij ξ een of ander getal tussen a en b is.

Bewijs

Als x samenvalt met een der basispunten x_i , $i = 0(1)n - 1$, is de stelling onmiddellijk duidelijk. Dan geldt immers $R_n(x) = \pi_n(x) = 0$ en elke ξ tussen a en b voldoet. Nu het geval $x = x_0, \dots, x_{n-1}$. We voegen een extra basispunt x_n toe en krijgen dan de $(n + 1)$ -punts formule:

$$f(x) = f_n^{**}(x) + f[x_0, \dots, x_{n-1}, x_n] \pi_n(x) + R_{n+1}(x).$$

Hierbij is f_n^{**} gebaseerd op de punten x_i , $i = 0(1)n - 1$ en de tweede term zorgt voor de toevoeging van het basispunt x_n (vgl. 10.7 en 11.1). Blijkbaar geldt: $R_{n+1}(x) = 0$ voor $x = x_i$, $i = 0(1)n$, m.a.w. R_{n+1} heeft $n + 1$ nulpunten. Dus heeft de afgeleide functie R_{n+1}' , volgens de stelling van Rolle, n nulpunten (die elk netjes tussen twee opeenvolgende nulpunten van R_{n+1} liggen).

Het argument herhalende vinden we tenslotte:

De n -de afgeleide $R_{n+1}^{(n)}$ heeft een nulpunt ξ voldoende aan $\min(x_0, \dots, x_n) < \xi < \max(x_0, \dots, x_n)$. Dus

$$0 = R_{n+1}^{(n)}(\xi) = f^{(n)}(\xi) - \left[\frac{d^n}{dx^n} f_n^{**}(x) \right]_{x=\xi} - f[x_0, \dots, x_n] \pi_n^{(n)}(\xi).$$

Omdat f_n^{**} een polynoom van graad $< n$ is, is zijn n -de afgeleide identiek 0. Omdat $\pi_n(x)$ een polynoom van graad n is met x^n -coëfficiënt 1, is zijn n -de afgeleide constant en wel gelijk aan $n!$. Uit een en ander volgt:

$$f[x_0, \dots, x_{n-1}, x_n] = \frac{f^{(n)}(\xi)}{n!}.$$

Nu vervangen we x_n door x (dit mag want $x \neq x_i$, $i = 0(1)n - 1$) en vermenigvuldigen met $\pi_n(x)$.

Wegens (11.2) krijgen we dan:

$$R_n(x) = f[x_0, \dots, x_{n-1}, x] \pi_n(x) = \frac{f^{(n)}(\xi) \pi_n(x)}{n!},$$

waarbij nu ξ voldoet aan: $a < \xi < b$. q.e.d.

Toepassingen

Deze formule voor de restterm is zeer geschikt voor het vinden van een bovengrens van de interpolatie-fout. Het enige, dat we hiertoe van f moeten weten, is een bovengrens voor zijn n -de afgeleide in het beschouwde interval.

Voorbeeld 1. Zij gegeven een tabel van $\sin x$ voor $x = 0(0.2)2$ (in radialen). Welke nauwkeurigheid is bereikbaar voor willekeurig argument met 4-punts interpolatie?

Het tabelinterval h is gelijk aan 0.2 . Neem als basispunten de 4 dichtsbijzijnde, die we nummeren $i = -1, 0, 1, 2$ en voldoen aan $x_i = x_0 + ih$ (vgl. sectie 9). Dan ligt x tussen x_0 en x_1 (behalve misschien aan de uiteinden der tabel).

Als we definiëren

$$p = (x - x_0)/h, \text{ geldt: } \pi_4(x) = (p + 1)p(p - 1)(p - 2) h^4.$$

Voor p tussen 0 en 1 bereikt $|\pi_4(x)|$ zijn maximum voor $p = \frac{1}{2}$ (afgeleide = 0), zodat voor $x_0 \leq x \leq x_1$ geldt:

$$|\pi_4(x)| \leq \frac{9}{16} h^4 = 9_{10}^{-4}.$$

Omdat $\frac{d^4}{dx^4} \sin(x) = \sin(x)$, wat in absolute waarde hoogstens 1 is, hebben we dus

$$|R_n(x)| \leq \frac{1}{24} * 9_{10}^{-4} \approx .0004,$$

m.a.w. de bereikbare precisie is 3 decimalen.

Voorbeeld 2. Hoe fijn moet $\ln x$ getabelleerd worden voor $1 \leq x \leq 10$ om te bereiken, dat lineaire interpolatie een precisie van 6 decimalen levert?

De maximale toegestane fout is dus $.5_{10}^{-6}$, m.a.w. de eis luidt:

$$R_2(x) = \frac{f''(\xi)}{2!} \pi_2(x) \leq .5_{10}^{-6}.$$

Nu geldt: $|\pi_2(x)| \leq h^2/4$ en $|f''(\xi)| = \frac{1}{\xi^2} \leq 1$, dus het tabelinterval h moet voldoen aan: $\frac{1}{2!} * \frac{h^2}{4} \leq .5_{10}^{-6}$ of $h^2 \leq 4_{10}^{-6}$, dus $h \leq .002$.

12. Interpolatieformule van Aitken

Deze formule bestaat uit een herhaald toepassen van lineaire interpolatie. Schrijven we deze in Lagrange-vorm, dan hebben we:

$$f(x) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1).$$

Omdat hier de basispunten x_0 en x_1 gebruikt zijn, noemen we dit y_{01} .

Om in de stijl te blijven noemen we de functiewaarden $f(x_i)$ nu y_i .

We schrijven de formule nu in determinantvorm:

$$y_{01} = \frac{1}{x_1 - x_0} \begin{vmatrix} y_0 & x_0 - x \\ y_1 & x_1 - x \end{vmatrix}.$$

En algemener:

$$y_{i, \dots, i+k} = \frac{1}{x_{i+k} - x_i} \begin{vmatrix} y_i, \dots, i+k & x_i - x \\ y_{i+1}, \dots, i+k & x_{i+k} - x \end{vmatrix}.$$

Men moet hierbij wel in het oog houden, dat $y_i, \dots, i+k$ voor $k > 0$ van x afhangt. Het blijkt een polynoom in x te zijn van graad $\leq k$, dat in de $k + 1$ punten $x_i, x_{i+1}, \dots, x_{i+k}$ met f overeenstemt.

De interpolatie-formule is dus mathematisch equivalent met de reeds besproken polynoom interpolaties van de orde $k + 1$. Op de basispunten x_0, \dots, x_{n-1} krijgen we dus

$$(12.1) \quad f(x) = y_0, \dots, x_{n-1} + R_n(x).$$

De formule wordt als volgt gebruikt. Laat de functie f door een tabel gegeven zijn. Kies telkens als nieuw basispunt een dicht bij x gelegen tabel-argument. De argumenten worden hier niet volgens grootte, maar in volgorde van keuze genummerd.

De achtereenvolgens berekende y -waarden worden als volgt opgeschreven:

x_0	$x_0 - x$	f_0			
			y_{01}		
x_1	$x_1 - x$	f_1		y_{012}	
			y_{12}		y_{0123}
x_2	$x_2 - x$	f_2		y_{123}	
			y_{13}		
x_3	$x_3 - x$	f_3			

Aitken's formule schijnt voordelig te zijn als we voor een enkele waarde van x willen interpoleren met de hand (en tafelmachine). De y -waarden gaan namelijk in steeds meer cijfers overeenstemmen, die we niet telkens hoeven neer te schrijven. De mate van overeenstemming geeft een idee van de bereikte precisie.

Willen we evenwel voor verscheidene waarden van x interpoleren (en tevens een idee van de bereikte precisie krijgen), dan kunnen we beter het gedeelde differentie-schema opbouwen en Newton toepassen. Kennen we de precisie reeds (b.v. uit resttermbeschouwing of tabel-gegevens) dan is de snelste formule hetzij Grünert's vorm (mits het aantal x -waarden groot is) hetzij in het equidistante geval Lagrange, mits we de Lagrange-coëfficiënten uit een tabel halen.

Na deze snelheids-overwegingen nog het volgende

Voorbeeld van Aitken-interpolatie.

Gevraagd in onderstaande tabel $f(21)$ te bepalen. (De argumenten zijn alvast volgens keuze gerangschikt):

x_i	$x_i - x$	f_i			
19	- 2	33312			
			36860		
23	2	40408	36854		
			--833	36855	
28	7	49345	---	52	-----5
			--928	---	4
				-----4	-----5
13	- 8	22738	---	41	-----4
			--990	-----5	
33	12	58367	---	82	
			37071		
7	-14	12225			

Resultaat $f(21) \approx 36855$.

Inverse interpolatie

Hieronder verstaat men het zoeken van een benaderde argument-waarde, waarvoor de functie een bepaalde waarde aanneemt. Dit kan men doen met een der bovengenoemde interpolatie-formules, door de rollen van functie-waarden en argumenten te verwisselen. Zij een functie f in een tabel gegeven. Gewone interpolatie bestaat uit het berekenen van benaderde waarden $y = f(x)$, inverse interpolatie uit het berekenen van $x = f^{-1}(y)$, m.a.w. interpolatie van de inverse functie f^{-1} . Hier is echter een waarschuwing op zijn plaats. Weet men dat een getabelleerde functie f voldoende nauwkeurig interpoleerbaar is, dan geeft dit geen garantie, dat dit ook geldt voor de inverse functie f^{-1} . Zelfs een hogere orde formule geeft niet altijd een bevredigende inverse interpolatie.

Voorbeeld

x	f(x) = x ³
0	0
1	1
2	8
3	27
4	64

Gevraagd x zodat $f(x) = 20$.

Vorm de Lagrange-coëfficiënten:

$$L_0^5(20) = \frac{(20-1)(20-8)(20-27)(20-64)}{(0-1)(0-8)(0-27)(0-64)} = 5.07986$$

$$L_1^5(20) = \frac{(20-0)(20-8)(20-27)(20-64)}{(1-0)(1-8)(1-27)(1-64)} = -6.44689$$

$$L_2^5(20) = \frac{(20-0)(20-1)(20-27)(20-64)}{(8-0)(8-1)(8-27)(8-64)} = 1.96429$$

$$L_3^5(20) = \frac{(20-0)(20-1)(20-8)(20-64)}{(27-0)(27-1)(27-8)(27-64)} = 0.40656$$

$$L_4^5(20) = \frac{(20-0)(20-1)(20-8)(20-27)}{(64-0)(64-1)(64-8)(64-27)} = -0.00382$$

Somcontrôle geeft 1.00000 en de benaderde waarde van x is

$$x \approx f^{-1*}(20) = L_0^5(20) * 0 + L_1^5(20) * 1 + L_2^5(20) * 2 + L_3^5(20) * 3 + \\ + L_4^5(20) * 4 = -1.3139$$

in plaats van $x = \sqrt[3]{20} = 2.7144$.

De fout is aanzienlijk. Dit komt doordat de inverse functie $x = \sqrt[3]{y}$ zich in de buurt van $y = 0$ allerm minst als een polynoom gedraagt.

De eerste afgeleide is reeds oneindig voor $y = 0$.

Een belangrijk bijzonder geval van inverse interpolatie is het zoeken van een argument x , waarvoor $f(x) = 0$, m.a.w. het bepalen van een wortel van een vergelijking. Dit zal later uitvoerig behandeld worden.

Opgaven

40) Zij gegeven een tabel van een functie f in twee arrays P en Q zodanig, dat voor $i = 0(1)N - 1$ de bij het argument $P[i]$ horende functie-waarde $Q[i]$ is.

a) Schrijf een Algol-procedure, die het volledige schema van gedeelde differenties uitrekent en in een 2-dimensionaal array aflevert.

b) Schrijf een Algol-procedure, die uit dit array der gedeelde differenties de Newton-coëfficiënten licht en in een 1-dimensionaal array $A[0 : N - 1]$ plaatst.

c) Schrijf een functie procedure, die voor elk gegeven argument x en orde n ($n \leq N$) de geïnterpoleerde waarde $f_n^{**}(x)$ berekent met Newton's formule en gebruik makend van de in A staande coëfficiënten.

d) Schrijf vervolgens een blok, waarin (na de nodige voorbereidings-aanroepen van de procedures uit a en b) voor $n = 5$ en 6 worden berekend de waarden $f_n^{**}(x)$, waarbij $x = 0(.1)1.5$.

Laat de resultaten afleveren in twee passende 1-dimensionale arrays.

41) Zij gegeven de tabel

x	$f(x)$
0	0
2	10
3	30
4	68
5	130
10	1010

Bereken een volledig schema van gedeelde differenties en daarna $f(1)$ en $f(8)$ met Newton's interpolatie-formule.

Zoek vervolgens het polynoom van laagste graad, dat met $f(x)$ in de gegeven punten overeenstemt, en schrijf dit expliciet uit (in Grünert's vorm).

- 42) Bewijs dat de restterm horende bij lineaire interpolatie gebaseerd op de punten x_0 en x_1 met $x_0 \leq x \leq x_1$ in absolute waarde hoogstens is $\frac{1}{8} M(x_1 - x_0)^2$, waarbij $M = \text{maximum van } |f''(x)|$ in het interval $[x_0, x_1]$.
Geldt dit resultaat ook voor extrapolatie, d.w.z. voor x buiten het interval $[x_0, x_1]$?
- 43) Zij gegeven een tabel van $\exp(x)$ voor $x = .50(.01)1.00$.
Welke precisie kan worden bereikt
a) met lineaire interpolatie,
b) met 3-punts interpolatie,
c) met 4-punts interpolatie?
- 44) a) Hoe fijn moet de functie $^{10}\log(x)$ voor $1 \leq x \leq 10$ getabelleerd worden, opdat zij in 5 decimalen nauwkeurig lineair interpoleerbaar is?
b) Dezelfde vraag voor de functie $\sin\left(\frac{\pi x}{180}\right)$ (dat is sinus met argument in graden) en $0 \leq x \leq 45$.
- 45) Bewijs dat de 3-punts Aitken-formule voor y_{012} inderdaad in x_0 , x_1 en x_2 exact met de functie f overeenstemt.
- 46) Bereken met behulp van Aitken interpolatie $f(16)$ uit de tabel gegeven in het voorbeeld aan het eind van sectie 12.

14. Samenvallende basispunten

Tot nu toe werd steeds verondersteld, dat de basispunten verschillend zijn. Dit is nodig, om te zorgen dat de op te lossen lineaire stelsels een determinant $\neq 0$ en een unieke oplossing bezitten.

We kunnen echter nagaan, wat er met de interpolatieformules gebeurt bij limiet-overgang naar een situatie, waar sommige basispunten komen samen te vallen. Doet men dit uitgaande van de formule van Lagrange, dan ontstaat de interpolatie-formule van Lagrange-Hermite, waarop we later hopen terug te komen. Eenvoudiger is het uit te gaan van Newton's formule. We gaan eerst na welk effect limietovergang heeft op de gedeelde differenties. Voor de eerste gedeelde differentie geldt, volgens de definitie van de afgeleide, mits deze bestaat in x_i :

$$\lim_{x_{i+1} \rightarrow x_i} f [x_i, x_{i+1}] = \lim_{x_{i+1} \rightarrow x_i} \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} = f'(x_i) .$$

Voor de k -de gedeelde differentie geldt, mits de k -de afgeleide bestaat (vgl. het bewijs van stelling 11.3):

$$f [x_i, x_{i+1}, \dots, x_{i+k}] = \frac{f^{(k)}(\xi)}{k!} ,$$

waarbij $\min(x_i, \dots, x_{i+k}) < \xi < \max(x_i, \dots, x_{i+k})$.

Als nu x_{i+1}, \dots, x_{i+k} allemaal tot x_i naderen, geldt dus, mits de k -de afgeleide van f bestaat in een omgeving van x_i en continu is in x_i :

$$\lim_{x_{i+1}, \dots, x_{i+k} \rightarrow x_i} f [x_i, x_{i+1}, \dots, x_{i+k}] = \frac{f^{(k)}(x_i)}{k!} .$$

Daarom definiëren we voor de samenvallende basispunten $x_i, x_{i+1}, \dots, x_{i+k}$, mits f voldoende vaak continu differentieerbaar is:

$$(14.1) \quad f [x_i, x_{i+1}, \dots, x_{i+k}] = \text{def} \frac{f^{(k)}(x_i)}{k!} .$$

Met deze formule kan nu ook in het geval van samenvallende basispunten het gedeelde differentieschema worden opgebouwd. De basispunten worden

zo genummerd, dat samenvallende punten steeds opeenvolgende indices krijgen. Bij de opbouw van het schema passen we (14.1) toe, als $x_i = x_{i+1} = \dots = x_{i+k}$; zo niet, dan is $x_i \neq x_{i+k}$ (volgens bovengenoemde nummerings-afspraken) en passen we (10.3) toe. Na het gedeelde differentie-schema aldus opgebouwd te hebben, kunnen we voor gegeven x de Newton-formule (10.5) of (10.6) berekenen.

Voorbeeld.

Gevraagd een polynoom van zo laag mogelijke graad te construeren, zodat

$$\begin{aligned} f(0) = f'(0) = 0, \quad f''(0) = 1, \\ f(1) = f'(1) = 1, \quad f''(1) = -5. \end{aligned}$$

Het gedeelde differentie-schema komt er als volgt uit te zien.

x	$f(x)$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$		
0	0				
0	0	0			
0	0		.5		
0	0	0		.5	
1	1	1	1	-1.5	
1	1		0		-1.5
1	1	1		-2.5	
1	1		-2.5		
1	1	1			

$$\begin{aligned} \text{Dus } f(x) &= 0 + 0 * x + .5 * x^2 + .5 * x^3 - 1.5 x^3(x-1) + 0 * x^3(x-1)^2 = \\ &= -\frac{1}{2} x^2 (3x^2 - 4x - 1). \end{aligned}$$

Laten we alle basispunten samenvallen (met x_0), dan gaat voor n maal continu differentieerbare f de formule van Newton (10.5) over in (vgl. stelling 11.3):

$$f(x) = f(x_0) + \frac{f'(x_0)}{1!} (x - x_0) + \frac{f''(x_0)}{2!} (x - x_0)^2 + \dots$$

$$+ \frac{f^{(n-1)}(x_0)}{(n-1)!} (x - x_0)^{n-1} + \frac{f^{(n)}(\xi)}{n!} (x - x_0)^n,$$

waarbij $\min(x_0, x) \leq \xi \leq \max(x_0, x)$.

Dit is de Taylor-ontwikkeling van f in het punt x_0 met restterm van Lagrange.

Bijna samenvallende basispunten.

We veronderstellen nu dat alle basispunten en het punt x , waar we willen interpoleren, bijna samenvallen, d.w.z. dat het kleinste interval $[a, b]$, dat x en alle basispunten bevat, zo klein is, dat $f^{(n)}(x)$ op dit interval nagenoeg constant is. Vergelijken we nu formule (10.7) met de formule uit stelling (11.3). Enerzijds levert toevoeging van x_n een n -de Newton term:

$$f[x_0, \dots, x_n] \pi_n(x) = \frac{f^{(n)}(\xi_1)}{n!} \pi_n(x)$$

(vgl. het bewijs van stelling 11.3);

anderzijds is de correctie-term

$$R_n(x) = \frac{f^{(n)}(\xi_2)}{n!} \pi_n(x).$$

Als nu $f^{(n)}(x)$ in het relevante interval nagenoeg constant is, zijn deze twee termen dus bijna gelijk.

M.a.w. als alle basispunten en het punt x dicht genoeg bij elkaar liggen, dan is voor zekere n de n -de Newton-term nagenoeg gelijk aan de correctie-term.

Hierop berust het praktisch gebruik van de formules van Newton en Aitken. Bij interpolatie (maar niet extrapolatie) in een voldoende fijne tabel geeft de laatste Newton-term (resp. de overeenstemming in de Aitken-waarden) een idee van de bereikte precisie.

Opmerking.

Voor de keuze der basispunten zijn allerlei systemen te bedenken. Bepaalde in zeker opzicht optimale keuzen (zoals b.v. Chebyshev-interpolatie) zullen we later behandelen. Thans gaan we aandacht schenken aan Newton interpolatie op equidistant gekozen basispunten.

Opgaven.

47) Zij gegeven de volgende tabel van de exponentiële functie

x	exp(x)
.0	1.00000
.1	1.10517
.2	1.22140
.3	1.34986
.4	1.49182
.5	1.64872
.6	1.82212
.7	2.01375
.8	2.22554
.9	2.45960
1.0	2.71828

- Welke precisie is bereikbaar met 4-punts polynoom-interpolatie?
- Bereken met 4-punts Lagrange benaderde waarden $\exp(x)$ voor $x = .125, .25, .375$ en controleer of de resultaten kloppen met de relatie $\exp(.375) = \exp(.125) * \exp(.25)$.
- Eveneens voor $x = .693, .694, .693147$ ($\approx \ln(2)$) en ga na of het laatste resultaat voldoende met 2 overeenstemt.

48) Zij gegeven de volgende tabel van de gemodificeerde Besselfunctie $10^{-22} x^{20} K_{20}(x)$ (NBS - AMS 55 pag. 426).

x	f(x)
18.0	.132454
18.2	.122321
18.4	.112891
18.6	.104124
18.8	.095978
19.0	.088414
19.2	.081397
19.4	.074892
19.6	.068865
19.8	.063285
20.0	.058124

$$\left[\begin{array}{c} (-4)4 \\ 4 \end{array} \right]$$

Het symbool onder de tabel betekent, dat lineaire interpolatie een precisie 4_{10}^{-4} levert en dat 4-punts Lagrange de volle tabel-precisie (dus in dit geval $.5_{10}^{-6}$) levert.

Bereken met 4-punts Lagrange $f(x)$ voor $x = 18.5, 19.1, 19.25, 19.3, 19.35$ en voor $x = 19.7312$.

15. Formule van Newton op equidistante basispunten

Evenals in sectie 9 voor de equidistante Lagrange-interpolatie stellen we, als h de afstand tussen 2 opeenvolgende basispunten is:

$$(15.1) \quad x_i = x_0 + ih, \quad i = k(1)l,$$

$$(15.2) \quad x = x_0 + ph.$$

We zullen k en l voorlopig weer vrij houden, waarbij natuurlijk de orde n gelijk is aan $l-k+1$.

Bij de equidistante Newton-formules treedt een kleine complicatie op, doordat we te maken hebben met twee ordeningen, nl. de natuurlijke volgorde en de volgorde van toevoeging. Blijkens (15.1) komt de index i overeen met de natuurlijke volgorde.

De toevoeging laten we steeds starten met x_0 en op een of andere systematische wijze verder gaan. De verschillende gekozen toevoegingsvolgorden leiden dan tot verschillende equidistante Newton-formules. Kijken we nu naar de gedeelde differenties. De berekening hiervan vergt steeds een deling door een veelvoud van h . Omdat h constant is (en delen met de hand onplezierig) stellen we deze deling liever uit. In plaats van gedeelde differenties gebruiken we daarom in het equidistante geval steeds differenties.

