Praktikum Numerieke Gewone differentiaalvergelijkingen.

Korte handleiding bij opgave 2.

Maak aantekeningen bij je experimenten!

Bij dit praktikum gebruiken we 10 kleine matlab-programma's (zgn. m-files):
`ASv.m`, `Av.m`, `ff.m`, `op2ab.m`, `op2cn.m`,
`op2gr.m`, `op2s1.m`, `op2s2.m`, `op2s22.m` en `op2sb.m`.[1]
Matlab kun je opstarten door `matlab` in te toetsen in een 'commandtool'.

Ad d. Start matlab op. Het stabiliteitsgebied van het Adams-Bashforth schema krijg je te zien door op2sb in te tikken na $>>$.

De experimenten met het schema kun je uitvoeren door op2ab aan te roepen. Je krijgt dan in één plaatje de exacte oplossing en benaderingen van de exacte oplossing in de punten $j\Delta x$ op het tijdstip `T` te zien. De grootheden `dx` (Δx), `dt` (Δt) en `r` verschijnen op het scherm. Het aantal gridpunten `N`, de stapgrootte `dt`, en het eindtijdstip `T` kun je aanpassen in de file `op2ab.m`. Zorg ervoor dat `T/dt` geheel is!

Ad e. Roep nu op2s1 aan. Je krijgt dan in één plaatje de exacte oplossing en benaderingen op het tijdstip `T` te zien. De grootheden `dx`, `dt`, `r`, alsmede de kleinste en de grootste eigenwaarde van AS, verschijnen op het scherm. De waardes voor `N`, `dt`, `T` en `epsilon` kun je aanpassen in de file `op2s1.m`. Zorg ervoor dat `T/dt` geheel is!

Ad f. De experimenten met de impliciete smoother kun je uitvoeren m.b.v. de m-file `op2s2.m`. Deze m-file werkt precies hetzelfde als `op2s1.m`.

Door op2gr aan te roepen krijg je 9 grafieken van de functie φ_ε (voor verschillende ε) te zien. Eventueel kun je de waardes voor ε (viz. `epsilon1`, ..., `epsilon9`) nog aanpassen in de file `op2gr.m`.

Ad g. Maak een keuze voor `N`, `s`, `T` en `delta` in de file `op2s22.m` (nb: alles wat na een %-teken staat is commentaar; dit kun je gebruiken om de verschillende keuzes voor `delta` aan c.q. uit te zetten). Als je nu op2s22 aanroept, krijg je in één plaatje de exacte oplossing en benaderingen op het tijdstip `T` te zien. Tevens verschijnen `dx`, `dt`, `r`, `s`, `delta`, `epsilon`, alsmede de kleinste en de grootste eigenwaarde van AS, op het scherm.

Ad h. Roep op2cn aan; deze m-file werkt precies hetzelfde als op2ab.

[1] Kopieer deze files naar je home-directory. Deze m-files zijn te vinden in de directory
`http://www.math.uu.nl/people/sleijpen/Opgaven/NumGDV/Opgave2.tar.gz`
Kopieer deze files naar de directory waarin je voor dit praktikum wilt werken.
Met `gunzip Opgave2.tar.gz` en `tar xvf Opgave2.tar` kan je de files 'uitpakken'. (De m-files `ASv.m`, `Av.m` en `ff.m` hoef je niet zelf aan te roepen tijdens dit praktikum; ze worden gebruikt in de andere m-files, dus je hebt ze wel nodig!)

Bij opgave 2 heb je de volgende matlab-subroutines tot je beschikking:

1. `adamsbas` (Adams-Bashforth)
   `cranknic` (trapeziumregel)
2. `laplace`
   `inhom`
   `smooth1`
   `smooth2`

De routines in groep 1 zijn rekenroutines, terwijl de routines in groep 2 zgn. matrixgeneratoren zijn, d.w.z. de belangrijkste output is een matrix waarmee je vervolgens gaat rekenen door een routine uit groep 1 aan te roepen. Om deze matrices te kunnen 'bouwen' heb je vooraf een aantal parameters nodig, nl:

- N aantal inwendige discretisatiepunten op het (ruimte-)interval $[0, \pi]$ (denk aan N tussen 10 en 100)
- T lengte tijdinterval (= tijdstip waarop de oplossing berekend/benaderd wordt)
- dt grootte van de tijdstap. Let erop dat het aantal tijdstappen T/dt geheel is, niet te klein (dan zijn instabiliteitsverschijnselen nauwelijks waarneembaar) en niet te groot (dan verricht je erg veel rekenwerk). Met ca. 100 tijdstappen kom je meestal wel goed uit.
- ep deze parameter heb je alleen nodig wanneer je smooth1 of smooth2 aanroept, alle waarden tussen 0 en 1 zijn in principe denkbaar.

