Oplossingen oefenzittingen

Numerieke Wiskunde - H0M71

Dit document bevat de einduitkomsten van (een deel van) de oefeningen van de oefenzittingen. Zorg ervoor dat je deze zelf kunt bekomen én dat je de oplossingen ook kan interpreteren.

In oefenzittingen 8, 9 en 10 wordt er gevraagd de convergentiefactor af te lezen uit een grafiek van de benaderingsfout voor verschillende iteratieve processen. De convergentiefactor is gedefinieerd als

$$\rho = \lim_{k\to\infty} \frac{\epsilon^{(k+1)}}{\epsilon^{(k)}}.$$

Merk op dat uit de definitie van orde (zie handboek) volgt dat als de orde groter is dan 1, de convergentiefactor ρ gelijk is aan 0!

Als de orde 1 is, dan mag je er voor k "voldoende groot" (meestal) vanuit gaan dat

$$\rho \approx \frac{\epsilon^{(k+1)}}{\epsilon^{(k)}} \implies \rho^m \approx \frac{\epsilon^{(k+m)}}{\epsilon^{(k)}}. \tag{1}$$

In één stap wordt de fout ongeveer vermenigvuldigd met ρ, dus in m stappen wordt de fout ongeveer vermenigvuldigd met ρ^m. Indien je dus de waarde van de fout kan aflezen in stappen k en $k+m$, dan vind je de benadering voor ρ

$$\rho \approx \left(\frac{\epsilon^{(k+m)}}{\epsilon^{(k)}}\right)^{\frac{1}{m}}. \tag{2}$$

Door m groot genoeg te nemen, middel je afleesfouten uit en krijg je een nauwkeurigere benadering.

Eigenlijk komt dit neer op het schatten van de richtingscoëfficiënt van de rechte die je bekomt in een grafiek van de fout met logaritmische schaal op de y-as. Uit vergelijking (1) volgt namelijk dat

$$\log\left(\epsilon^{(k+1)}\right) \approx \log(\rho) + \log\left(\epsilon^{(k)}\right)$$

en bijgevolg

$$\log\left(\epsilon^{(k)}\right) \approx k\log(\rho) + \log\left(\epsilon^{(0)}\right).$$

Hieraan zie je dat je een rechte bekomt met richtingscoëfficiënt $\log(\rho)$. De richtingscoëfficiënt kan je nu schatten als

$$\log(\rho) \approx \frac{\log\left(\epsilon^{(k+m)}\right) - \log\left(\epsilon^{(k)}\right)}{m}.$$

Deze formule is iets minder handig dan formule (2), want je moet de log van de fout aflezen, en nadien moet je nog ρ halen uit $\log(\rho)$. Toon nu zelf eens aan dat deze laatste formule equivalent is met formule (2).

1 Bewegende kommavoorstelling en foutenanalyse

Probleem 1. EP getallen: 24 bit, DP getallen: 53 bit

Probleem 2. $2^{52} - 1$ tussen de getallen 1 en 2. $2^{50} + 2^{49} - 1$ tussen de getallen 7 en 9.

Probleem 3. $x_n = 1000$ voor $n \geq 1000$.

Probleem 5.

$$\left|\frac{\bar{y} - y}{y}\right| \leq \left(\frac{3}{2}\frac{\sqrt{x+1}}{\sqrt{x+1}+1} + 2\right)\epsilon_{mach} \leq \frac{7}{2}\,\epsilon_{mach}$$

Probleem 6.

$$\left|\frac{\bar{y} - y}{y}\right| \leq \left(\left|\frac{\cos(x)}{1-\cos(x)}\right| + 3\right)\epsilon_{mach}$$

Probleem 8.

(a) $$\left|\frac{\bar{y} - y}{y}\right| \leq 2\,\epsilon_{mach}$$

(b) $$\left|\frac{\bar{y} - y}{y}\right| \leq \left(\left|\frac{\cos(x)}{1-\cos(x)}\right| + 3\right)\epsilon_{mach}$$

(c) $$\left|\frac{\bar{y} - y}{y}\right| \leq \left(\left|\frac{e^{-2x}}{1-e^{-2x}}\right| + 2\right)\epsilon_{mach}$$ (Geen fout voor 2*x in basis 2.)

(d) $$\left|\frac{\bar{y} - y}{y}\right| \leq \left(\left|\frac{1}{x}\right| + |\log(y)| + 1\right)\epsilon_{mach}$$

(e) $$\left|\frac{\bar{y} - y}{y}\right| \leq \left(\left|\frac{1}{2}\frac{e^x}{e^x - 1}\right| + \frac{3}{2}\right)\epsilon_{mach}$$

(f) $$\left|\frac{\bar{y} - y}{y}\right| \leq \left(\left|\frac{1}{x}\cot\left(\frac{1}{x}\right)\right| + 1\right)\epsilon_{mach}$$

(g) $$\left|\frac{\bar{y} - y}{y}\right| \leq \left(\left|\log(y) + \frac{x^4}{1+x^2}\right| + x^2 + 1\right)\epsilon_{mach}$$ (Dezelfde ϵ_i voor de bewerking x^2!)

(h) $$\left|\frac{\bar{y} - y}{y}\right| \leq \left(\left|\frac{e^{x^2} + e^{-x^2}}{e^{x^2} - e^{-x^2}} \cdot x^2 - 1\right| + \left|\frac{2e^{x^2}}{e^{x^2} - e^{-x^2}}\right| + 2\right)\epsilon_{mach}$$ (id.)

