Numerieke methoden 2

27 november 2019

De inhoud van dit dictaat is niet nieuw. Het vormt slechts een overzicht van de onderwerpen die tijdens het college behandeld zullen worden. Sommige delen van deze tekst zijn gebaseerd op het uitstekende boek [**?**].

Dit dictaat is uitsluitend bedoeld om te gebruiken binnen het college *Numerieke methoden 2* aan UHasselt in het studiejaar 2018/19.

1 Stelsels lineaire vergelijkingen

In deze sectie willen we methodes beschouwen om een stelsel lineaire vergelijking op te lossen:

Probleem 1. *Gegeven een* $A \in \mathbb{R}^{n\times n}$ *met* $\det(A) \neq 0$ *en een* $b \in \mathbb{R}^n$*. Zoek een* $\overline{x} \in \mathbb{R}^n$ *zodanig dat*

$$A\overline{x} = b. \tag{1}$$

We hebben in het vak 'Numerieke methoden 1' al de LU-decompositie beschouwd. In dit opleidingsonderdeel voegen wij er nog een tweede bij, de QU decompositie. Vervolgens bespreken wij wat iteratieve methoden voor de benadering van oplossingen voor Probleem 1.

1.1 Het konditiegetal

Beschouw de volgende vergelijking:

$$x_1 + (1+\varepsilon)x_2 = 2, \tag{2}$$
$$x_1 + x_2 \qquad\quad = 2 + \delta. \tag{3}$$

De oplossing is gelijk aan

$$x_1 = 2 + \frac{\delta + \varepsilon\delta}{\varepsilon}, \qquad x_2 = -\frac{\delta}{\varepsilon}.$$

We beschouwen δ als kleine storing in de data, dus wat we eigenlijk willen oplossen is het stelsel voor $\delta = 0$ (met oplossing $\overline{x} := (2, 0)$). De 'storing' in de data is gelijk aan

$$x_1 - \overline{x}_2 = \frac{\delta + \varepsilon\delta}{\varepsilon}, \qquad x_2 - \overline{x}_2 = -\frac{\delta}{\varepsilon}.$$

Voor $\varepsilon = \mathcal{O}(1)$ (dus niet klein!) is $\|x - \overline{x}\| = \mathcal{O}(\delta)$, een kleine storing in δ impliceert een kleine storing in x. Het ziet er anders uit voor ε klein, want dan is een kleine storing in δ niet meer noodzakelijk een kleine storing in x. Denk aan $\varepsilon = \delta = 10^{-8}$ (ongeveer single precision). Dan is $\|x - \overline{x}\| = \mathcal{O}(1)$. Dus ook al hebben wij een relatief kleine storing in δ, is de storing in x vrij groot. We bespreken vervolgens wat er hier de reden voor is.

Definitie 1. *De norm van een matrix, ten opzichte van een vectornorm, is gedefinieerd door*

$$\|A\| := \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|} \tag{4}$$

Lemma 1 (Eigenschappen van matrix normen)**.** $\|\cdot\|$ *is inderdaad een norm op* $\mathbb{R}^{n\times n}$ *(ga het na), dus er geldt:*

1. $\|\cdot\| : \mathbb{R}^{n\times n} \to \mathbb{R}^{\geq 0}$.
2. $\|A\| = 0$ *als en alleen als* $A = 0$.
3. $\|\lambda A\| = |\lambda|\|A\|$ *voor elk* $\lambda \in \mathbb{R}$.
4. $\|A + B\| \leq \|A\| + \|B\|$ *(*driehoeksongelijkheid*).*

Er geldt bovendien:

$$\|Ax\| \leq \|A\|\|x\|, \quad \forall x \in \mathbb{R}^n,$$

en vervolgens

$$\|AB\| \leq \|A\|\|B\|$$

voor matrices $A, B \in \mathbb{R}^{n\times n}$.

Bewijs. Oefening. □

De representatie (4) is niet altijd echt handig. Gelukkig bestaan er voor enkele normen wat eenvoudigere uitdrukkingen.

Lemma 2 (Representatie van sommige matrix normen)**.** *Er geldt:*

- $\|A\|_\infty = \max_{i=1,\ldots,n} \sum_{k=1}^n |a_{ik}|$.
- $\|A\|_1 = \max_{k=1,\ldots,n} \sum_{i=1}^n |a_{ik}|$.
- $\|A\|_2 = \sqrt{\lambda_{\max}(A^T A)}$.

Met $\|A\|_p$ *betekenen wij de matrix norm gedefinieerd door de vector norm* $\|\cdot\|_p$ *in* (4) *te gebruiken.*

Bewijs. Oefening. □

Een belangrijk getal voor een matrix is het conditiegetal. We zullen zien dat dit getal een maat voor de storingen in de oplossing van (1) is als gegeven data gestoord is.

Definitie 2 (Conditiegetal van een matrix)**.** *Stel dat* $A \in \mathbb{R}^{n\times n}$ *een inverteerbare matrix is. Dan is het conditiegetal gedefinieerd door*

$$\kappa(A) := \|A\|\|A^{-1}\|.$$

Lemma 3. *Voor een* $A \in \mathbb{R}^{n\times n}$*,* A *inverteerbaar, geldt:*

- $\kappa(A) = \kappa(A^{-1})$.
- $\kappa(A) \geq 1$.

Bewijs. Oefening. □

Vanaf nu beschouwen we - indien niet expliciet anders gezegd - alleen maar inverteerbare matrices.

Overeenkomst: De matrix A^{-1} bestaat.

Vaak zijn alleen maar benaderingen van A en b bekend. We beschouwen eerst het geval dat A exact bekend is en dat we alleen maar een benadering van b ter beschikking hebben.

Lemma 4. *Stel dat $\overline{x} + \Delta x$ de oplossing van de gestoorde lineaire vergelijking*

$$A(\overline{x} + \Delta x) = b + \Delta b$$

voorstelt. Dan geldt:

$$\frac{\|\Delta x\|}{\|\overline{x}\|} \leq \kappa(A) \frac{\|\Delta b\|}{\|b\|}.$$

Bewijs. Omdat $\overline{x} = A^{-1}b$ geldt $\Delta x = A^{-1}\Delta b$, dus we hebben al de afschatting voor de absolute fout,

$$\|\Delta x\| \leq \|A^{-1}\| \|\Delta b\|.$$

Nu geldt omwille van $A\overline{x} = b$ ook $\|b\| \leq \|A\| \|\overline{x}\|$ oftewel $\|\overline{x}\| \geq \frac{\|b\|}{\|A\|}$ en bijgevolg

$$\frac{\|\Delta x\|}{\|\overline{x}\|} \leq \|A^{-1}\| \frac{\|\Delta b\|}{\|\overline{x}\|} \leq \|A^{-1}\| \|A\| \frac{\|\Delta b\|}{\|b\|}.$$

□

Opmerking 1. *We beschouwen nogmaals het voorbeeld (2); veronderstel $\varepsilon > 0$. Hier is*

$$A = \begin{pmatrix} 1 & 1+\varepsilon \\ 1 & 1 \end{pmatrix},$$

met als conditiegetal (ten opzichte van de $\|\cdot\|_\infty$ norm)

$$\kappa_\infty(A) = \frac{1}{\varepsilon} \left\| \begin{pmatrix} 1 & 1+\varepsilon \\ 1 & 1 \end{pmatrix} \right\|_\infty \left\| \begin{pmatrix} 1 & -(1+\varepsilon) \\ -1 & 1 \end{pmatrix} \right\|_\infty = \frac{(2+\varepsilon)^2}{\varepsilon} = \mathcal{O}(\varepsilon^{-1}).$$

Het probleem is voor $\varepsilon \ll 1$ dus slecht geconditioneerd, dit is precies wat wij in het voorbeeld gezien hebben.

In de praktijk is niet alleen b, maar normaal ook A gestoord. De formule wordt dan wat moeilijker:

Lemma 5. *Stel dat $\overline{x} + \Delta x$ de oplossing is van de gestoorde vergelijking*

$$(A + \Delta A)(\overline{x} + \Delta x) = b + \Delta b.$$

Stel bovendien dat $\kappa(A)\frac{\|\Delta A\|}{\|A\|} < 1$. Dan geldt

$$\frac{\|\Delta x\|}{\|\overline{x}\|} \leq \frac{\kappa(A)}{1 - \kappa(A)\frac{\|\Delta A\|}{\|A\|}} \left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \right)$$

Opmerking 2. *De beperking $\kappa(A)\frac{\|\Delta A\|}{\|A\|} < 1$ is in de praktijk natuurlijk vaak moeilijk aan te wijzen.*

1.2 De QU decompositie

In deze sectie beschouwen wij een decompositie van een matrix A in een orthogonale matrix Q en een bovendriehoeksmatrix U, dus $A = QU$. Orthogonale matrices zijn bijzonder nuttig wanneer men met de euclidische norm werkt. We zullen dus in deze sectie slechts de $\|\cdot\|_2$ norm beschouwen. Met $\kappa_2(A)$ wordt het conditiegetal van A ten opzichte van de euclidische norm betekend, dus

$$\kappa_2(A) := \|A\|_2\|A^{-1}\|_2.$$

Om wat inzicht te krijgen even het volgende lemma:

Lemma 6. *Stel dat $A \in \mathbb{R}^{n\times n}$ symmetrisch positief definiet is. Er geldt*

$$\kappa_2(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}. \tag{5}$$

Bewijs. We weten al (zie Lemma 2) dat $\|A\|_2 = \sqrt{\lambda_{\max}(A^TA)}$. Omdat A spd is geldt $A^TA = A^2$ en alle eigenwaarden zijn positief, dus

$$\|A\|_2 = \lambda_{\max}(A), \qquad \|A^{-1}\|_2 = \lambda_{\max}(A^{-1}).$$

De eigenwaarden van A^{-1} zijn de inverse van de eigenwaarden van A. Dat bewijst het lemma. □

We beginnen eerst met de definitie van een orthogonale matrix:

Definitie 3. *Een matrix $Q \in \mathbb{R}^{n\times n}$ is orthogonal indien $Q^T = Q^{-1}$.*

Stelling 1. *Stel dat $Q, \overline{Q} \in \mathbb{R}^{n\times n}$ orthogonale matrices zijn. Er geldt:*

1. *$\|Qx\|_2 = \|x\|_2, \forall x \in \mathbb{R}^n$ en bijgevolg $\|QA\|_2 = \|A\|_2$ voor $A \in \mathbb{R}^{n\times n}$.*
2. *$\kappa_2(Q) = 1$.*
3. *$Q\overline{Q}$ is orthogonal.*

Bewijs. Oefening. □

Om de matrix A middels elementaire operaties in bovendriehoeksvorm te brengen bestaan er enkele methodes, vaak geometrisch gemotiveerd. We beperken dit college tot de Givens rotaties, Householder transformaties zullen in de zelfstudie beschouwd worden. Het volgende lemma definieert de Givens rotaties:

Lemma 7. *Gegeven zijn de waarden $a, b \in \mathbb{R}$. Stel dat $r := \sqrt{a^2 + b^2}$, $c := \frac{a}{r}$ en $s := \frac{b}{r}$. Er geldt:*

$$\underbrace{\begin{pmatrix} c & s \\ -s & c \end{pmatrix}}_{=:Q} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} r \\ 0 \end{pmatrix}.$$

Q is een orthogonale matrix en wordt Givens rotatie genoemd.