(15.3) Definitie voorwaartse differenties.

$$\text{eerste differentie: } \Delta f_i = \underset{\text{def}}{\Delta} f_{i+1} - f_i$$

(Iets duidelijker zou zijn de notatie $(\Delta f)_i$, maar deze haken worden gewoonlijk weggelaten.)

$$\begin{aligned} \text{tweede differentie: } \Delta^2 f_i &= \underset{\text{def}}{\Delta} \Delta f_{i+1} - \Delta f_i = \\ &= f_{i+2} - 2f_{i+1} + f_i \end{aligned}$$

$$\text{k-de differentie : } \Delta^k f_i = \underset{\text{def}}{\Delta} \Delta^{k-1} f_{i+1} - \Delta^{k-1} f_i$$

Volledigheidshalve moeten we ook definiëren de nulde differentie

$$\Delta^0 f_i = \underset{\text{def}}{\Delta} f_i$$

Als duidelijk is welke functie bedoeld wordt, laat men ook vaak het functie-symbool f weg en schrijft Δ_i , Δ_i^k , enz.

Blijkbaar geldt nu:

$$(15.4) \quad f [x_i, x_{i+1}] = (\Delta f_i)/h.$$

En voor de k -de differenties ($k \geq 0$) geldt:

$$(15.5) \quad f [x_i, \dots, x_{i+k}] = (\Delta^k f_i)/(k! h^k).$$

Dit laatste bewijzen we door inductie naar k . Het geldt blijkbaar voor $k = 0$ en voor $k = 1$ (zie formule 15.4). Nemen we nu aan dat formule (15.5) geldt voor $k-1$, dan hebben we voor k :

$$\begin{aligned} f [x_i, \dots, x_{i+k}] &= \frac{f [x_{i+1}, \dots, x_{i+k}] - f [x_i, \dots, x_{i+k-1}]}{x_{i+k} - x_i} = \\ &= \frac{(\Delta^{k-1} f_{i+1})/((k-1)! h^{k-1}) - (\Delta^{k-1} f_i)/((k-1)! h^{k-1})}{kh} = (\Delta^k f_i)/(k! h^k), \end{aligned}$$

waarmee de formule bewezen is.

Nu gaan we een equidistante Newton-formule opstellen, waarbij de basispunten worden toegevoegd in de natuurlijke volgorde:

x_0, x_1, \dots, x_{n-1} (dus $k = 0$ en $l = n-1$). We hebben dan:

$\pi_k(x) = (x - x_0) \dots (x - x_{k-1}) = p(p-1) \dots (p-k+1) h^k$ en

$$\begin{aligned} (15.6) \quad f_n^{**}(x_0 + ph) &= \sum_{k=0}^{n-1} f [x_0, \dots, x_k] \pi_k(x) = \\ &= \sum_{k=0}^{n-1} \frac{\Delta^k f_0}{k! h^k} p(p-1) \dots (p-k+1) h^k. \end{aligned}$$

Definiëren we de binomiaalcoëfficiënt $\binom{p}{k}$ voor k geheel ≥ 0 en p willekeurig:

$$(15.7) \quad \binom{p}{k} = \text{def} \frac{p(p-1) \dots (p-k+1)}{k!}.$$

Voor $k = 0$ hebben we te maken met lege producten, die altijd 1 zijn, zodat dus $\binom{p}{0} = 1$.

Als p geheel ≥ 0 is kunnen we teller en noemer vermenigvuldigen met $(p - k)!$ en krijgen we de compacte formule

$$(15.8) \quad \binom{n}{k} = \frac{n!}{k! (n - k)!}.$$

Formule (15.6) krijgt dan de gedaante

$$(15.9) \quad f_n^*(x_0 + ph) = \sum_{k=0}^{n-1} \binom{p}{k} \Delta^k f_0 = \\ = f_0 + p \Delta f_0 + \frac{p(p-1)}{2} \Delta^2 f_0 + \dots + \frac{p(p-1) \dots (p-n+2)}{(n-1)!} \Delta^{n-1} f_0.$$

Dit is de voorwaartse formule van Newton van de orde n .

De restterm luidt, mits $f^{(n)}$ bestaat:

$$R_n(x) = R_n(x_0 + ph) = \frac{f^{(n)}(\xi)}{n!} \pi_n(x) = \binom{p}{n} h^n f^{(n)}(\xi), \text{ waarbij}$$

$$\min(x_0, x) < \xi < \max(x_{n-1}, x).$$

Hier wordt duidelijk, waarom deze formule "van de orde n " heet. Als h naar nul gaat, gaat de restterm evenredig met h^n naar nul, m.a.w. de restterm is van de orde h^n .

Dit wordt wel genoteerd met het symbool O :

$$R_n(x_0 + ph) = O(h^n), \text{ d.w.z. } \lim_{h \rightarrow 0} \frac{R_n(x_0 + ph)}{h^n} \text{ bestaat en is ongelijk aan } 0.$$

Evenzo duidt het symbool o aan, dat de limiet nul is.

16. Functies, functionalen en operatoren

In de inleiding hebben we gesproken over functies en in de eerste plaats vermeld functies met reëel argument en reële waarde. De verzameling van deze functies hebben we aangeduid met $R \rightarrow R$, waarbij R de verzameling der reële getallen voorstelt. We gaan nu ook andere soorten functies beschouwen.

(16.1) Definitie: Een functionaal is een functie, waarvan het argument tot de verzameling $R \rightarrow R$ behoort en de functie-waarde een reëel getal is.

De verzameling der functionalen duiden we aan met $(R \rightarrow R) \rightarrow R$.

Voorbeelden van functionalen:

1) bepaalde integraal $I_{a,b}$ (a en b vast) gedefiniëerd door:

$$I_{a,b}(f) = \int_a^b f(x) dx \text{ (of kortweg aangeduid met } \int_a^b f \text{)}.$$

De functie f is argument en het reële getal $\int_a^b f(x) dx$ is waarde van de functionaal $I_{a,b}$.

2) de functionaal m gedefiniëerd door:

$$m(f) = \max_{x \geq 0} |f(x)|.$$

Deze functionaal voegt aan iedere begrensde reële functie een reëel getal toe.

3) de evaluator e_x (x vast) gedefiniëerd door:

$$e_x(f) = f(x).$$

Deze functionaal voegt aan iedere reële functie f , die x in zijn domein heeft, het reële getal $f(x)$ toe.

(16.2) Definitie: Een operator is een functie, waarvan argument en waarde beide tot de verzameling $R \rightarrow R$ behoren.

De verzameling der operatoren duiden we aan met $(R \rightarrow R) \rightarrow (R \rightarrow R)$.

Voorbeelden van operatoren:

1) de differentiatie-operator D , die aan elke differentiëerbare functie f toevoegt de afgeleide functie f' , gedefiniëerd door:

$$f'(x) = \frac{d f(x)}{dx}.$$

Notatie: $f' = D(f)$ of ook Df en voor de waarde van f' in x :

$$f'(x) = \frac{d f(x)}{dx} = (Df)(x).$$

2) de voorwaartse differentie operator Δ , die, bij gegeven h , aan een functie f toevoegt de functie Δf , gedefiniëerd door:

$$(\Delta f)(x) = f(x+h) - f(x).$$

Als index i de waarde in het punt x_i aanduidt, hebben we (vgl. 15.3):

$$\Delta f_i = (\Delta f)_i = f(x_i + h) - f(x_i) = f_{i+1} - f_i.$$

3) De schuif-operator E , gedefiniëerd door:

$$Ef(x) = f(x+h), \text{ dus } Ef_i = f_{i+1}.$$

16.3 Compositie en machten van operatoren.

De compositie of het product van twee operatoren A en B wordt verkregen door ze achter elkaar uit te voeren:

$$(AB)(f) = \text{def } A(B(f)).$$

Vaak laten we haken weg en schrijven $AB(f)$ of ABf .

Bij k gelijke operatoren spreken we van de k -de macht van een operator, die we gewoon met exponent k aanduiden:

$$A^2 = AA, \quad A^k = \underbrace{A \dots A}_{k \text{ stuks}}.$$

Exponent nul, "nul keer de operator toepassen", beduidt altijd de identieke operator I , die aan iedere f toevoegt f zelf. Dus

$A^0 = I$, oftewel $A^0(f) = I(f) = f$. Voor de identieke operator schrijft men vaak 1 .

Voorbeelden van compositie van operatoren:

$$D^k(f) = f^{(k)}, \text{ de } k\text{-de afgeleide, of voor de waarde in } x:$$

$$D^k(f)(x) = f^{(k)}(x).$$

$$\Delta^2 f = \Delta(\Delta f), \text{ of voor de waarde in } x:$$

$$\Delta^2 f(x) = \Delta f(x+h) - \Delta f(x) = f(x+2h) - 2f(x+h) + f(x).$$

$$\Delta^k f = \Delta(\Delta^{k-1} f), \text{ of voor de waarde in } x_i \text{ (vgl. 15.3):}$$

$$\Delta^k f_i = (\Delta^k f)_i = \Delta(\Delta^{k-1} f)_i = \Delta^{k-1} f_{i+1} - \Delta^{k-1} f_i.$$

Machtreeksen van operatoren

Men kan een operator vaak in een machtreeks ontwikkelen, die met de operator equivalent is, als het argument van de operator een polynoom is. Deze machtreeksen lijken vaak op machtreeksen van gewone functies en zijn daardoor een geschikt hulpmiddel om formules te onthouden. Hier volgt een enkel voorbeeld. De Taylor-ontwikkeling van $f(x + h)$ in het punt x (die o.a. voor polynomen exact is) kan als volgt worden geschreven:

$$Ef(x) = f(x + h) = \sum_{k=0}^{\infty} \frac{h^k}{k!} D^k f(x) = (\exp(hD)f)(x).$$

M.a.w. de schuifoperator E kan worden geschreven als:

$$E = \exp(hD) = \sum_{k=0}^{\infty} \frac{h^k}{k!} D^k.$$

17. De differentie-operatoren

Behalve voorwaartse differenties, worden ook achterwaartse en centrale differenties gebruikt. Ze kunnen alle worden uitgedrukt in de schuifoperator E gedefiniëerd door

$$(17.1) \quad Ef(x) = f(x + h).$$

Dus $\Delta f(x) = f(x + h) - f(x) = (E - I) f(x)$.

Dit wordt kortweg uitgedrukt door de formule

$$(17.2) \quad \Delta = E - I.$$

Voor de machten van E geldt:

$$E^k f(x) = f(x + kh).$$

Als k geheel ≥ 0 , volgt dit uit de definitie in (16.3), maar we gebruiken de formule ook voor negatieve of niet gehele exponent p .

Hiervoor moeten we dus definiëren:

$$(17.3) \quad E^p f(x) = f(x + ph), \quad p \text{ willekeurig.}$$

De achterwaartse differentie wordt gedefiniëerd door:

$$(17.4) \quad \nabla = I - E^{-1},$$

$$\text{d.w.z.} \quad \nabla f(x) = f(x) - f(x - h),$$

of voor de waarde in het punt x_i :

$$\nabla f_i = f_i - f_{i-1}.$$

En de centrale differentie δ wordt gedefiniëerd door

$$(17.5) \quad \delta = E^{\frac{1}{2}} - E^{-\frac{1}{2}},$$

$$\text{d.w.z.} \quad \delta f(x) = f\left(x + \frac{h}{2}\right) - f\left(x - \frac{h}{2}\right),$$

$$\text{ofwel} \quad \delta f_{i+\frac{1}{2}} = f_{i+1} - f_i,$$

$$\delta^2 f_i = \delta f_{i+\frac{1}{2}} - \delta f_{i-\frac{1}{2}} = f_{i+1} - 2f_i + f_{i-1}.$$

(17.4) Relatie tussen Δ , ∇ en δ

Uit het bovenstaande volgt:

$$\Delta = \nabla E = \delta E^{\frac{1}{2}},$$

$$\text{ofwel:} \quad \Delta f_i = \nabla f_{i+1} = \delta f_{i+\frac{1}{2}} = f_{i+1} - f_i.$$

M.a.w. de drie soorten differenties zijn slechts verschillende namen voor dezelfde getallen. Uitgeschreven krijgen we de volgende schema's, waarin elke differentie gelijk is aan de links-onderstaande minus de links-bovenstaande.

Voorwaartse differenties

f_{-2}			
	Δ_{-2}		
f_{-1}		Δ_{-2}^2	
	Δ_{-1}		Δ_{-2}^3
f_0		Δ_{-1}^2	
	Δ_0		Δ_{-1}^3
f_1		Δ_0^2	
	Δ_1		Δ_0^3
f_2		Δ_1^2	
	Δ_2		Δ_1^3
		Δ_2^2	
			Δ_2^3

Centrale differenties

f_{-2}		δ_{-2}^2	
	$\delta_{-3/2}$		$\delta_{-3/2}^3$
f_{-1}		δ_{-1}^2	
	$\delta_{-1/2}$		$\delta_{-1/2}^3$
f_0		δ_0^2	
	$\delta_{1/2}$		$\delta_{1/2}^3$
f_1		δ_1^2	
	$\delta_{3/2}$		$\delta_{3/2}^3$
f_2		δ_2^2	

Achterwaartse
differenties

f_{-2}		∇_{-1}^2	
	∇_{-1}		∇_{-1}^2
f_{-1}		∇_0^2	
	∇_0		∇_0^2
f_0		∇_1^2	
	∇_1		∇_1^2
f_1		∇_2^2	
	∇_2		∇_2^2
f_2			

18. De vier equidistante Newton-interpolatie-formules

We zagen reeds, dat toevoeging der basispunten in de natuurlijke volgorde x_0, x_1, \dots, x_{n-1} leidt tot de voorwaartse formule van Newton

$$(18.1) \quad f^*(x_0 + ph) = \sum_{k=0}^{n-1} \frac{p(p-1)\dots(p-k+1)}{k!} \Delta^k f_0 = \sum_{k=0}^{n-1} \binom{p}{k} \Delta^k f_0$$

met restterm $R_n(x_0 + ph) = \binom{p}{n} h^n f^{(n)}(\xi)$.

Deze formule wordt gebruikt voor interpolatie aan het begin van een tabel. Zij is gemakkelijk te onthouden door vergelijking met de machtsreeks

$$(1+x)^p = \sum_{k=0}^{\infty} \binom{p}{k} x^k.$$

Laten we in (18.1) n naar ∞ gaan, dan convergeert de reeks meestal niet naar $f(x_0 + ph)$, maar wel als f een polynoom is. Dan krijgen we

$$E^p f_0 = f(x_0 + ph) = \sum_{k=0}^{\infty} \binom{p}{k} \Delta^k f_0 = (I + \Delta)^p f_0.$$

Of in operator-taal:

$$E^p = (I + \Delta)^p = \sum_{k=0}^{\infty} \binom{p}{k} \Delta^k.$$

Dit laatste hadden we formeel meteen kunnen afleiden uit (17.2).

Nu gaan we de basispunten toevoegen in de tegen-natuurlijke volgorde: $x_0, x_{-1}, \dots, x_{1-n}$ (dus nu geldt voor de range der indices $k = 1 - n$ en $l = 0$). In dit geval gebruiken we het achterwaartse differentie-schema en krijgen de achterwaartse formule van Newton

$$(18.2) \quad f^{**}(x_0 + ph) = \sum_{k=0}^{n-1} \frac{p(p+1) \dots (p+k-1)}{k!} \nabla^k f_0 = \\ = \sum_{k=0}^{n-1} \binom{-p}{k} (-1)^k \nabla^k f_0.$$

Deze formule wordt gebruikt aan het eind van een tabel. Om deze te onthouden gaan we uit van (17.4) die onmiddellijk levert:

$$E^{-1} = I - \nabla.$$

$$\text{Dus } f(x_0 + ph) = E^p f_0 = (E^{-1})^{-p} f_0 = (I - \nabla)^{-p} f_0 = \sum_{k=0}^{\infty} \binom{-p}{k} (-1)^k \nabla^k f_0,$$

welke relatie geldt voor polynomen f .

Voor interpolatie in het midden van een tabel gebruiken we liever een formule, waar x_0 ongeveer midden tussen x_k en x_l ligt. Wij beschouwen de toevoegingsvolgorde $x_0, x_1, x_{-1}, x_2, x_{-2}, \dots$.

Dan geldt als n oneven is: $k = (1 - n)/2$, $l = (n - 1)/2$

en als n even is: $k = 1 - n/2$, $l = n/2$.

De formule die ontstaat heet de voorwaarts centrale formule van Newton of ook voorwaartse formule van Gauss:

$$(18.3) \quad f^{**}(x_0 + ph) = f_0 + \binom{p}{1} \delta_{\frac{1}{2}} + \binom{p}{2} \delta_0^2 + \binom{p+1}{3} \delta_{\frac{3}{2}} \\ + \binom{p+1}{4} \delta_0^4 + \dots$$

De coëfficiënt van δ_0^{2i} is $\binom{p+i-1}{2i}$, de coëfficiënt van $\delta_{\frac{1}{2}}^{2i+1}$ is $\binom{p+i}{2i+1}$.

De restterm voor n maal differentieerbare f luidt:

$$\binom{p + (n-1) \div 2}{n} h^n f^{(n)}(\xi).$$

Tenslotte beschouwen we de toevoegingsvolgorde

$$x_0, x_{-1}, x_1, x_{-2}, x_2, \dots$$

Dan geldt voor oneven n : $k = (1 - n)/2$, $l = (n - 1)/2$

en voor even n : $k = -n/2$, $l = n/2 - 1$.

Dit leidt tot de achterwaarts centrale formule van Newton of ook achterwaartse formule van Gauss:

$$(18.4) \quad f^{**}(x_0 + ph) = f_0 + \binom{p}{1} \delta_{-\frac{1}{2}} + \binom{p+1}{2} \delta_0^2 + \binom{p+1}{3} \delta_{-\frac{3}{2}} \\ + \binom{p}{4} \delta_0^4 + \dots$$

De coëfficiënt van δ_0^{2i} is nu $\binom{p+i}{2i}$ en de coëfficiënt van $\delta_{-\frac{1}{2}}^{2i+1}$ is $\binom{p+i}{2i+1}$.

De restterm luidt nu: $\binom{p + n \div 2}{n} h^n f^{(n)}(\xi)$.

Opgaven

49) Bewijs de formules:

$$\Delta^k f_j = \sum_{i=0}^k \binom{k}{i} (-1)^i f_{j+k-i} ;$$

$$\nabla^k f_j = \sum_{i=0}^k \binom{k}{i} (-1)^i f_{j-i} .$$

50) Bereken het differentie-schema van de tabel, gegeven in opgave 47, totdat de differenties verwaarloosbaar worden.

Bereken vervolgens hieruit met een geschikte Newton-formule $\exp(x)$ voor $x = 0.01, 0.05, 0.09531$ en voor $x = 0.95, 0.99$ en 0.90469 .

51) Bereken het differentie-schema van de tabel in opgave 48, totdat de differenties verwaarloosbaar worden.

Bereken hieruit met Newton-interpolatie $f(x)$ voor $x = 18.1, 18.01, 18.88, 19.9$ en 19.925 .

19. Gesymmetriseerde centrale interpolatie-formules

De centrale formules van Newton zijn niet helemaal symmetrisch. Voor- dat we symmetrische formules gaan afleiden voeren we in de gemiddelde operator μ , gedefiniëerd door:

$$(19.1) \quad \mu f_{i+\frac{1}{2}} = \frac{1}{2} (f_i + f_{i+1}).$$

Of in operator-taal: $\mu = \frac{1}{2} (E^{-\frac{1}{2}} + E^{\frac{1}{2}})$.

De formule van Stirling

Nemen we nu het gemiddelde van de voorwaartse en achterwaartse Gauss-formules (18.3 en 18.4), dan ontstaat de formule van Stirling:

$$(19.2) \quad f^*(x_0 + ph) = f_0 + p\mu\delta f_0 + S_2(p)\delta_0^2 + S_3(p)\mu\delta_0^3 + \dots,$$

waarbij $S_{2i}(p) = \binom{p+i-1}{2i-1}$ en $S_{2i+1}(p) = \binom{p+i}{2i+1}$.

Deze formule is symmetrisch rond $x = x_0$, dus $p = 0$.

De formule van Bessel

Een veel belangrijker formule, die symmetrisch is rond $x = (x_0 + x_1)/2$, dus $p = \frac{1}{2}$, ontstaat als we het gemiddelde nemen van de voorwaartse Gauss-formule (18.3) en de verschoven achterwaartse Gauss-formule startend in x_1 , waar dus de punten worden toegevoegd in de volgorde:

$$x_1, x_0, x_2, x_{-1}, x_3, x_{-2}, x_4, x_{-3}, \dots$$

De verschoven achterwaartse Gauss ziet er dus zo uit:

$$f^*(x_0 + ph) = f^*(x_1 + (p-1)h) = f_1 + \binom{p-1}{1}\delta_{\frac{1}{2}} + \binom{p}{2}\delta_1^2 + \binom{p}{3}\delta_{\frac{1}{2}}^3 + \\ + \binom{p+1}{4}\delta_1^4 + \dots$$

De coëfficiënt van δ_1^{2i} is $\binom{p+i-1}{2i}$ en die van $\delta_{\frac{1}{2}}^{2i+1}$ is $\binom{p+i-1}{2i+1}$.

Het gemiddelde van deze en (18.3) levert de formule van Bessel:

$$(19.3) f^{**}(x_0 + ph) = u f_{\frac{1}{2}} + (p - \frac{1}{2})\delta_{\frac{1}{2}} + B_2(\delta_0^2 + \delta_1^2) + B_3 \delta_{\frac{1}{2}}^3 + \\ + B_4(\delta_0^4 + \delta_1^4) + \dots = f_0 + p \delta_{\frac{1}{2}} + B_2(\delta_0^2 + \delta_1^2) + B_3 \delta_{\frac{1}{2}}^3 + \dots$$

De Besselcoëfficiënten $B_i = B_i(p)$ zijn dus:

$$(19.4) B_{2i}(p) = \frac{1}{2} \binom{p+i-1}{2i}, B_{2i+1}(p) = \frac{p-\frac{1}{2}}{2i+1} \binom{p+i-1}{2i} = \\ = \frac{2p-1}{2i+1} B_{2i}(p).$$

Vanwege de symmetrie in $p = \frac{1}{2}$ voldoen zij aan de symmetrie-relaties:

$$(19.5) B_{2i}(1-p) = B_{2i}(p), B_{2i+1}(1-p) = -B_{2i+1}(p).$$

De Besselcoëfficiënten B_i zijn getabelleerd in "Interpolation & Allied Tables" [3]. Wegens de relaties (19.5) hoeven zij, om het interval $[0,1]$ te bestrijken, slechts getabelleerd te zijn voor $0 \leq p \leq \frac{1}{2}$.