Probleem 9.

$$\left|\frac{\overline{P}-P}{P}\right| \leq (n-1)\,\epsilon_{mach}$$

$$\begin{aligned}\left|\overline{SP}-SP\right| &= \left|\sum_{i=1}^{n}\epsilon_i(a_ib_i)+\sum_{i=1}^{n-1}\delta_i\left(\sum_{k=1}^{i+1}a_kb_k\right)\right| \\ &\leq \left(\sum_{i=1}^{n}|a_ib_i|+\sum_{i=1}^{n-1}\left|\sum_{k=1}^{i+1}a_kb_k\right|\right)\epsilon_{mach}\end{aligned}$$

2 Conditie en stabiliteit

Probleem 1. $\dfrac{f'(x)x}{f(x)} = \dfrac{2x^2+4x+3}{(1+x)(3+2x)}$, slechte conditie voor $x \approx -1$ en $x \approx \frac{-3}{2}$

1. $\left|\dfrac{\bar{y}-y}{y}\right| \leq \left(1+\dfrac{|1+2x||1+x|}{|3+2x||x|}+\dfrac{2}{|3+2x||x|}\right)\epsilon_{mach}$ (geen fout bij vermenigvulding met 2)

   - onstabiel voor $x \approx 0$
   - zwak stabiel voor $x \approx \frac{-3}{2}$
   - voorwaarts stabiel voor andere waarden van x

   Vraag: geeft dit algoritme een nauwkeurig resultaat voor $x \approx 1$?

2. $\left|\dfrac{\bar{y}-y}{y}\right| \leq 4\,\epsilon_{mach}$ (geen fout bij vermenigvulding met 2)

   - voorwaarts stabiel voor alle waarden van x

   $\rightarrow$ beter algoritme

Opmerking: wanneer je op een machine met basis 2 vermenigvuldigt met of deelt door 2, dan maak je geen relatieve fout, waarom? Als je hier toch een relatieve fout zou doorvoeren dan moet je in de tweede term van 1. $|1+2x|$ vervangen door $(|1+2x|+|2x|)$ en in 2. staat er dan $(4+|2x|/|3+2x|)\epsilon_{mach}$, waardoor algoritme 2. zwak stabiel is voor $x \approx -3/2$.

Probleem 2. $\delta_c y = \big(1+x\cot(x)\big)\delta x$, slechte conditie voor $x \approx k\pi$, $k \neq 0$. (Waarom $k \neq 0$?)

$\left|\dfrac{\bar{y}-y}{y}\right| \leq 2\,\epsilon_{mach}$, voorwaarts stabiel algoritme

Probleem 3. $\delta_c x_{12} = \dfrac{\frac{d(x_{12})}{dt}\cdot t}{x_{12}(t)} = \mp\dfrac{t}{2\sqrt{1-t}\left(1\pm\sqrt{1-t}\right)}$, slechte conditie voor $t \approx 1$, voor $t = 1$ heeft de veelterm een dubbel nulpunt

Probleem 4.

(a) $\delta_c y = \dfrac{2(1+x)}{2+x}$

- **eval1** $\left|\dfrac{\bar{y}-y}{y}\right| \leq \left(\dfrac{3(1+x)^2}{|2+x||x|}+1\right)\epsilon_{mach}$
- **eval2** $\left|\dfrac{\bar{y}-y}{y}\right| \leq 2\,\epsilon_{mach}$

(b) $\delta_c y = \dfrac{-2e^{2x}x}{e^{2x}-1}\delta x$, slechte conditie voor $x \to +\infty$, goede conditie voor $x \approx 0$.

- **eval1** $\left|\dfrac{\bar{y}-y}{y}\right| \leq \left(2\left|\dfrac{e^{2x}}{e^{2x}-1}\right| + (e^x+1) + |e^x-1| + 1\right)\epsilon_{mach}$
  - onstabiel voor $x \approx 0$
  - onstabiel voor $x \to +\infty$ (in de praktijk voor $x >> 1$)

    Waarom niet zwak stabiel?

    Voor grote positieve waarden van x is de conditie slecht én is de relatieve fout $\delta_s y$ door afrondingsfouten groot. We moeten echter kijken naar de volgende verhouding

    $$\lim_{x\to+\infty} \frac{|\delta_s y|}{\left|\delta_c y_{(\delta x \approx \epsilon_{mach})}\right|} = \lim_{x\to+\infty} \frac{(e^x+1)+|e^x-1|}{2|x|} = +\infty.$$

    Enkel wanneer deze verhouding in grootte-orde kleiner of gelijk is aan 1, spreken we van zwak stabiel.
  - Voorwaarts stabiel voor $x \to -\infty$, en merk op dat de conditie voor $x \to -\infty$ ook goed is.

- **eval2** $\left|\dfrac{\bar{y}-y}{y}\right| \leq \left(\left|\dfrac{e^{2x}}{e^{2x}-1}\right| + 2\right)\epsilon_{mach}$
  - onstabiel voor $x \approx 0$
  - voorwaarts stabiel voor andere waarden van x

Formule **eval2** is dus beter voor grote waarden van x.

3 Veeltermінterpolatie

Probleem 1.

(a) Beide methoden geven $y_1(x) = 4x - 3$.

(b) $[-3 \;\; 4]^T$ is een oplossing van het Vandermonde stelsel.