Met de Givens rotatie kan men een matrix A in bovendriehoeksvorm brengen. We laten dit aan een voorbeeld zien. (Merk op: het bevel `sym` zorgt ervoor dat bv. breuken en wortels door Matlab niet als numerieke waarden geïnterpreteerd worden, maar als 'echte' breuken en wortels.)

Voorbeeld 1.
```
A=sym([3,1,2;4,3,2;12,11,10])
A =
[  3,  1,  2]
[  4,  3,  2]
[ 12, 11, 10]

>> Q1=sym([3/5,4/5,0;-4/5,3/5,0;0,0,1])
Q1 =
[  3/5, 4/5, 0]
[ -4/5, 3/5, 0]
[    0,   0, 1]

>> R1=Q1*A
R1 =
[  5,  3, 14/5]
[  0,  1, -2/5]
[ 12, 11,   10]

>> Q2=sym([5/13,0,12/13;0,1,0;-12/13,0,5/13])
Q2 =
[   5/13, 0, 12/13]
[      0, 1,     0]
[ -12/13, 0,  5/13]

>> R2=Q2*R1
R2 =
[ 13, 147/13, 134/13]
[  0,      1,   -2/5]
[  0,  19/13,  82/65]

>> r=sym(sqrt(1^2+(19/13)^2)); c=1/r; s = 19/13/r; Q3=sym([1,0,0;0,c,s;0,-s,c])
Q3 =
[ 1,                    0,                   0]
[ 0,  (13*530^(1/2))/530, (19*530^(1/2))/530]
[ 0, -(19*530^(1/2))/530, (13*530^(1/2))/530]

>> R=Q3*R2
R =
[ 13,       147/13,               134/13]
[  0, 530^(1/2)/13, (122*530^(1/2))/3445]
[  0,            0,   (12*530^(1/2))/265]

>> eval(R)
ans =
   13.0000   11.3077   10.3077
         0    1.7709    0.8153
         0         0    1.0425
```

Het is dus middels Givens rotaties mogelijk om een matrix A in bovendriehoeksvorm te brengen. Dat geldt zelfs voor niet-kwadratische matrices A, wat nog zeer belangrijk gaat worden voor least-squares problemen. We hebben dus de volgende stelling:

Stelling 2. *Stel dat $A \in \mathbb{R}^{m\times n}$. Dan bestaat er een orthogonale matrix $Q \in \mathbb{R}^{m\times m}$ en een bovendriehoeksmatrix $U \in \mathbb{R}^{m\times n}$ zodanig dat*

$$A = QU.$$

Een bovendriehoeksmatrices U is gedefinieerd door $U_{ij} = 0$ indien $i > j$. Dus, voor $m > n$ ziet ze er

zo uit:

$$U = \begin{pmatrix} u_{11} & u_{12} & \dots & \dots & \dots & u_{1n} \\ 0 & u_{22} & u_{23} & \dots & \dots & u_{2n} \\ 0 & 0 & u_{33} & u_{34} & \dots & u_{3n} \\ 0 & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & \dots & \dots & u_{nn} \\ 0 & 0 & \dots & \dots & \dots & 0 \\ 0 & 0 & \dots & \dots & \dots & 0 \\ 0 & 0 & \dots & \dots & \dots & 0 \end{pmatrix}.$$

1.3 Lineaire iteratieve methodes

In dit hoofdstuk bespreken wij de oplossing van Prob. 1 via iteratieve methodes. Matrices die in de praktijk voorkomen zijn vaak *ijl* (ook *dunbezette* of *schaarse* matrix genoemd, Engels: *sparse*), dat wil zeggen dat $A_{ij} = 0$ voor de meeste i, j. Een klassiek voorbeeld is de discretisatie van $-u_{xx} = f$ met behulp van eindige differenties. Deze levert een matrix $A \in \mathbb{R}^{n\times n}$ met $A_{ii} = \frac{2}{\Delta x^2}$ en $A_{i,i\pm1} = -\frac{1}{\Delta x^2}$. Dus per lijn (met n elementen) zijn er slechts drie elementen verschillend van nul. Dit impliceert dat een matrix-vector multiplicatie grofweg $3n$ berekeningen nodig heeft in plaats van n^2. Voor een groot n maakt dat natuurlijk een verschil en we willen gebruik maken van dit verschil om oplossingen van vergelijkingen sneller (of in vele gevallen zelfs mogelijk) te maken. Merk op dat de LU decompositie voor een ijle matrix niet noodzakelijk ijl is, zie `ijl.m`. Tijdens de zelfstudie gaan we er dieper op in hoe zo'n ijle matrix eruit ziet.

In dit hoofdstuk zullen we het hebben over iteratieve methoden voor de benadering van $\overline{x} \in \mathbb{R}^n$, waar

$$A\overline{x} = b,$$

met $b \in \mathbb{R}^n$ en $A \in \mathbb{R}^{n\times n}$ inverteerbaar. In tegenstelling tot wat we tot nu toe hebben gezien, leveren deze methoden in het algemeen nooit de exacte oplossing, maar (hopelijk) een benadering ervan. We beschouwen dus, voor een gegeven ψ, de iteratieve methode

$$x_{k+1} := \psi(x_k), \quad k \geq 0 \tag{6}$$

met fout

$$e_k := x_k - \overline{x}.$$

Definitie 4. *De iteratieve methode* (6) *is* lineair*, indien er een iteratiematrix C bestaat zodanig dat*

$$e_{k+1} = Ce_k, \quad k \geq 0.$$

Omdat $e_k = Ce_{k-1} = \ldots = C^k e_0$ geldt ook:

Lemma 8. *De lineaire iteratieve methode* (6) *convergeert voor elk $x_0 \in \mathbb{R}^n$ als en slechts als $C^k \to 0$, dus*

$$\lim_{k\to\infty} e_k = 0, \ \forall e_0 \in \mathbb{R}^n \qquad \Leftrightarrow \qquad \lim_{k\to\infty} C^k = 0.$$

We moeten dus de eigenschap $C^k \to 0$ bestuderen. Zonder bewijs het volgende lemma:

Lemma 9. *$\forall B \in \mathbb{R}^{n\times n}$, $\forall \varepsilon > 0$, bestaat er een matrix norm $\|\cdot\|_\#$ zodanig dat*

$$\|B\|_\# \leq \rho(B) + \varepsilon,$$

met $\rho(B) := \max\{|\lambda| \mid \lambda \in \mathbb{C}$ eigenwaarde van $B\}$.

Het voorafgaande lemma wordt gebruikt om het volgende lemma te bewijzen en een criterium te krijgen voor een convergente iteratieve methode.

Lemma 10 (Stabile matrices)**.** *Stel dat $B \in \mathbb{R}^{n\times n}$. Dan geldt dat*

$$\lim_{k\to\infty} B^k = 0 \qquad \Leftrightarrow \qquad \rho(B) < 1.$$

Een matrix B met $\rho(B) < 1$ wordt stabile matrix *genoemd.*

Bewijs. "$\Leftarrow$" We beschouwen een $\varepsilon > 0$ zodanig dat $\rho(B) + \varepsilon < 1$ en gebruiken de norm van La. 9. Er geldt:

$$\|B^k\|_\# \le \|B\|_\#^k \le (\rho(B) + \varepsilon)^k \to 0, \quad k \to \infty.$$

Vervolgens geldt $\lim_{k\to\infty} \|B^k\|_\# \to 0$ en dus $\lim_{k\to\infty} B^k = 0$.

"$\Rightarrow$" Definieer

$$\sigma(B) := \{\lambda \mid \lambda \in \mathbb{C} \text{ eigenwaarde van } B\}. \tag{7}$$

Voor elke $\lambda \in \sigma(B)$ bestaat er een $0 \neq v \in \mathbb{C}^n$ zodanig dat $Bv = \lambda v$. We kiezen λ zodanig dat $|\lambda| = \rho(B)$. Er geldt

$$\|B\|_{\infty,\mathbb{C}} := \max_{0\neq x\in\mathbb{C}^n} \frac{\|Bx\|_\infty}{\|x\|_\infty} \ge \frac{\|\lambda v\|_\infty}{\|v\|_\infty} = \rho(B).$$

Omdat $B^k \to 0$ geldt uiteraard ook $\|B^k\|_{\infty,\mathbb{C}} \to 0$. Dus

$$\rho(B)^k = \rho(B^k) \le \|B^k\|_{\infty,\mathbb{C}} \to 0.$$

Omdat $\rho(B)^k \to 0$ moet $\rho(B) < 1$ zijn.

□

Stelling 3. *Een lineaire iteratieve methode is convergent als en slechts als $\rho(C) < 1$, met $C \in \mathbb{R}^{n\times n}$ de iteratiematrix.*

We beschouwen nu enkele voorbeelden, die allemaal op een splitsing

$$A = M - N$$

gebaseerd zijn. Merk op dat $A\overline{x} = b$ vervolgens equivalent is met

$$M\overline{x} = N\overline{x} + b.$$

We maken de twee belangrijke aannamen:

- De matrix M is niet-singulier.
- De vergelijking $Mx = y$ kan voor elke $y \in \mathbb{R}^n$ 'eenvoudig' opgelost worden.

Vervolgens kan een iteratieve methode aangegeven worden door

$$Mx_{k+1} = Nx_k + b, \quad k \ge 0$$

voor een gegeven $x_0 \in \mathbb{R}^n$. Dat is een lineaire iterative methode omdat (merk op: $A\overline{x} = b$)

$$\begin{aligned} x_{k+1} - \overline{x} &= M^{-1}\left(Nx_k + b - M\overline{x}\right) = M^{-1}\left(Nx_k + A\overline{x} - M\overline{x}\right) \\ &= M^{-1}\left(-Ax_k + Mx_k + A\overline{x} - M\overline{x}\right) = M^{-1}(M - A)(x_k - \overline{x}) \end{aligned}$$

$C := \mathrm{Id} - M^{-1}A = M^{-1}N$ is dus de iteratiematrix.
We beschouwen enkele voorbeelden:

Voorbeeld 2. • *(Methode van Richardson) We splitsen A op in $M = \frac{1}{\omega}\mathrm{Id}$ en $N = \frac{1}{\omega}\mathrm{Id} - A$. $\omega \in \mathbb{R}^{\neq 0}$ is een parameter die men kan kiezen. De iteratie is heel eenvoudig:*

$$x_{k+1} = x_k - \omega(Ax_k - b).$$

- *(Methode van Jacobi) We splitsen A op in*

$$A = D - L - U,$$

waar $D_{ii} = A_{ii}\ \forall i = 1, \dots n$, $L_{ij} = -A_{ij}, i > j$ en $U_{ij} = -A_{ij}, i < j$. We moeten aannemen dat $\det(D) \neq 0$ en zetten vervolgens $M = D$, $N = U + L$. De iteratie is

$$x_{k+1} = D^{-1}(L + U)x_k + D^{-1}b.$$

(Merk op: D^{-1} is eenvoudig te bepalen.) Er bestaat ook een gedempte versie met $M := \frac{1}{\omega}D$, $N = M - A$. (Ze heet gedempt omdat men meestal $\omega \leq 1$ kiest.)

- *(Methode van Gauß-Seidel) We splitsen $A = D - L - U$ (zie boven) en zetten $M = D - L$ en $N = U$. Een stelsel lineaire vergelijkingen $Mx = y$ is eenvoudig op te lossen omdat M een benedendriehoeksmatrix is. De methode is gegeven door*

$$(D - L)x_{k+1} = Ux_k + b,$$

of, in index-notatie:

$$a_{ii}(x_i)_{k+1} = -\sum_{j<i} a_{ij}(x_j)_{k+1} - \sum_{j>i} a_{ij}(x_j)_k + b_i.$$

We beschouwen de convergentie van de gedempte methode van Jacobi.