Nu iets over de Restterm van Bessel's formule.

Als de orde n even is ($n = 2l$ zeg), zijn zowel de voorwaartse Gauss als de verschoven achterwaartse Gauss-formule gebaseerd op de basispunten x_j , $j = 1 - n/2 (1) n/2 = 1 - l (1) l$.

De even orde Bessel-formule is dus ook op deze punten gebaseerd.

De restterm luidt, mits $f^{(2l)}$ bestaat en continu is in het interval, dat alle basispunten en het punt x bevat:

$$(19.6) R_{2l}(x) = \binom{p+l-1}{2l} h^{2l} \frac{f^{(2l)}(\xi_1) + f^{(2l)}(\xi_2)}{2} = \\ = \binom{p+l-1}{2l} h^{2l} f^{(2l)}(\xi) = 2B_{2l} h^{2l} f^{(2l)}(\xi).$$

Hierbij wordt gebruikt de middelwaarde-stelling, toegepast op de continue $2l$ -de afgeleide van f en ξ ligt weer in het bovengenoemde interval.

Als de orde n oneven is ($n = 2l + 1$ zeg) zijn de gebruikte basispunten x_j , $j = -l (1) l + 1$, waarbij evenwel de buitenste punten ieder slechts in een der betreffende Gauss-formules gebruikt worden. De oneven Bessel-formule is dus geen interpolatie-formule in de boven gedefiniëerde zin

(zie sectie 5), wat overigens aan haar praktische bruikbaarheid niets afdoet. De restterm heeft evenwel in dit geval een iets minder prettige vorm:

$$(19.7) \quad R_{2l+1}(x) = h^{2l+1} \binom{p+1}{2l+1} f^{(2l+1)}(\xi_1) + \\ + \binom{p+1-1}{2l+1} f^{(2l+1)}(\xi_2).$$

De formule van Everett:

In de formule van Bessel kunnen alle oneven differenties weggewerkt worden door toepassing van de relatie:

$$\delta_{\frac{1}{2}}^{2i+1} = \delta_1^{2i} - \delta_0^{2i}.$$

Hierdoor ontstaat de formule van Everett:

$$(19.8) \quad f''(x_0 + ph) = (1-p)f_0 + p f_1 + E_2 \delta_0^2 + F_2 \delta_1^2 + E_4 \delta_0^4 + F_4 \delta_1^4 + \dots,$$

waarbij voor de Everett-coëfficiënten $E_{2i} = E_{2i}(p)$ en $F_{2i} = F_{2i}(p)$ geldt:

$$(19.9) \quad E_{2i} = B_{2i} - B_{2i+1} = - \binom{p+i-1}{2i+1}; \quad F_{2i} = B_{2i} + B_{2i+1} = + \binom{p+i}{2i+1}.$$

Vanwege de symmetrie in $p = \frac{1}{2}$ voldoen deze aan de symmetrie-relatie:

$$(19.10) \quad E_{2i}(1-p) = F_{2i}(p).$$

De Everett-coëfficiënten zijn getabelleerd in [3].

Gebruik in de praktijk

In de praktijk van het rekenen met tafel-machines worden de formules van Bessel en Everett vaak gebruikt.

De formule van Everett vereist iets minder werk dan die van Bessel. Is echter de laatste differentie, die in rekening gebracht moet worden van even orde (dus de formule van oneven orde), dan kan men gebruiken de Everett-formule met een Besselterm. Bijvoorbeeld als 5-de orde formule krijgt men dan:

$$f(x_0 + ph) \approx (1-p)f_0 + p f_1 + E_2 \delta_0^2 + F_2 \delta_1^2 + B_4(\delta_0^4 + \delta_1^4).$$

Opgaven

- 52) Bewijs de symmetrie-relaties (19.5) en (19.10) direkt uit de formules (19.4) resp. (19.9).
- 53) Bereken de waarden $B_k(0.5)$ voor $k = 2(1)7$ en schrijf de zo verkregen middenpunt-formule van Bessel uit.
- 54) Zoek voor $k = 2, 3, 4$ de waarde p_m van p in het interval $[0,1]$, waar de Newton-coëfficiënten $\binom{p}{k}$ en de Bessel-coëfficiënten $B_k(p)$ hun grootste absolute waarde bereiken en bereken ook de bijbehorende waarden $\binom{p}{k_m}$ en $B_k(p_m)$.
- 55) Zij gegeven de volgende tabel van $f(x) = \sin(\pi x/180)$:

x	$\sin(\pi x/180)$
28	.469472
31	.515038
33	.544639
37	.601815

Bepaal $f(30)$ met 2, 3, 4-punts Lagrange,
 met opeenvolgende Newton-benaderingen,
 met de herhaalde lineaire interpolatie volgens Aitken.

- 56) Bepaal met behulp van gedeelde differenties het 3^e-graads polynoom f , dat voldoet aan

$$f(0) = f'(0) = 4,$$

$$f(1) = f'(1) = 10.$$

57) Zij gegeven de volgende tabel van $f(x) = \sin(x)$:

x	sin (x)
.5	.47943
.7	.64422
.9	.78333
1.1	.89121
1.3	.96356
1.5	.99749
1.7	.99166

Bereken benaderde waarden $f(x)$ met een geschikte Newton-formule voor $x = .5(.05).7$ en $1.5(.05)1.7$, en vervolgens met Bessel's formule voor $x = .9(.05)1.1$.

Erratum

pagina 78 formule (18.4) $\binom{p}{4} \delta_0^4$ moet zijn $\binom{p+2}{4} \delta_0^4$.

20. Het gedrag van de differenties

Beschouw een gegeven functie f , die voldoende vaak differentieerbaar is, en een equidistante interpolatie-formule van de orde n .

Als het tabelinterval h zo klein is, dat de n -de afgeleide van f nagenoeg constant is, geeft de laatst in rekening gebrachte Newton-term een goed idee van de bereikte precisie (zie p. 66).

Zowel de restterm $R_n(x)$ als de n -de differentie zijn van de orde h^n , d.w.z. dat zij na deling door h^n als limiet een getal $\neq 0$ opleveren (vgl. formule 15.5 en de in secties 15, 18 en 19 vermelde restterm-formules). B.v. geldt voor de Bessel-formule van orde n (als we p vast houden):

$$(20.1) \quad \lim_{h \rightarrow 0} \frac{R_n(x_0 + ph)}{h^n} = 2B_n f^{(n)}(x_0) = 2B_n(p) \lim_{h \rightarrow 0} \frac{\delta^n f_0}{h^n}.$$

Hieruit volgt, dat voor voldoende vaak differentieerbare f en voldoende kleine h de differenties van de orde $\leq n$ in absolute waarde steeds kleiner worden. De hogere orde differenties gaan, evenals de hogere afgeleiden van f , meestal weer toenemen. Dit gebeurt alleen niet bij zeer tamme functies, in de analyse bekend als "gehele analytische functies", dat zijn functies die als machtreeks met oneindige convergentie-straal voorstelbaar zijn (b.v. \exp , \cos en natuurlijk de polynomen).

Als we de orde van een interpolatie-formule naar oneindig laten gaan krijgen we dan ook meestal geen convergente reeks, tenzij voor de bovengenoemde tamme functies. De praktische bruikbaarheid ligt bij een geschikt gekozen orde en tabel-interval, beide afhankelijk van de gewenste nauwkeurigheid.

Invloed van afrondingsfouten

In de praktijk hebben we gewoonlijk te maken met een (getabelleerde) functie f_T , die de som is van een nette, vaak differentieerbare, functie $f(x)$ en een zich allerminst netjes gedragende storing $\epsilon(x)$.

Dus: $f_{\mathbb{T}}(x_i) = f(x_i) + \varepsilon(x_i)$.

Als de storing $\varepsilon(x)$ alleen door afronding van het eindresultaat ontstaat is zij in absolute waarde \leq de helft van de laatste getabelleerde eenheid (b.v. voor een tabel in 6 decimalen: $|\varepsilon(x)| \leq 0.5_{10}^{-6}$).

Vaak wordt de storing $\varepsilon(x)$ mede veroorzaakt door andere effecten, zoals afronding van tussen-resultaten, in welk geval de bovengrens voor ε iets hoger uitvalt.

Bekijken we nu het differentie-schema van de storingsfunctie $\varepsilon(x)$. Het meest ongunstige beeld ontstaat, als de storingen $\varepsilon(x_i)$ voor opeenvolgende waarden van i alternerend van teken en gelijk in absolute waarde zijn. Houden we het op de bovengenoemde grens van 0.5 maal de laatste getabelleerde eenheid, dan ziet dit ongunstige beeld er zo uit:

ε	δ	δ^2	δ^3	δ^4
+.5				
	-1			
-.5		+2		
	+1		-4	
+.5		-2		+8
	-1		+4	
-.5		+2		
	+1			
+.5				

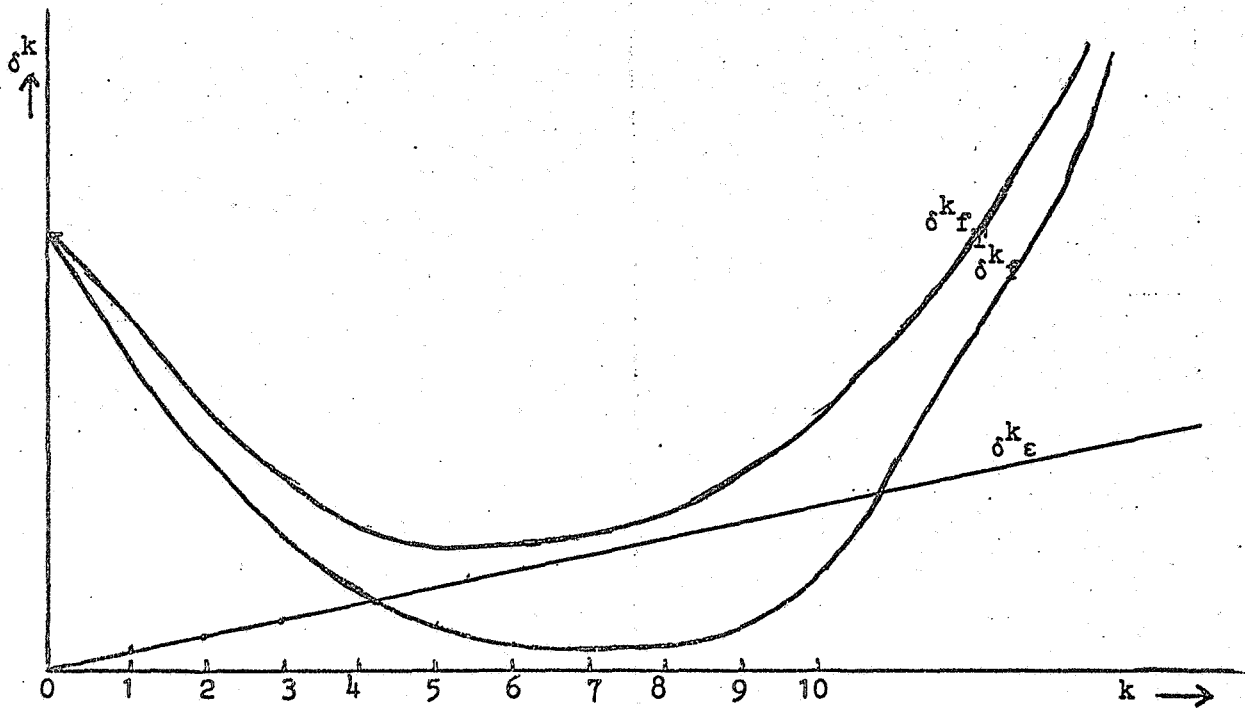
Kortom, als de storing $\varepsilon(x)$ uitsluitend ontstaat door afronding der functiewaarden, geldt:

$$(20.2) \quad |\delta^k \varepsilon_i| \leq 2^{k-1} \times \text{de laatste getabelleerde eenheid.}$$

Omdat de differentie-operator een lineaire operator is, geldt voor de differenties van de getabelleerde functie $f_{\mathbb{T}}$

$$(20.3) \quad \delta^k f_{\mathbb{T}} = \delta^k f + \delta^k \varepsilon.$$

Het totale beeld komt er dus zo uit te zien:



Interpoleren zal dus, bij gegeven tabelinterval en tabel-precisie, het beste resultaat opleveren voor een orde k , waarbij $\delta^k f$ en $\delta^k \epsilon$ ongeveer even groot zijn.

De blunder

Een fout in een enkele getabelleerde functie-waarde is aan het differentie-schema vrij gemakkelijk te onderkennen en zelfs te corrigeren. Beschouwen we hiertoe het differentie-schema van de blunder-functie $b(x)$, die 1 is in het punt x_0 en overall elders 0:

b	δ	δ^2	δ^3	δ^4	δ^5	δ^6
0		0		0		1
0	0		0		1	
0	0	0		1		-6
0	0		1		-5	
0	1			-4		15
1		-2		6		-20
	-1		3		-10	
0		1		-4		15
	0		-1		5	
0		0		1		-6
	0		0		-1	
0		0		0		1

De k -de differentie van de blunder $b(x)$ blijkt, op teken na, een binomiaal-coëfficiënt te zijn; in formule (vgl. opgave 49):

$$(20.4) \quad \nabla^k b_i = (-1)^i \binom{k}{i}.$$

Het optreden van een fout uit zich dus in een plaatselijk toenemen van de k -de differentie. Is k zo groot, dat de differentie elders verwaarloosbaar wordt, dan zit de fout in het midden van de bult en wordt de correctie het nauwkeurigst berekend uit de topwaarde gedeeld door de betreffende binomiaal-coëfficiënt.

Voorbeeld

x	f(x)	δ	δ^2
0	61644		
1	62450	806	-10
2	63246	796	-11
3	64031	785	-9
4	64807	776	-36
5	65547	740	+45
6	66332	785	-35
7	67082	750	0
8	67832	750	-5
9	68577	745	-40
10	69282	705	+13
11	70000	718	-7
12	70711	711	-8
13	71414	703	-6
14	72111	697	

f(5) is fout en moet zijn 65574, wat wijst op verwisseling van 2 cijfers;
 f(8) is fout en moet zijn 67824 of 67823 (weer verwisseling van 2
 cijfers?); tenslotte zit er nog een fout in f(9).

Opgaven

58) Doe opgaven 39, 47 en 48 nu met de formule van Bessel of die van Everett.

59) Enige van de volgende 20 opeenvolgende waarden, horende bij equidistante argumenten, zijn incorrect wegens typische overschrijffouten.

Localiseer en corrigeer deze.

17278	48818	79779	112630
23424	54440	86249	119398
29585	60723	92752	126246
35764	67041	99318	133180
41964	73398	105937	140206

21. Het gedrag van de interpolatie-coëfficiënten

We zullen nu vergelijken het gedrag van de coëfficiënten van de voorwaartse Newton-formule en een centrale formule, nl. Bessel. Hiertoe berekenen wij de maximale absolute waarde dezer coëfficiënten in het interval $0 \leq p \leq 1$ voor orde ≤ 6 , wat wij vinden door als p_{\max} een nulpunt van de afgeleide te nemen (vgl. opgave 54).

We krijgen dan het volgende tabelletje:

orde k	p_{\max}	$ \binom{p}{k} _{\max}$	p_{\max}	$ B_k _{\max}$	$ \delta^k _{\max}$ zodat term $< .5$
2	.500	.12500	.500	.06250	4
3	.423	.06415	.211 & .789	.00802	60
4	.382	.04167	.500	.01172	20
5	.356	.03026	.219 & .781	.00087	500
6	.337	.02347	.500	.00244	100

Men ziet, dat de Bessel-coëfficiënten veel sneller afnemen dan de coëfficiënten van de voorwaartse Newton, waarbij echter bedacht moet worden, dat de even orde Bessel-coëfficiënten vermenigvuldigd worden met de som van twee differenties.

De laatste kolom geeft het maximum voor $|\delta^k|$, waarvoor de k-de Bessel-term een verwaarloosbare bijdrage $< 0.5 \times$ laatste eenheid levert.

(Zie ook [3] pag. 56).

Omdat de coëfficiënten van centrale formules veel sneller afnemen dan die van de voorwaartse en achterwaartse formules, gebruikt men in de praktijk altijd een centrale formule (gewoonlijk Bessel of Everett), tenzij dit onmogelijk is (aan het begin of eind van een tabel).

Opmerking

Het komt nogal eens voor, dat een functie-waarde $f_p = f(x_0 + ph)$ berekend moet worden voor een waarde van p , waarvoor de betreffende interpolatie-coëfficiënten (van Bessel, Everett en Lagrange) niet getabelleerd zijn. Men kan dan het beste de functie berekenen voor 3 naburige getabelleerde waarden van p . Bereken van deze fijnere tabel de eerste en tweede differenties. Meestal zal de tweede differentie verwaarloosbaar (< 4 eenheden) zijn, zodat f_p kan worden verkregen door lineaire interpolatie in de fijnere tabel.

Bv.: stel dat men een tabel van Bessel-coëfficiënten heeft voor $p = 0(.01)1.00$ en men moet interpoleren voor $p = .3333$.

Bereken dan f_p voor $p = .33, .34$ en $.35$. Als de tweede differentie < 4 eenheden is, verkrijgt men f_p door lineair te interpoleren tussen $.33$ en $.34$. Is de tweede differentie niet verwaarloosbaar, dan berekene men nog f_p voor $p = .32$, kijk of de derde differentie verwaarloosbaar is, enz.

22. Throwback, het terugwerpen van een differentie op een van lagere orde

Deze truc van Comrie stelt ons in staat, met vrijwel dezelfde hoeveelheid rekenwerk en schijnbaar dezelfde interpolatie-formule een veel hogere precisie te halen. Nemen we in de formule van Bessel de termen met tweede en vierde differenties samen, dan hebben wij:

$$(22.1) \frac{1}{2} \binom{p}{2} (\delta_0^2 + \delta_1^2) + \frac{1}{2} \binom{p+1}{4} (\delta_0^4 + \delta_1^4) = \\ = B_2 \left(\binom{2}{0} + \binom{2}{1} + \frac{1}{12} (p+1)(p-2)(\delta_0^4 + \delta_1^4) \right).$$

In het interval $0 \leq p \leq 1$ verandert de factor $\frac{1}{12} (p+1)(p-2)$ maar weinig (maximum = $-\frac{3}{16}$ voor $p = \frac{1}{2}$, minimum = $-\frac{1}{6}$ voor $p = 0$ en 1).

Vervangen we deze factor door een geschikte constante C , dan kunnen we op deze manier de vierde orde term grotendeels verdisconteren. In plaats van (22.1) nemen we dus

$$B_2(\delta_0^2 + C \delta_0^4 + \delta_1^2 + C \delta_1^4),$$

waarbij we de volgende fout maken in de coëfficiënt van de vierde differenties:

$$B_2\left(\frac{1}{12}(p+1)(p-2) - C\right).$$

We kiezen C zo, dat deze fout extremen van wisselend teken en gelijke absolute waarde heeft en krijgen dan als beste waarde

$$(22.2) \quad C = -\frac{3 + \sqrt{2}}{24} \approx -.184.$$

De fout in de vierde orde term is dan 50 maal zo klein als de vierde orde term zelf, dus verwaarloosbaar, als $\delta^4 < 1000$.

We modificeren nu de tweede differenties als volgt:

$$(22.3) \quad \delta_{im}^2 = \delta_i^2 - .184 \delta_i^4 \quad (i = 0, 1)$$

en krijgen dan de volgende gewijzigde Bessel-formule

$$(22.4) \quad f(x_0 + ph) \approx f_0 + p \delta_{\frac{1}{2}} + B_2(\delta_{0m}^2 + \delta_{1m}^2) + B_3 \delta_{\frac{1}{2}}^3.$$

Vervangen we nu ook $\delta_{\frac{1}{2}}^3$ door de overeenkomstige $\delta_{\frac{1}{2m}}^3$, wat mag als $\delta_{\frac{1}{2}}^5$ verwaarloosbaar is, maar anders meestal ook geen kwaad kan, dan krijgen we de volgende gewijzigde Everett-formule:

$$(22.5) \quad f(x_0 + ph) \approx f_0 + p \delta_{\frac{1}{2}} + E_2 \delta_{0m}^2 + F_2 \delta_{1m}^2.$$

Deze formule mag worden gebruikt als $\delta^4 < 1000$ en $\delta^5 < 500$.

Het hier geschetste procédé heet "terugwerpen van de vierde differenties op de tweede differenties".

Het idee kan worden uitgebreid tot terugwerpen van hogere differenties op de tweede of op de tweede en vierde differenties, enz. Voor bijzonderheden, zie [3] pag. 57.

Sommige tabellen geven naast de functie-waarden ook de gemodificeerde tweede en vierde differenties. In deze tabellen kan men direct de betreffende Everett-formule gebruiken. Worden gewijzigde differenties opgegeven, dan mag men hiermee natuurlijk niet hogere differenties maken.

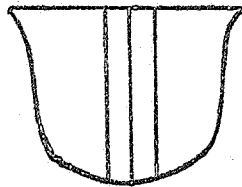
Voorbeeld, tabel van $\sin(\pi x/180)$ in 5 decimalen.

x	$\sin(\pi x/180)$	δ^2	δ^4	δ_m^2
0	.00000	0	0	0
15	.25882	-1764	121	-1786
30	.50000	-3407	231	-3450
45	.70711	-4819	329	-4880
60	.86603	-5902	402	-5976
75	.96593	-6583	450	-6666
90	1.00000	-6814	462	-6899

Opmerking

Throwback leidt tot formules, die niet voldoen aan de definitie van interpolatie in sectie 5. De gewijzigde formules stemmen nog wel met f overeen in de basispunten x_0 en x_1 , maar niet meer in de andere gebruikte basispunten. Om het onderste uit de kan te krijgen kan men het idee van exacte overeenstemming geheel loslaten en zoeken naar een z.g.n. "geëconomiseerd" polynoom, dat in een zeker interval zo min mogelijk van de gegeven functie afwijkt.

Zie bv. [3] pag. 60, waar de coëfficiënten van geëconomiseerde polynomen zijn uitgedrukt in differenties.



Hoofdstuk 3. Numeriek differentiëren en integreren

0. Inleiding

Afgeleiden en bepaalde integralen zijn samen te vatten onder het begrip functionaal. Zoals we in het vorige hoofdstuk, sectie 16, gezien hebben is een functionaal een functie, waarvan het argument een reële functie (behorend tot de verzameling $R \rightarrow R$) en de functie-waarde een reëel getal is. De bepaalde integraal \int_a^b is de functionaal, die aan elke op $[a, b]$ integreerbare reële functie f toevoegt het reële getal $\int_a^b f(x) dx$ (kortweg geschreven als $\int_a^b f$, mits duidelijk is wat de integratie-variabele is). De k -de afgeleide in het punt a , aangeduid als D_a^k , is de functionaal, die aan elke k maal differentieerbare reële functie f toevoegt het reële getal $D_a^k(f) = \left(\frac{d^k}{dx^k} f(x) \right)_{x=a} = f^{(k)}(a)$.