(c) $f''(x) = 4 \Longrightarrow$ formule (4.10) : $E_1(x) = 2(x+1)(x-1) = f(x) - y_1(x)$.

Probleem 2. $y_2(x) = f(x)$, $f^{(3)}(x) = 0 \Longrightarrow$ formule (4.10) : $E_2(x) = 0 = f(x) - y_2(x)$. (Leg het verband met Probleem 4.)

Probleem 3. $y_3(x) = f(x)$. Verklaar.

Probleem 4. Als $f = p$ een veelterm is van graad $\leq n$, dan geldt er dat $y_n = p$. Neem $p(x) = 1$ en $p(x) = x^k$.

Probleem 5. Bewijs dat het rechterlid de interpolerende veelterm van graad n is. Dit houdt in dat y_n een veelterm moet zijn van graad n die voldoet aan de interpolatievoorwaarden.

Probleem 6. Het volstaat te bewijzen dat $\pi'(x_i) = \prod_{k=0,k\neq i}^{n}(x_i - x_k)$. Hiervoor pas je de regel $(f \cdot g)' = f'g + fg'$ toe op $\pi'(x) = \Big((x - x_i)\prod_{k=0,k\neq i}^{n}(x - x_k)\Big)'$.

Probleem 7. Algoritme 4.3 in het handboek. (Probeer dit eerst zelf op te stellen.)

4 (PC) Bewegende kommavoorstelling en benaderings- en afrondingsfouten

Probleem 2. $(0.1)_{10} = (1.100110011001\ldots)_2 \times 2^{-4}$, $fl(0.1) = (1.10011001\ldots10011010)_2 \times 2^{-4}$ (52 cijfers na de komma)

$\Longrightarrow fl(0.1) - 0.1 \approx (0.00000000\ldots000000001)_2 \times 2^{-4}$ met 1 het 53ste cijfer na de komma

$$= 2^{-53} \cdot 2^{-4} = 2^{-57}$$

$\Longrightarrow$ relatieve fout $= 2^{-57}/0.1 \approx 6.9 \cdot 10^{-17}$

Probleem 3. Formule (1) heeft orde 1, want de benaderingsfout neemt af zoals $\mathcal{O}(h)$ als $h \to 0$. Dit wordt bewezen in de opgave. Bewijs zelf m.b.v. Taylor reeksen dat de benaderingsfout voor formule (2) afneemt zoals $\mathcal{O}(h^2)$ als $h \to 0$ en de orde dus gelijk is aan 2.

Als de benaderingsfout voldoet aan $E_{benadering} = Ch^p$ [1] voor een constante C, dan geldt er dat $\log(E_{benadering}) = \log(C) + p\log(h)$ en dan krijg je in een `loglog` grafiek van $E_{benadering}$ i.f.v. h dus een rechte met richtingscoëfficiënt gelijk aan p. Als de richtingscoëfficiënt van de rechte die overeenkomt met de benaderingsfout van formule (1) gelijk is aan 1, dan is het duidelijk uit de figuur dat de afrondingsfouten ongeveer op een rechte liggen met richtingscoëfficiënt gelijk aan -1. De afrondingsfouten nemen dus toe zoals $\mathcal{O}(h^{-1})$ als $h \to 0$. Als je een foutenanalyse zou doen, dan bekom je inderdaad dat voor een constante D

$$E_{afronding} \approx Dh^{-1}\epsilon_{mach}.$$

[1] Dit is ongeveer hetzelfde als $E_{benadering} = \mathcal{O}(h^p)$.

Wat is nu de kleinste fout die je kan bekomen met beide formules? De fout bestaat uit de benaderingsfout en de afrondingsfouten

$$E \approx Ch^p + Dh^{-1}\epsilon_{mach}.$$

Als h groot is domineert de eerste term en als h klein is de tweede. We vinden de kleinste fout als beide termen ongeveer even groot zijn

$$Ch^p \approx Dh^{-1}\epsilon_{mach} \Leftrightarrow h \approx \left(\frac{D}{C}\right)^{\frac{1}{p+1}} (\epsilon_{mach})^{\frac{1}{p+1}}.$$

Voor formule 1 is $p = 1$, en krijgen we dus de kleinste fout bij $h \approx (\epsilon_{mach})^{\frac{1}{2}} \approx 10^{-8}$ en de fout is dan $E \approx 10^{-8}$, voor formule (2) krijgen we $h \approx (\epsilon_{mach})^{\frac{1}{3}} \approx 10^{-5}$ en de fout is dan $E \approx 10^{-10}$. Dit komt overeen met de observaties op de figuur. Formule (2) geeft een nauwkeuriger resultaat, voor een grotere waarde van h.
(Omdat enkel de grootte ordes ons interesseren, hebben we de constanten C en D van grootte orde $\mathcal{O}(1)$ genomen. Voor ϵ_{mach} hebben we 10^{-16} genomen voor formule (1) en 10^{-15} voor formule (2) om het rekenwerk te vereenvoudigen.)

5 Numerieke integratie

Noot over het gebruik van een basis bij de voorwaarden voor de nauwkeurigheidsgraad: Dat de definitie van de nauwkeurigheidsgraad met formule (5) equivalent is met de voorwaarden bovenaan p. 2 volgt rechtstreeks uit lineaire algebra toegepast op veeltermen. Voor de duidelijkheid geven we hier nog even een uitgeschreven bewijs van deze equivalentie. Uit dit bewijs kan je ook goed zien dat je niet per se de basis $1, x, x^2, \ldots$ hoeft te gebruiken, maar ook een andere basis kan gebruiken zoals in Probleem 5. Wat krijg je bijvoorbeeld als je de Lagrange basis gebruikt?