Lemma 11. *Stel dat $B \in \mathbb{R}^{n\times n}$ een symmetrisch positief definiete matrix is. We noemen grootste en kleinste eigenwaarde $\lambda_{\max}(B)$ en $\lambda_{\min}(B)$, respectivelijk. Er geldt:*

$$\rho(\mathrm{Id} - \omega B) < 1 \qquad \Leftrightarrow \qquad 0 < \omega < \frac{2}{\lambda_{\max}(B)}. \tag{8}$$

en voor $w_{opt} := \frac{2}{\lambda_{\min}(B)+\lambda_{\max}(B)}$

$$\min_{\omega\in\mathbb{R}} \rho(\mathrm{Id} - \omega B) = \rho(\mathrm{Id} - w_{opt}B) = 1 - \frac{2}{\kappa_2(B) + 1}.$$

Bewijs. We noemen de verzameling van eigenwaarden van een matrix B $\sigma(B)$, zie (7). Er geldt

$$\sigma(\mathrm{Id} - \omega B) = \{1 - \omega\lambda \mid \lambda \in \sigma(B)\}.$$

Bijgevolg is

$$\rho(Id - \omega B) = \max\{|1 - \omega\lambda_{\min}(B)|, |1 - \omega\lambda_{\max}(B)|\} = \begin{cases} 1 - \omega\lambda_{\max}(B), & \omega \leq 0, \\ 1 - \omega\lambda_{\min}(B), & 0 \leq \omega \leq \omega_{opt} \\ \omega\lambda_{\max}(B) - 1, & \omega_{opt} \leq \omega \end{cases}.$$

Bijgevolg is $\rho(\mathrm{Id} - \omega B) < 1$ als en slechts als $\omega > 0$ en $\omega\lambda_{\max}(B) - 1 < 1$. Dat levert (8). Bovendien geldt nog

$$\min_{\omega} \rho(\mathrm{Id} - \omega B) = 1 - \omega_{opt}\lambda_{\min}(B) = 1 - \frac{2\lambda_{\min}(B)}{\lambda_{\min}(B) + \lambda_{\max}(B)} = 1 - \frac{2}{\kappa_2(B) + 1}.$$

(Hier hebben we gebruik gemaakt van (5).) □

Hieruit volgt meteen de volgende stelling:

Stelling 4. *Stel dat A symmetrisch positief definiet is. De iteratiematrix van de gedempte Jacobi methode wordt gegeven door*

$$C_\omega := \mathrm{Id} - \omega D^{-1} A.$$

Indien $0 < \omega < \frac{2}{\lambda_{\max}(D^{-1}A)}$ geldt $\rho(C_\omega) < 1$ en de gedempte Jacobie methode is convergent.

Bewijs. Merk op dat $B := D^{-1/2} A D^{-1/2}$ een symmetrisch positief definiete matrix is. Nu pas La. 11 op B toe. Merk op dat $\sigma(B) = \sigma(D^{-1}A)$ en $\sigma(\mathrm{Id} - \omega B) = \sigma(\mathrm{Id} - \omega D^{-1} A)$. □

1.4 Niet-lineaire iteratieve technieken voor lineaire stelsels

Om de behandeling van lineaire stelsels af te ronden beschouwen wij in deze sectie nog twee belangrijke niet-lineaire iteratieve technieken om een vergelijking $Ax = b$ op te lossen. Vanaf nu beschouwen wij een symmetrisch positief definiete (spd) matrix $A \in \mathbb{R}^{n \times n}$.

We zullen hier technieken uit de optimalisatie gebruiken. Dat mag omwille van het volgende lemma:

Lemma 12. *Definieer $F : \mathbb{R}^n \to \mathbb{R}$ voor een zekere $A \in \mathbb{R}^{n \times n}$, A spd, door*

$$F(x) := \frac{1}{2}(x, Ax) - (x, b). \tag{9}$$

F bezit een uniek minimum in een punt $\overline{x} \in \mathbb{R}^n$ en

$$F(\overline{x}) = \min_{x \in \mathbb{R}^n} F(x) \quad \Leftrightarrow A\overline{x} = b.$$

Opmerking 3. *Merk op: $(x, y) := x^T y$.*

Bewijs. Definieer $\overline{x} := A^{-1} b$ en merk op dat

$$F(x) = \frac{1}{2}(x - \overline{x}, A(x - \overline{x})) + c \tag{10}$$

met $c = -\frac{1}{2}\overline{x}^T A \overline{x}$ onafhankelijk van x. Omdat $y^T A y > 0$ voor elke $y \in \mathbb{R}^n \setminus \{0\}$ is het gemakkelijk te zien dat $\overline{x}$ ook een minimum voorstelt en dat dit mimimum ook een oplossing van $A\overline{x} = b$ is. □

Het idee van een eerste methode - de *steepest descent methode* (*methode van de steilste helling* of *methode der steilste afdaling*) - is nu, gebaseerd op een eerste schatting $x_0 \in \mathbb{R}^n$ de steilste richting te volgen en daar het minimum te vinden.

Lemma 13. *Stel dat $F : \mathbb{R}^n \to \mathbb{R}^n$ een continu afleidbare functie is. Er geldt:*

$$v^* := \operatorname*{argmax}_{v \in \mathbb{R}^n, \|v\|_2 = 1} F'(x) v = \frac{F'(x)^T}{\|F'(x)\|_2}.$$

Bewijs. Eenerzijds: $|F'(x)v^*| = \frac{\|F'(x)\|_2^2}{\|F'(x)\|_2} = \|F'(x)\|_2$. Anderzijds: voor een $v \in \mathbb{R}^n$ met $\|v\|_2 = 1$ geldt

$$|F'(x)v| \leq \|F'(x)\|_2 \|v\|_2 = \|F'(x)\|_2.$$

Het maximum wordt door v^* aangenomen. □

Opmerking 4. *Merk op dat voor F gegeven in (9) geldt: $F'(x)^T = Ax - b$.*

We beschouwen nu het volgende algoritme:

Algoritme 1 (Methode van de steilste helling)**.** *Gegeven $x_0 \in \mathbb{R}^n$ en een spd matrix A. Stel dat F gegeven is als in* (9)*. Voor $k \geq 0$ definieer:*

$$r_k := F'(x_k)^T, \qquad t_k := \operatorname*{argmin}_{t \in \mathbb{R}} F(x_k + t \cdot r_k),$$

en vervolgens

$$x_{k+1} := x_k + t_k \cdot r_k.$$

Opmerking 5. *Het minimum van de functie $F(x_k + t \cdot r_k)$ is uniek en er geldt $t_k = \frac{-r_k^T r_k}{r_k^T A r_k}$. ($t_k = 0$ indien $r_k = 0$.) De methode is dus goed gedefinieerd voor elke $x_0 \in \mathbb{R}^n$.*

Zonder bewijs nog een stelling over de convergentiesnelheid van de methode van de steilste helling:

Stelling 5. *Stel dat A symmetrisch positief definiet is. Dan convergeert de rij $(x_k)_{k \in \mathbb{N}}$, gedefinieerd in Alg. 1 voor elke $x_0 \in \mathbb{R}^n$ naar $\overline{x}$. Voor $e_k := x_k - \overline{x}$ geldt*

$$\|e_k\|_A \leq \left(\frac{\kappa_2(A) - 1}{\kappa_2(A) + 1} \right)^k \|e_0\|_A$$

(Merk op dat $\|e_k\|_A^2 := (e_k, Ae_k)$ een norm definieert omdat A symmetrisch positief definiet is.)

Een nadeel van de methode van de steilste helling is dat de optimalisatie procedure in stap k 'vergeet' dat er ook andere stappen $x_0, \ldots, x_{k-1}$ waren. We beschouwen nu de methode van de *geconjugeerde gradiënten* (CG). Om het wat gemakkelijker te houden veronderstellen we dat $x_0 = 0 \in \mathbb{R}^n$.

Opmerking 6. *De aanname $x_0 = 0$ is geen echte beperking. Om met een arbitraire $x_0 \in \mathbb{R}^n$ te kunnen werken beschouwt men het systeem*

$$A\widetilde{x} = \widetilde{b}$$

met $\widetilde{x} := \overline{x} - x_0$ en $\widetilde{b} := b - Ax_0$.

We beginnen op dezelfde manier als voor de methode van de steilste helling en zetten de eerste 'richting' gelijk aan $r_0 := F'(x_0)^T = -b$. We definiëren $U_0 := \operatorname{span}\{b\}$. We zoeken vervolgens een x_1 zodanig dat

$$x_1 = \operatorname*{argmin}_{x \in U_0} F(x). \tag{11}$$

Deze x_1 komt overeen met de x_1 die in Algoritme 1 berekend werd. We kunnen dit echter wat anders schrijven (zie (10)):

$$\operatorname*{argmin}_{x \in U_0} F(x) = \operatorname*{argmin}_{x \in U_0} \left(\frac{1}{2} (x - \overline{x}, A(x - \overline{x})) + c \right) = \operatorname*{argmin}_{x \in U_0} \|x - \overline{x}\|_A^2, \tag{12}$$

waar we de 'energie-norm' $\| \cdot \|_A$ met

$$\|x\|_A^2 := (x, x)_A, \qquad (x, x)_A := (x, Ax),$$

gebruikt hebben. De rechterzijde van (12) geeft de motivatie om de methode van de steilste helling te verbeteren. Nu hebben we via (11) een x_1 berekent. We volgen de methode van de steilste helling en definiëren een nieuwe richting $r_1 := Ax_1 - b$ en definiëren $U_1 := U_0 \oplus \operatorname{span}\{Ax_1 - b\}$. In tegenstelling tot algoritme 1 berekenen wij nu het minimum niet via de eendimensionale optimisatie, maar via

$$x_2 = \operatorname*{argmin}_{x \in U_1} \|x - \overline{x}\|_A.$$

Over het algemeen verschilt x_2 van deze methode nu echt van de x_2 van de methode van de steilste helling.