Voor $k = 0$ is dit de evaluator e_a , die de waarde van f levert in het punt a :

$$e_a(f) = D_a^0(f) = f(a).$$

Al deze functionalen hebben de prettige eigenschap van lineariteit, d.w.z.:

(0.1) Definitie. Een functionaal p is lineair, als voor elke f en g uit het domein van p en voor elke reële α en β geldt:

$$p(\alpha f + \beta g) = \alpha p(f) + \beta p(g).$$

Anders gezegd, p is lineair als voor elke f en g uit zijn domein

$$p(f + g) = p(f) + p(g)$$

en bovendien voor elke reële α

$$p(\alpha f) = \alpha p(f).$$

De tweede formulering is equivalent met de eerste. De analyse leert, dat zowel (bepaalde) integraal, als differentiëren lineaire functionalen zijn, in formule:

$$\int_a^b \alpha f(x) + \beta g(x) dx = \alpha \int_a^b f(x) dx + \beta \int_a^b g(x) dx,$$

$$\frac{d}{dx} (\alpha f(x) + \beta g(x)) = \alpha \frac{df(x)}{dx} + \beta \frac{dg(x)}{dx}.$$

Voor het berekenen van een benaderde waarde van $p(f)$ gaan we uit van een interpolatie-formule voor f . Deze heeft, zoals we in het vorige hoofdstuk steeds gezien hebben, de gedaante:

$$f(x) = \sum_{j=0}^{n-1} a_j G_j(x) + R_n(x)$$

waarbij a_j van f , maar niet van x afhangen en G_j functies van x onafhankelijk van f zijn.

Voor een lineaire functionaal p hebben we dan:

$$p(f) = \sum_{j=0}^{n-1} a_j p(G_j) + p(R_n).$$

M.a.w. uit een interpolatie-formule voor f wordt een benaderde waarde van $p(f)$ verkregen, door de functionaal p op de basis-functies toe te passen en daarvan de betreffende lineaire combinatie te berekenen. Dit principe gaan we nu toepassen voor het berekenen van afgeleiden en bepaalde integralen.

1. Numeriek differentiëren

In tegenstelling tot analyse is in de numerieke analyse het differentiëren moeilijker dan het integreren. We zullen hier alleen equidistante formules beschouwen. Voor het bereiken van voldoende precisie is enerzijds een klein tabelinterval h nodig. Anderzijds vallen bij kleiner wordende h steeds meer cijfers weg en worden de afrondingsfouten met een factor $\frac{1}{h}$ vermenigvuldigd. In de praktijk tracht men dan ook numerieke differentiatie te vermijden. De formules kunnen echter goed bruikbaar

zijn in iteratieve processen. Bij voorbeeld kan men numerieke differentiatie-formules gebruiken om formules af te leiden voor het oplossen van differentiaal-vergelijkingen. We zullen nu enige formules opstellen. Restterm-beschouwingen laten we achterwege, omdat de fout veel meer wordt bepaald door het wegvallen van cijfers.

1.1. Voorwaartse differentiatie-formules

Gaan we uit van de voorwaartse formule van Newton, dan krijgen we, als $x = x_0 + ph$:

$$\begin{aligned} \frac{df(x)}{dx} &= \frac{1}{h} \frac{df(x_0 + ph)}{dp} = \\ &= \frac{1}{h} \left(\Delta_0 + \frac{2p-1}{dp} \Delta_0^2 + \frac{3p^2-6p+2}{6} \Delta_0^3 + \frac{2p^3-9p^2+11p-3}{12} \Delta_0^4 + \dots \right), \end{aligned}$$

$$\frac{d^2 f(x)}{dx^2} = \frac{1}{h^2} \left(\Delta_0^2 + (p-1) \Delta_0^3 + \frac{6p^2-18p+11}{12} \Delta_0^4 + \dots \right),$$

enz. In het bijzonder voor $p = 0$:

$$f'(x_0) = \frac{1}{h} \left(\Delta_0 - \frac{1}{2} \Delta_0^2 + \frac{1}{3} \Delta_0^3 - \frac{1}{4} \Delta_0^4 + \dots \right),$$

$$f''(x_0) = \frac{1}{h} \left(\Delta_0^2 - \Delta_0^3 + \frac{11}{12} \Delta_0^4 - \dots \right).$$

Hierbij zij opgemerkt, dat de formule voor $f'(x_0)$ overeen komt met de reeksontwikkeling van de operator $\frac{1}{h} \ln(I + \Delta)$, welke gelijk is aan de operator D blijkens de relatie $I + \Delta = E = \exp(hD)$.

Algemeen geldt voor de k -de afgeleide:

$$\frac{d^k f(x)}{dx^k} = \frac{1}{h^k} \left(\Delta_0^k + \dots \right),$$

waarbij ... termen met hogere orde differenties voorstellen.

Dergelijke formules ontstaan ook, als we uitgaan van de achterwaartse Newton-formule. In plaats van Δ 's krijgen we dan D 's en alle - tekens worden + tekens. (Ga dit na).

1.2. Centrale differentiatie-formules

Hier nemen we Stirling's formule (zie p. 80) als uitgangspunt en krijgen (nog steeds met $x = x_0 + ph$):

$$\frac{df(x)}{dx} = \frac{1}{h} (\mu\delta_0 + p\delta_0^2 + \frac{3p^2-1}{6} \mu\delta_0^3 + \frac{2p^3-p}{12} \delta_0^4 + \frac{5p^4-15p^2+4}{120} \mu\delta_0^5 + \dots)$$

$$\frac{d^2f(x)}{dx^2} = \frac{1}{h^2} (\delta_0^2 + p\mu\delta_0^3 + \frac{6p^2-1}{12} \delta_0^4 + \frac{20p^3-30p}{120} \mu\delta_0^5 + \dots).$$

En voor $p = 0$:

$$f'(x_0) = \frac{1}{h} (\mu\delta_0 - \frac{1}{6} \mu\delta_0^3 + \frac{1}{30} \mu\delta_0^5 - \dots)$$

$$f''(x_0) = \frac{1}{h^2} (\delta_0^2 - \frac{1}{12} \delta_0^4 + \frac{1}{90} \delta_0^6 - \dots).$$

Men kan ook van Bessel's formule uitgaan, die we voor deze gelegenheid als volgt schrijven:

$$f(x) = f_0 + p\delta_{\frac{1}{2}} + 2B_2 \mu\delta_{\frac{1}{2}}^2 + B_3 \delta_{\frac{1}{2}}^3 + 2B_4 \mu\delta_{\frac{1}{2}}^4 + \dots$$

Hieruit volgt:

$$\frac{df(x)}{dx} = \frac{1}{h} (\delta_{\frac{1}{2}} + \frac{2p-1}{2} \mu\delta_{\frac{1}{2}}^2 + \frac{6p^2-6p+1}{12} \delta_{\frac{1}{2}}^3 + \frac{4p^3-6p^2-2p+2}{24} \mu\delta_{\frac{1}{2}}^4 + \dots)$$

$$\frac{d^2f(x)}{dx^2} = \frac{1}{h^2} (\mu\delta_{\frac{1}{2}}^2 + \frac{12p-6}{12} \delta_{\frac{1}{2}}^3 + \frac{12p^2-12p-2}{24} \mu\delta_{\frac{1}{2}}^4 + \dots).$$

Deze formules worden prettig voor $p = \frac{1}{2}$, het symmetrie-punt van Bessel's formule:

$$f'(x_0 + \frac{1}{2}h) = \frac{1}{h} (\delta_{\frac{1}{2}} - \frac{1}{24} \delta_{\frac{1}{2}}^3 + \frac{3}{640} \delta_{\frac{1}{2}}^5 - \dots)$$

$$f''(x_0 + \frac{1}{2}h) = \frac{1}{h^2} (\mu\delta_{\frac{1}{2}}^2 - \frac{5}{24} \mu\delta_{\frac{1}{2}}^4 + \dots).$$

Voorbeeld

Nemen we één term van de centrale formule voor $f'(x_0)$, dan krijgen we de volgende ruwe benadering:

$$f'(x_0) = \frac{1}{h} \mu \delta_0 = \frac{f(x_0 + h) - f(x_0 - h)}{2h}.$$

We zullen met deze formule berekenen $f'(x_0)$, waarbij $f(x) = \sin(x)$ en $x_0 = 1$.

Dus $f'(x_0) = \cos(1) \approx .5403023$.

Voor h nemen we steeds kleiner wordende waarden, tot het resultaat niet meer verbetert.

h	$\frac{f(x_0 + h) - f(x_0 - h)}{2h}$	
1	$\frac{.90930 - 0}{2}$	= .45465
.5	$\frac{.99749 - .47943}{1}$	= .51806
.2	$\frac{.93204 - .71736}{.4}$	= .53670
.1	$\frac{.89121 - .78333}{.2}$	= .53940
.05	$\frac{.86742 - .81342}{.1}$	= .54000
.02	$\frac{.85211 - .83050}{.04}$	= .54025
.01	$\frac{.84683 - .83603}{.02}$	= .54000

Er vallen 2 cijfers weg en we krijgen dus niet meer dan 3 cijfers:

$$f'(x_0) \approx .540.$$

Besluit. Men kan ook differentiatie-formules opstellen uitgaande van een Lagrange-formule. Dan ontstaan formules uitgedrukt, niet in differenties, maar in functie-waarden. Het gebruik van deze formules is nog riskanter, omdat men dan geen idee krijgt van de precisie. Verscheidene formules, zowel in Newton- als in Lagrange-vorm staan in [3] p. 61-64.

2. Numerieke integratie of quadratuur

Het numeriek berekenen van bepaalde integralen, ook wel "quadratuur" genoemd, verloopt veel prettiger dan het differentiëren. Afrondingsfouten worden enigszins vereffend, zodat de integraal vaak zelfs iets nauwkeuriger is dan de precisie van de integrand. Numeriek integreren wordt dan ook veel gedaan en verscheidene integratie-processen zijn beschreven in de vorm van Algol-procedures. Voor het afleiden van integratie-formules gaan we weer uit van een interpolatie-formule en krijgen een benaderde waarde van een functionaal \int_a^b door deze op de basis-functies toe te passen en vervolgens de betreffende lineaire combinatie te nemen.

2.1. Newton-Cotes formules (gesloten type)

Deze keer gaan we uit van equidistante Lagrange-formules van de orde n op de basispunten

$$x_i = x_0 + ih, \quad i = 0(1)n - 1.$$

In de gesloten integratie-formules zijn de uiterste basispunten tevens de integratie-grenzen, dus $a = x_0$ en $b = x_{n-1}$. Dan hebben we dus:

$$(2.1.0) \quad \int_{x_0}^{x_{n-1}} f(x) dx = h \int_0^{n-1} f(x_0 + ph) dp =$$

$$h \sum_{j=0}^{n-1} f_j \int_0^{n-1} L_j^n(p) dp + h \int_0^{n-1} R_n(x_0 + ph) dp =$$

$$h \sum_{j=0}^{n-1} f_j NC_j^n + h E_n.$$

Dit is de n -punts Newton-Cotes formule.

De Newton-Cotes coëfficiënten zijn

$$(2.1.1) \quad NC_j^n = \int_0^{n-1} L_j^n(p) dp,$$

die voor elke n eenvoudig kunnen worden berekend (zie lijstje hieronder).

Voor $n = 2$ kan een formule voor de restterm, mits f'' bestaat, gemakkelijk worden verkregen door toepassen van de

Middelwaarde stelling uit de integraal-rekening.

Als f en g continu zijn op het interval $[a, b]$ en g wisselt aldaar niet van teken (dus op het hele interval geldt hetzij $g(x) \geq 0$ hetzij $g(x) \leq 0$), dan is er een punt ξ in het interval zó, dat

$$\int_a^b f(x)g(x) dx = f(\xi) \int_a^b g(x) dx.$$

We hebben namelijk:

$$h E_2 = h \int_0^1 R_n(x_0 + ph) dp = h \int_0^1 \frac{h^2}{2!} p(p-1) f''(\xi_p) dp.$$

Omdat $p(p-1)$ niet van teken wisselt mogen we bovengenoemde stelling toepassen en krijgen dan:

$$h E_2 = \frac{h^3}{2!} f''(\xi) \int_0^1 p(p-1) dp = -\frac{h^3}{12} f''(\xi).$$

Voor grotere waarde van n kunnen we de middelwaarde stelling helaas niet meer toepassen. De formules zijn uiteraard exact voor polynomen van de graad $< n$; voor oneven n echter is de formule, bij verrassing, ook exact voor n -de graads polynomen.

De restterm $h E_n$ blijkt voor voldoende vaak differentieerbare f de volgende vorm te hebben:

$$\begin{aligned} \text{als } n \text{ even is: } h E_n &= C h^{n+1} f^{(n)}(\xi), \\ \text{als } n \text{ oneven is: } h E_n &= C h^{n+2} f^{(n+1)}(\xi), \end{aligned}$$

waarbij ξ in het integratie-interval ligt. De orde van de n -punts Newton-Cotes formule is per definitie de h -exponent in de fout-term E_n van de

formule voor $\frac{1}{h} \int_{x_0}^{x_{n-1}} f(x) dx$.

De n-punts Newton-Cotes formule is dus van de orde n, als n even is en van de orde n+1, als n oneven is. We leiden deze restterm-formules niet af. De betreffende constanten C kunnen worden verkregen door voor f te nemen een polynoom van de graad n resp. n+1. Hier volgen de Newton-Cotes formules voor lage n met bijbehorende restterm.

$$(2.1.2) \quad \int_{x_0}^{x_1} f(x) dx = \frac{h}{2} (f_0 + f_1) - \frac{h^3}{12} f''(\xi) \quad (\text{trapezium-regel})$$

$$(2.1.3) \quad \int_{x_0}^{x_2} f(x) dx = \frac{h}{3} (f_0 + 4f_1 + f_2) - \frac{h^5}{90} f^{(4)}(\xi) \quad (\text{formule van Simpson})$$

$$(2.1.4) \quad \int_{x_0}^{x_3} f(x) dx = \frac{3h}{8} (f_0 + 3f_1 + 3f_2 + f_3) - \frac{3h^5}{80} f^{(4)}(\xi) \quad \left(\frac{3}{8} - \text{regel}\right)$$

$$(2.1.5) \quad \int_{x_0}^{x_4} f(x) dx = \frac{4h}{90} (7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4) - \frac{8h^7}{945} f^{(6)}(\xi).$$

In de 9-punts formule zijn sommige coëfficiënten negatief. Hierdoor gaan cijfers wegvallen, wat verlies aan precisie oplevert. Daarom kan men beter een formule van bescheiden orde kiezen en deze aaneenrijgen. Houden we hierbij het interval constant dan krijgen we, voor n = 2 en 3, de volgende "geregen" formules:

$$\int_{x_0}^{x_m} f(x) dx = h \left(\frac{1}{2} f_0 + f_1 + \dots + f_{m-1} + \frac{1}{2} f_m \right) - \frac{x_m - x_0}{12} h^2 f''(\xi) \quad (\text{trapezium-regel}),$$

$$\int_{x_0}^{x_{2m}} f(x) dx = \frac{h}{3} (f_0 + 4f_1 + 2f_2 + 4f_3 + \dots + 4f_{2m-1} + f_{2m}) -$$

$$\frac{x_{2m} - x_0}{180} h^4 f^{(4)}(\xi) \quad (\text{Simpson}).$$

Opgaven

60) Leidt af formules voor $f'(x)$ en $f''(x)$ uitgaande van de achterwaartse formule van Newton tot en met de ∇^5 -term.

61) Bereken de n -punts Newton-Cotes coëfficiënten (zie formule 2.1.1) voor $n = 4$ en $n = 5$. Bereken de restterm voor een polynoom van de graad 4 resp. 6 en vind hieruit de constante C in de boven vermelde restterm-formules.

62) Schrijf een Algol-procedure, die $\int_a^b f(x) dx$ berekent met Simpson's formule, gebruikend $2m+1$ basispunten. Schrijf hieromheen een programma, dat

$$4 \int_0^1 \sqrt{1-x^2} dx$$

berekent, waarbij achtereenvolgens 3, 5, 9 en 17 basispunten gebruikt worden.

2.2 Richardson-correctie

De gegeven formules met constant interval h hebben een ernstig nadeel. Als de functie niet overal even tam is, wordt toch overal h zo klein gekozen, dat ook het wildste gedeelte nog nauwkeurig genoeg geïntegreerd wordt. Liever willen we h zodanig variëren, dat de fout per staplengte overal ongeveer constant is. Dan wordt h dus alleen klein, waar de functie wild is. Hiertoe zullen we echter een idee van de fout moeten hebben. Meestal is de tweede of vierde afgeleide niet zo eenvoudig te berekenen. Een idee van de fout kan evenwel handig als volgt worden verkregen.

Correctieterm bij trapeziumregel.

We berekenen de integraal over een zeker traject $2h$ met de trapeziumregel op 2 manieren, nl. over twee stappen h en over één stap $2h$, dus

$$I_1 = \frac{h}{2} (f_0 + 2f_1 + f_2) ,$$

$$I_2 = h (f_0 + f_2) .$$

Dan hebben we dus voor de integraal:

$$\int_{x_0}^{x_2} f(x) dx = I_1 + 2Ch^3 = I_2 + 8Ch^3 + O(h^5) ,$$

waarbij $C = -f''(\xi)/12$, voor zekere ξ in het interval (mits f'' continu is).

Dus
$$I_1 - I_2 = 6Ch^3 + O(h^5) .$$

Nemen we nu als correctie-term:

$$(2.2.1) \quad RC_2 = 2Ch^3 = \frac{I_1 - I_2}{3} = -\frac{h}{6} \Delta^2 f_0 ,$$

dan ontstaat de 4^e orde formule

$$(2.2.2) \quad I = I_1 + \frac{I_1 - I_2}{3} + O(h^5) = \frac{h}{3} (f_0 + 4f_1 + f_2) + O(h^5) .$$

Dit is kennelijk Simpson's formule, maar nu met een idee van de precisie, nl. de aangebrachte correctie-term. Deze correctie heet Richardson-correctie. Eigenlijk deed Huygens dit al in 1654 in een artikel "De circuli magnitudinis inventa", waar hij met deze methode 9 decimalen van π vindt. In zijn geval zijn I_1 en I_2 het oppervlak van een ingeschreven regelmatige $2n$ -hoek resp. n -hoek.

Bijvoorbeeld $n = 12$ (cirkel met straal 1).

Oppervlak regelmatige 12-hoek: $I_2 = 3$.

Oppervlak regelmatige 24-hoek: $I_1 = 3.1058$.

Huygens' correctie = $\frac{I_1 - I_2}{3} = .0353$.

Dus $\pi \approx 3.1058 + .0353 = 3.1411$.

Correctieterm bij Simpson

Nu-gaan we op dezelfde manier Simpson's formule beschouwen.

Zij nu

$$I_1 = \frac{h}{3} (f_0 + 4f_1 + 2f_2 + 4f_3 + f_4),$$

$$I_2 = \frac{h}{3} (2f_0 + 8f_2 + 2f_4).$$

Dan geldt:

$$\int_{x_0}^{x_4} f(x) dx = I_1 + 2Ch^5 = I_2 + 32Ch^5 + o(h^7),$$

waarbij nu $C = -f^{(4)}(\xi)/90$.

Dus $I_1 - I_2 = 30Ch^5 + o(h^7)$.

De Richardson-correctieterm is nu

$$(2.2.3) \quad RC_4 = 2Ch^5 = \frac{I_1 - I_2}{15} = -\frac{h}{45} \Delta^4 f_0$$

en we krijgen de 6^e orde formule

$$(2.2.4) \quad I = I_1 + \frac{I_1 - I_2}{15} + o(h^7).$$

Deze formule is equivalent met de 5-punts Newton-Cotes formule (2.1.5). Ga dit na.

Correctieterm bij 2n-de orde formule

Laten I_1 en I_2 twee benaderingen van een integraal over een zeker traject zijn, verkregen met een formule van de orde $2n$, waarbij de afstand tussen de basispunten h is voor I_1 en $2h$ voor I_2 . Dan luidt de Richardson-correctie

$$(2.2.5) \quad RC_{2n} = \frac{I_1 - I_2}{4^n - 1},$$

waarmee de volgende benadering van de orde $2n+2$ wordt verkregen

$$(2.2.6) \quad I = I_1 + \frac{I_1 - I_2}{4^n - 1}.$$

De formule, die op deze wijze uit een Newton-Cotes formule van de orde $2n$ ontstaat, is meestal niet met een Newton-Cotes formule equivalent. Dat lukt alleen bij de trapezium-regel en bij Simpson.

2.3 Integreren met automatisch variërende staplengte

Nu we een schatting van de fout (nl. de Richardson-correctieterm) hebben, kunnen we maatregelen nemen, om deze juist beneden een van te voren gestelde limiet te houden.

Zij gevraagd \int_a^b in een absolute precisie ϵ .

Wij kiezen Simpson's formule met Richardson-correctie en gaan stap voor stap integreren. Stel dat $\int_a^{x_0}$ reeds berekend is en dat we in x_0 een geschat interval h hebben. We berekenen de correctieterm (2.2.3).

Het bij interval h behorende integratie-traject heeft een lengte $4h$. Om de absolute fout $< \epsilon$ te krijgen moet voor dit traject gelden:

$$|RC_4| = \frac{h}{45} |\Delta^4 f_0| \leq \frac{4h\epsilon}{b-a}.$$

$$\text{Dus } |\Delta^4 f_0| < \frac{180 \varepsilon}{b-a} = \Delta_{\max}.$$

Het procédé van Fröberg is nu als volgt. Als $\Delta^4 f_0$ te groot is, halveren we h en integreren opnieuw (vanaf x_0). Is $\Delta^4 f_0$ wel klein genoeg dan accepteren wij de stap en gaan verder integreren, dus vanaf x_0+4h , totdat het eindpunt b bereikt wordt. We moeten evenwel zorgen, dat h weer verdubbeld wordt, als $|\Delta^4 f_0|$ al te klein is. Omdat Δ^4 van de orde h^4 is, moeten wij hierbij een veiligheidsfactor > 16 in acht nemen; 20 schijnt een goede waarde te zijn. Dan luidt dus het criterium voor verdubbelen:

$$|\Delta^4 f_0| < \Delta_{\max} / 20.$$

In Algol 60 kan dit als volgt worden beschreven (zie pag. 108):

De procedure van Schönfeld en Argelo

Wanneer de integraal ingewikkeld is, moeten we zorgen zo min mogelijk integrand-waarden te gebruiken (dus h in elk stadium zo groot mogelijk te houden), en ook geen enkele integrand-waarde weg te gooien (wat in de vorige procedure gebeurt bij verwerping). Dit ideaal wordt goed benaderd in het volgende stukje Algol-poëzie (zie pag. 109).

comment

INT: = integral from a to b of y dx with an absolute tolerance eps.