- $\Rightarrow$: Als formule (5) geldt voor alle veeltermen $p(x)$ van graad $\leq d$, dan moet de formule uiteraard ook gelden voor alle veeltermen in een basis voor P_d, de deelruimte van alle veeltermen van graad $\leq d$. (Merk op dat de dimensie van deze deelruimte $d + 1$ is.)

- $\Leftarrow$: Omgekeerd, stel dat formule (5) geldt voor alle veeltermen in zo'n basis, die we nu even noteren als $\{q_0(x), q_1(x), \ldots q_d(x)\}$, d.w.z.

  $$H_0 q_k(x_0) + H_1 q_k(x_1) + \ldots + H_n q_k(x_n) = \int_a^b q_k(x)dx$$

  geldt voor alle $k \in \{0, 1, \ldots, d\}$. Elke veelterm $p(x)$ van graad $\leq d$ kan je uiteraard schrijven als een lineaire combinatie van de basisveeltermen

  $$p(x) = \sum_{k=0}^{d} a_k q_k(x)$$

en bijgevolg geldt er dat

$$\begin{aligned}\int_a^b p(x)dx &= \int_a^b \sum_{k=0}^{d} a_k q_k(x)dx = \sum_{k=0}^{d} \int_a^b a_k q_k(x)dx = \sum_{k=0}^{d} a_k \int_a^b q_k(x)dx \\ &= \sum_{k=0}^{d} a_k \Big(H_0 q_k(x_0) + H_1 q_k(x_1) + \ldots + H_n q_k(x_n) \Big) \\ &= H_0 \sum_{k=0}^{d} a_k q_k(x_0) + H_1 \sum_{k=0}^{d} a_k q_k(x_1) + \ldots + H_n \sum_{k=0}^{d} a_k q_k(x_n) \\ &= H_0 p(x_0) + H_1 p(x_1) + \ldots + H_n p(x_n).\end{aligned}$$

We hebben dus bewezen dat als formule (5) geldt voor elke basisveelterm, dan geldt ze voor alle veeltermen van graad $\leq d$. Dit volgt eigenlijk direct uit de lineariteit van de som en de integraal, die we hierboven hebben toegepast.

Probleem 2. Een interpolerende kwadratuurformule van $n+1$ punten heeft minstens nauwkeurigheidsgraad n. Een kwadratuurformule met $n+1$ punten die een nauwkeurigheidsgraad heeft van n of hoger is interpolerend.

Probleem 3. $H_0 = H_1 = 1$, de nauwkeurigheidsgraad is 1.

Probleem 4. $a = 1, b = 1, c = \dfrac{\sqrt{3}}{3}$, de nauwkeurigheidsgraad is 3.

Probleem 5. $H_0 = -\dfrac{2}{3}h, H_{-\frac{1}{2}} = H_{\frac{1}{2}} = \dfrac{4}{3}h$, de nauwkeurigheidsgraad is 3.

6 (PC) Het oplossen van stelsels lineaire vergelijkingen

Probleem 1. (a) $L * U \approx A$ tot op machine-nauwkeurigheid. Indien je L en U kent, dan kan je de determinant van A berekenen als `prod(diag(U))`.
(b) L is niet benedendriehoeks. Dit komt omdat er pivotering is toegepast zoals in Algoritme 3.4 in het handboek. Je krijgt $PA = LU \iff A = P^{-1}LU$, en de `L` die je van Matlab terugkrijgt is gelijk aan $P^{-1}L$. De matrix P^{-1} permuteert enkel de rijen van L. Hoe weet je dat P^{-1} net zoals P een permutatiematrix is?

Probleem 2. De onderstaande tabel geeft de relatieve fouten, residu's en conditiegetallen weer. Merk op dat dit grootte-ordes zijn en dat de precieze waarden kunnen verschillen per uitvoer, omdat de matrices randomheid bevatten. De tabel leert ons het volgende:

- Voor de eerste matrix zijn alledrie de methoden achterwaarts stabiel, want de residu's zijn klein. Dit geldt ook voor de laatste matrix. Voor de tweede matrix is gauss1 echter niet meer achterwaarts stabiel, wat je kan zien aan het 'veel grotere' residu.

De instabiliteiten zijn hier te wijten aan het feit dat gauss1 geen optimale rijpivotering toepast, en de tweede matrix op een bepaald moment een zeer kleine pivot zal bevatten. Wanneer optimale pivotering wel wordt toegepast, in gauss2, dan krijgen we wel een stabiele methode[2].

- Het conditiegetal van de matrix A geeft een verband tussen het residu en de relatieve fout op de berekende oplossing x, de ongelijkheid in formule (2) in de opgave. Je kan dit verband inderdaad nagaan voor de waarden in de tabel.
- Indien het conditiegetal van de matrix A groot is, zoals voor de derde matrix, dan kan zelfs een achterwaarts stabiele methode een grote relatieve fout op de berekende oplossing geven.