We kunnen nu een eerste versie van de methode van de geconjugeerde gradiënten aangeven:

Algoritme 2 (Geconjugeerde gradiënten methode – versie 1)**.** *Voor $k \in \mathbb{N}^{\geq 0}$ zijn de x_{k+1} gedefinieerd door*

$$x_{k+1} = \operatorname*{argmin}_{x \in U_k} \|x - \overline{x}\|_A, \tag{13}$$

met

$$U_k := \operatorname{span}\{b, Ax_1 - b, Ax_2 - b, \ldots, Ax_k - b\}. \tag{14}$$

Het is tot nu toe nog niet duidelijk hoe men de x_k in (13) efficient kan berekenen. We maken nu de aanname dat U_k gegeven is door

$$U_k = \operatorname{span}\{p_0, \ldots, p_k\} \tag{15}$$

met $(p_i, p_j)_A = 0$ voor $i \neq j$. We voeren de volgende definitie in:

Definitie 5. *We zeggen dat twee vectoren $x \in \mathbb{R}^n$ en $y \in \mathbb{R}^n$* geconjugeerde richtingen *zijn indien $(x, y)_A = 0$.*

Opmerking 7. *De methode van de steilste helling wordt vaak ook* gradiëntenmethode *genoemd. De richtingen (dus de gradiënten) zijn 'onafhankelijk' van elkaar. De verbetering van de CG methode is nu dat we richtingen gaan kiezen die gebaseerd zijn op de gradiënt in stap k en op de gradiënten in stappen j, $j < k$.*

Indien U_k van vorm (15) is kunnen wij x_{k+1} gemakkelijk aangeven door

$$x_{k+1} = \sum_{j=0}^{k} \frac{(p_j, \overline{x})_A}{(p_j, p_j)_A} p_j. \tag{16}$$

(Waarom?) Merk op: We kunnen $(p_j, \overline{x})_A = (p_j, b)$ berekenen zonder $\overline{x}$ expliciet te kennen! Een consequentie uit (16) is

$$x_{k+1} = \sum_{j=0}^{k-1} \frac{(p_j, \overline{x})_A}{(p_j, p_j)_A} p_j + \frac{(p_k, \overline{x})_A}{(p_k, p_k)_A} p_k =: x_k + \alpha_k p_k. \tag{17}$$

Het moeilijke deel is dus de berekening van de p_k. Stel dat we U_{k-1} al hebben berekend. U_k is bijgevolg

$$U_k = \operatorname{span}\{p_0, \ldots, p_{k-1}, Ax_k - b\}.$$

De bedoeling is nu een p_k te vinden zodanig dat $\operatorname{span}\{p_0, \ldots, p_k\} = U_k$ en $(p_k, p_j)_A = 0$ voor $j \leq k-1$. Ook die is gemakkelijk aan te geven omdat een goede keuze

$$p_k := Ax_k - b - \sum_{j=0}^{k-1} \frac{(p_j, Ax_k - b)_A}{(p_j, p_j)_A} p_j \tag{18}$$

is. Men moet echter nog heel veel inproducten berekenen. Het succes van de CG methode ligt in de volgende stelling:

Stelling 6. *Stel dat x_k berekend wordt via Algoritme 2 en de p_k via* (18)*. Stel bovendien dat $Ax_k - b \neq 0$ voor elke k. (Indien $Ax_k - b = 0$ geldt sowieso $x_k = \overline{x}$.) Er geldt:*

$$(Ax_k - b, p_j)_A = 0, \qquad 0 \leq j \leq k-2.$$

Bewijs. We beschouwen een zekere $k \geq 2$. Wegens de definitie van x_k geldt

$$x_k - \overline{x} \perp_A U_{k-1} \qquad \Rightarrow \qquad Ax_k - b \perp U_{k-1}$$

en dus

$$Ax_k - b \perp Ax_j - b, \qquad 0 \leq j \leq k-1. \tag{19}$$

Omdat we verondersteld hebben dat $Ax_k - b \neq 0$ moet dus $Ax_k - b \neq Ax_j - b$ zijn en daarom $x_k \neq x_j$. Bijgevolg geldt $\alpha_j \neq 0$, $j \leq k-1$, en we kunnen (17) ook schrijven als

$$Ax_{j+1} - b = Ax_j - b + \alpha_j Ap_j. \tag{20}$$

Vervolgens geldt voor $j \leq k-2$:

$$\begin{aligned}(Ax_k - b, p_j)_A = (Ax_k - b, Ap_j) &\overset{(20)}{=} \left(Ax_k - b, \frac{1}{\alpha_j}\left(Ax_{j+1} - b - (Ax_j - b)\right)\right) \\ &= \frac{1}{\alpha_j}\left((Ax_k - b, Ax_{j+1} - b) - (Ax_k - b, Ax_j - b)\right) \overset{(19)}{=} 0.\end{aligned}$$

□

Stelling 6 maakt de berekening van bijna alle formules veel eenvoudiger, bijvoorbeeld wordt uit (18)

$$p_k = Ax_k - b - \frac{(p_{k-1}, Ax_k - b)_A}{(p_{k-1}, p_{k-1})_A} p_{k-1}. \tag{21}$$

Algoritme 3 (Geconjugeerde gradiënten methode – versie 2)**.** *Bereken de p_k via* (21) *en dan x_k via* (16)*. Indien $Ax_k - b = 0$ stop.*

2 Kleinste-kwadratenmethode

Vele praktische toepassingen leiden tot een overbepaald lineair stelsel van vergelijkingen.

Voorbeeld 3. *We beschouwen een natuurkundig experiment. Een massa m aan een veer (zie Fig. 1) - we veronderstellen dat er geen demping plaats vindt - beschrijft een harmonische oscillatie. De frequentie T, dus de duur van een op-en-neer gaande beweging, is afhankelijk van de massa m en de veerconstante D via*

$$T = 2\pi\sqrt{\frac{m}{D}},$$

Gegeven zijn (echte, ik heb die niet verzonnen!) waarden van een experiment:

i	1	2	3	4	5	6	7
m_i *in kg*	0.02	0.04	0.06	0.08	0.10	0.12	0.14
T_i *in s*	0.6034	0.863	0.925	1.063	1.175	1.166	1.375

Hoe kan men D bepalen? Idee: Men schrijft gewoon

$$0.6034 = 2\pi\sqrt{\frac{0.02}{D}}, \quad 0.863{=}2\pi\sqrt{\frac{0.04}{D}}, \quad 0.925{=}2\pi\sqrt{\frac{0.06}{D}}, \quad 1.063{=}2\pi\sqrt{\frac{0.08}{D}},$$
$$1.175 = 2\pi\sqrt{\frac{0.10}{D}}, \quad 1.166{=}2\pi\sqrt{\frac{0.12}{D}}, \quad 1.375{=}2\pi\sqrt{\frac{0.14}{D}}.$$

Dat is een (niet-lineair) stelsel van vergelijkingen zonder oplossing. We kunnen het ook schrijven als $F(x) = 0$, voor de functie $F : \mathbb{R} \to \mathbb{R}^7$ met $F_i(x) := T_i - 2\pi\sqrt{m_i}x$, met $x := \sqrt{D^{-1}}$. (Zo hebben we het probleem lineair gemaakt, F is dus van vorm $F(x) = Ax - b$ voor $A \in \mathbb{R}^{7\times 1}$.) Omdat meten alleen met een fout kan heeft $F(x) = 0$ geen oplossing. Een idee van Gauß was nu om niet $F(x) = 0$ op te lossen, maar om een x te vinden zodanig dat $\|F(x)\|_2^2 = \sum_i F_i(x)^2$ zo klein mogelijk wordt. Dat levert meteen het volgende probleem:

Probleem 2. *Gegeven een $A \in \mathbb{R}^{m\times n}$ met* $\operatorname{rang}(A) = n$ *(dus $m \geq n$) en een $b \in \mathbb{R}^m$. Zoek een $\overline{x} \in \mathbb{R}^n$ zodanig dat*

$$\|A\overline{x} - b\|_2 = \min_{x\in\mathbb{R}^n} \|Ax - b\|_2. \tag{22}$$

Voor dit optimalisatieprobleem bestaat er een (unieke) oplossing $\overline{x}$:

Stelling 7. *De unieke oplossing van Probleem 2 is gegeven door $\overline{x}$, waar $\overline{x}$ voldoet aan*

$$A^T A\overline{x} = A^T b. \tag{23}$$

Deze vergelijking wordt normaalvergelijking *genoemd.*

Figuur 1: Een massa m aan een veer beschrijft een harmonische oscillatie.

Bewijs. De matrix A^TA is symmetrisch (dat is duidelijk) en positief definiet, omdat

$$x^TA^TAx = (Ax, Ax) = \|Ax\|_2^2.$$

A heeft volledige rang en dus geldt $Ax = 0$ alleen voor $x = 0$.

Het probleem (22) verandert niet als we het kwadraat beschouwen en met $\frac{1}{2}$ vermeningvuldigen, dus we beschouwen $F(x) := \frac{1}{2}\|Ax - b\|^2 \to \min$. We kunnen F ook schrijven als

$$F(x) = \frac{1}{2}(Ax - b, Ax - b) = \frac{1}{2}(Ax, Ax) - (Ax, b) + \frac{1}{2}(b, b) = \frac{1}{2}(x, A^TAx) - (x, A^Tb) + \frac{\|b\|_2^2}{2}.$$

Dat is van vorm $F(x) = \frac{1}{2}(x, Bx) - (x, \widetilde{b}) + c$ (met $B = A^TA$ en $\widetilde{b} = A^Tb$, zie ook (9)), we weten dus dat het minimum gelijk is aan $\overline{x}$ met

$$A^TA\overline{x} = A^Tb.$$

□

Opmerking 8. *We weten nu dat de oplossing van* $\|Ax - b\|_2 \to \min$ *uniek is. Dit resultaat is niet noodzakelijk waar voor andere normen! (Oefening.)*

Om de conditie van (22) te beschouwen hebben we het conditiegetal $\kappa_2(A)$ nodig. Het probleem is echter dat Def. 2 niet meer van toepassing is, omdat A^{-1} niet bestaat. We moeten de definitie dus veralgemenen:

Definitie 6. *Voor een matrix* $A \in \mathbb{R}^{m\times n}$ *met* $\operatorname{rang}(A) = n$ *definiëren wij*

$$\kappa_2(A) := \frac{\max_{x\neq 0} \frac{\|Ax\|_2}{\|x\|_2}}{\min_{x\neq 0} \frac{\|Ax\|_2}{\|x\|_2}} \tag{24}$$

Lemma 14. *Voor* A *inverteerbaar (dus* $m = n$*) zijn Def. 2 en Def. 6 equivalent.*

Bewijs. Oefening. □

Lemma 15 (Conditie van het kleinste-kwadraten probleem)**.** *Stel dat* $\overline{x}$ *de oplossing is van* $\|Ax - b\|_2 \to \min$*, en* $\overline{x} + \Delta x$ *de oplossing van* $\|A(x + \Delta x) - (b + \Delta b)\|_2 \to \min$*. Stel bovendien dat* θ *gekozen werd zodanig dat* $\cos(\theta) = \frac{\|A\overline{x}\|_2}{\|b\|_2}$*. Er geldt:*

$$\frac{\|\Delta x\|_2}{\|\overline{x}\|_2} \leq \frac{\kappa_2(A)}{\cos(\theta)} \frac{\|\Delta b\|_2}{\|b\|_2}. \tag{25}$$

Bewijs. Oefening. □

Wat zijn nu de methoden om een kleinste-kwadraten-probleem op te lossen? Het meest voor de hand liggende is om de normaalvergelijking (23) te gebruiken en dan het lineaire stelsel met een van de methoden uit het voorafgaande hoofdstuk op te lossen. Dat is in de meeste gevallen niet geschikt, omdat het vrij duur is de matrix A^TA te berekenen en omdat

$$\kappa_2(A^TA) = \kappa_2(A)^2.$$

Vergelijk dat met het conditiegetal $\frac{\kappa_2(A)}{\cos(\theta)}$ van het probleem, zie (25); in de meeste gevallen is $\kappa_2(A)^2$ veel groter. (Er bestaan echter gevallen waar het toch een goed idee kan zijn om ze te gebruiken, bv. als A een zekere struktuur heeft die A^TA eenvoudig maakt. De oplossing kan dan via Cholesky-decompositie, omdat de matrix A^TA altijd spd is.)