The method is Simpsons rule with Richardson correction using Froebergs technique of halving and doubling the interval. The total amount of Richardson corrections is smaller than or equal to eps;

```

real procedure INT(x,a,b,y,eps); value a,b,eps; real x,a,b,y,eps;
begin Boolean ls; integer signh;
real bd,h,H,f0,f2,f4,F,T,I,max,min;
  H: = b-a; signh: = sign(H);
  max: = 180 * eps/abs(H); min: = max/20;
  I: = 0; x: = a; f0: = y;
m1: bd: = b; H: = b-a; ls: = true;
m2: h: = H/4;
  x: = a+h; F: = f0 + 4 * y;
  x: = x+h; f2: = y; F: = F + 2 * f2;
  x: = x+h; F: = F + 4 * y;
  x: = bd; f4: = y; F: = F + f4;
  T: = 8 * f2 + 2 * (f0+f4) - F;
  if abs(T) > max then
reject: begin ls: = false; H: = H/2; bd: = a+H end else
accept: begin a: = bd; f0: = f4; I: = (F-T/15) * h + I;
  if ls then go to m3;
  if abs(T) < min then H: = 2 * H;
  bd: = bd + H; if sign(b-bd) ≠ signh then go to m1
  end;
  go to m2;
m3: INT: = I/3
end INT;

```

comment

INT: = integral from a to b of y dx with absolute tolerance eps.

The method is Simpsons rule with Richardson correction. If the fourth difference is too big, the total interval is split in two equal parts and the integration process is invoked recursively. The total amount of Richardson corrections is smaller than or equal to eps;

```

real procedure INT(x,a,b,y,eps);
value a,b,eps; real x,a,b,y,eps;
begin real fa,fb,fb,deltamax,T;
    real procedure I(x0,x4,f0,f2,f4);
    real x0,x4,f0,f2,f4;
    begin real f1,f3,x2;
        x2: = (x0+x4)/2;
        x: = (x0+x2)/2; f1: = y;
        x: = (x2+x4)/2; f3: = y;
        T: = 6 * f2 - 4 * (f1+f3) + f0 + f4;
        I: = if abs(T) < deltamax then
            (x4-x0) * (4 * (f1+f3) + 2 * f2 + f0 + f4 - T/15) else
            I(x0,x2,f0,f1,f2) + I(x2,x4,f2,f3,f4)
    end I;
    deltamax: = 180 * eps/abs(b-a);
    x: = a; fa: = y; x: = b; fb: = y;
    x: = (a+b)/2; fm: = y;
    INT: = I(a,b,fa,fb,fb)/12
end INT;

```

Critiek

Geen van beide procedures zijn waterdicht. Zij nemen als eerste schatting $h = (b-a)/4$. Als nu de integrand per ongeluk in de basispunten nul is, gaat het mis.

Bijvoorbeeld $\int_0^{4\pi} x \sin x \, dx = -4\pi$.

Bovendien nemen de procedures geen maatregelen tegen singulariteiten, of precieser gezegd: tegen gevallen, waar de vierde differentie weigert klein genoeg te worden. De procedures komen dan nooit klaar. Behalve met halveren (en verdubbelen) kan men de staplengte ook iets soepeler regelen. Hierop hopen wij terug te komen bij het oplossen van differentiaalvergelijkingen.

2.4. Romberg integratie

Uitgaande van de trapezium-regel kunnen we de Richardson-correctie herhaald toepassen om formules van steeds hogere orde te krijgen. Zij $I_{p,q}$ de integraal verkregen na p halveringen en q Richardson correcties.

M.a.w. $I_{p,0} = \int_a^b$ volgens trapeziumregel op 2^{p+1} basispunten $(p=0,1,2,\dots)$,

$$I_{p,1} = I_{p,0} + RC_2 = I_{p,0} + \frac{I_{p,0} - I_{p-1,0}}{3}.$$

En algemeen voor $q = 1(1)p$ (vgl. formule 2.2.5)

$$I_{p,q} = I_{p,q-1} + RC_{2q} = I_{p,q-1} + \frac{I_{p,q-1} - I_{p-1,q-1}}{4^q - 1}.$$

Voorbeeld

Nog eens π , nu als $4 \int_0^1 \frac{dx}{1+x^2}$.

$I_{p,0}$	$I_{p,1}$	$I_{p,2}$	$I_{p,3}$	$I_{p,4}$
3				
3.1	3.13333			
3.13118	3.14157	3.14212		
3.13899	3.14159	3.14159	3.14158	
3.14094	3.14159	3.14159	3.14159	3.14159

$$\pi = 3.14159265\dots$$

De Romberg integratie is alleen aantrekkelijk als de integrand over het hele interval vrij tam is. Anders kan men beter met een variërende stapgrootte werken. Misschien kunnen de ideeën van Romberg-integratie en stap-variëring gecombineerd worden.

2.5. Steffenson formules (open type)

Ook hier gaan we uit van een equidistante n -punts Lagrange formule. De basispunten zijn $x_i = x_0 + ih$, waarbij ditmaal $i = 1(1)n$. In de open formules zijn de integratie-grenzen zelf geen basispunten, maar juist een afstand h verwijderd van de uiterste basispunten, m.a.w. $a = x_0$ en $b = x_{n+1}$. Dan hebben we

$$(2.5.1) \quad \int_{x_0}^{x_{n+1}} f(x) dx = h \int_0^{n+1} f(x_0+ph) dp =$$

$$= h \sum_{j=1}^n f_j \int_0^{n+1} L_j^n(p) dp + h \int_0^{n+1} R_n(x_0+ph) dp.$$

Dit is de n -punts Steffenson formule (ook wel open Newton-Cotes formule genoemd).

Voor enige lage waarden van n zien ze er zo uit:

$$(2.5.2) \quad \int_{x_0}^{x_3} f(x) dx = \frac{3h}{2} (f_1 + f_2) + \frac{h^3}{4} f''(\xi) .$$

$$(2.5.3) \quad \int_{x_0}^{x_4} f(x) dx = \frac{4h}{3} (2f_1 - f_2 + 2f_3) + \frac{28}{90} h^5 f^{(4)}(\xi) .$$

Dit soort open type formules worden gebruikt bij het oplossen van differentiaalvergelijkingen.

Opgaven

- 63) a) Zij gegeven een niet-equidistante tabel van een functie f , m.a.w. de argumenten x_i en bijbehorende functiewaarden f_i voor $i = 0(1)N-1$.
Schrijf een programma, dat uit deze gegevens een equidistante tabel maakt voor argument $x = a(h)b$ en deze tabel netjes uittipt.
Gebruik de procedures van opg. 40 (p. 62) en de (zonder declaratie beschikbare) in/uitvoerprocedures read, NLCR en print (zie cursus Dijkstra achterin).
- b) Gebruik dit programma voor het vervaardigen van een tabel van $f(x) = \sqrt[3]{x}$ voor $x = 1(h)10$ waarbij h zo gekozen wordt, dat 4-punts Lagrange-interpolatie in 10 decimalen mogelijk is. Of bedenk een andere leuke functie om dit programma te testen.
- 64) a) Schrijf een integratie-procedure volgens de trapeziumregel met Richardson-correctie, hierbij de staplengte zodanig variërend, dat de correctie per lengte-eenheid nagenoeg constant blijft en de totale correctie kleiner dan een gegeven ϵ is.
- b) Schrijf een programma om deze procedure te testen. Kies hierin een geschikte integrand en integratie-grenzen, laat ϵ variëren en zorg dat bij elke berekende integraal het aantal benodigde integrand-aanroepen wordt geteld en getypt.
- 65) Schrijf een programma, dat een van de in sectie 2.3 vermelde procedures test, zoals in 64b) aangegeven. Of schrijf een programma, dat enige integratie-procedures vergelijkt wat betreft precisie en efficiency.

3. Bepaalde en onbepaalde integratie in Newton-vorm

Voor het verkrijgen van integratie-formules, uitgedrukt in differenties gaan we uit van een equidistante Newton formule, hetzij een voor- of achterwaartse, hetzij een (gesymmetriseerd) centrale, zoals Bessel. Er bestaan verscheidene bruikbare formules van dit type, zowel voor bepaalde als voor onbepaalde integralen. Vele van deze formules staan in Interpolation and Allied Tables [3], p. 66-70. Ter inleiding eerst iets over

3.0 Onbepaalde integralen en andere inverse operatoren

We beschouwen een continue functie f .

3.0.0 Definitie. Een onbepaalde integraal of stamfunctie van een functie f is een functie F met de eigenschap

$$DF = F' = f.$$

Onbepaalde integratie is dus de inverse operatie van differentiëren en we schrijven

$$F = D^{-1}(f).$$

De stamfunctie F is op een additieve constante na bepaald. Immers als C een constante is, hebben we

$$(F + C)' = F' + C' = F' = f,$$

want $C' = 0$. De constante C heet integratie-constante.

De relatie met bepaalde integraal blijkt uit de volgende

3.0.1 Stelling. Als F een stamfunctie is van een continue functie f , geldt:

$$\int_a^b f(x)dx = F(b) - F(a).$$

Door het vormen van dit verschil raken we de integratie-constante kwijt. Vanwege deze stelling schrijven we ook:

$$D^{-1}(f)(x) = F(x) = \int^x f(t)dt \quad (\text{ofwel} = \int f(x)dx)$$

en
$$\int_a^b f(x)dx = \int^b f(x)dx - \int^a f(x)dx.$$

Algemeen hebben we

Definitie. De inverse A^{-1} van een operator A is een operator, die voldoet aan de relatie

3.0.2
$$AA^{-1} = I.$$

M.a.w., als f een functie is:

3.0.3
$$AA^{-1}(f) = f.$$

In het algemeen geldt niet $A^{-1}A = I$ en bij gegeven f hoeft $A^{-1}(f)$ niet uniek bepaald te zijn door bovengenoemde relatie. Voor de onbepaalde integraal-operator D^{-1} en de nu te behandelen inverse differentie-operatoren geldt, dat $A^{-1}(f)$ slechts op een additieve constante na bepaald is.

Beschouwen we nu de inverse voorwaartse differentie Δ^{-1} . Blijkens 3.0.3 moet gelden

3.0.4
$$\Delta\Delta^{-1} f_i = f_i.$$

Dus
$$\Delta^{-1} f_{i+1} - \Delta^{-1} f_i = f_i.$$

Hieruit volgt

3.0.5
$$\Delta^{-1} f_k - \Delta^{-1} f_i = f_i + f_{i+1} + \dots + f_{k-1}.$$

Vanwege deze betrekking heet Δ^{-1} ook voorwaartse somfunctie. Voor één waarde van i mag $\Delta^{-1} f_i$ vrij gekozen worden, alle andere $\Delta^{-1} f_k$ liggen dan vast door 3.0.5.

Evenzo moet voor de inverse achterwaartse differentie ∇^{-1} gelden

$$3.0.6 \quad \nabla \nabla^{-1} f_k = f_k.$$

$$\text{Dus} \quad \nabla^{-1} f_k - \nabla^{-1} f_{k-1} = f_k$$

en

$$3.0.7 \quad \nabla^{-1} f_k - \nabla^{-1} f_i = f_k + f_{k-1} + \dots + f_{i+1}.$$

De operator ∇^{-1} heet dan ook achterwaartse somfunctie.

Tenslotte geldt voor de inverse centrale differentie δ^{-1}

$$3.0.8 \quad \delta \delta^{-1} f_i = f_i.$$

$$\text{Dus ook} \quad \delta \delta^{-1} \mu f_{i+\frac{1}{2}} = \mu f_{i+\frac{1}{2}}.$$

Oftewel (μ en δ^{-1} mogen wel verwisseld worden):

$$\mu \delta^{-1} f_{i+1} - \mu \delta^{-1} f_i = \mu f_{i+\frac{1}{2}}.$$

Sommeren we dit, dan ontstaat

$$3.0.9 \quad \mu \delta^{-1} f_k - \mu \delta^{-1} f_i = \frac{1}{2} f_i + f_{i+1} + \dots + f_{k-1} + \frac{1}{2} f_k = \sum_{j=i}^{k''} f_j,$$

Hierin herkennen we de trapezium-benadering van $\frac{1}{h} \int_{x_i}^{x_k} f$. Hiervoor voeren we bij deze het symbool \int'' in, waarbij het dubbele x_i accentje betekent dat eerste en laatste term van de som gehalveerd zijn.

3.1 Voorwaartse integratie-formules

Integreren we de voorwaartse formule van Newton, dan krijgen we

$$3.1.0 \quad \frac{1}{h} \int_{x_j}^{x_{j+1}} f(x) dx = \int_0^1 f(x_j + ph) dp = \sum_{s=0}^{n-1} \left(\int_0^1 \binom{p}{s} dp \right) \Delta^s f_j + E_n$$

$$= f_j + \frac{1}{2} \Delta_j - \frac{1}{12} \Delta_j^2 + \frac{1}{24} \Delta_j^3 - \frac{19}{720} \Delta_j^4 + \dots \quad (\text{formule van Laplace}).$$

Sommeren we dit, dan krijgen we

$$\begin{aligned}
 3.1.1 \quad \frac{1}{h} \int_{x_i}^{x_k} f(x) dx &= f_i + f_{i+1} + \dots + f_{k-1} + \frac{1}{2} (f_k - f_i) - \frac{1}{12} (\Delta_k - \Delta_i) + \dots \\
 &= \Delta^{-1} f_k - \Delta^{-1} f_i + \frac{1}{2} (f_k - f_i) - \frac{1}{12} (\Delta_k - \Delta_i) + \dots
 \end{aligned}$$

Dit is dus het trapezium-stuk plus correcties uitgedrukt in voorwaartse differenties. Uit deze formule halen we een formule voor de onbepaalde integraal (vgl. stelling 3.0.1):

$$3.1.2 \quad \frac{1}{h} \int_{x_i}^{x_i} f(x) dx = (\Delta^{-1} + \frac{1}{2} I_1 - \frac{1}{12} \Delta + \frac{1}{24} \Delta^2 - \frac{19}{720} \Delta^3 + \dots) f_i.$$

Willen we een formule voor de onbepaalde integraal in een niet-getabelleerd argument, dan integreren we nog even door tot x_p en krijgen:

$$\begin{aligned}
 3.1.3 \quad \frac{1}{h} \int_{x_0}^{x_p} f(x) dx &= \frac{1}{h} \int_{x_0}^{x_0} f(x) dx + \frac{1}{h} \int_{x_0}^{x_p} f(x) dx = \\
 &(\Delta^{-1} + (p + \frac{1}{2})I + (\frac{1}{2} p^2 - \frac{1}{12})\Delta + (\frac{1}{6} p^3 - \frac{1}{4} p^2 + \frac{1}{24})\Delta^2 + \dots) f_0.
 \end{aligned}$$

3.2 Achterwaartse integratie-formules en formule van Gregory

Op precies dezelfde wijze krijgen we uit de achterwaartse Newton-formule:

$$3.2.0 \quad \frac{1}{h} \int_{x_{j-1}}^{x_j} f(x) dx = f_j - \frac{1}{2} \nabla_j - \frac{1}{12} \nabla_j^2 - \frac{1}{24} \nabla_j^3 - \frac{19}{720} \nabla_j^4 - \dots$$

Dus

$$3.2.1 \quad \frac{1}{h} \int_{x_i}^{x_k} f(x) dx = \nabla^{-1} f_k - \nabla^{-1} f_i - \frac{1}{2} (f_k - f_i) - \frac{1}{12} (\nabla_k - \nabla_i) - \dots$$

En we krijgen voor de onbepaalde integraal

$$3.2.2 \quad \frac{1}{h} \int_{x_i}^{x_k} f(x) dx = (v^{-1} - \frac{1}{2} \Delta - \frac{1}{12} \nabla - \frac{1}{24} \nabla^2 - \frac{19}{720} \nabla^3 - \dots) f_k.$$

De formules 3.1.1 en 3.2.1 gebruiken (voor de differenties) functie-waarden, die aan één kant buiten het integratie-traject liggen. Als de functie buiten het traject niet bestaat of zich slecht gedraagt is zo'n formule dus niet bruikbaar. Een formule, die zich beperkt tot functie-waarden binnen het integratie-traject ontstaat door de onbepaalde integralen 3.1.2 en 3.2.2 te combineren tot

$$3.2.3 \quad \frac{1}{h} \int_{x_i}^{x_k} f(x) dx = v_k^{-1} - \Delta_i^{-1} - \frac{1}{2} (f_k + f_i) - \frac{1}{12} (\nabla_k - \Delta_i) - \\ \frac{1}{24} (\nabla_k^2 + \Delta_i^2) - \frac{19}{720} (\nabla_k^3 - \Delta_i^3) - \dots$$

Dit is de formule van Gregory. Eigenlijk moeten we nog nagaan, of v_k^{-1} en Δ_i^{-1} goed op elkaar aansluiten en er geen additieve constante meer overblijft. Dit blijkt in orde te zijn en we krijgen weer het trapezium-stuk plus differentie-correcties:

$$\frac{1}{h} \int_{x_i}^{x_k} f(x) dx = \sum_{j=i}^k f_j - \frac{1}{12} (\nabla_k - \Delta_i) - \text{enz.}$$

Voor $k = i+2$, $i+3$ is deze formule Simpson resp. $\frac{3}{8}$ -regel met differentie-correcties (ga dit na).

3.3 Centrale integratie-formules

Integreren we de formule van Bessel, dan ontstaat

$$3.3.0 \quad \frac{1}{h} \int_{x_0}^{x_1} f(x) dx = \mu f_{\frac{1}{2}} - \frac{1}{12} \mu \delta_{\frac{1}{2}}^2 + \frac{11}{720} \mu \delta_{\frac{1}{2}}^4 - \frac{191}{60480} \mu \delta_{\frac{1}{2}}^6 + \dots$$

De oneven orde termen ontbreken blijkbaar. Uit deze formule volgt:

$$3.3.1 \quad \frac{1}{h} \int_{x_i}^{x_k} f(x) dx = \mu \delta^{-1} f_k - \mu \delta^{-1} f_i - \frac{1}{12} (\mu \delta_k - \mu \delta_i) + \frac{11}{720} (\mu \delta_k^3 - \mu \delta_i^3) - \dots \quad (\text{formule van Gauss}).$$

Het verschil van de twee δ^{-1} -termen is weer het trapezium-stuk (vgl. formule 3.0.9). De onbepaalde integraal luidt nu

$$3.3.2 \quad \frac{1}{h} \int^{x_k} f(x) dx = \mu \delta^{-1} f_k - \frac{1}{12} \mu \delta_k + \frac{11}{720} \mu \delta_k^3 - \frac{191}{60480} \mu \delta_k^5 + \dots$$

Volkomen analoog kunnen we, uitgaande van Stirling's formule, een formule krijgen voor de bepaalde integraal van $x_{-\frac{1}{2}}$ tot $x_{+\frac{1}{2}}$, waaruit we de volgende formule halen voor onbepaalde integraal tot het halfweg-punt $x_{\frac{1}{2}}$:

$$3.3.3 \quad \frac{1}{h} \int^{x_{\frac{1}{2}}} f(x) dx = (\delta^{-1} + \frac{1}{24} \delta - \frac{17}{5760} \delta^3 + \frac{367}{967680} \delta^5 - \dots) f_{\frac{1}{2}}.$$

Integreren we nu nog even door tot x_p , dan ontstaat de volgende formule voor onbepaalde integraal in niet-getabelleerd punt:

$$\frac{1}{h} \int^{x_p} f(x) dx = \frac{1}{h} \int^{x_{\frac{1}{2}}} f(x) dx + \frac{1}{h} \int_{x_{\frac{1}{2}}}^{x_p} f(x) dx = \delta_{\frac{1}{2}}^{-1} + \frac{1}{2}(p - \frac{1}{2})(f_0 + f_1) + A_1 \delta_{\frac{1}{2}} + A_2(\delta_0^2 + \delta_1^2) + A_3 \delta_{\frac{1}{2}}^3 + A_4(\delta_0^4 + \delta_1^4) + \dots$$

Deze formule en tabellen van de coëfficiënten A_i zijn te vinden in [3], p. 68-69.

Besluit

Analoog aan de formules van Gregory en Gauss bestaat er ook een formule, die een bepaalde integraal benadert door een trapezium-stuk plus correcties uitgedrukt in de afgeleiden van de integraal in de uiteinden van het integratie-traject. Deze formule (Euler-Maclaurin) zullen we later behandelen. We hebben ons hier beperkt tot formules op equidistante basispunten. Later zullen we formules bespreken op niet-equidistante punten, waarin de punten in zekere zin optimaal worden gekozen (Gauss quadratuur).

Opgaven

66a) Stel een formule op voor een onbepaalde integraal in x_p uitgedrukt in achterwaartse differenties met index 0. Werk hierbij de coëfficiënten tot en met die van v^3 volledig uit. Leid vervolgens hieruit af een formule voor $\int_{x_0}^{x_1}$.

b) Werk de coëfficiënten A_i van formule 3.3.4 uit voor $i = 1(1)4$.

67) Schrijf een Algol-programma, dat van een gegeven equidistant getabelde functie f het differentie-schema tot en met orde 4, alsmede de eerste en tweede somfunctie berekent en uittipt.

Als functie f kiest U enige geschikte functies, zoals $\frac{1}{x}$, $\ln x$, $\cos x$, en een passend interval.

Of U laat het programma de functie-waarden van een band inlezen. In dit geval maakt U ook een paar "getallenbanden" voor dit programma op.

Voor in- en uitvoer gebruikt U de procedures, vermeld achterin de cursus van Dijkstra.

68) Schrijf een Algol-programma, dat met behulp van de formule van Gregory berekent

$$\int_{-1}^{+1} (1-x)^\alpha (1+x)^\beta dx$$

voor $\alpha, \beta = .5(1)5$.

Doe de berekening voor enige geschikte waarden van h en neem telkens de differenties mee, tot ze verwaarloosbaar worden in zeg 6 decimalen.

69) Schrijf een Algol-programma, dat met behulp van de formule van Gauss (3.3.1) berekent

$$\int_0^x (t^2 + 1)^m dt$$

voor $m = -1, -\frac{1}{2}, \frac{1}{2}, \frac{3}{2}$ en voor $x = 0(.5)5(1)10$.

Doe de berekening voor enige geschikte waarden van h en neem telkens de differenties mee, tot ze verwaarloosbaar zijn in 6 decimalen.

Hoofdstuk 4. Lineaire vergelijkingen, Determinanten, Matrix inversie

0. Literatuur

Behalve het op pag. 21 genoemde [1], volgen hier nog enige boeken, die uitvoerig de lineaire algebra en de matrix-rekening behandelen; 22-24 behandelen theorie, de andere boeken ook numerieke methoden.