	$\kappa_2(A)$		gauss1	gauss2	qr
genmatrix1	10^2	$\|\|\delta x\|\|_2$	10^{-15}	10^{-15}	10^{-15}
		$\|\|r\|\|_2/\|\|b\|\|_2$	10^{-16}	10^{-16}	10^{-16}
genmatrix2	10^1	$\|\|\delta x\|\|_2$	10^{-7}	10^{-15}	10^{-15}
		$\|\|r\|\|_2/\|\|b\|\|_2$	10^{-8}	10^{-16}	10^{-16}
genmatrixc	10^{11}	$\|\|\delta x\|\|_2$	10^{-5}	10^{-6}	10^{-5}
		$\|\|r\|\|_2/\|\|b\|\|_2$	10^{-16}	10^{-17}	10^{-16}

7 Het oplossen van niet-lineaire vergelijkingen

Probleem 1. Stel dat de orde gelijk is aan p en er bijgevolg geldt dat

$$\lim_{k\to\infty} \frac{\varepsilon^{(k+1)}}{[\varepsilon^{(k)}]^p} = \rho_p, \quad \text{met} \quad 0 < \rho_p < \infty.$$

We kunnen dan het volgende schrijven:

$$\begin{aligned}
\lim_{k\to\infty} \frac{\varepsilon^{(k+1)}}{[\varepsilon^{(k)}]^n} &= \lim_{k\to\infty} \left(\frac{\varepsilon^{(k+1)}}{[\varepsilon^{(k)}]^p} \frac{1}{[\varepsilon^{(k)}]^{n-p}} \right) \\
&= \lim_{k\to\infty} \left(\frac{\varepsilon^{(k+1)}}{[\varepsilon^{(k)}]^p} \right) \lim_{k\to\infty} \left(\frac{1}{[\varepsilon^{(k)}]^{n-p}} \right) \\
&= \rho_p \lim_{k\to\infty} \left(\frac{1}{[\varepsilon^{(k)}]^{n-p}} \right).
\end{aligned}$$

Aangezien het hier gaat over een convergerende rij van benaderingen is $\lim_{k\to\infty} \varepsilon^{(k)} = 0$, dus is de bovenstaande limiet gelijk aan ∞ voor $n > p$ en gelijk aan 0 voor $n < p$.

[2]Ter info: gauss2 is voor bijna alle matrices achterwaarts stabiel, het algoritme met qr is voor alle matrices achterwaarts stabiel.

Probleem 2. (a) volledig consistent, spiraalvormige divergentie, $F'(x^*) = -1.763\ldots$
(b) volledig consistent, spiraalvormige convergentie, $F'(x^*) = -0.5671\ldots$

Probleem 3. Voor deze oefening moet je eerst $F(x)$ bepalen volgens de methode van Newton-Raphson, $F(x) = x - f(x)/f'(x)$. Uit de gevallenstudie van Newton-Raphson volgt dat de methode altijd consistent is, maar over reciproke consistentie weet je nog niets. Er volgt ook uit dat de convergentie voor enkelvoudige nulpunten kwadratisch is.

- $f(x) = 1 - \frac{a}{x^2}$, $F(x) = \frac{x}{2}\left(3 - \frac{x^2}{a}\right)$, consistent, niet reciprook consistent, want $x = 0$ is een vast punt van F maar geen nulpunt van f.
- De convergentiefactor is gelijk aan $F'(x^*) = 0$, dus de orde is 2.
- Je zou m.b.v. een zelfgetekende figuur toch zeker het gebied van divergentie $\{x : |x| > \sqrt{5a}\}$ moeten kunnen aanduiden en de gebieden van convergentie $\{x : x > 0, x < \sqrt{3a}\}$ en $\{x : x < 0, x > -\sqrt{3a}\}$. Als je de figuur goed kan interpreteren zijn de precieze waarden van de intervallen minder belangrijk.

Probleem 4. Gebruik de afleiding in het handboek bij de gevalstudie van de methode van Newton-Raphson. Voor de duidelijkheid noemen we $\hat{m}$ de multipliciteit van de wortel x^* van f en m de waarde van de parameter in de formule voor $F(x)$. Het is duidelijk dat

$$F'(x) = 1 - \left(m\frac{f(x)}{f'(x)}\right)'.$$

Bij Newton-Raphson zou m gelijk zijn aan 1 in bovenstaande formule en volgens de gevallenstudie geldt er dat $F'(x^*) = 1 - 1/\hat{m}$. Bijgevolg geldt er voor de aangepaste methode dat

$$F'(x^*) = 1 - \frac{m}{\hat{m}}.$$

Indien $m = \hat{m}$ dan is de methode kwadratisch. **Antwoord denkvraag:** Meestal weet men echter niet op voorhand wat de multipliciteit zal zijn van de wortels die men zoekt en kan men de waarde van m moeilijk vastleggen. Indien men bovendien $m > 1$ zou nemen, dan geldt er voor enkelvoudige wortels ($\hat{m} = 1$) dat $F'(x^*) = 1 - m <= -1$, en dan is er zelfs geen convergentie.

Probleem 5. (a) volledig consistent, monotone convergentie, $F'(x^*) = 0.216\ldots$
(b) consistent want $x = 0$ is een vast punt, maar geen nulpunt[3], spiraalvormige convergentie, $F'(x^*) = -0.3816\ldots$ Merk op dat de methode hier convergeert voor alle waarden van $x \in (0, 1]$. Dit volgt niet uit een stelling, maar kan je grafisch afleiden.

Probleem 6.