`laplace` levert de matrix A uit onderdeel c, deze matrix hangt uitsluitend af van de discretisatie in de x-richting (parameter N).
`inhom` levert een matrix F waaruit de waarden voor de inhomogene term $f = f(x, t)$ op alle discretisatiepunten in zowel ruimte- als tijdrichting kunnen worden afgelezen (parameters N, T en dt).
`smooth1` levert de expliciete smoother S1, `smooth2` levert de impliciete smoother S2 (parameters N, ep), omdat S2 weer afhangt van A kan `smooth2` pas worden aangeroepen nadat (opnieuw) `laplace` is aangeroepen.
`adamsbas` gebruikt automatisch een smoother S; bij expliciet smoothen kies je dus S = S1, bij impliciet smoothen S = S2 en zonder smoothen S = eye(N) (= identiteitsmatrix).

1. Voorbeelden. Adams-Bashforth zonder smoothen, exact en benaderd in een figuur:

```
N = ..
laplace            (discretisatie in de x-richting)
S = eye(N);        (niet smoothen)
T = ..
dt = ..            (discretisatie in de t-richting)
inhom              (levert tevens de exacte oplossing y)
adamsbas           (levert benaderende oplossing yab)
figure(1)          (opent plaatje)
restart            (maakt figuur schoon)
plot(x,y)          (plot exacte oplossing in t=T)
plot(x,yab,'--')   (plot benaderende oplossing in t=T)
```

idem met expliciet smoothen, nieuwe benadering in zelfde figuur:

```
ep = ..
smooth1
S = S1;
adamsbas
plot(x,yab,':')
```

Adams-Bashforth met expliciet smoothen en een nieuwe dt, exact + nieuwe benadering in nieuwe figuur:

```
dt = ..
inhom
adamsbas
figure(2)
restart
```

```
plot(x,y)
plot(x,yab,':')
```

Adams-Bashforth met expliciet smoothen en een nieuwe dx, nieuwe benadering in tweede figuur:

```
N = ..
laplace
smooth1
S = S1;
inhom
adamsbas
plot(x,yab,'--')
```

Adams-Bashforth met impliciet smoothen voor de oude en een nieuwe waarde van ep; wissen van figuur 1 en nieuwe benaderingen in figuur 1:

```
smooth2
S = S2;
adamsbas
figure(1)
restart
plot(x,yab,'--')
ep = ..
smooth2
S = S2;
adamsbas
plot(x,yab,':')
```

Adams-Bashforth met impliciet smoothen, oude waarde voor ep en nieuwe waarde voor dx, resultaat in zelfde figuur:

```
N = ..
laplace
smooth2
S = S2;
inhom
adamsbas
plot(x,yab)
```

2. [Semi-diskretisatie van een diffusievergelijking].
Bij dit praktikum wordt matlab gebruikt. De opgaves waarvoor je matlab nodig hebt zijn gemerkt met een ^{m}. Tijdens het praktikum wordt een korte handleiding over het gebruik van matlab (bij dit praktikum) uitgedeeld. Probeer de opgaves **a**, **b**, **c**, en de onderdelen van **d** t/m **h** waarbij je geen computer nodig hebt, *voor* het praktikum uit te werken!