Om (22) op te lossen maken wij gebruik van de QU decompositie. Stel dat we al weten dat $A = QU$ met $U = \begin{pmatrix} \widetilde{U} \\ \mathbf{0} \end{pmatrix}$, $\widetilde{U} \in \mathbb{R}^{n\times n}$ en $\mathbf{0} \in \mathbb{R}^{(m-n)\times n}$. De matrix $\widetilde{U}$ is inverteerbaar omdat we geëist hebben dat de rang van A gelijk is aan n. Vervolgens geldt

$$\|Ax - b\|_2 = \|QUx - b\|_2 = \|Q(Ux - Q^T b)\|_2 = \|Ux - Q^T b\|,$$

omdat $\|Qy\|_2 = \|y\|_2$ voor elke $y \in \mathbb{R}^m$. We kunnen dus het probleem $\|Ux - \widetilde{b}\|_2 \to \min$ met $\widetilde{b} := Q^T b$ beschouwen. Dit probleem is van vorm

$$\left\| \begin{pmatrix} \widetilde{U} \\ \mathbf{0} \end{pmatrix} x - \begin{pmatrix} \widetilde{b}_1 \\ \widetilde{b}_2 \end{pmatrix} \right\| \to \min .$$

Merk op dat

$$\left\| \begin{pmatrix} \widetilde{U} \\ \mathbf{0} \end{pmatrix} x - \begin{pmatrix} \widetilde{b}_1 \\ \widetilde{b}_2 \end{pmatrix} \right\|^2 = \|\widetilde{U}x - \widetilde{b}_1\|^2 + \|\widetilde{b}_2\|^2.$$

Uiteraard woord de uitdrukking minimaal voor $\overline{x} := \widetilde{U}^{-1}\widetilde{b}_1$, dus $\overline{x}$ is de oplossing van het stelsel lineaire vergelijkingen $\widetilde{U}\overline{x} = \widetilde{b}_1$. Het conditiegetal van dit stelsel is $\kappa_2(\widetilde{U}) = \kappa_2(U) = \kappa_2(A)$, normal is dat veel kleiner dan $\kappa_2(A)^2$. De oplossing kan gemakkelijk bepaald worden omdat U al in bovendriehoeksvorm gegeven is. De euclidische norm van $\widetilde{b}_2$ is het zogenoemde *residuum*, dus

$$\|A\overline{x} - b\|_2 = \|\widetilde{b}_2\|_2.$$

'Optimaal' is een waarde van nul: Dan waren we in staat om het stelsel exact op te lossen. In de praktijk komt nul echter nauwelijks voor.

De QU decompositie moet niet op voorhand berekend worden, ze kan in een algorithme geïntegreerd worden. We sluiten dit hoofdstuk af met een voorbeeld. De methode om Q en U te bepalen is via Givens rotaties.

Voorbeeld 4. *We beschouwen de matrix A uit voorbeeld 1 zonder de derde colom; en de vector $b := (1, 2, 3)^T$. Het idee is om de matrix $[A|b]$ via Givens rotaties op bovendriehoeksvorm te brengen.*

```
 >> A = [3 1; 4 3; 12 11]
A =
     3     1
     4     3
    12    11
>> b = [1; 2; 3]
b =
     1
     2
     3
>> Ab = [A b]
Ab =
     3     1     1
     4     3     2
    12    11     3
>> Q1=[3/5,4/5,0;-4/5,3/5,0;0,0,1];
>> Ab=Q1*Ab
Ab =
    5.0000    3.0000    2.2000
   -0.0000    1.0000    0.4000
   12.0000   11.0000    3.0000
>> Q2=[5/13,0,12/13;0,1,0;-12/13,0,5/13];
>> Ab = Q2 * Ab
```

```
Ab =
   13.0000   11.3077    3.6154
   -0.0000    1.0000    0.4000
         0    1.4615   -0.8769
>> r=sqrt(1^2+(19/13)^2); c=1/r; s = 19/13/r; Q3=[1,0,0;0,c,s;0,-s,c];
>> Ab = Q3 * Ab
Ab =
   13.0000   11.3077    3.6154
   -0.0000    1.7709   -0.4979
    0.0000         0   -0.8253
>> U = Ab(1:2,1:2)
U =
   13.0000   11.3077
         0    1.7709
>> b1 = Ab(1:2,3)
b1 =
    3.6154
   -0.4979
>> x = U\b1
x =
    0.5226
   -0.2811
```

We moeten de Q matrix niet opslaan. Zelf de berekening via matrix-matrix-multiplicaties is veel te ingewikkeld. De Givens matrices zijn zo eenvoudig opgebouwd, dat men alleen rijen moet opsommen.

3 Gauss kwadratuur

We definiëren de vectorruimte van veeltermen tot en met graad n door

$$\Pi_n := \{f : \mathbb{R} \to \mathbb{R} \mid f \text{ is een veelterm van graad maximaal } n\}.$$

In dit hoofdstuk beschouwen wij de integraal

$$\mathcal{I}_\omega(f) := \int_{-1}^{1} \omega(x) f(x) \mathrm{d}x,$$

met een gladde positieve gewichtsfunctie $\omega : [-1, 1] \to \mathbb{R}^{>0}$. Ons doel is om $\mathcal{I}_\omega$ zo nauwkeurig mogelijk door de formule

$$\mathcal{I}_\omega(f) \approx \sum_{i=0}^{m} \alpha_i f(x_i) \tag{26}$$

te benaderen.

Definitie 7 (Nauwkeurigheidsgraad)**.** *De kwadratuurformule* (26) *is van* nauwkeurigheidsgraad n *indien er voor elk* $p \in \Pi_n$ *geldt:*

$$\mathcal{I}_\omega(p) = \sum_{i=0}^{m} \alpha_i p(x_i)$$

Bij de Newton-Cotes formules waren de x_i beperkt tot equidistant verdeelte punten. Die restrictie schrappen wij in het volgende en stellen ons de vraag: Welke nauwkeurigheidsgraad n is überhaupt mogelijk?

Lemma 16. *Een formule van type* (26) *kan maximale nauwkeurigheidsgraad van* $2m + 1$ *hebben.*

Bewijs. Beschouw de veelterm $p(x) := \prod_{i=0}^{m}(x - x_i)^2$ van graad $2m + 2$. Er geldt

$$\sum_{i=0}^{m} \alpha_i p(x_i) = 0 \neq \mathcal{I}_\omega(p),$$

omdat $\mathcal{I}_\omega(p) > 0$. □

Herhaal de definitie van de Lagrange veeltermen behorende tot $x_0, \ldots, x_m$,

$$l_{im}(x) := \prod_{j=0, j\neq i}^{m} \frac{x - x_j}{x_i - x_j}.$$

Om minstens van nauwkeurigheidsgraad m te zijn moet

$$\alpha_i = \mathcal{I}_\omega(l_{im}), \qquad 0 \leq i \leq m, \tag{27}$$

gelden. We gaan vanaf nu ervan uit dat hieraan voldaan is.

Lemma 17. *Veronderstel dat* (27) *geldt. De formule* (26) *bezit nauwkeurigheidsgraad* $2m + 1$ *als en slechts als* $q_{m+1}(x) := \Pi_{i=0}^{m}(x - x_i)$ *(een veelterm van graad* $m + 1$*)* $\omega-$*orthogonaal op veeltermen van graad* m *staat, d.w.z.*

$$\int_{-1}^{1} \omega(x) q_{m+1}(x) p(x) \mathrm{d}x = 0, \qquad \forall p \in \Pi_m.$$

Bewijs. Beschouw een veelterm p van graad Π_{2m+1}. Deze kan geschreven worden als

$$p(x) = q_{m+1}(x)p_1(x) + p_2(x),$$

met $p_1, p_2 \in \Pi_m$. Merk op dat $q_{m+1}(x_i) = 0$, $0 \leq i \leq m$. Bijgevolg:

$$\sum_{i=0}^{m} \alpha_i p(x_i) = \sum_{i=0}^{m} (q_{m+1}(x_i)p_1(x_i) + p_2(x_i)) = \sum_{i=0}^{m} p_2(x_i) = \mathcal{I}_\omega(p_2).$$

De laatste gelijkheid geldt omdat (27) geldt en bijgevolg de kwadratuurformule nauwkeurig is van graad m. Vervolgens:

$$\mathcal{I}_\omega(p) = \mathcal{I}_\omega(q_{m+1}p_1) + \mathcal{I}_\omega(p_2).$$

Nu kan $\sum_{i=0}^{m} \alpha_i p(x_i) = \mathcal{I}_\omega(p)$ voor elk p van graad $2m+1$ enkel gelden als en slechts als $\mathcal{I}_\omega(q_{m+1}p_1) = 0$. □

Opmerking 9. *Onder de voorwaarden van La. 17, dus de q_{m+1} zijnde een orthogonaal veelterm, zijn de α_i positief.*

Opmerking 10. *We kunnen op de ruimte van veeltermen een inproduct definiëren door*

$$(f, g) := \int_{-1}^{1} \omega(x)f(x)g(x)\mathrm{d}x.$$

De punten x_i zijn bijgevolg de nulpunten van een veelterm van graad $m+1$ die orthogonaal op alle veeltermen van graad n staat. Er kan bewezen worden dat die nulpunten bestaan, in $(-1, 1)$ liggen en eenvoudig zijn. Gebruikelijke ω zijn de volgende:

- $\omega(x) \equiv 1$. *In dit geval zijn de zogenoemde* Legendre *veeltermen een orthogonale basis. De eerste Legendre veeltermen zijn*

$$L_0(x) := 1, \qquad L_1(x) := x, \qquad L_2(x) := \frac{1}{2}\left(3x^2 - 1\right), \qquad L_3(x) := \frac{1}{2}\left(5x^3 - 3x\right), \qquad ... \tag{28}$$

- $\omega(x) \equiv \frac{1}{\sqrt{1-x^2}}$. *Hier vormen de Chebyshev veeltermen een orthogonale basis. We gaan hier op dit moment niet dieper op in, zie [**?**].*

Opmerking 11. *Zij $\omega \equiv 1$. De nulpunten van de Legendre veeltermen vanaf graad één, zie* (28)*, zijn $\{0\}$, $\{\pm\frac{1}{\sqrt{3}}\}$, $\{0, \pm\sqrt{\frac{3}{5}}\}$, ... We kunnen dus de volgende Gauss kwadraturen bepalen:*

1. $\mathcal{I}_\omega(f) \approx \alpha_0 f(0)$. $\alpha_0 = 2$ *om de functie $f(x) = 1$ exact te integreren.*
2. $\mathcal{I}_\omega(f) \approx \alpha_0 f\left(-\frac{1}{\sqrt{3}}\right) + \alpha_1 f\left(\frac{1}{\sqrt{3}}\right)$. *Er moet gelden $\alpha_0 + \alpha_1 = 2$ en $\alpha_0 - \alpha_1 = 0$, dus $\alpha_{0,1} = 1$.*
3. *... (Oefening!)*
4. *...*

Als een laatste stap bekijken wij nog een foutenformule voor de Gauss kwadratuur:

Stelling 8. *Zij $\omega \equiv 1$ en $f \in C^{2m+2}([-1, 1])$. Voor de Gauss kwadratuur* (26) *met α_i zoals in* (27) *en x_i zoals in La. 17 geldt:*

$$\left| \mathcal{I}_\omega(f) - \sum_{i=0}^{m} \alpha_i f(x_i) \right| \leq \frac{2^{2m+3}}{(2m+2)!} \left\| f^{(2m+2)} \right\|_\infty .$$

Opmerking 12. *Voor een functie* $f : [-1, 1] \to \mathbb{R}$ *is*

$$\|f\|_\infty := \sup_{x\in[-1,1]} |f(x)|.$$

Bewijs. We beschouwen de veelterm $q \in \Pi_{2m+1}$ gedefinieerd door

$$q(x_i) = f(x_i), \quad q'(x_i) = f'(x_i), \qquad 0 \le i \le m.$$

Uit 'Numerieke methoden 1' (zie ook [**?**, blz. 349]) weten wij dat, voor een $\xi \in [-1, 1]$,

$$|f(x) - q(x)| = \left| \frac{f^{(2m+2)}(\xi)}{(2m+2)!} \prod_{i=0}^{m} (x - x_i)^2 \right| \le \frac{2^{2m+2}}{(2m+2)!} \left\| f^{(2m+2)} \right\|_\infty .$$

Merk op dat er geldt:

$$\sum_{i=0}^{m} \alpha_i f(x_i) = \sum_{i=0}^{m} \alpha_i q(x_i) = \mathcal{I}_\omega(q),$$

de laatste gelijkheid omdat $q \in \Pi_{2m+1}$. Bijgevolg is

$$\begin{aligned} \left| \mathcal{I}_\omega(f) - \sum_{i=0}^{m} \alpha_i f(x_i) \right| &= |\mathcal{I}_\omega(f) - \mathcal{I}_\omega(q)| = |\mathcal{I}_\omega(f - q)| \\ &\le \int_{-1}^{1} \frac{2^{2m+2}}{(2m+2)!} \left\| f^{(2m+2)} \right\|_\infty \mathrm{d}x = \frac{2^{2m+3}}{(2m+2)!} \left\| f^{(2m+2)} \right\|_\infty . \end{aligned}$$

□

Opmerking 13. *De afschatting in Stl. 8 is niet optimaal, omdat ze geen gebruik maakt van eigenschappen van de Legendre veeltermen.*

Opmerking 14. *Indien er sprake is van een integratiegebied* $[c, d]$ *kan de transformatie*

$$x \mapsto 2\frac{x - \frac{c+d}{2}}{d - c}$$

gebruikt worden. Oefening: Hoe verandert de foutafschatting?