[3]Je kan ook argumenteren dat $F(x)$ niet gedefinieerd is voor $x = 0$, waardoor dit geen vast punt is. Indien je $F(0)$ definieert als de waarde van de limiet van F in 0, dan is $x = 0$ wel een vast punt van F.

- $f(x) = x - \dfrac{a}{x}, F(x) = \dfrac{2ax}{x^2 + a}$, consistent, niet reciprook consistent, want $x = 0$ is een vast punt van F maar geen nulpunt van f
- De convergentiefactor is $F'(x^*) = 0$, dus de orde is 2.
- Je zou m.b.v. een zelfgetekende figuur moeten kunnen zien dat de convergentie monotoon verloopt naar de nulpunten en dat er voor alle $x \neq 0$ convergentie is naar $\sqrt{a}$ voor $x > 0$ en naar $-\sqrt{a}$ voor $x < 0$.

Probleem 7.

- het moeilijkere deel van de verklaring van (4): Vermits $\Delta p(x)$ een kleine perturbatie is van een veelterm, is ook de afgeleide $\Delta p'(x)$ een veelterm met kleine coëfficiënten van ongeveer dezelfde grootte-orde als $\Delta p(x)$. Bijgevolg is $\Delta p'(c)\Delta c$ een tweede orde term die we mogen verwaarlozen.
- Indien c een meervoudig nulpunt is, dan geldt er dat $p'(c) = 0$, waardoor de tweede orde termen niet mogen worden verwaarloosd.
- In Sectie 18 van H2 van Deel 2 krijg je een benadering voor $x^* - x^*(\epsilon)$. Formule (5) van de opgave is een speciaal geval hiervan, waarbij $x^* = c$, $x^*(\epsilon) = c + \Delta c$, $f(x) = p(x)$ en $\epsilon g(x) = \Delta p(x)$.

Probleem 8.

1. $\Delta c \approx -2.5 \cdot 10^{-7}$
2. Er geldt dat $\lim_{c \to 1} -\frac{\Delta p(c)}{p'(c)} = -\infty$, wat duidelijk geen goede schatting is van Δc. Voor $t = 1$ heeft $p(x)$ een dubbele wortel $c = 1$, waardoor formule (4) niet meer geldt.

8 Stelsels niet-lineaire vergelijkingen

Probleem 1. $x^{(0)}, x^{(1)}, f(x^{(0)})$ zijn vectoren in $\mathbb{R}^n$, terwijl de functie $J(x)$ die je meegeeft als input, een matrix in $\mathbb{R}^{n \times n}$ teruggeeft. I.p.v. een deling door de afgeleide in Newton-Raphson voor één veranderlijke, krijg je nu een formule met de inverse van de Jacobiaan. Hier los je uiteraard een stelsel op. Tenslotte gebruik je $||x^{(1)} - x^{(0)}|| < \epsilon$ als absoluut stopcriterium of $||x^{(1)} - x^{(0)}|| \leq \epsilon ||x^{(0)}||$ als relatief stopcriterium of een combinatie van beiden.

Probleem 2.

(a) $J = \begin{bmatrix} 2x + 1 & -2y \\ -2x\cos(x^2) & 1 \end{bmatrix}$

(b) orde 2, want in figuur 1 zie je dat (voor k groot genoeg) de fout in elke stap wordt gekwadrateerd. Door de log schaal op de verticale as komt dit erop neer dat de afstand op de grafiek van 10^0 tot de waarde van de fout in elke stap verdubbelt. (Verklaar dit!) Als de orde 2 is, dan moet de convergentiefactor ρ gelijk zijn aan 0. In tabel 1 zie je ongeveer een verdubbeling van het aantal juiste beduidende cijfers. Dit aantal is voor $x^{(k)}$ (ongeveer) gelijk aan $1, 3, 6, 12$ voor respectievelijk $k = 6, 7, 8, 9$, waarbij we vergelijken met de waarde voor $k = 10$.

(c) Evalueren van $\det(J)$ voor $(x, y) = \left(x^{(10)}, y^{(10)}\right)$ geeft ongeveer de waarde 1.1897. Omdat de Jacobiaan is duidelijk regulier is in de oplossing, is de convergentie kwadratisch.

(d) In de figuur kan je goed zien dat het sterk afhangt van de startwaarde of er al dan niet convergentie zal optreden. Zelfs voor een startwaarde dichtbij de oplossingen is divergentie mogelijk.

Probleem 3.

(a) $J = \begin{bmatrix} 2x & 8y \\ y^2 - 1 & 2xy \end{bmatrix}$

(b) orde 1, want in figuur 2 zie je dat (voor k groot genoeg) de fout in elke stap met een vast getal kleiner dan 1 wordt vermenigvuldigd. Door de log schaal op de verticale as komt dit erop neer dat de afstand op de grafiek van 10^0 tot de waarde van de fout in elke stap vermeerdert met een vast getal. (Verklaar dit!) Je kan een schatting van de convergentiefactor aflezen van de grafiek, je bekomt $\rho \approx 0.5$. In tabel 2 zie je dat de fout in elke stap inderdaad ongeveer halveert.