- 1 National Physical Laboratories, Modern Computing Methods (London 1962).
- 22 A.C. Aitken, Determinants and matrices.
- 23 A.S. Householder, The theory of matrices in numerical analysis (1964).
- 24 R. Zurmühl, Matrizen.
- 25 C. Lanczos, Applied Analysis (1957).
- 26 E. Stiefel, Einführung in die numerische Mathematik (1961).
- 27 V.N. Faddeeva, Computational methods of linear algebra (1959).
- 28 D.K. Faddeev & V.N. Faddeeva, Computational methods of linear algebra (1963).

1. Inleiding

Bij de interpolatie-theorie is al even de theorie van de lineaire vergelijkingen ter sprake gekomen. Nu gaan we numerieke oplossingsmethoden bespreken. Een stelsel van n lineaire vergelijkingen met n onbekenden, die we x_1, \dots, x_n noemen, ziet er in het algemeen zo uit:

$$1.0 \quad \begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases}$$

De coëfficiënten a_{ij} ($i, j = 1, \dots, n$) vormen een matrix, die we A noemen en ook wel noteren als (a_{ij}) .

Deze matrix heet de matrix (ook wel kleine matrix) van het stelsel.

De rechterleden b_i vormen een vector, die we b noemen. (Matrix A aangevuld met vector b tot een n bij $n+1$ matrix heet ook wel grote matrix.) De onbekenden x_i vormen een vector, die x heet. De vectoren b en x worden opgevat (en geschreven) als kolom-vectoren. Dientengevolge kan het stelsel (1.0) compact worden geschreven in de vorm

$$1.1 \quad Ax = b.$$

Dit wil zeggen: gegeven een n bij n matrix A en een n -vector b , gevraagd een n -vector x zó, dat matrix A maal kolom-vector x de kolom-vector b oplevert. Het getal n heet de orde van het stelsel en ook de orde van matrix A .

In het eenvoudigste geval, orde $n = 1$, heeft het stelsel de gedaante $a_{11}x_1 = b_1$. Mits $a_{11} \neq 0$, luidt de oplossing: $x_1 = b_1/a_{11}$. Gezien de definitie van deling is dit alleen maar een andere schrijfwijze. Deze notatie zullen we ook gebruiken voor een stelsel van de orde n :

$$1.2 \quad x = b/A.$$

Dit wil zeggen, x is een (de?) oplossingsvector van het stelsel (1.1). Zoals we gezien hebben, kan een lineair stelsel worden opgelost door eliminatie, gevolgd door terug-substitutie. We gaan nu dit procédé wat gedetailleerder bekijken.

2. Gauss' eliminatie

2.1 Eliminatie zonder verwisselingen

De onbekenden worden in de natuurlijke volgorde geëlimineerd en voor het elimineren van x_k wordt de k -de vergelijking gebruikt. De eerste eliminatiestap verloopt dus als volgt.

Er wordt verondersteld, dat $a_{11} \neq 0$. De onbekende x_1 wordt uit alle vergelijkingen, behalve de eerste, geëlimineerd, d.w.z., men berekent voor $i = 2(1)n$ de factoren $m_{i1} = -a_{i1}/a_{11}$, vermenigvuldigt de eerste vergelijking met m_{i1} , en telt dit bij de i -de vergelijking op. Voorzien we de gewijzigde elementen van een boven-index 1 dan komt het resultaat er zo uit te zien

$$\begin{array}{ccccccc}
 & a_{11} & a_{12} & \dots & \dots & \dots & a_{1n} & b_1 \\
 & 0 & a_{22}^{(1)} & \dots & \dots & \dots & a_{2n}^{(1)} & b_2^{(1)} \\
 2.1.0 & \text{-----} & & & & & & \\
 & 0 & a_{n2}^{(1)} & \dots & \dots & \dots & a_{nn}^{(1)} & b_n^{(1)}
 \end{array}$$

Algemener: in de k -de eliminatie-stap, $k = 1(1)n-1$, wordt x_k met behulp van de k -de vergelijking geëlimineerd uit alle volgende vergelijkingen. (De in deze stap gewijzigde elementen zullen we aanduiden met boven-index k .) Hierbij wordt verondersteld, dat $a_{kk}^{(k-1)} \neq 0$. Men berekent, voor $i = k+1(1)n$, de factoren $m_{ik} = -a_{ik}^{(k-1)} / a_{kk}^{(k-1)}$, vermenigvuldigt hiermee de k -de vergelijking en telt het resultaat bij de i -de vergelijking op. Zo doorgaande krijgt het stelsel tenslotte de driehoeksvorm, die er zo uit ziet

$$\begin{array}{ccccccc}
 & a_{11} & a_{12} & a_{13} & \dots & \dots & \dots & a_{1n} & b_1 \\
 & 0 & a_{22}^{(1)} & a_{23}^{(1)} & \dots & \dots & \dots & a_{2n}^{(1)} & b_2^{(1)} \\
 2.1.1 & 0 & 0 & a_{33}^{(2)} & \dots & \dots & \dots & a_{3n}^{(2)} & b_3^{(2)} \\
 & | & & & & & & & | \\
 & | & & 0 & & & & & | \\
 & | & & & & & & & | \\
 & | & & & & & & & | \\
 & 0 & 0 & \dots & \dots & \dots & 0 & a_{nn}^{(n-1)} & b_n^{(n-1)}
 \end{array}$$

Dit stelsel is mathematisch equivalent met het oorspronkelijke stelsel. De determinant is tijdens het elimineren niet veranderd en dus hebben we (vgl. de definitie van determinant op pag. 26):

$$2.2.2 \quad \det(A) = a_{11} a_{22}^{(1)} \dots \dots \dots a_{nn}^{(n-1)} .$$

Als alle diagonaal-elementen van de driehoeksmatrix ongelijk aan 0 zijn, is dus ook $\det(A)$ ongelijk aan 0. Het stelsel heeft dan precies één oplossing, die wordt verkregen door terugsubstitueren:

$$x_n = b_n^{(n-1)} / a_{nn}^{(n-1)}$$

$$x_{n-1} = (b_{n-1}^{(n-2)} - a_{n-1,n}^{(n-2)} x_n) / a_{n-1,n-1}^{(n-2)}$$

$$x_1 = (b_1 - a_{1n} x_n - \dots - a_{12} x_2) / a_{11}$$

2.2 Som-controle

Bij het handrekenen moet elke stap gecontroleerd worden. Hiertoe vullen we het stelsel aan met een extra kolom-vector s , welks elementen zijn

$$s_i = \sum_{j=1}^n a_{ij} + b_i$$

Deze elementen transformeren we in elke eliminatie-stap op dezelfde wijze als de b_i . Na elke stap moet de som in elke rij kloppen, natuurlijk op afrondingsfouten na.

Het resultaat van de terugsubstitutie kan worden gecontroleerd, door na te gaan of deze het oorspronkelijke stelsel bevredigen. Men kan ook alle oorspronkelijke vergelijkingen optellen en hierin de verkregen oplossingsvector x invullen. Dus als

$$t_j = \sum_{i=1}^n a_{ij} \quad , \quad c = \sum_{i=1}^n b_i$$

dan moet gelden (op afrondingsfouten na).

$$\sum_{j=1}^n t_j x_j = c$$

2.3 Rij-verwisselingen

Bij het elimineren volgens boven geschetst schema kunnen we geluk hebben en nooit een 0 op de hoofddiagonaal aantreffen. Zo niet, dan zullen we enige vergelijkingen moeten verwisselen. Rekenen we exact

en is de determinant niet 0, dan bereiken we op deze wijze de verlangde driehoeksvorm en de gezochte oplossing. De praktijk vergt echter meestal, bij het elimineren afrondingsfouten te maken. We moeten dan zorgen, de invloed van deze fouten klein te houden. Hiertoe is het nodig niet alleen nullen, maar ook betrekkelijk kleine elementen op de hoofddiagonaal te vermijden. Beschouwen we nu de k -de eliminatie-stap, waar dus x_k geëlimineerd zal worden. We kiezen nu uit de mogelijke kandidaten (dus de elementen $a_{ik}^{(k-1)}$, $i = k(1)n$) een grootste in absolute waarde. Laat $a_{pk}^{(k-1)}$ zo'n maximaal element zijn. Als $p \neq k$, verwisselen we nu de k -de en de p -de vergelijking en gaan daarna op de in (2.1) geschetste wijze x_k elimineren. Het gekozen element $a_{pk}^{(k-1)}$ heet ook wel pivot en het verwisselen wordt ook wel pivoten genoemd. Bij verwisseling keert de determinant van teken om. Hebben we de driehoeksvorm verkregen met eliminatie met verwisselingen dan geldt dus

$$\det(A) = (-1)^{\uparrow \text{aantal verwisselingen}} \times \text{product der pivots.}$$

Omdat $a_{pk}^{(k-1)}$ maximaal gekozen is, geldt voor de factoren

$$|m_{ik}| = |a_{ik}^{(k-1)} / a_{pk}^{(k-1)}| \leq 1.$$

Deze relatie is prettig indien we (met de hand) met vaste komma rekenen. Rekenen we met drijvende komma, dan helpt deze relatie ons niet. Evenwel, ook in een drijvende komma-systeem, waar de getallen worden voorgesteld met een vaste relatieve precisie, is het van belang de pivots maximaal te kiezen. Dit moge blijken uit volgend voorbeeld (uit [1] pag. 51).

m_{i1}		A		b
	.000003	.213472	.332147	.235262
-71837.3	.215512	.375623	.476625	.127653
-57752.3	.173257	.663257	.625675	.285321

$$a_{22}^{(1)} = a_{22} + m_{21} a_{12} = .375623 - 15335.25 \approx -15334.9$$

$$a_{23}^{(1)} = a_{23} + m_{21} a_{13} = .476625 - 23860.6 \approx -23860.1$$

$$b_2^{(1)} = b_2 + m_{21} b_1 = .127653 - 16900.6 \approx -16900.5.$$

Doordat m_{21} maal de eerste vergelijking veel grotere coëfficiënten heeft dan de tweede vergelijking, wordt deze bijna helemaal uitgeschoven. Van de coëfficiënten der tweede vergelijking komt maar één cijfer terecht in de afgeronde resultaten. Ditzelfde gebeurt met de derde vergelijking. De gewijzigde matrix $A^{(1)}$ en het gewijzigde rechterlid $b^{(1)}$ zien er dan zo uit:

	$A^{(1)}$		$b^{(1)}$
.000003	.213472	.332147	.235262
0	-15334.9	-23860.1	-16900.5
0	-12327.8	-19181.6	-13586.6

Gaan we hiermee verder, dan vinden we een oplossing, die de tweede en derde vergelijking van het oorspronkelijke stelsel slechts in één decimaal nauwkeurig bevredigt. Met verwisseling (van 1e en 2e vergelijking) krijgen we echter wel een goed resultaat.

2.4 Voorbeeld

Hier volgt een volledig voorbeeld van Gauss' eliminatie met rijverwisselingen en som-controle uit [1] pag. 6. De pivots zijn onderstreept. De gecontroleerde getallen zijn met \checkmark aangeduid. De berekeningen worden gedaan in één decimaal extra.

m		A			b	s	
	<u>.4096</u>	.1234	.3678	.2943	.4043	1.5994	
-.54834	.2246	.3872	.4015	.1129	.1550	1.2812	
-.88989	.3645	.1920	.3728	.0643	.4240	1.4176	
-.43555	.1784	.4002	.2786	.3927	-.2557	.9942	
-.92230		.31953	.19982	-.04848	-.06669	.40418(9)	✓
-.23723		.08219	.04550	-.19759	.06422	-.00568(9)	✓
		<u>.34645</u>	.11840	.26452	-.43179	.29758	✓
			<u>.09062</u>	-.29245	.33155	.12972	✓
-.19212			.01741	-.26034	.16665	-.07628(7)	✓
				<u>-.20415</u>	.10295	-.10120	✓

Terugsubstitueren levert:

x_1	x_2	x_3	x_4
-0.00593	-1.55547	2.03123	-0.50429

Check met de som der oorspronkelijke vergelijkingen:

$$1.1771 x_1 + 1.1028 x_2 + 1.4207 x_3 + .8642 x_4 = .7276 (.72761)$$

In 6 decimalen nauwkeurig luidt de oplossing:

$$-.006124 \quad -1.555598 \quad 2.031468 \quad -.504263.$$

De precisie is dus iets lager, dan we, met onze extra decimaal, zouden verwachten. Het is dus raadzaam een of twee extra decimalen mee te nemen.

Opgaven

- 70) Schrijf een Algol-programma, dat een tabel uittijpt (vgl. Barlow's Tables) van n , n^2 , n^3 , \sqrt{n} , $\sqrt{10n}$, $\sqrt[3]{n}$, $1/n$, voor $n = n_0(1+n_0) + 50$. Zorg dat om de 5 regels een regel blank wordt ingelast.
- 71) Schrijf een Algol-programma, dat een tabel maakt van \sin , \tan , \cot , en \cos met argument θ in graden voor $\theta = \theta_0(1 - \text{minuut})\theta_0 + 30$ minuten. Las om de 10 regels een regel blank in.
- 72) Schrijf een Algol-programma, dat een tabel maakt van de (equidistante) Lagrange-coëfficiënten van de orde n voor interpolatiefunctie $p = p_0 + (p_1 - p_0)x$. Let hierbij op de afronding en zorg dat de som gelijk aan 1 blijft. Las om de 5 regels een regel blank in.
- 73) Er is een procedure voor het oplossen van een lineair stelsel van de orde n , met matrix in array $A[1 : n, 1 : n]$ en rechterlid in array $b[1 : n]$. De heading van de procedure is als volgt:
"real procedure DETSOL (A, b, n); value n; integer n; array A, b;"
- De waarde van de functie procedure is $\det(A)$, de oplossingsvector x van het stelsel " $\sum_j A[i, j] \times x[j] = b[i]$ " worden over vector b geschreven.
- Schrijf een programma, dat DETSOL toetst voor enige matrices, door de oplossing in het oorspronkelijke systeem in te vullen en de afwijkingen in de rechterleden (de residuen) uit te typen.

2.5 Gauss-Jordan eliminatie

Dit is een variant op het eliminatie-schema volgens Gauss. Hier wordt niet naar driehoeksvorm, maar naar diagonaalvorm getransformeerd. M.a.w. in de k -de eliminatie-stap wordt (na eventuele rijverwisseling als boven) x_k geëlimineerd uit de i -de vergelijkingen niet alleen voor $i = k+1(1)n$, maar ook voor $i = 1(1)k-1$. Na n stappen heeft men dan een equivalent stelsel

$$2.5.1 \quad A^{(n)} x = b^{(n)}$$

waarbij $A^{(n)}$ een diagonaal-matrix is. De oplossing is dan eenvoudig

$$2.5.2 \quad x_i = b_i^{(n)} / a_{ii}^{(n)}, \quad i = 1(1)n.$$

Vergeleken met Gauss' eliminatie kan deze variant het voordeel hebben, dat het iets compacter geprogrammeerd kan worden. Immers, in plaats van de terugsubstitutie hebben we hier de veel eenvoudiger formule (2.5.2). Voor handrekenen heeft Gauss-Jordan het nadeel, dat veel meer tussenresultaten moeten worden opgeschreven, dan bij Gauss' eliminatie. We gaan nu een methode beschouwen, waarin juist aanzienlijk minder tussenresultaten moeten worden genoteerd.

3. Ontbinding in driehoeksvorm (methode van Crout)

Deze methode is mathematisch equivalent met Gauss' eliminatie en kan, als het spel goed gespeeld wordt, er zelfs numeriek mee equivalent zijn. Ter inleiding gaan we eerst Gauss' eliminatie nader beschouwen en wel

3.1 Gauss' eliminatie, geschreven als matrix-vermenigvuldiging

We zien voorlopig af van de rij-verwisselingen. In de eerste eliminatiestap wordt m_{i1} maal de eerste rij van A opgeteld bij de i -de rij van A . Dit kan worden geschreven als voor-vermenigvuldigen van A met de matrix

$$M_1 = \begin{pmatrix} 1 & & & & 0 \\ m_{21} & 1 & & & \\ m_{31} & 0 & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{n1} & 0 & \dots & 0 & 1 \end{pmatrix}$$

Voor de nieuwe matrix $A^{(1)}$ geldt dus $A^{(1)} = M_1 A$.
Evenzo kan de 2e stap worden geschreven als $A^{(2)} = M_2 A^{(1)}$, waarbij

$$M_2 = \begin{pmatrix} 1 & & & & 0 \\ 0 & 1 & & & \\ \vdots & m_{32} & 1 & & \\ \vdots & \vdots & 0 & \ddots & \vdots \\ 0 & m_{n2} & 0 & \dots & 1 \end{pmatrix}$$

Algemeen $A^{(k)} = M_k A^{(k-1)}$, voor $k = 1(1)n-1$, waarbij

$$M_k = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & m_{k+1,k} & 1 & \\ & & \vdots & \vdots & \ddots \\ & & m_{n,k} & 0 & \dots & 1 \end{pmatrix}$$

Na $n-1$ stappen ontstaat de boven-driehoeksmatrix $A^{(n-1)}$, die we ook U noemen (naar "upper triangle"). Hiervoor geldt dus:

$$3.1.0 \quad U = A^{(n-1)} = M_{n-1} \dots M_2 M_1 A = MA,$$

als $M = M_{n-1} \dots M_2 M_1$. Evenals de matrices M_k is ook matrix M een onder-driehoek met uitsluitend enen op de hoofddiagonaal ("unit lower triangle"). Dit volgt uit de

3.1.1 Stelling. Het product van 2 onderdriehoeksmatrices is weer een onder-driehoek. Hebben beide matrices uitsluitend enen op de hoofddiagonaal, dan geldt dit ook voor de product-matrix.

De stelling volgt direct uit de definitie van matrix-vermenigvuldiging (ga dit na). Iets dergelijks geldt natuurlijk ook voor bovendriehoeken.

We hebben dus nu de Gauss' eliminatie zonder verwisselingen geschreven als een matrix-vermenigvuldiging:

$$3.1.2 \quad MA = U,$$

waarbij A de matrix van het stelsel is en M een zodanige onder-driehoek met uitsluitend enen op de hoofddiagonaal, dat de product-matrix U een bovendriehoek is.

Het resulterende equivalente stelsel is dus

$$MAx = Mb$$

$$\text{ofwel} \quad Ux = b^{(n-1)},$$

welk stelsel wordt opgelost door terugsubstitutie.

3.2 Methode van Crout

We vermenigvuldigen nu in de bovengenoemde relatie (3.1.2) beide leden met M^{-1} , de inverse van M, en krijgen dan

$$3.2.0 \quad M^{-1}MA = A = M^{-1}U.$$

M^{-1} is weer een onderdriehoek met enen op de hoofddiagonaal blijkens de volgende

3.2.1 Stelling. De inverse van een onder-driehoeksmatrix zonder nullen op de hoofddiagonaal (want anders bestaat de inverse niet) is weer een onder-driehoek. Is een matrix een onder-driehoek met uitsluitend enen op de hoofddiagonaal, dan geldt dit ook voor haar inverse.

De stelling volgt gemakkelijk uit de definitie van inverse matrix (ga dit na).

De onderdriehoek M^{-1} noemen we ook L (naar "lower triangle") en we schrijven dus $A = LU$.

In de methode van Crout nu wordt deze splitsing rechtstreeks berekend, stap voor stap de definitie van matrix-vermenigvuldiging gebruikend.

In de k -de stap, $k = 1(1)n$, wordt u_{kj} , $j \geq k$ berekend uit de relatie

$$3.2.2 \quad a_{kj} = l_{k1}u_{1j} + \dots + l_{k,k-1}u_{k-1,j} + l_{kk}u_{kj}$$

en l_{ik} , $i \geq k$, uit de relatie

$$3.2.3 \quad a_{ik} = l_{i1}u_{1k} + \dots + l_{i,k-1}u_{k-1,k} + l_{ik}u_{kk}$$

Aannemende $l_{kk} = 1$ krijgen we dan, voor $k = 1(1)n$,

$$3.2.4 \quad \begin{cases} u_{kj} = a_{kj} - l_{k1}u_{1j} - \dots - l_{k,k-1}u_{k-1,j}, & j \geq k, \\ l_{ik} = (a_{ik} - l_{i1}u_{1k} - \dots - l_{i,k-1}u_{k-1,k})/u_{kk}, & i > k \end{cases}$$

De elementen van L en U , die rechts van de gelijkttekens staan, zijn al berekend. Zo krijgen we na n stappen de LU-ontbinding van A . Rekenend met een tafel-rekenmachine hoeven we als tussenresultaten slechts op te schrijven de elementen van L en U (in totaal dus n^2 getallen), wat aanzienlijk voordeliger is, dan bij Gauss' eliminatie. Rekenend met een automatische rekenmachine, is Crout's methode te verkiezen boven eliminatie, indien scalaire producten (in dit geval rijen van L met kolommen van U) worden berekend in extra precisie.

Determinant. Na de ontbinding $A = LU$ vinden we ook hier de determinant van A op eenvoudige wijze, dank zij de

Product-stelling voor determinanten. Zijn P en Q twee n bij n matrices, dan geldt:

$$\det (PQ) = \det (P) \times \det (Q).$$

Dus in ons geval

$$\det (A) = \det (L) \times \det (U) = \det (U) = u_{11} \dots u_{nn}$$

want $\det (L) = 1$.

Heen- en terug-substitutie

Na de splitsing $A = LU$ moeten we oplossen het stelsel

$$LUx = b.$$

Dit gaat in twee etappes, nl. eerst wordt berekend $y = b/L$, daarna $x = y/U$. De berekening van y noem ik heen-substitutie (forward substitution) en de daarop volgende berekening van x is de reeds bij Gauss' eliminatie genoemde terug-substitutie (back substitution).

3.3 Voorbeeld

We lossen het stelsel, gegeven in 2.4, nu op met Crout's methode (uit [1], pag. 12). We houden weer een som-kolom bij, die we als extra rechterlid-vector mee laten doen.

De LU-ontbinding van dit stelsel luidt:

LU					b/L	s/L
	.40960	.12340	.36780	.29430	.40430	1.59940
1		.31953	.19982	-.04848	-.06669	.40418
.54834	1		-.00590	-.18513	.08137	-.10966
.88989	.25721	1		3.40003	-1.71453	1.68550
.43555	1.08426	16.65290	1			

Uit $y = b/L$ vinden we dan tenslotte door terugsubstitutie $x = y/U$:

$$x_1 = -.00609, \quad x_2 = -1.55560, \quad x_3 = 2.03144, \quad x_4 = -.50427.$$

Berekenen we uit $z = s/L$ op dezelfde wijze $w = z/U$, dan moet (op afrondingsfouten na) gelden $w_i = 1 + x_i$.

$$\text{Immers } (Aw)_i = \sum_j a_{ij}(1 + x_i) = \sum_j a_{ij} + b_i = s_i.$$

Omdat hier geen rijen zijn verwisseld, komt er een iets kleinere pivot (nl. u_{33}), dan in het Gauss' schema van dit voorbeeld. Hier zijn echter minder tussenresultaten afgerond, zodat toch het antwoord nauwkeuriger is (zie pag. 127).