4 Gewone differentiaalvergelijkingen

4.1 Inleiding

Het laatste hoofdstuk in dit college zal het hebben over eerste-orde gewone differentiaalvergelijkingen

$$\begin{aligned} y'(t) &= f(t, y(t)), \quad \forall t \in [0, T], \\ y(0) &= y_0. \end{aligned} \tag{29}$$

$f : [0, T] \times \mathbb{R}^n \to \mathbb{R}^n$ en $y_0 \in \mathbb{R}^n$ zijn gegeven, de onbekende is de functie $y : [0, T] \to \mathbb{R}^n$.

Voorbeeld 5.

- *$y'(t) = f(t)$ heeft de oplossing $y(t) = y_0 + \int_0^t f(\tau)\mathrm{d}\tau$. Dit is zeer eenvoudig.*
- *Bijna even eenvoudig: $y'(t) = \lambda y(t)$ met oplossing $y(t) = e^{\lambda t} y_0$.*
- *Wat ingewikkelder:*

  $$y'(t) = y(t)(1 - y(t)).$$

  Om dit op te lossen maken we gebruik van scheiding der variabelen *(links y, rechts t; de formele redenering komt later in je studie):*

  $$\frac{y'(t)}{y(t)(1 - y(t))} = 1.$$

  Integratie ten opzichte van t levert

  $$\int \frac{y'(t)}{y(t)(1 - y(t))}\mathrm{d}t = \int 1\mathrm{d}t.$$

  De linke integraal is gelijk aan (substitutie $z = y(t)$!)

  $$\int \frac{1}{z(1 - z)}\mathrm{d}z = \ln(z) - \ln(1 - z) + c = \ln\left(\frac{z}{1 - z}\right) + c \stackrel{!}{=} \int 1\mathrm{d}t = t.$$

  Oplossen naar z en terugsubstitutie ($z = y(t)$):

  $$y(t) = \frac{1}{1 + e^{c - t}}.$$

  We vinden c door de beginconditie $y(0) = y_0$, dus

  $$c = \ln\left(\frac{1 - y_0}{y_0}\right).$$

  Uiteindelijk levert dit de oplossing

  $$y(t) = \frac{1}{1 + \frac{1 - y_0}{y_0} e^{-t}}.$$

Opmerking 15. *We beschouwen alleen maar eerste orde stelsels van gewone differentiaalvergelijkingen. Dat is echter geen beperking. Beschouw de algemene gewone differentiaalvergelijking*

$$y^{(m)} = g(t, y, y', y'', \ldots, y^{(m-1)})$$

met beginvoorwaarden voor y, $y', \ldots, y^{(m-1)}$. Definieer de hulpvariabelen $z_1 = y'$, $z_2 = y'' = z_1'$, …, $z_{m-1} = y^{(m-1)}$ en krijg het stelsel van eerste-orde gewone differentiaalvergelijkingen

$$y' = z_1, \quad z_1' = z_2, \quad \ldots \quad z_{m-2}' = z_{m-1}, \quad z_{m-1}' = g(t, y, z_1, z_2, \ldots, z_{m-1}).$$

Voorbeeld 6. *De dynamica van een slinger kan beschreven worden door de gewone differentiaalvergelijking*

$$y''(t) = -c\sin(y(t)), \qquad y(0) = y_0, \quad y'(0) = 0.$$

$c := \frac{g}{l}$ *voor* g *de zwaartekrachtsversnelling en* l *de slingerlengte.* y_0 *is de uitwijking van de slinger. We kunnen dit ook schrijven als*

$$y'(t) = z(t), \quad z'(t) = -c\sin(y(t)), \qquad y(0) = y_0, \quad z(0) = 0.$$

Om wat plaats te besparen zullen wij vaak y in plaats van $y(t)$ schrijven. y blijft echter een functie van tijd.

Unieke oplosbaarheid werd in het college 'Differentialen en differentiaalvergelijkingen' beschreven. Voor het gemak geven wij de volgende stelling weer:

Stelling 9 (Picard-Lindelöf)**.** *Stel dat er een* $a > 0$ *bestaat zodanig dat de functie* $f(t, y)$ *continu is op*

$$\mathcal{S} := \{(t, y) \mid 0 \leq t \leq a, \quad y \in \mathbb{R}^n\}.$$

Bovendien veronderstellen wij dat f*, ten opzichte van het tweede argument, Lipschitz-continu is, er geldt dus*

$$\|f(t, y) - f(t, z)\| \leq L\|y - z\| \qquad \forall (t, y), (t, z) \in \mathcal{S}.$$

Vervolgens bestaat er voor $t \in [0, a]$ *een unieke oplossing* $y(t)$ *voor het probleem* (29)*.*

Bewijs. Het idee van het bewijs is om (29) te schrijven als

$$y(t) = y_0 + \int_0^t f(\tau, y(\tau))\mathrm{d}\tau. \tag{30}$$

De functie y is dus een vast punt van de operator

$$\Phi : C^0([0, a]) \to C^0([0, a]), \quad y \mapsto y_0 + \int_0^t f(\tau, y(\tau))\mathrm{d}\tau.$$

$C^0([0, a])$ is, samen met de oneindig norm $\|f\|_\infty := \max_{x\in[0,a]} |f(x)|$ een Banach ruimte. Voor een Banach ruimte geldt de stelling van Banach evenwel. De voorwaarden van de stelling aan te tonen is een oefening. □

Lemma 18 (Conditie van de gewone differentiaalvergelijking)**.** *We beschouwen de oplossing* y *van* (29) *en de oplossing* z *van* (29) *met* y_0 *vervangen door* z_0*. Onder de voorwaarden van Stelling 9 geldt*

$$\|y(t) - z(t)\| \leq e^{L\cdot t}\|y_0 - z_0\| \qquad \forall t \in [0, a].$$

Bewijs. Er geldt:

$$y(t) = y_0 + \int_0^t f(\tau, y(\tau))\mathrm{d}\tau, \qquad z(t) = z_0 + \int_0^t f(\tau, z(\tau))\mathrm{d}\tau$$

en bijgevolg

$$\|y(t) - z(t)\| = \|y_0 - z_0 + \int_0^t f(\tau, y(\tau)) - f(\tau, z(\tau))\mathrm{d}\tau\| \leq \|y_0 - z_0\| + L\int_0^t \|y(\tau) - z(\tau)\|\mathrm{d}\tau.$$

We definiëren $\varphi(t) = \|y(t) - z(t)\|$, voor deze functie geldt

$$\varphi(t) \leq \|y_0 - z_0\| + L\int_0^t \varphi(\tau)\mathrm{d}\tau.$$

De rest kan nu via de ongelijkheid van Gronwall bewezen worden, die we hier zonder bewijs aangeven. □

Lemma 19. *(Gronwall) Stel dat $p : \mathbb{R}^+ \to \mathbb{R}$ een niet-negatieve, niet-dalende integreerbare functie is. Veronderstel bovendien dat $g, \varphi : \mathbb{R}^+ \to \mathbb{R}$ continue functies zijn. Indien*

$$\varphi(t) \leq g(t) + \int_0^t p(\tau)\varphi(\tau)\mathrm{d}\tau, \qquad \forall t \in [0, T],$$

geldt ook

$$\varphi(t) \leq g(t)e^{\int_0^t p(\tau)\mathrm{d}\tau}, \qquad \forall t \in [0, T].$$

Om het nog wat gemakkelijker te houden bespreken wij alleen de numerieke oplossing van *autonome differentiaalvergelijkingen*. Dat zijn differentiaalvergelijkingen van vorm

$$\begin{aligned} y'(t) &= f(y(t)), \quad \forall t \in [0, T], \\ y(0) &= y_0. \end{aligned} \tag{31}$$

We maken dus de aanname:

f hangt niet meer expliciet van de tijd af en f is Lipschitz-continu.

Opmerking 16. *Autonome differentiaalvergelijkingen te beschouwen kan zonder verlies van algemeenheid, omdat men de tijd (differentiaalvergelijking $t' = 1, t(0) = 0$) als extra variabele kan beschouwen.*

4.2 Één-stap methoden

Alvorens we beginnen moeten wij de interval $[0, T]$ discretiseren. Gegeven dus $N \in \mathbb{N}$ waarden t^i met

$$0 = t^0 < t^1 < \ldots < t^N = T.$$

Het eenvoudigste geval is indien de t^n gelijkmaatig verdeelt zijn, dus de afstand $t^n - t^{n-1}$ is een constante waarde die wij Δt noemen. Vanaf nu veronderstellen wij dit en definiëren voor $\Delta t := \frac{T}{N}$ het zogenoemde rooster

$$\mathcal{T}_{\Delta t} := \left\{ t^n := n\Delta t = \frac{nT}{N}, \mid 0 \leq n \leq N \right\}.$$

Het idee achter elke discretisatie methode om (29) op te lossen is om *discrete* waarden y^n op dit rooster te berekenen, die een benadering aan $y(t^n)$ voorstellen.