(c) $J(2, \sqrt{3})) = \begin{bmatrix} 4 & 8\sqrt{3} \\ 2 & 4\sqrt{3} \end{bmatrix} \longrightarrow \det(J) = 0$, omdat de Jacobiaan singulier is in de oplossing is de convergentie lineair. De partiële afgeleiden van een functie in een punt bepalen de richtingscoëfficiënt van de raaklijn aan die functie in dat punt. Als twee functies in een snijpunt proportionele partiële afgeleiden hebben, d.w.z. de Jacobiaan is singulier, dan hebben ze in dit snijpunt dus dezelfde raaklijn. In figuur 2 zie je dat de twee functies elkaar inderdaad raken.

Probleem 4.

1. De formules voor totale stap zijn

$$x^{(k+1)} = x^{(k)} - \frac{y^{(k)} - \sin(\pi x^{(k)})}{-\pi \cos(\pi x^{(k)})}$$
$$y^{(k+1)} = y^{(k)} - \frac{(x^{(k)} - 1)^2 + (y^{(k)} - 1)^2 - 1/3}{2(y^{(k)} - 1)}$$

Voor enkelvoudige stap moet in de tweede formule $x^{(k)}$ vervangen worden door $x^{(k+1)}$

2. Voor totale stap kan je de fouten aflezen voor $k = 13$ en $k = 21$ als respectievelijk 10^{-6} en 10^{-10}. Hiermee bekom je een schatting $\rho \approx 0.3$. Voor enkelvoudige stap kan je de fouten aflezen voor $k = 8$ en $k = 14$ als respectievelijk 10^{-6} en 10^{-12}. Hiermee bekom je een schatting $\rho \approx 0.1$.

3. In $(0.8\ldots, 0.4\ldots)$ heeft de eerste vergelijking, waaruit we x oplossen, de steilste helling. Om te convergeren naar het andere nulpunt moeten we x oplossen uit de tweede vergelijking die in dat nulpunt de steilste helling heeft.

4. voor $x^{(0)} \approx 1/2 + k$ met k een geheel getal krijg je een deling door een heel kleine waarde die bijna nul is, idem voor $y^{(0)} \approx 1$.

9 Iteratieve methoden voor het oplossen van stelsels lineaire vergelijkingen

Probleem 2. Voor de exacte oplossing X van het stelsel geldt er $AX = B$ en dus $UX + DX + LX = B$. Trek deze vergelijking (na wat herschikken) af van de iteratieformules voor Jacobi en Gauss-Seidel in matrix notatie. Je bekomt, respectievelijk voor Jacobi en Gauss-Seidel, een verband tussen de opeenvolgende iteratiefouten. Hieruit haal je gemakkelijk het te bewijzen.

Probleem 3. Jacobi: $G = \begin{bmatrix} 0 & 1/2 \\ 1/2 & 0 \end{bmatrix}$, $||G||_\infty = 1/2 < 1 \Longrightarrow$ convergentie. Dit moet uiteraard ook blijken uit de spectraalradius. De eigenwaarden van G kan je berekenen als $\lambda = \pm 1/2$, bijgevolg $\rho(G) = 1/2 < 1$.
Gauss-Seidel: $G = \begin{bmatrix} 0 & 1/2 \\ 0 & 1/4 \end{bmatrix}$, $||G||_\infty = 1/2 < 1 \Longrightarrow$ convergentie. De eigenwaarden van G kan je berekenen als 0 en $1/4$, bijgevolg $\rho(G) = 1/4 < 1$.

Probleem 4. Jacobi: $G = \begin{bmatrix} 0 & -3/2 \\ -1/2 & 0 \end{bmatrix}$, $||G||_\infty = 3/2 > 1 \longrightarrow$ zegt niets over convergentie. De eigenwaarden van G kan je berekenen als $\lambda = \pm\sqrt{3}/2$, bijgevolg $\rho(G) = \sqrt{3}/2 < 1$. De methode van Jacobi convergeert dus, hoewel $||G||_\infty > 1$. Hier zie je duidelijk dat dit een voldoende, maar niet nodige voorwaarde is.
Gauss-Seidel: $G = \begin{bmatrix} 0 & -3/2 \\ 0 & 3/4 \end{bmatrix}$, $||G||_\infty = 3/2 > 1 \longrightarrow$ zegt niets over convergentie. De eigenwaarden van G kan je berekenen als 0 en $-3/4$, bijgevolg $\rho(G) = 3/4 < 1$. De methode van Gauss-Seidel convergeert dus, hoewel $||G||_\infty > 1$.
Uit de figuur kan je de convergentiefactoren aflezen. Voor Jacobi kan je de fouten aflezen voor $k = 0$ en $k = 80$ als respectievelijk 10^0 en 10^{-5}. Hiermee bekom je een schatting $\rho \approx 0.866$. Voor Gauss-Seidel kan je de fouten aflezen voor $k = 40$ en $k = 80$ als respectievelijk 10^{-5} en 10^{-10}. Hiermee bekom je een schatting $\rho \approx 0.75$. De convergentiefactoren zijn dus precies de spectraalradii, zoals ook blijkt uit de volgende oefenzitting!