Opgaven

74) Bewijs de stellingen 3.1.1 en 3.2.1.

Schrijf de inverse matrix van een onderdriehoeksmatrix P van de orde 4 expliciet uit, waarbij dus

$$P = \begin{pmatrix} p_{11} & 0 & 0 & 0 \\ p_{21} & p_{22} & 0 & 0 \\ p_{31} & p_{32} & p_{33} & 0 \\ p_{41} & p_{42} & p_{43} & p_{44} \end{pmatrix}$$

waarbij de hoofddiagonaal-elementen van P niet 0 zijn.

75) Bewijs, dat voor $k = 1(1)n-1$ geldt (zie sectie 3.1):

$$M_k^{-1} = \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & \ddots \\ 0 & & & & -m_{k+1,k} \\ & & & & \vdots \\ & & & & -m_{nk} \\ & & & & & \ddots \\ & & & & & & 0 \\ & & & & & & & \ddots \\ & & & & & & & & 1 \end{pmatrix}.$$

Bewijs bovendien (ter illustratie, hoe nauw Crout met Gauss' eliminatie samenhangt):

$$M^{-1} = M_1^{-1} M_2^{-1} \dots M_{n-1}^{-1} = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & -m_{21} & & \\ & & \vdots & & \\ & & & & -m_{n,n-1} \\ & & & & & \ddots \\ & & & & & & 1 \end{pmatrix}.$$

76) Bewijs dat verwisseling van de p -de rij en de k -de rij van een matrix A kan worden geschreven als voorvermenigvuldiging van A met een permutatie-matrix $P^{(pk)}$, die in 4 elementen van de eenheidsmatrix verschilt, nl. $P_{kk}^{(pk)} = P_{pk}^{(pk)} = 0$ en $P_{pk}^{(pk)} = P_{kp}^{(pk)} = 1$. Wat is de determinant van $P^{(pk)}$?

3.4 Varianten van Crout

We hebben hierboven gekozen $l_{ii} = 1$, omdat dit het nauwste aansluit bij het Gauss-schema. Men kan in plaats hiervan evengoed kiezen $u_{ii} = 1$ of een voorwaarde, die zorgt dat $|l_{ii}|$ en $|u_{ii}|$ (ongeveer) even groot zijn. Een andere mogelijkheid is de matrix te schrijven als LDU, waarbij D diagonaal is en $l_{ii} = u_{ii} = 1$.

3.5 Crout met rij-vertalingen

Evenals bij Gauss'eliminatie moet ook de splitsing in driehoeken worden uitgevoerd met vertalingen, teneinde kleine elementen op de hoofddiagonaal te vermijden. We moeten zodanig vertalen, dat in iedere kolom van L het grootste element op de hoofddiagonaal terecht komt. Het is hierbij iets handiger de variant $u_{ii} = 1$ te kiezen. Uit (3.2.2) en (3.2.3) komen dan de volgende formules voor de k-de stap, $k = 1(1)n$:

$$3.5.1 L \quad l_{ik} = a_{ik} - l_{i1}u_{1k} - \dots - l_{i,k-1}u_{k-1,k}, \quad i \geq k.$$

Zoek hieruit het element l_{pk} met maximale absolute waarde en vertal de rijen p en k. Bereken daarna (de indices duiden op de positie na de vertaling):

$$3.5.1 U \quad u_{kj} = (a_{kj} - l_{k1}u_{1j} - \dots - l_{k,k-1}u_{k-1,j})/l_{kk}, \quad j > k.$$

Voor een beschrijving van dit proces in Algol, zie de procedure DET hieronder. De bijbehorende heen- en terug-substitutie wordt gedaan in SOL en DETSOL combineert beide. De procedures zijn getest en geregeld in gebruik in het X1-Algol systeem van het Mathematisch Centrum. Voor de berekening der scalaire producten werd en wordt hierbij gebruikt een machine-code procedure INPROD, die voor enige decimalen extra precisie zorgt.

In plaats van deze machine-code procedure zou men de volgende declaratie kunnen gebruiken.

```

comment INPROD := the sum over k from a through b of x * y;
real procedure INPROD (k, a, b, x, y);
value a, b; integer k, a, b; real x, y;
begin real r; r := 0;
    for k := a step 1 until b do r := x * y + r;
    INPROD := r
end INPROD;

```

comment AP 204

DET := determinant of the n-th order matrix given in array A[1 : n, 1 : n]. The first index of the array elements is the row index, the second one the column index. The method is triangular decomposition according to Crout with row interchanges. This process yields a lower triangle L and a unit triangle U such that L * U equals matrix A with interchanged rows. Together with each row interchange the sign of the non-pivotal row is reversed, so that the determinant equals the product of the diagonal elements of L. At each stage the pivot is chosen in a column of L such that its modulus divided by the Euclidean norm of the corresponding row of A is maximal. The integer array p[1 : n] is an output vector in which the pivotal row indices are recorded. The elements of A are replaced by the corresponding calculated elements of L and U. So enough information is retained for subsequent solution of linear systems and for matrix inversion. DET uses the non-local real procedure INPROD;

```

real procedure DET (A, n, p);
value n; integer n; array A; integer array p;
begin integer i, j, k; real d, r, s; array v[1 : n];
  for i := 1 step 1 until n do
    v[i] := sqrt (INPROD (j, 1, n, A[i,j], A[i,j]));
    d := 1;
    for k := 1 step 1 until n do
      begin r := -1;
        for i := k step 1 until n do
          begin A[i,k] :=
            A[i,k] - INPROD (j, 1, k - 1, A[i,j], A[j,k]);
            s := abs (A[i,k]) / v[i];
            if s > r then begin r := s; p[k] := i end
          end LOWER;
        v[p[k]] := v[k];
        for j := 1 step 1 until n do
          begin r := A[k,j];
            A[k,j] := if j ≤ k then A[p[k],j] else
              (A[p[k],j] - INPROD (i, 1, k - 1, A[k,i], A[i,j]))
              / A[k,k]; if p[k] ≠ k then A[p[k],j] := -r
          end UPPER;
        d := A[k,k] × d
      end LU;
    DET := d
  end DET;

```

comment AP 205

SOL replaces the vector given in array $b[1 : n]$, by the solution vector x of the linear system $L \times U \times x = b$ with interchanged (and possibly sign-reversed) elements. Here L is the lower triangle and U the unit upper triangle which are given in array $LU[1 : n, 1 : n]$ such that the elements with first index $<$ second index are elements of U and the other elements of LU are elements of L . The integer array $p[1 : n]$ defines the interchanges with sign reversions of the elements of b in correspondence with the pivoting administration in DET. Hence a call $DET(A, n, p)$, followed by a call $SOL(A, b, n, p)$, has the effect that b is replaced by the solution vector x of the linear system: $\text{Sigma}(A[i,j] \times x[j]) = b[i]$. The procedure SOL leaves the elements of LU and p unaltered and uses the non-local real procedure INPROD;

```
procedure SOL (LU, b, n, p);
value n; integer n; array LU, b; integer array p;
begin integer i, k; real r;
    for k:= 1 step 1 until n do
        begin r:= b[k]; b[k]:=
            (b[p[k]] - INPROD (i, 1, k - 1, LU[k,i], b[i])) / LU[k,k];
            if p[k]  $\neq$  k then b[p[k]]:= -r
        end;
    for k:= n step - 1 until 1 do
        b[k]:= b[k] - INPROD (i, k + 1, n, LU[k,i], b[i])
    end SOL;
```

comment AP 207

DETSOL:= determinant of the n-th order matrix given in array
 A[1 : n, 1 : n]. Moreover the vector, given in array b[1 : n],
 is replaced by the solution vector x of the linear system:
 $\text{Sigma } (A[i,j] \times x[j]) = b[i]$. For further details see DET and SOL,
 which are used by DETSOL;

```
real procedure DETSOL (A, b, n); value n; integer n; array A, b;
begin integer array p[1 : n];
      DETSOL:= DET (A, n, p); SOL (A, b, n, p)
end DETSOL;
```

4. Symmetrische matrices

4.0 Inleiding

Er zijn methodes speciaal voor symmetrische matrices, waarin gedurende het proces de symmetrie (met de daaraan verbonden voordelen) wordt gehandhaafd. De LU-ontbinding leent zich uitstekend voor symmetrisering. Er is daarbij echter geen plaats voor rij-verwisselingen (tenzij we ook de overeenkomstige kolom-verwisselingen zouden doen, wat niet genoeg helpt). Helaas zijn er symmetrische matrices, waarvoor verwisseling nodig is, b.v.:

$$\begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 3 \\ 2 & 3 & 0 \end{pmatrix}.$$

Dergelijke matrices laten zich dus niet op symmetrische wijze als een LU-product schrijven. Er is echter een belangrijke klasse van symmetrische matrices, waarvoor het wonderwel gaat, nl. de positief definitie matrices. Daarom ter inleiding de volgende definities en stellingen.

Zij A een reële symmetrische matrix van de orde n en x een willekeurige n -vector, die opgevat wordt als kolom-vector en $\underline{0}$ de n -vector, waarvan alle elementen 0 zijn.

4.1.0. Definitie. Matrix A heet positief semidefinit als voor elke x geldt $x^T A x \geq 0$.

4.1.1. Definitie. Matrix A heet positief definit als voor elke $x \neq \underline{0}$ geldt $x^T A x > 0$.

In de praktijk is het vaak lastig vast te stellen of een matrix deze prettige eigenschap(en) bezit.

Een goed bruikbare voorwaarde biedt de volgende

4.1.2. Stelling. Als A geschreven kan worden als het product van een reële (rechthoekige) matrix F met zijn getransponeerde, in formule: $A = F^T F$, dan is A positief semidefinit. Zijn de kolommen van F lineair onafhankelijk, dan is A zelfs positief definit.

Bewijs. Zij $A = F^T F$. Dan $x^T A x = x^T F^T F x = (F x)^T F x \geq 0$, want $(F x)^T F x$ is een som van quadraten. Dus A is positief semidefinit. Laten nu bovendien de kolommen van F lineair onafhankelijk zijn. Dan volgt uit $x \neq 0$ dat ook $F x \neq 0$, dus $(F x)^T F x > 0$, m.a.w. A is dan positief definit.

Ook het omgekeerde geldt, waarbij we zelfs kunnen eisen, dat F een boven-driehoeksmatrix is:

4.1.3. Stelling. Een positief definitie matrix A kan worden geschreven als een product $U^T U$, waarbij U een reële boven-driehoeksmatrix is. (Deze stelling bewijzen we niet.)

4.1.4. Stelling. Als A positief definit is, dan ook al zijn submatrices, die ontstaan door een aantal rijen en de overeenkomstige kolommen van A te schrappen.

Dit volgt uit de definitie (4.1.1).

4.1.5. Stelling. Als A positief definit is, dan geldt $\det(A) > 0$. Dit volgt uit stelling (4.1.3).

4.2. De wortel-methode (Cholesky) voor symmetrische positief-definitie matrices

Zij A een symmetrische positief definitie matrix. We schrijven $A = LU$ en handhaven de symmetrie door te eisen $L = U^T$, dus $A = U^T U$. Deze ontbinding is mogelijk volgens stelling 4.1.3. Dan hebben we dus voor $k = 1(1)n$ (vgl. 3.2.2 en 3.2.3):

$$4.2.0 \begin{cases} u_{kk} = \sqrt{a_{kk} - u_{1k}^2 - \dots - u_{k-1,k}^2} \\ u_{kj} = (a_{kj} - u_{1k}u_{1j} - \dots - u_{k-1,k}u_{k-1,j})/u_{kk}, j > k. \end{cases}$$

In elk stadium zijn de elementen van U , die rechts van het = teken staan, reeds bekend. Omdat A positief definit is, is het getal onder het wortel-teken altijd (althans bij exact rekenen, zie echter onderstaande waarschuwing) positief en U blijft dus reëel. Bovendien blijken verwisselingen in dit geval overbodig te zijn.

Weet men niet van te voren of A positief definitief is, dan kan men dit proces voortzetten, zolang de worteltrekking geen argument ≤ 0 krijgt. Blijft het tot en met $k = n$ goed gaan, dan is blijkbaar de $U^T U$ -ontbinding gelukt en de matrix, volgens stelling 4.1.2, positief definitief. Ontmoeten we een argument ≤ 0 voor de worteltrekking, dan kunnen we meestal beter overgaan op Gauss' eliminatie of Crout met verwisselingen.

Waarschuwing. Het kan heel goed voorkomen, dat een matrix positief definitief is, maar door afrondingsfouten toch aanleiding geeft tot een niet-positief argument voor de worteltrekking. Men moet dan wel rekenen op precisieverlies, zie sectie 7, slechte conditie.

5. Meerdere rechterleden

Beschouwen we nu een stelsel met een vaste matrix A en meerdere rechterleden (aantal m). De rechterleden vormen een matrix B met n rijen en m kolommen; de oplossing is een met B gelijkvormige matrix X . Het stelsel schrijven we dan als

$$5.0.1 \quad AX = B$$

en de oplossing als

$$X = B/A.$$

Zowel in het Gauss' schema als in de heen-substitutie van Crout's methode kunnen alle rechterleden tegelijk meegenomen worden. Als somkolom nemen we nu de som van alle kolommen van A en B samen.

Het kan nodig zijn de rechterleden afzonderlijk na elkaar te behandelen. Dit is b.v. het geval, als het volgende rechterlid afhangt van de vorige oplossings-vector, wat nogal eens voorkomt in een iteratieproces. In dat geval doet men een keer hetzij de Gauss' eliminatie, daarbij de driehoek $A^{(n-1)} = U$ en alle factoren m_{ik} bewarend, of de LU-decompositie, hierbij L en U bewarend.

Bovendien noteert men in beide gevallen welke verwisselingen zijn uitgevoerd. Met deze gegevens kan men daarna voor elk nieuw rechterlid

met weinig moeite (d.w.z. met betrekkelijk weinig bewerkingen, dus in kortere tijd) de oplossing berekenen. Na Gauss' eliminatie van de matrix moeten namelijk de overeenkomstige eliminatie-stappen in het rechterlid gedaan worden, gevolgd door terugsubstitutie; na LU-ontbinding heeft men voor elk rechterlid de heen- en terug-substitutie te doen.

6. Matrix-inversie.

Een belangrijk speciaal geval van de matrix-vergelijking 5.0.1 is het geval $B = I$ (eenheidsmatrix), dus

$$6.0.1 \quad AX = I.$$

De oplossing $X = I/A$ heet inverse van A en wordt genoteerd als A^{-1} . Mathematisch geldt: als $AX = I$, dan ook $XA = I$. Numeriek hoeft dit echter niet te gelden.

De matrix X wordt berekend door het stelsel (6.0.1) op te lossen met een of andere numerieke methode, zoals Gauss' eliminatie, Crout, en als A positief definit is, Cholesky.

In een machine procedure kan de zaak zo gearrangeerd worden, dat A^{-1} over A heen geschreven wordt, bijna zonder extra werkruimte te gebruiken. (Naast matrix A en een n -vector voor het noteren van de verwisselingen blijkt er nog slechts één extra n -vector als werkruimte nodig te zijn.)

Opmerking. $A^{-1}b$ betekent iets anders dan b/A . Het eerste betekent, dat eerst A^{-1} wordt berekend en daarna $A^{-1}b$ wordt verkregen door matrix maal vector vermenigvuldiging. Het tweede betekent, dat het stelsel $Ax = b$ wordt opgelost, zonder A^{-1} te berekenen. Aangezien het eerste ongeveer tweemaal zoveel werk vergt en ook meestal minder nauwkeurig is, verdient de tweede aanpak altijd de voorkeur, tenzij men A^{-1} toch ook nodig heeft.

7. Slechte conditie

Delen door 0 mag niet. Aangezien delen het oplossen van een lineair stelsel van de orde 1 is, valt te verwachten dat het oplossen van stelsels van de orde $n > 1$ ook moeilijkheden kan opleveren.

We weten reeds, dat een stelsel $Ax = b$ geen eenduidige oplossing heeft (maar oneindig veel of geen enkele oplossing), als $\det(A) = 0$.

Numerieke moeilijkheden ontstaan echter ook, als $\det(A)$ dicht bij 0 ligt, d.w.z. aanzienlijk kleiner uitvalt, dan men op grond van de grootte der elementen van A zou verwachten. Men spreekt dan van slechte conditie en de matrix zowel als het stelsel heten slecht geconditioneerd. Bijvoorbeeld het stelsel

$$\begin{pmatrix} 1 & -1 \\ -101 & 102 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

heeft als oplossing $x^T = (1, 1)$.

Variëren we dit tot $y^T = (1.101, 1.100)$, dan blijkt, dat de residu-vector $r = b - Ay$ heel kleine elementen heeft, nl. $r^T = (-.001, -.001)$.

Het residu is dus een factor 100 kleiner, dan het verschil tussen y en de exacte oplossing. Dit komt, doordat de determinant nogal klein is (vergelijk $\det(A) = 1$, $a_{11} a_{22} = 102$).

Later zullen we het begrip "slechte conditie" nauwkeuriger definiëren en onderzoeken. Hier vermelden we het slechts als een waarschuwing.

Een numerieke oplossing van een stelsel moet men kritisch bezien.

Bovendien mag men niet uit het klein zijn van het residu zonder meer besluiten, dat de oplossings-vector nauwkeurig genoeg is.

Opgaven

- 77) Bewijs stelling (4.1.4) uit de definitie (4.1.1) door geschikte vectoren x te kiezen.
- 78) Bewijs dat een reële symmetrische matrix van de orde 2 dan en slechts dan positief definitief is, als beide hoofddiagonaal-elementen en de determinant positief zijn.
- 79) Welke van deze matrices zijn positief definitief, welke zijn wel positief semidefinitief, maar niet positief definitief?

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 2 \\ 3 & 2 & 1 \end{pmatrix}, \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{pmatrix}.$$

- 80) Schrijf een Algol-procedure, die een driehoeksmatrix van de orde n invertteert en daaromheen een programma, dat deze procedure gebruikt voor het inverteren van de bovendriehoeksmatrices A en B , gedefiniëerd door: $A_{ij} = 1$ voor $i \leq j$, $B_{ij} = j - i + 1$ voor $i \leq j$.

8. Iteratieve heen- en terug-substitutie

Als het stelsel slecht geconditioneerd is, is het residu vaak veel kleiner, dan het verschil tussen berekende en exacte oplossing. Men kan dan een betere benadering vinden als volgt.

Zij het stelsel $Ax = b$. Splits eerst de matrix A in driehoeken, dus $A = LU$.

Bereken vervolgens door heen- en terug-substitutie

$$x^{(0)} = b/A.$$

Bereken daarna (bij voorkeur in extra precisie rekenend) het residu

$$r^{(1)} = b - Ax^{(0)}.$$

Bereken dan door heen- en terug-substitutie

$$d^{(1)} = r^{(1)}/A.$$

Deze oplossing brengen we als correctie aan op $x^{(0)}$, dus

$$x^{(1)} = x^{(0)} + d^{(1)}.$$

Meestal is $x^{(1)}$ veel beter dan $x^{(0)}$. Als $x^{(1)}$ nog niet goed genoeg is herhalen we het procedé met $x^{(1)}$ in plaats van $x^{(0)}$, enz.

Deze iteratieve heen- en terug-substitutie is dus een toepassing van de in sectie 5 geschetste verwerking van meerdere rechterleden, die niet allen tegelijk bekend zijn.

9. Iteratieve oplossings-methoden

Hier zullen we een paar iteratieve methoden bespreken ter oplossing van een stelsel $Ax = b$. Ze convergeren goed, als de hoofddiagonaal van A sterk overheerst, d.w.z. dat de elementen op de hoofddiagonaal in absolute waarde veel groter zijn dan alle andere elementen van A . Een voordeel van deze methoden is, dat zij profijt kunnen trekken van elementen die 0 zijn, ook indien het 0-patroon onregelmatig is.

Dit voordeel wordt beduidend bij grote matrices met veel nullen erin, die natuurlijk alleen buiten de hoofddiagonaal mogen staan.

De iteratie begint met een geschikte vector $x^{(0)}$ als beginschatting.

In elke iteratie-stap wordt uit vector $x^{(i)}$ een nieuwe benadering $x^{(i+1)}$ berekend, waarbij $x^{(i)}$ de i -de iterate (= iteratie-resultaat) heet. Er zijn verschillende formules voor de iteratie-stap, waarvan we hier enige bespreken.

9.1. Jacobi iteratie

Hier luidt de iteratie-stap als volgt:

$$9.1.0 \quad x_k^{(i+1)} = (b_k - \sum_{\substack{j=1 \\ j \neq k}}^n a_{kj} x_j^{(i)}) / a_{kk}, \quad k := 1(1)n.$$

M.a.w. voor het berekenen van x_k brengen we in de k -de vergelijking alle termen, behalve de k -de, naar rechts en bepalen hieruit een nieuwe waarde voor x_k , waarbij we voor x_j ($j \neq k$) de oude waarden gebruiken.

Voorbeeld. Het stelsel

$$\begin{pmatrix} 4 & 1 & -2 \\ 2 & 7 & -1 \\ -3 & 1 & -8 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 8 \\ 17 \\ 7 \end{pmatrix}$$

heeft als oplossing $x^T = (1, 2, -1)$.

Wij schrijven nu

$$x_1 = (8 - x_2 + 2x_3)/4$$

$$x_2 = (17 - 2x_1 + x_3)/7$$

$$x_3 = -(7 + 3x_1 - x_2)/8.$$

Iteratie startend met $x^{(0)} = \underline{0}$ levert:

$$\begin{array}{l}
 x_1 \quad 2.0 \quad .95 \quad .91 \quad .9975 \quad 1.0093 \quad 1.0006 \quad .9991 \quad .9999 \quad 1.0001 \quad 1.0000 \\
 x_2 \quad 2.4 \quad 1.73 \quad 1.97 \quad 2.0229 \quad 2.0050 \quad 1.9979 \quad 1.9994 \quad 2.0002 \quad 2.0001 \quad 2.0000 \\
 x_3 \quad -.9 \quad -1.32 \quad -1.02 \quad -.9700 \quad -.9962 \quad -1.0029 \quad -1.0005 \quad -.9997 \quad -.9999 \quad -1.0000
 \end{array}$$

Schrijven we matrix A als $A = L + D + U$, waarbij D een diagonaal-matrix is, L een onderdriehoek en U een bovendriehoek, beide met louter nullen op de hoofddiagonaal, dan kunnen we formule (9.1.0) schrijven in de gedaante

$$9.1.1 \quad x^{(i+1)} = (b - (L + U)x^{(i)})/D,$$

waarbij / weer betekent oplossen van een lineair stelsel, ditmaal met de diagonaal-matrix D als matrix, dus eenvoudig door te delen door $d_{kk} = a_{kk}$.