We hebben al gezien dat de differentiaalvergelijking (29) equivalent is met een integraalvergelijking (30). Gebaseerd op bekende kwadratuurformules kunnen wij dus de eerste één-stap methoden noteren:

Euler methoden Gegeven een $y(t^n)$. Uitgaande van dit is de exacte oplossing $y(t^{n+1})$ gegeven door (zie ook (30))

$$y(t^{n+1}) = y(t^n) + \int_{t^n}^{t^{n+1}} f(y(t))\mathrm{d}t.$$

Gebaseerd op de (linke) rechthoekregel kunnen wij dit benaderen door

$$y(t^{n+1}) \approx y(t^n) + \Delta t f(y(t^n)).$$

Hier kunnen wij meteen een algoritme van maken:

Algoritme 4 (Expliciete Euler methode)**.** *Gegeven $N \in \mathbb{N}$ en $\Delta t := \frac{T}{N}$. Een benadering van (31) is gegeven door $y^0 = y_0$ en*

$$y^{n+1} = y^n + \Delta t f(y^n).$$

Uiteraard kunnen wij hetzelfde met de rechte rechthoekregel doen en krijgen de benadering

$$y(t^{n+1}) \approx y(t^n) + \Delta t f(y(t^{n+1})),$$

en vervolgens het algoritme

Algoritme 5 (Impliciete Euler methode)**.** *Gegeven $N \in \mathbb{N}$ en $\Delta t := \frac{T}{N}$. Een benadering van* (31) *is gegeven door $y^0 = y_0$ en*

$$y^{n+1} = y^n + \Delta t f(y^{n+1}).$$

Opmerking 17. *Om y^{n+1} van de impliciete Euler methode te bepalen moet men een vergelijking op lossen. Daarom het woord 'impliciet'.*

Nog een derde benaderingsmethode komt door de middelpuntsregel op de integraal toe te passen, dus

$$y(t^{n+1}) \approx y(t^n) + \Delta t f(y(t^{n+1/2})).$$

Dat is wat ingewikkelder, omdat wij (nog) geen benadering van $y(t^{n+1/2})$ ter beschikking hebben. We kunnen echter een expliciete Euler stap doorvoeren en krijgen het volgende algoritme:

Algoritme 6 (Verbeterde Euler methode)**.** *Gegeven $N \in \mathbb{N}$ en $\Delta t := \frac{T}{N}$. Een benadering van* (31) *is gegeven door $y^0 = y_0$ en*

$$y^{n+1} = y^n + \Delta t f\left(y^n + \frac{\Delta t}{2} f(y^n)\right).$$

Voorbeeld 7. *We beschouwen een van de waarschijnlijk eenvoudigste differentiaalvergelijkingen:*

$$y' = \lambda y, \qquad y(0) = y_0$$

voor een $\lambda \in \mathbb{R}$. Deze vergelijking wordt ook Dahlquist-vergelijking *genoemd. De exacte oplossing is $y(t) = e^{\lambda t} y_0$. We passen de verschillende methoden op deze vergelijking toe. Omdat de vergelijking zo eenvoudig is, is het hier nog mogelijk om expliciet waarden voor y^n aan te geven.*

- *(Expliciete Euler methode)*

$$y^{n+1} = y^n + \Delta t \lambda y^n = (1 + \Delta t \lambda) y^n = (1 + \Delta t \lambda)^{n+1} y^0.$$

- *(Impliciete Euler methode)*

$$y^{n+1} = y^n + \Delta t \lambda y^{n+1} \quad \Rightarrow \quad y^{n+1} = \frac{y^n}{1 - \Delta t \lambda} = \frac{1}{(1 - \Delta t \lambda)^{n+1}} y^0.$$

- *(Verbeterde Euler methode)*

$$y^{n+1} = y^n + \Delta t \lambda \left(y^n + \frac{\Delta t}{2} \lambda y^n\right) = \left(1 + \Delta t \lambda + \frac{\Delta t^2 \lambda^2}{2}\right) y^n = \left(1 + \Delta t \lambda + \frac{\Delta t^2 \lambda^2}{2}\right)^{n+1} y^0.$$

We beschouwen $T = 1$, $\lambda = 2$, $y^0 = 1$ en berekenen de verschillende benaderingen. De fouten zijn in Tabel 1 genoteerd. Merk op: de fout voor de verbeterde Euler methode daalt veel sneller dan de andere.

De methoden die wij tot nu toe gezien hebben zijn voorbeelden van één-stap methoden:

Δt	expliciete Euler	impliciete Euler	verbeterde Euler
0.1	1.20E+00	1.92E+00	8.44E-02
0.05	6.62E-01	8.36E-01	2.28E-02
0.025	3.49E-01	3.92E-01	5.93E-03
0.0125	1.79E-01	1.90E-01	1.51E-03

Tabel 1: Numerieke oplossingen van de vergelijking $y' = 2y$ met exakte oplossing $y(t) = e^{2t}$. Berekend wordt tot $T = 1$, genoteerd de fouten $|e^2 - y^N|$ voor verschillende benaderingen en Δt. Merk op: Δt wordt in elke stap halveert. De fouten voor expliciete en impliciete Euler gedragen zich ogenschijnlijk als $\mathcal{O}(\Delta t)$, die van de verbeterde Euler methode als $\mathcal{O}(\Delta t^2)$.

Definitie 8 (Één-stap methode)**.** *Een* één-stap methode *voor de benadering van een oplossing tot de gewone differentiaalvergelijking* (31) *is gegeven door*

$$y^{n+1} = y^n + \Delta t \phi(t^n, y^n, \Delta t). \tag{32}$$

Voorbeeld 8.

- *(Expliciete Euler methode)*

$$\phi(t^n, y^n, \Delta t) := f(y^n).$$

- *(Impliciete Euler methode)*

$$\phi(t^n, y^n, \Delta t) := f(y^n + \Delta t \phi(t^n, y^n, \Delta t)).$$

- *(Verbeterde Euler methode)*

$$\phi(t^n, y^n, \Delta t) := f\left(y^n + \frac{\Delta t}{2} f(y^n)\right).$$

Opmerking 18. *Bij impliciete methoden wordt y^{n+1} door een (waarschijnlijk niet-lineair) stelsel van vergelijkingen bepaald. Hoe garandeerd men dat de oplossing daarvan eenduidig is? Beschouw de impliciete Euler methode $y^{n+1} = y^n + \Delta t f(y^{n+1})$, dus we moeten de vergelijking*

$$x = y^n + \Delta t f(x)$$

oplossen. Dit is uiteraard een vast punt probleem, dus de iteratieve methode

$$x^\nu = y^n + \Delta t f(x^\nu) =: \Phi(x^\nu), \quad x^0 = y^n$$

ligt voor de hand. Indien f Lipschitz-continu is, en indien Δt voldoende klein, is er aan de voorwaarden van de stelling van Banach voldaan! (Oefening)

We hebben al gezien dat verschillende methoden ook verschillende convergentie snelheid hebben. In het volgende willen we nu bestuderen *of* een één-stap methode convergeert en, indien wel, hoe snel. Uiterst belangrijk is hiervoor het concept van de locale afbreekfout:

Definitie 9. *Zij $\tilde{y}^{n+1} := y(t^n) + \Delta t \phi(t^n, y(t^n), \Delta t)$. (Merk op: We hebben rechts y^n door $y(t^n)$ vervangen, de exacte oplossing dus!) De locale afbreekfout is gedefinieerd door*

$$R(t^n, \Delta t) := \left\| \tilde{y}^{n+1} - y(t^{n+1}) \right\|. \tag{33}$$

(R hangt uiteraard nog van y, ϕ, ... af, maar voor het gemak maken wij deze relatie niet expliciet.) Indien voor een gladde oplossing y en $t^n \in [0, T]$ een $q \in \mathbb{N}$ bestaat zodanig dat

$$R(t^n, \Delta t) = \mathcal{O}(\Delta t^{q+1}) \tag{34}$$

geldt, zeggen wij dat de methode consistent *is met* consistentieorde q.

Voorbeeld 9. *We beschouwen de Euler methoden als voorbeelden.*

- *(Expliciete Euler methode)*

$$\widetilde{y}^{n+1} = y(t^n) + \Delta t f(y(t^n)) = y(t^n) + \Delta t y'(t^n) = y(t^{n+1}) + \mathcal{O}(\Delta t^2).$$

De methode heeft dus minstens consistentieorde $q = 1$. (Ga na dat het niet $q > 1$ kan zijn. Beschouw hiervoor de Dahlquist-vergelijking.)

- *(Impiciete Euler methode)*

$$\begin{aligned}\widetilde{y}^{n+1} = y(t^n) + \Delta t f(\widetilde{y}^{n+1}) &= y(t^n) + \Delta t f(y(t^n) + \Delta t f\left(\widetilde{y}^{n+1}\right)) \\ &= y(t^n) + \Delta t f(y(t^n)) + \mathcal{O}(\Delta t^2).\end{aligned}$$

De methode heeft dus minstens consistentieorde $q = 1$. (Ga na dat het niet $q > 1$ kan zijn. Beschouw hiervoor de Dahlquist-vergelijking.)

- *(Verbeterde Euler methode)*

$$\begin{aligned}\widetilde{y}^{n+1} &= y(t^n) + \Delta t f\left(y(t^n) + \frac{\Delta t}{2} f(y(t^n))\right) \\ &= y(t^n) + \Delta t \left(f(y(t^n)) + \frac{\Delta t}{2} f'(y(t^n)) f(y(t^n)) + \mathcal{O}(\Delta t^2)\right) \\ &= y(t^n) + \Delta t f(y(t^n)) + \frac{\Delta t^2}{2} f'(y(t^n)) f(y(t^n)) + \mathcal{O}(\Delta t^3) \\ &= y(t^n) + \Delta t y'(t^n) + \frac{\Delta t^2}{2} y''(t^n) + \mathcal{O}(\Delta t^3).\end{aligned}$$

(Merk op: $y' = f(y)$ en dus $y'' = f'(y)y' = f'(y)f(y)$.) De methode is dus consistent met consistentieorde minstens $q = 2$. (Ga na dat het niet $q > 2$ kan zijn. Beschouw hiervoor de Dahlquist-vergelijking.)

Lemma 20. *Een één-stap methode is consistent als en slechts als*

$$\phi(t, y(t), \Delta t = 0) = f(y(t)).$$

Bewijs. We gebruiken de definitie van R (zie (33)), die voor een één-stap methode kan geschreven worden als

$$\begin{aligned}R(t^n, \Delta t) &= \left\| y(t^n) + \Delta t \phi(t^n, y(t^n), \Delta t) - y(t^{n+1}) \right\| \\ &= \left\| y(t^n) + \Delta t \phi(t^n, y(t^n), \Delta t) - y(t^n) - \Delta t f(y(t^n)) + \mathcal{O}(\Delta t^2) \right\| \overset{!}{=} \mathcal{O}(\Delta t^2).\end{aligned}$$

Het laatste geldt omdat de methode consistent is en dus $q \geq 1$ in (34). Bijgevolg:

$$\|\phi(t^n, y(t^n), \Delta t) - f(y(t^n))\| = \mathcal{O}(\Delta t).$$

□

Consistentie is een *locale* eigenschap. We zijn echter geïnteresseerd in convergentie, een *globale* eigenschap:

Definitie 10. *Beschouw y^n gedefineerd door (32) met een gegeven y_0. Zij $y(t)$ de oplossing van (31) met $y(0) = y_0$. Definieer*

$$e^n := y^n - y(t^n).$$

De methode is convergent van orde $p \in \mathbb{N}$ indien

$$\|e^n\|_\infty := \max_{n=0,\ldots,N} \|e^n\| = \mathcal{O}(\Delta t^p).$$

Het verrassende is dat consistentie (een lokale eigenschap) convergentie (een globale eigenschap) impliceert. Dat is een algemeen principe in de numerieke wiskunde, een beetje overdreven geformuleerd als

$$\text{Consistentie} + \text{Stabiliteit} = \text{Convergentie}.$$

We zullen zien dat Lipschitz-continuiteit van f en ϕ al genoeg is om stabiliteit te garanderen.