10 (PC) Het berekenen van eigenwaarden

Probleem 1. De eigenwaarden van A vind je als de nulpunten van de karakteristieke veelterm $\det(\lambda I - A)$ en zijn gelijk aan $4, 3$ en -2. De eigenvector(en) x bij een eigenwaarde λ vind je als de oplossing(en) van $(\lambda I - A)x = 0$, waarbij $x \neq 0$, of dus als de vector(en) die de nulruimte van $(\lambda I - A)$ opspant (opspannen). De eigenvectoren bij 4,3 en -2 zijn respectievelijk

$$\begin{bmatrix}1\\1\\0\end{bmatrix}, \begin{bmatrix}1\\0\\0\end{bmatrix} \text{ en } \begin{bmatrix}0\\1\\1\end{bmatrix}.$$

Een veelvoud van een eigenvector is ook een eigenvector. Matlab geeft altijd eigenvectoren terug die genormaliseerd zijn, d.w.z. dat $\|x\|_2 = 1$. Voor de eerste eigenvector krijg je dan

$$\begin{bmatrix}1/\sqrt{2}\\1/\sqrt{2}\\0\end{bmatrix}.$$

Probleem 2.

- Voor de methode van de machten is er uiteraard convergentie naar de grootste eigenwaarde $\lambda = 4$ en bijhorende eigenvector. De methode van de inverse machten is eigenlijk de methode van de machten toegepast op de matrix $(\sigma I - A)^{-1}$. De eigenwaarden van deze matrix zijn $1/(\sigma - \lambda_i)$. Voor de waarde $\sigma = 3.8$ komt de grootste eigenwaarde van deze matrix overeen met $\lambda_i = 4$, dus is er ook convergentie naar deze eigenwaarde. Ook de adaptieve methode convergeert naar de eigenwaarde die het dichtst bij de initiële waarde van de verschuiving σ ligt.

- De convergentiefactoren die je zou moeten aflezen zijn $\rho_1 \approx 0.75$ en $\rho_2 \approx 0.25$. Deze waarden komen overeen met de theoretische convergentiefactoren

$$\rho_1 = \frac{\lambda_2(A)}{\lambda_1(A)} \quad \text{en} \quad \rho_2 = \frac{\lambda_2\left((\sigma I - A)^{-1}\right)}{\lambda_1\left((\sigma I - A)^{-1}\right)}.$$

  Je besluit verder ook dat $\rho_3 = 0$.

- In de methode `invmachten_adaptief` zal de matrix $(\sigma_i I - A)^{-1}$ singulier worden, wanneer σ_i een eigenwaarde van A tot op machine-nauwkeurigheid benadert. De waarde van `x0` zal op dat moment de bijhorende eigenvector benaderen tot op machine nauwkeurigheid. In de stappen

```
x1 = (sigma(i)*eye(size(A,1))-A)\x0;
mu = 1/norm(x1);
```

zal `x1` daarom oneindig groot worden (verklaar dit!), waardoor in Matlab de waarde `Inf` wordt toegekend aan elementen van `x1`, en bijgevolg krijgt `mu` de waarde nul. Bekijk de rest van de code. Doordat $\mu = 0$, blijft $\sigma_{i+1} = \sigma_i$ en zal `x0` gelijk worden aan de nulvector. In de volgende stap wordt dan een stelsel opgelost met als matrix een singuliere matrix en als rechterlid de nulvector, wat niet gedefinieerd is en `NaN` (Not a number) waarden geeft in Matlab. Dit is analoog aan de operatie $0/0$.

Probleem 3. Als je de iteratiematrix G berekend hebt en de eigenwaarden en eigenvectoren, dan kan je de spectraalradius bekomen als de grootste eigenwaarde in absolute waarde

$$\rho(G) = \max_i \left(|\lambda_i(G)|\right) = 1.33\ldots = \frac{4}{3}.$$

Uit de opgave van de vorige oefenzitting besluit je dat de methode van Jacobi niet convergeert voor elke mogelijke keuze van de startvector. Inderdaad, kijk je naar de formule van $E^{(k)}$

$$E^{(k)} = G^k E^{(0)} = \alpha_1 \lambda_1^k V_1 + \alpha_2 \lambda_2^k V_2 + \ldots + \alpha_n \lambda_n^k V_n,$$

dan zie je dat de eerste term oneindig groot zal worden voor $k \to \infty$, op voorwaarde dat $\alpha_1 \neq 0$. Merk op dat er startvectoren zijn waarvoor $\alpha_1 = 0$. Als doorheen de berekening deze component nul blijft, dan krijg je convergentie als de tweede grootste eigenwaarde (in modulus) voldoet aan $|\lambda_2| < 1$. Merk echter op dat door afrondingsfouten, de eerste component op een bepaald moment verschillend kan worden van 0!

Bekijk je nu de startvectoren `x0a` en `x0b`, dan hangt de convergentie af van de waarde van α_1 in de ontbinding van de fouten `e0a = x0a-xe` en `e0b = x0b-xe` als lineaire combinatie van de eigenvectoren van de iteratiematrix G. De coëfficiënten α_i kan je vinden als oplossing van de stelsels

```
V\e0a          V\e0b
```

met als oplossingen, respectievelijk,

$$\begin{bmatrix} 0 \\ 1.3556\ldots \\ -2.0401\ldots \end{bmatrix} \quad \text{en} \quad \begin{bmatrix} 0.0000577\ldots \\ 1.3557\ldots \\ -2.0401\ldots \end{bmatrix}.$$

Voor de eerste startvector krijg je dus $\alpha_1 = 0$, waardoor de convergentie afhangt van de tweede grootste eigenwaarde van G, die in dit geval kleiner is dan 1. De methode van Jacobi convergeert, want $|\lambda_2| < 1$. Voor de tweede startvector krijg je een kleine waarde van α_1, waardoor er divergentie optreedt, maar de term die divergeert begint pas na een bepaald aantal stappen te domineren over de andere, convergerende termen.