9.2. Gauss-Seidel iteratie

Hier berekenen we in elke stap de nieuwe x_k -waarden ook in de volgorde $k = 1(1)n$, maar nu gebruiken we steeds de nieuwste x_j -waarden voor $j \neq k$. M.a.w. de iteratie-stap luidt

$$9.2.0 \quad x_k^{(i+1)} = (b_k - \sum_{j=1}^{k-1} a_{kj} x_j^{(i+1)} - \sum_{j=k+1}^n a_{kj} x_j^{(i)})/a_{kk}, \quad k = 1(1)n.$$

Met de bovengenoemde schrijfwijze voor A kunnen we dit in de volgende vorm schrijven:

$$9.2.1 \quad x^{(i+1)} = (b - Lx^{(i+1)} - Ux^{(i)})/D.$$

Omdat hier steeds de meest recente waarden gebruikt worden, zal deze methode meestal iets sneller convergeren.

Voorbeeld. Hetzelfde stelsel als in (9.1), wederom startend met de nul-vector, levert nu:

$$\begin{array}{l}
 x_1 \quad 2.0000 \quad .8393 \quad 1.0320 \quad .9941 \quad 1.0011 \quad .9994 \quad 1.0000 \\
 x_2 \quad 1.8571 \quad 1.9898 \quad 1.9993 \quad 2.0000 \quad 2.0020 \quad 2.0002 \quad 2.0000 \\
 x_3 \quad -1.3929 \quad -.9410 \quad -1.0121 \quad -.9978 \quad -1.0002 \quad -.9998 \quad -1.0000
 \end{array}$$

9.3. Relaxatie

De beide voorgaande methoden werken de vergelijkingen af in een vaste volgorde. Bij relaxatie pakken we de vergelijking met in absolute waarde grootste residu en maken dit residu 0 door die onbekende te variëren, die in deze vergelijking de grootste coëfficiënt heeft.

In formule:

Bereken voor $k = 1(1)n$ de residuen

$$r_i = b_i - \sum_{k=1}^n a_{ik} x_k.$$

Bepaal i , waarvoor $|r_i|$ maximaal is.

Bepaal daarna k , waarvoor $|a_{ik}|$ maximaal is en bereken vervolgens voor deze k een nieuwe x_k uit de i -de vergelijking, door hierin voor de andere x_j de nieuwste waarden te substitueren:

$$x_k := (b_i - \sum_{\substack{j=1 \\ j \neq k}}^n a_{ij} x_j) / a_{ik}.$$

Als de hoofddiagonaal overheerst, is natuurlijk altijd $k = i$.

In dit schema kan het voorkomen, dat sommige x_k maar zelden gevariëerd worden.

Voorbeeld. Nemen we weer het stelsel van (9.1) met de nul-start (hier is dus steeds $k = i$) dan luiden de iteraties en de bijbehorende residuen:

x_1	0	0	1.4	1.4	1.4	.98	.98	.980	.980	1.001	1.0010	1.0010
x_2	0	2.4	2.4	2.4	1.87	1.87	1.99	1.990	2.007	2.007	2.0006	2.0006
x_3	0	0	0	-1.1	-1.1	-1.1	-1.1	-.994	-.994	-.994	-.994	-1.0003
r_1	8	5.6	0	-2.2	-1.67	.01	-.11	.102	.085	.001	.0074	-.0052
r_2	17	.2	-2.6	-3.7	.01	.85	.01	.116	-.003	-.045	-.0002	-.0065
r_3	7	4.6	8.8	0	.53	-.73	-.85	-.002	-.019	.044	.0504	0

Opgaven

81) Bepaal met de eliminatie-methode van Gauss de oplossing van het volgende stelsel

$$\begin{pmatrix} 2.85714 & 4.09524 & 7.57143 & 10.45238 \\ 3.52941 & 9.76471 & 3.70588 & 2.35294 \\ 1.53846 & 2.36923 & 12.90769 & 5.55385 \\ 9.88769 & 4.90237 & 2.00376 & 1.00000 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 116.90476 \\ 106.88235 \\ 107.95385 \\ 77.99846 \end{pmatrix}$$

82) Bepaal met de eliminatie-methode van Gauss-Jordan de inverse van de matrix

$$\begin{pmatrix} .921653 & .058760 & .383541 \\ .058760 & .955930 & -.287656 \\ -.383541 & .287656 & .877583 \end{pmatrix}$$

83) Bepaal met de methode van Crout de inverse van de matrix

$$\begin{pmatrix} 8 & 6 & 4 & 2 \\ 6 & 12 & 8 & 4 \\ 4 & 8 & 12 & 6 \\ 2 & 4 & 6 & 8 \end{pmatrix}$$

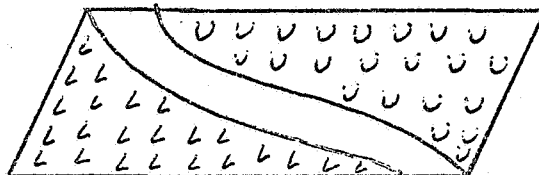
De gegeven matrix-elementen zijn exact.

84) Bepaal in 4 decimalen nauwkeurig de oplossing van het stelsel

$$\begin{aligned} 10x - 3y &= 24.8125 \\ -x + 6y - 2z &= -5.0959 \\ -4y + 8z - t &= 24.5219 \\ -2z + 9t &= 9.7089 \end{aligned}$$

met de iteratieve methoden van

a) Jacobi, b) Gauss-Seidel.



Fracticum proefwerk

1) Bereken het Rayleigh-quotiënt $(v^T A u) / (v^T u)$, waarbij

$$v = \begin{pmatrix} +2 \\ -4 \\ +6 \\ +8 \end{pmatrix}, \quad A = \begin{pmatrix} -13.873 & -4 & +6 & +8 \\ +6 & -27.873 & +18 & +24 \\ -10 & +20 & -45.873 & -40 \\ +14 & -28 & +42 & +40.127 \end{pmatrix}$$

$$u = \begin{pmatrix} +.1091089 \\ +.3273268 \\ -.5455447 \\ +.7637626 \end{pmatrix}$$

2) Zij gegeven de volgende tabel van $f(x) = x^3$:

x	f(x)
20	8000
21	9261
22	10648
23	12167
24	13824
25	15625
26	17576
27	19683

Bereken hieruit door inverse interpolatie een benaderde waarde van $\sqrt[3]{13500}$ in 4 cijfers nauwkeurig.

Aanwijzing: voeg de basispunten toe in een geschikte volgorde, maak gedeelde differenties (niet meer dan nodig blijkt) en gebruik Newton's formule term voor term, totdat precisie bereikt is.

Z.O.Z.

- 3) Zij gegeven de volgende tabel van $\sin(\pi x/180)$ met gemodificeerde tweede differenties, die in 5 decimalen interpoleerbaar is d.m.v. de gemodificeerde vierde orde Everett-formule.

x	$\sin(\pi x/180)$	δ_m^2
0	.00000	0
15	.25882	-1786
30	.50000	-3450
45	.70711	-4880
60	.86603	-5976
75	.96593	-6666
90	1.00000	-6899

Bereken met behulp van deze tabel $y = \tan(\pi/18)$, d.i. $\tan(10^\circ)$ en vervolgens:

$$z = 1 - 1.1499196 y + .6774323 y^2 - .2080030 y^3 + .1268089 y^4.$$

Practicum proefwerk, groep II

- 1) In C. Hastings, Approximations for digital computers staat de volgende benadering voor de error-functie

$$\phi(X) = \frac{2}{\sqrt{\pi}} \int_0^X e^{-t^2} dt.$$

Als $p = \frac{1}{1 + p^X}$, dan luidt de benadering

$$\phi^{**}(X) = 1 - (a_1 p + a_2 p^2 + a_3 p^3 + a_4 p^4 + a_5 p^5) \phi'(X),$$

waarbij $p = .3275911$,

$$a_1 = +.2258\ 36846$$

$$a_2 = - .2521\ 28668$$

$$a_3 = +1.2596\ 95130$$

$$a_4 = -1.2878\ 22453$$

$$a_5 = + .9406\ 46070$$

en ϕ' is de afgeleide van ϕ .

a) Bereken uit deze benaderings-formule $\frac{\sqrt{\pi}}{2}$, door te kijken wat er voor $X = 0$ uit moet komen.

b) Bereken $\phi^{**}(x)$ voor $X = .1$ en $X = 1$, als gegeven is:

$$\exp(-.01) \approx .99004\ 98337$$

$$\exp(-1) \approx .36787\ 94412.$$

- 2) Bereken uit de volgende (exacte) tabel $f(160000)$ met Aitken's interpolatie-formule.

x	f(x) = x ↑ (2/3)
125000	2500
132651	2601
140608	2704
148877	2809
157464	2916
166375	3025
175616	3136
185193	3249
195112	3364
205379	3481
216000	3600

Vereiste relatieve precisie in 5 cijfers.

Z.O.Z.

2) De volgende tabel van de Bessel-functie $Y_2(x)$ is in de tabel-precisie interpoleerbaar met 6-punts Lagrange.

x	$Y_2(x)$
6.0	.22985 790
6.1	.20392 273
6.2	.17660 555
6.3	.14815 715
6.4	.11883 613
6.5	.08890 666
6.6	.05863 613
6.7	+.02829 284
6.8	-.00185 639
6.9	-.03154 852
7.0	-.06052 661

Bovendien zijn gegeven de volgende 6-punts Lagrange-coëfficiënten:

P	L_{-2}	L_{-1}	L_0	
.32	.01077 69446	-.09470 64832	.78132 84864	.68
.33	.01092 66459	-.09571 08458	.77148 74242	.67
.34	.01106 46105	-.09660 89124	.76150 55448	.66
.35	.01119 08672	-.09740 19922	.75138 67969	.65

	L_3	L_2	L_1	p
p	L_1	L_2	L_3	
.32	.36768 39936	-.07441 22368	.00932 92954	.68
.33	.37998 63433	-.07622 48054	.00953 52378	.67
.34	.39229 07352	-.07798 55076	.00973 35295	.66
.35	.40459 28906	-.07969 25391	.00992 39766	.65

	L_0	L_{-1}	L_{-2}	p
--	-------	----------	----------	---

Gevraagd te berekenen $Y_2\left(6\frac{2}{3}\right)$.

1) Zij gegeven het volgende Algol-programma:

```
"begin comment Testprogramma voor PROD;  
    integer n; real result;  
    real procedure PROD(k, a, b, fk);  
    value a, b; integer k, a, b; real fk;  
    begin real p; p:= 1;  
        for k: = a step 1 until b do p: = fk * p;  
        PROD: = p  
    end PROD;  
    n: = 0;  
    for n: = n + 1 while result < 106 do  
        begin result := PROD(i, 1, n-1, 10 * i);  
        print (result)  
    end  
end"
```

Hierin is "print (result)" een statement, die zorgt voor het uittypen van de waarde van "result".

a) Welke getallen typt dit programma uit?

b) Wat is de betekenis van "value a, b"?

Waarom is het aan te bevelen, hier b in de value lijst te plaatsen?

Waarom kunnen k en fk in dit geval niet in de value-lijst worden opgenomen?

c) Wat is het verschil tussen specificaties en declaraties? Noem de in dit programma voorkomende specificaties. Welke van deze mogen worden weggelaten?

- 2) In C. Hastings, Approximations for digital computers (p. 173) staat een benadering voor de functie

$$E(k) = \int_0^{\pi/2} \sqrt{1 - k^2 \sin^2 \phi} \, d\phi$$

geldig voor $0 \leq k < 1$. Als $\eta = 1 - k^2$, dan luidt de benadering:

$$E^*(k) = \{1 + a_1 \eta + a_2 \eta^2\} + \{b_1 \eta + b_2 \eta^2\} \ln \frac{1}{\eta},$$

waarbij: $a_1 = .4630151$ $b_1 = .2452727$
 $a_2 = .1077812$ $b_2 = .0412496$.

- a) Schrijf een procedure declaratie, waarin $E^*(k)$ wordt berekend.
 b) Schrijf een blok, dat deze procedure gebruikend de waarden $E^*(k)$ berekent voor $k = \sin\left(\frac{\pi\theta}{180}\right)$, waarbij $\theta = 0(10)90$.
 Laat de resultaten afleveren in een passend array.
- 3) Hoe luidt de formule van Lagrange van de orde 3
- a) voor het algemene geval?
 b) voor het geval, dat de basispunten equidisant zijn?
 c) Als een functie f gegeven is voor de argumenten $10.0(0.2)20.0$ en $f^*(x)$ wordt gevraagd voor $x = 10.54321$, welke basispunten kiest U dan voor interpolatie van de orde 3 en wat is dan de waarde van p ?
- 4) Bereken voor het equidisante geval en voor interpoleerfractie $p = \frac{1}{2}$ de Lagrange-coëfficiënten van de orde 3 en 4.
 Schrijf de hierbij horende "middenpunt"-formules voor $f^*(x_0 + \frac{1}{2}h)$ uit.

- 5) a) Hoe luidt de interpolatie-formule van Newton van de orde n ?
- b) Hoe luidt de formule voor de bijbehorende restterm, uitgedrukt in de n -de afgeleide van de functie f ?
In welk interval moet $f^{(n)}$ bestaan, opdat deze formule geldig zij? Noem de stelling uit de analyse, die voor de afleiding van deze formule nodig is.

Cursus Wetenschappelijk Rekenaar A

Proefwerk Algol practicum (open boek). 15-6-'65

1. Op een in de bandlezer liggende band bevinden zich achtereenvolgens de getallen 3, 1, 3, 4, 2, 1, 5, 3, 1, 2.

Wat wordt door het volgende programma geponst?

```
begin  real procedure inprod (m,p,ap,bp);  
      value m; integer m, p; real ap, bp; .  
      begin real s; s := 0;  
          for p:= 1 step 1 until m do  
              s:= s + ap × bp;  
          inprod:= s  
      end inprod;  
      integer n; n:= read;  
      begin array A[1:n,1:n], b[1:n];  
          integer i, j;  
          for i:= 1 step 1 until n do  
              for j:= i step 1 until n do A[i,j]:= read;  
              for j:= 1 step 1 until n do b[j]:= read;  
              for i:= 1 step 1 until n do  
                  punch (inprod(n,j,b[j], if j < i then A[j,i] else A[i,j]))  
      end  
end
```

end

(Uit examen WRA, 1964).

2. Schrijf een ALGOL programma, dat van de functie

$$f(x) = (1 + x) \sqrt{1 - x^2}$$

een volledig differentie-schema tot en met orde 4, alsmede de eerste som-functie uittypt voor argument $x = -1(.05)+1$.

Bovendien moet het programma uit deze gegevens met behulp van de formule van Gregory ${}_{-1}^{+1} \int f$ berekenen en uittypen.

Proefwerk Algol practicum (vervolg)

3. Een tabel f_1, \dots, f_n stelt functie-waarden voor behorende bij equidistante abscissen x_1, \dots, x_n .

De tabel beslaat de plaatsen 1 t/m n van een array.

In datzelfde array op de plaatsen -3, -2, -1, 0 staan achtereenvolgens de waarden van n , x_1 , x_n en $x_2 - x_1$.

Gevraagd te schrijven een declaratie voor een functie-procedure, die als actuele parameters meekrijgt een argument x , de array-identificer en een label, die de plaats aangeeft waar het programma hervat moet worden als x buiten het segment $[x_1, x_n]$ ligt. De procedure-identificer moet als waarde krijgen de functie-waarde, die door lineaire interpolatie correspondeert met x .

Schrijf hieromheen een programma dat een array van 1 t/m 101 vult met de waarden e^x , waarbij $x = 0(.02)20$ en ook op de plaatsen -3 t/m 0 van het array passende waarden invult.

Vervolgens moet het programma de functie-procedure aanroepen voor $x = -1(0.03)23$ en voor deze waarden van x de geïnterpoleerde functie-waarden typen, of, als x buiten het interval ligt, een alarm-indicatie (een geschikt getal of een tekst) uittypen en daarna gewoon doorgaan met de volgende waarde van x .

Cursus Wetenschappelijk Rekenaar A

Proefwerk Numerieke Analyse (gesloten boek) 15-6-'65

1. a) Geef een definitie van functionaal en van lineaire functionaal.
Licht de definities toe met een voorbeeld.
 - b) Gegeven een operator A , hoe wordt de inverse operator A^{-1} gedefini-
eerd? Wordt door deze definitie A^{-1} ondubbelzinnig vastgelegd?
Licht Uw antwoorden toe met een voorbeeld.
 - c) Hoe verkrijgt men formules voor numeriek differentiëren?
Schrijf een formule neer voor $f'(x_0)$ van de orde 4. Waarom is
numeriek differentiëren niet erg aanlokkelijk?
2. Leidt de integratie-formule van Gregory af in de volgende etappes:
 - a) Vind uit de voorwaartse formule van Newton een formule voor
 $\int_{x_j}^{x_{j+1}} f$. Bereken hierbij de coëfficiënten expliciet t/m Δ^4 .
Leid hieruit af een formule voor $\int_{x_0}^{x_k} f$ en een formule voor de
onbepaalde integraal $\int_{x_k} f$.
 - b) Schrijf de overeenkomstige formule op van de onbepaalde integraal
 $\int_{x_k}^{x_i} f$ uitgedrukt in achterwaartse differenties. (Hier (in b) hoeft
men geen volledige afleiding te geven.)
 - c) Leid uit de formules voor onbepaalde integraal, gevonden in a en b,
af een formule $\int_{x_0}^{x_k} f$, uitgedrukt in voorwaartse differenties
in x_0 en achterwaartse differenties in x_k , zijnde de formule van
Gregory.

Proefwerk Numerieke Analyse (vervolg)

3. Ontbind de volgende matrix in driehoeken volgens Crout met rij-vertalingen en bereken de determinant.

$$\begin{pmatrix} 70.0 & 214.0 & -19.0 & -7.0 \\ 500.0 & 100.0 & 150.0 & -50.0 \\ -12.0 & -85.4 & -6.0 & -5.4 \\ 39.0 & 21.8 & 68.9 & 20.1 \end{pmatrix}$$

Opmerking: De matrix-elementen zijn exact. Doet men de ontbinding met enen op de hoofddiagonaal van de boven-driehoek dan hebben alle getallen, die men krijgt, hoogstens één decimaal achter de punt.

4. a) Geef een definitie van positief definitie symmetrische matrix:

b) Bewijs de stelling:

Als een matrix A van de orde n geschreven kan worden als $A = F^T F$, waarbij F een reële (rechthoekige) matrix van de rang n is, dan is A positief definitief.

c) Beschrijf een speciale methode voor het oplossen van een lineair stelsel, waarvan de matrix symmetrisch positief definitief is.

Licht de methode toe aan de hand van een matrix van de orde 3 met elementen a_{ij} .

Waarom gaat deze methode mis bij symmetrisch niet-positief-definitieve matrices?

5. Beschrijf de iteratieve methode van Gauss-Seidel.

Bij wat voor matrices weet men zeker, dat de methode convergeert?

Bij wat voor matrices biedt deze iteratieve methode voordelen boven niet-iteratieve methoden?

Errata

- pag.2 regel 19: $4xaxc$ moet zijn $4xaxc$
 regel 21-26 ($4x$): $2xa$ moet zijn $2xa$
- pag.3 regel 19: $9.34x10^{+(-7)}$ moet zijn $9.34x10^{+(-7)}$
 regel 24: $X, /, \frac{\circ}{\circ}$ moet zijn $\times, /, \frac{\circ}{\circ}$
 regel 27 (voorlaatste): $x / \frac{\circ}{\circ}$ moet zijn $\times / \frac{\circ}{\circ}$
- pag.4 opgave 2, laatste regel: $2.54x10^{+(-n)}$ moet zijn $2.54x10^{+(-n)}$

Verzamelde errata van Hoofdstuk 2

- pag. 21 "Honscholder" moet zijn "Householder";
pag. 24 opgave-nummers 16, 17, 18 moeten zijn resp. 21, 22, 23;
pag. 39 regel 5 van onder "det(M)" moet zijn " $|\det(M)|$ ";
pag. 46 opgave 31 moet luiden:

"Bewijs de formule $\sum_{j=0}^{n-1} x_j \int_j^n (x) = x$, voor $n > 1$ ";

- pag. 58 sectie 12 eerste formule: " $f(x)$ " moet zijn " $f_2^*(x)$ ";
pag. 58 regel 6 van onder, in de tweede determinant:
" $y_i, \dots, i+k \quad x_i - x$ " moet zijn " $y_i, \dots, i+k-1 \quad x_i - x$ ";
pag. 58 laatste regel " x_k " moet zijn " x_{i+k} ";
pag. 60 na het voorbeeld is het sectie-nummer 13 vergeten,
er moet dus staan: "13. Inverse interpolatie";
pag. 78 formule (18.4) " $\binom{p}{4} \delta_0^4$ " moet zijn " $\binom{p+2}{4} \delta_0^4$ ";
pag. 80 regel volgend op formule (19.2):
" $\binom{p+i-1}{2i-1}$ " moet zijn " $\binom{p}{2i} \binom{p+i-1}{2i-1}$ ";
pag. 82 in formule (19.7) moeten worden toegevoegd:
achter "=" een factor " $\frac{1}{2}$ ", achter " h^{2i+1} " een openingshaak
en achteraan een sluithaak;
pag. 85 formule (20.1) " $2B_n f^{(n)}(x_0)$ " moet zijn " $2B_n(p) f^{(n)}(x_0)$ ".

Errata hoofdstuk 3 & 4

pag. 97 in formule voor $\frac{df(x)}{dx}$: " $2p - 1$ " moet zijn $\frac{2p - 1}{2}$

in formule voor $f''(x_0)$: " $\frac{1}{h}$ " moet zijn " $\frac{1}{h^2}$ "

in voorlaatste regel: "D's" moet zijn "V's"

pag. 104 in formule (2.2.1) schrap: " $= 2Ch^3$ "

pag. 105 regel 5 "magnitudinis" moet zijn "magnitudine"

in formule (2.2.3) schrap: " $= 2Ch^5$ "

pag. 121 regel 6 moet luiden: "22 behandelt theorie, de andere boeken
zowel theorie als numerieke methoden."

pag. 129 in opschrift van sectie 3: "driehoeksvorm" moet zijn "driehoeken"

pag. 135 laatste regel van achter "gebruiken" toevoegen", die evenwel geen
extra precisie levert."

pag. 144 regel 17 achter " $r^T =$ " moet staan " (-.001, +.001) "

