OEFENINGEN NUMERIEKE WISKUNDE

OEFENZITTING 2: FOUTENANALYSE

Probleem 1. (Genormeerde bewegende kommavoorstelling) Analyseer de volgende algoritmen voor het bepalen van de basis, b, en het aantal cijfers in de mantisse, p:

- bepaal_b $<$uit: $b>$
  1. $A \leftarrow 1$
  2. while $(A+1)-A=1$
     - 2.1. $A \leftarrow 2*A$
  3. $i \leftarrow 1$
  4. while $(A+i)=A$
     - 4.1. $i \leftarrow i+1$
  5. $b \leftarrow (A+i)-A$
- bepaal_p $<$in: b; uit: $p>$
  1. $p \leftarrow 1$
  2. $z \leftarrow b$
  3. while $(z+1)-z=1$
     - 3.1. $p \leftarrow p+1$
     - 3.2. $z \leftarrow z*b$

Probleem 2. Beschouw volgende algoritmen voor het berekenen van product en scalair product:

- PRODUCT $<$in: $a_1,\ldots,a_n$; uit: $P=\Pi_{i=1}^n a_i>$
  1. $P \leftarrow a_1$
  2. voor $i=2:1:n$
     - 2.1 $P \leftarrow P*a_i$
- SCALAIR PRODUCT $<$in: $a_1,\ldots,a_n,b_1,\ldots,b_n$; uit: $SP=\sum_{i=1}^n a_ib_i>$
  1. $SP \leftarrow a_1b_1$
  2. voor $i=2:1:n$

  2.1 $SP \leftarrow SP+a_i*b_i$

Maak een afschatting van de fout wanneer de bewerkingen met eindige precisie in bewegende kommavoorstelling gebeuren. Veronderstel dat, met a en b exact voorgesteld in de computer, $\mathrm{fl}(a\circ b)=(a\circ b)(1+\epsilon)$ met $|\epsilon|\leq\epsilon_{\mathrm{mach}}$. Met $\circ$ bedoelen we zowel optelling als vermenigvuldiging.

Probleem 3. Beschouw de volgende algoritmen voor het evalueren van een veelterm $V=\sum_{i=0}^n a_ix^i$ in x:

- eval1 $<$in: $a_0,\ldots,a_n,x$; uit: $V>$
  1. $V \leftarrow a_nx^n$
  2. voor $i=n-1:-1:0$
     - 2.1 $V \leftarrow V+a_i*x^i$
- eval2 $<$in: $a_0,\ldots,a_n,x$; uit: $V>$
  1. $V \leftarrow a_n$
  2. voor $i=n-1:-1:0$
     - 2.1 $V \leftarrow V*x+a_i$
- eval3 $<$in: $a_0,\ldots,a_n,x$; uit: $V>$
  1. $p \leftarrow x$
  2. $V \leftarrow a_0$
  3. voor $i=1:1:n$

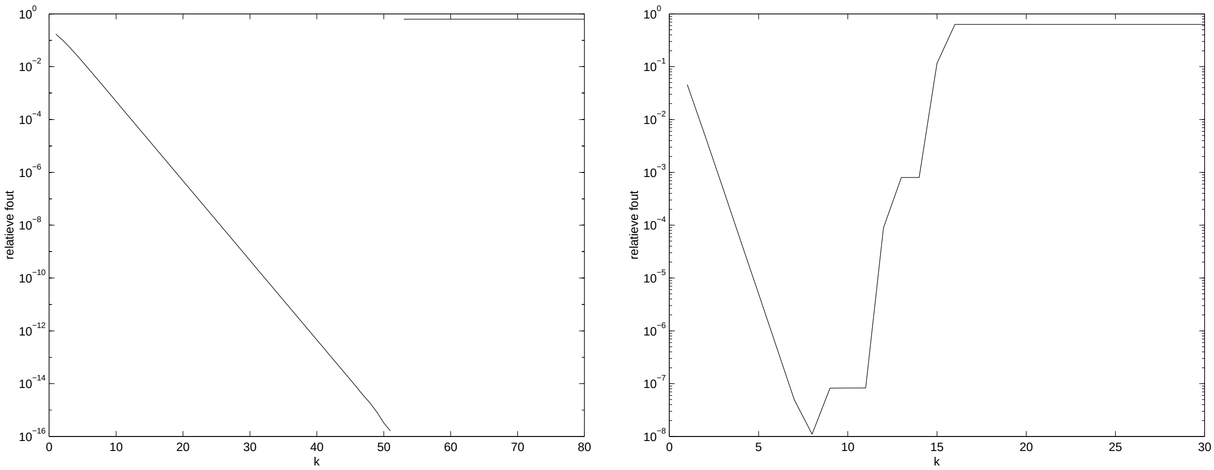
3.1 $V \leftarrow V + a_i * p$
3.2 $p \leftarrow p * x$

Welke methode is geimplementeerd (achterwaartse/voorwaartse evaluatie, horner)? Bepaal voor de drie algoritmen het aantal bewerkingen in functie van n en maak een afschatting van de fout indien de machineprecisie eindig is. Vergelijk de resultaten.

Probleem 4. (Examenvraag)
We weten dat $\lim_{x\to\infty}(1+1/x)^x = e^1$. We gaan de waarde e benaderen door $\tilde{e}_k = (1+1/x_k)^{x_k}$ uit te rekenen voor

- $x_k = 2^k$ en
- $x_k = 10^k$.

In de figuren hieronder geven we de relatieve fout weer, d.w.z. $\left|\frac{\tilde{e}_k - e}{e}\right|$. De eerste figuur geeft de relatieve fout voor de machten van 2 terwijl de tweede figuur de fouten voor machten van 10 weergeeft.



(a) Relatieve fout

(b) Relatieve fout

Waarom zijn de twee grafieken zo verschillend?
Kan men hieruit afleiden met welke basis en met hoeveel beduidende cijfers de computer werkt? Verklaar in detail je antwoord.

Probleem 5*. Analyseer het volgende recursieve algoritme:
SOM <in: $a_1, \ldots, a_n$; uit: $S = \sum_{i=1}^n a_i$ >
1. als $n = 1$
    1.1 $S \leftarrow a_1$
    1.2 anders
        $S \leftarrow S(a_1, \ldots, a_{n\mathrm{div}2}) + S(a_{n\mathrm{div}2+1}, \ldots, a_n)$

De bewerking 'div' levert het quotient van de gehele deling, dus $(2n+1)\mathrm{div}2 = (2n)\mathrm{div}2 = n,\ n \in \mathbb{N}$.
Toon aan dat in eindige precisie, de fout $S - \bar{S}$ voldoet aan:

$$n \le 2^k,\ k \in \mathbb{N} \rightarrow |S - \bar{S}| \le k\epsilon_{\mathrm{mach}}\left(|a_1| + |a_2| + \cdots + |a_n|\right).$$

Een recursief algoritme leent zich tot een bewijs door ...