Stelling 10. *Stel dat y de oplossing is van* (31) *met een gladde flux functie f. (Vervolgens is ook y glad.) Zij bovendien een één-stap methode* (32) *gegeven met een Lipschitz-continue flux ϕ (Lipschitz-constante L_ϕ) en consistentieorde $q \geq 1$. Dan is de methode convergent van orde q.*

Bewijs. Voor het gemak schrijven we de afhankelijkheid van ϕ van de tijd t en de tijdstap Δt niet in het argument, dus

$$\phi(y(t)) \equiv \phi(t, y(t), \Delta t).$$

Er geldt

$$e^n = y^n - y(t^n) = y^n - \widetilde{y}^n + \widetilde{y}^n - y(t^n) = y^n - \widetilde{y}^n + \mathcal{O}(\Delta t^{q+1}).$$

We weten dus dat er een Δt_0 bestaat zodanig dat voor elke $\Delta t \leq \Delta t_0$,

$$\|\widetilde{y}^n - y(t^n)\| \leq C\Delta t^{q+1}.$$

(Dit volgt uit de definitie van de grote-$\mathcal{O}$.) Bovendien:

$$\begin{aligned}\|y^n - \widetilde{y}^n\| &= \|y^{n-1} + \Delta t\phi(y^{n-1}) - y(t^{n-1}) - \Delta t\phi(y(t^{n-1}))\| \\ &\leq \|y^{n-1} - y(t^{n-1})\| + \Delta t\|\phi(y^{n-1}) - \phi(y(t^{n-1}))\| \\ &\leq \|e^{n-1}\| + \Delta t L_\phi\|e^{n-1}\| =: \alpha\|e^{n-1}\|.\end{aligned}$$

Bijgevolg geldt (herhaal: $\alpha := (1 + \Delta t L_\phi)$)

$$\begin{aligned}\|e^n\| &\leq C\Delta t^{q+1} + \alpha\|e^{n-1}\| \\ &\leq C\Delta t^{q+1} + \alpha\left(C\Delta t^{q+1} + \alpha\|e^{n-2}\|\right) \leq \dots \\ &\leq C\Delta t^{q+1}\left(1 + \alpha + \alpha^2 + \dots + \alpha^{n-1}\right) = C\Delta t^{q+1}\frac{\alpha^n - 1}{\alpha - 1}.\end{aligned}$$

(Merk op: $\|e^0\| = 0$!) Er geldt $\ln(1 + x) \leq x$ en bijgevolg

$$\alpha^n \leq \alpha^N = e^{N\ln(\alpha)} = e^{N\ln(1+\Delta t L_\phi)} \leq e^{N\Delta t L_\phi} = e^{TL_\phi}.$$

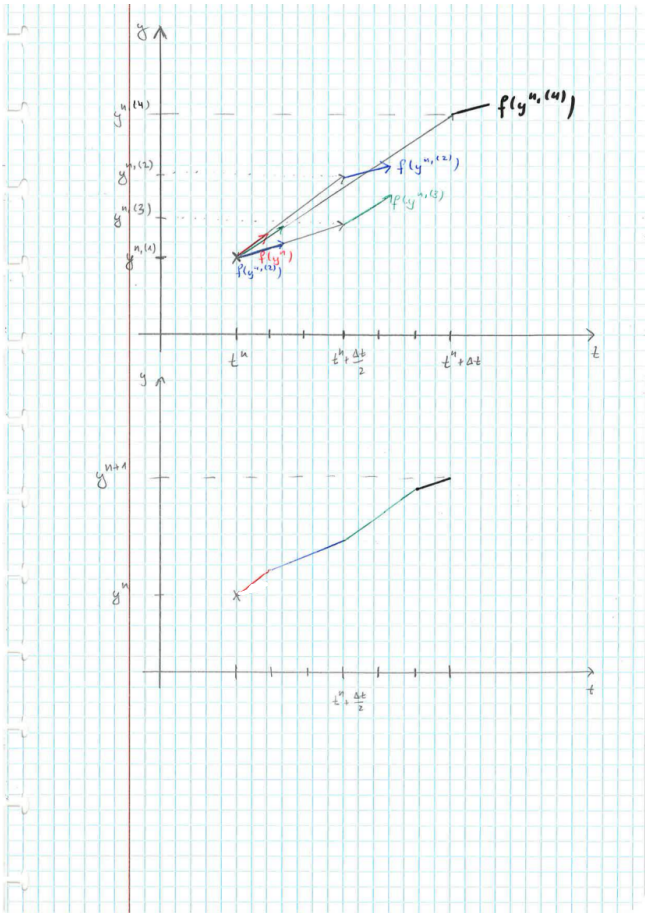
We sluiten het bewijs nu af door te berekenen:

$$\|e^n\| \leq C\Delta t^{q+1}\frac{\alpha^n - 1}{\alpha - 1} \leq C\Delta t^{q+1}\frac{e^{TL_\phi} - 1}{\Delta t L_\phi} \leq C(T, L_\phi)\Delta t^q.$$

□

4.3 Runge-Kutta methoden

Een belangrijk ingrediënt tot een goede numerieke methode is de consistentieorde. In deze sectie zullen wij het hebben over Runge-Kutta methoden, een klas van één-stap methoden waarmee het gemakkelijker wordt om (zeer) hoge consistentieorde te realiseren. Het trucje is om tussenstappen $y^{n,(i)}$ (de zogenoemde *trappen*) te definiëren, die benaderingen voorstellen aan $y(t^n + c_i\Delta t)$, met $0 \leq c_i \leq 1$. We beginnen met het voorbeeld van de 'klassieke' Runge-Kutta methode:



Figuur 2: De RK4 methode.

Algoritme 7 (Klassieke Runge-Kutta methode, RK4)**.** *Gegeven* $N \in \mathbb{N}$ *en* $\Delta t := \frac{T}{N}$*. Een benadering van* (31) *is gegeven door* $y^0 = y_0$ *en*

$$
\begin{aligned}
y^{n,(1)} &= y^n \\
y^{n,(2)} &= y^n + \frac{\Delta t}{2} f(y^{n,(1)}) \\
y^{n,(3)} &= y^n + \frac{\Delta t}{2} f(y^{n,(2)}) \\
y^{n,(4)} &= y^n + \Delta t f(y^{n,(3)}) \\
y^{n+1} &= y^n + \frac{\Delta t}{6} \left(f(y^{n,(1)}) + 2f(y^{n,(2)}) + 2f(y^{n,(3)}) + f(y^{n,(4)}) \right).
\end{aligned}
$$

Dat ziet er iets ingewikkelder uit dan een één-stap methode, het kan echter in de vorm van Def. 8 geschreven worden en is vervolgens een één-stap methode. (Ga het na!) Voor een poging tot een schets, zie Fig. 2 of het Matlab programma `rk4_demo.m`.

Stelling 11. *De consistentieorde van RK4 is gelijk aan vier.*

Bewijs. Voor het gemak beschouwen wij alleen de Dahlquist-vergelijking $y' = \lambda y$, het resultaat geldt

echter ook voor een niet-lineaire vergelijking. We krijgen voor $\widetilde{y}^{n+1}$:

$$\begin{aligned}
\widetilde{y}^{n,(1)} &= y(t^n),\\
\widetilde{y}^{n,(2)} &= y(t^n) + \frac{\lambda\Delta t}{2}\widetilde{y}^{n,(1)} = y(t^n) + \frac{\lambda\Delta t}{2}y(t^n) = \left(1 + \frac{\lambda\Delta t}{2}\right) y(t^n),\\
\widetilde{y}^{n,(3)} &= y(t^n) + \frac{\Delta t\lambda}{2}\widetilde{y}^{n,(2)} = y(t^n) + \frac{\Delta t\lambda}{2}\left(y(t^n) + \frac{\lambda\Delta t}{2}y(t^n)\right) = \left(1 + \frac{\Delta t\lambda}{2} + \frac{(\Delta t\lambda)^2}{4}\right) y(t^n)\\
\widetilde{y}^{n,(4)} &= y(t^n) + \Delta t\lambda\widetilde{y}^{n,(3)} = \left(1 + \Delta t\lambda + \frac{(\Delta t\lambda)^2}{2} + \frac{(\Delta t\lambda)^3}{4}\right) y(t^n)\\
\widetilde{y}(t^{n+1}) &= y(t^n) + \frac{\lambda\Delta t}{6}\left(\widetilde{y}^{n,(1)} + 2\widetilde{y}^{n,(2)} + 2\widetilde{y}^{n,(3)} + \widetilde{y}^{n,(4)}\right)\\
&= y(t^n) + \frac{\lambda\Delta t}{6}y(t^n)\left(1 + 2\left(1 + \frac{\lambda\Delta t}{2}\right) + 2\left(1 + \frac{\Delta t\lambda}{2} + \frac{(\Delta t\lambda)^2}{4}\right) + \left(1 + \Delta t\lambda + \frac{(\Delta t\lambda)^2}{2} + \frac{(\Delta t\lambda)^3}{4}\right)\right)\\
&= y(t^n)\left(1 + \lambda\Delta t\left(\frac{1}{6} + \frac{2}{6} + \frac{2}{6} + \frac{1}{6}\right) + (\lambda\Delta t)^2\left(\frac{1}{6} + \frac{1}{6} + \frac{1}{6}\right) + (\lambda\Delta t)^3\left(\frac{1}{12} + \frac{1}{12}\right) + (\lambda\Delta t)^4\frac{1}{24}\right)\\
&= y(t^n)\left(1 + \lambda\Delta t + \frac{(\lambda\Delta t)^2}{2} + \frac{(\lambda\Delta t)^3}{6} + \frac{(\lambda\Delta t)^4}{24}\right)
\end{aligned}$$

Merk op: voor de exacte oplossing $y(t) = e^{\lambda t}$ geldt

$$y(t^{n+1}) = e^{\lambda t^{n+1}} = \sum_{i=0}^{\infty} \frac{(\lambda\Delta t)^i}{i!} y(t^n).$$

De methode is dus – voor deze vergelijking – van consistentieorde $q = 4$. Dat geldt ook voor het algemene geval, is wel wat ingewikkelder te bewijzen: zelfstudie! □

RK4 is de 'originele' Runge-Kutta methode. De uitbreiding naar een klas van methoden gaat via de volgende algoritme:

Algoritme 8 (Algemene Runge-Kutta methode)**.** *Gegeven $N \in \mathbb{N}$ en $\Delta t := \frac{T}{N}$. Een benadering van* (31) *is gegeven door $y^0 = y_0$ en*

$$\begin{aligned}
y^{n,(i)} &= y^n + \Delta t\sum_{j=1}^{m} a_{ij} f(y^{n,(j)}), \qquad 1 \le i \le m\\
y^{n+1} &= y^n + \Delta t\sum_{j=1}^{m} b_j f(y^{n,(j)}).
\end{aligned}$$

De coëfficiënten a_{ij} en b_j bepalen de methode.

De coëfficienten A en b worden meestal in een *Butcher* tableau opgeslagen in vorm $\begin{array}{c|c} c & A \\ \hline & b^T. \end{array}$ c speelt een rol indien een niet-autonome differentiaalvergelijking beschouwd wordt, dus indien f van y en van t afhangt. In de algoritme moet dan $f(y^{n,(j)})$ door $f(t^n + c_j\Delta t, y^{n,(j)})$ vervangen worden. Meestal is

$$c_i = \sum_{j=1}^{m} a_{ij}, \tag{35}$$

omdat dan de Runge-Kutta methode – toegepast op de niet-autonome vergelijking – hetzelfde resultaat levert als de Runge-Kutta methode, toegepast op de autonome vergelijking met het trucje $t' = 1$.

Voor RK4 kunnen wij A, b en c aangeven:

0	0	0	0	0
1/2	1/2	0	0	0
1/2	0	1/2	0	0
1	0	0	1	0
	1/6	2/6	2/6	1/6