Beschouw voor een plaatsvariabele $x \in [0, \pi]$ en een tijdvariabele $t \geq 0$ de diffusievergelijking

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + f(x,t), \tag{2.1}$$

met begin- en randvoorwaarden

$$\begin{cases} u(x,0) = \varphi(x), \\ u(0,t) = u(\pi,t) = 0. \end{cases} \tag{2.2}$$

a. Ga na dat voor $f(x,t) = \sin x\,(1 + 2\sin t)$ en $\varphi(x) = 0$ de oplossing u^* gegeven wordt door

$$u^*(x,t) = \sin x\,(1 - \cos t + \sin t).$$

Bij semi-diskretisatie wordt eerst in de x-richting gediskretiseerd en vervolgens in de t-richting. Voor de diskretisatie in de x-richting verdelen we $[0,\pi]$ in N deelintervallen van gelijke lengte; definieer $\Delta x \equiv \pi/(N+1)$. Voor $t \geq 0$ zoeken we een rijtje $u_0(t), u_1(t), \ldots, u_{N+1}(t)$ dat de waarden van u^* in 0, Δx, $\ldots$, $(N+1)\Delta x$ op tijdstip t benadert. Voor $j = 0, 1, \ldots, N+1$ schrijven: we

$$u_j(t) \equiv u(j\Delta x, t), \quad u_j^*(t) \equiv u^*(j\Delta x, t), \quad f_j(t) \equiv f(j\Delta x, t).$$

Verder is

$$\vec{u} \equiv (u_1, \ldots u_N)^T, \quad \vec{u}^* \equiv (u_1^*, \ldots u_N^*)^T, \quad \vec{f} \equiv (f_1, \ldots f_N)^T.$$

Voor de plaatsafgeleide gebruiken we de volgende relatie:

$$\frac{\partial^2 u}{\partial x^2}(j\Delta x, t) = \frac{u_{j+1}(t) - 2u_j(t) + u_{j-1}(t)}{(\Delta x)^2} + \delta_j^x(t).$$

b. Toon aan dat voor de plaatsdiskretisatiefout δ_j^x geldt:

$$\delta_j^x(t) = \mathcal{O}(\Delta x^2).$$

Door nu tegelijkertijd in alle inwendige diskretisatiepunten de relatie

$$u_j' = \frac{u_{j+1} - 2u_j + u_{j-1}}{(\Delta x)^2}$$

op te leggen $\left(u_j' = \frac{\partial u}{\partial t}(j\Delta x, t) = \frac{\partial^2 u}{\partial x^2}(j\Delta x, t) = \frac{\partial^2 u_j}{\partial x^2}\right)$ ontstaat een gewone differentiaalvergelijking in $\vec{u}$ van de vorm

$$\begin{cases} \vec{u}\,' = \dfrac{1}{(\Delta x)^2} A\vec{u} + \vec{f}, \\ \vec{u}(0) = \vec{\varphi}. \end{cases} \tag{2.3}$$

c. Bepaal A, $\vec{f}$ en $\vec{\varphi}$. Reken na dat $(\vec{v}_\ell)_{1 \leq \ell \leq N}$ met

$$\vec{v}_\ell = (\sin \ell\Delta x, \sin 2\ell\Delta x, \ldots, \sin N\ell\Delta x)^T$$

eigenvectoren van A zijn en bepaal de bijbehorende eigenwaarden η_ℓ: $A\vec{v}_\ell = \eta_\ell \vec{v}_\ell$. Konkludeer dat

$$-4 \leq \eta_\ell \leq 0 \quad \text{voor alle} \quad \ell = 1, 2, \ldots, N.$$

Voor de tijdintegratie van de gewone differentiaalvergelijking (5) gebruiken we nu het expliciete 2-staps schema van Adams-Bashforth

$$\vec{u}_{n+1} = \vec{u}_n + \frac{3r}{2} A\vec{u}_n + \frac{3\Delta t}{2}\vec{f}_n - \frac{r}{2} A\vec{u}_{n-1} - \frac{\Delta t}{2}\vec{f}_{n-1} \tag{2.4}$$

met de startwaarde voor $\vec{u}_1$ verkregen uit $\vec{u}_0$ en $\vec{u}_{\frac{1}{2}}$ d.m.v een centrale differentie (midpuntregel); $\vec{u}_{\frac{1}{2}}$ is op zijn beurt weer verkregen uit $\vec{u}_0$ m.b.v. een voorwaartse differentie (Euler forward), d.w.z.

$$\begin{cases} \vec{u}_{\frac{1}{2}} = \vec{u}_0 + \dfrac{r}{2} A\vec{u}_0 + \dfrac{\Delta t}{2}\vec{f}_0, \\ \vec{u}_1 = \vec{u}_0 + rA\vec{u}_{\frac{1}{2}} + \Delta t\, \vec{f}_{\frac{1}{2}}. \end{cases}$$

$\mathbf{d}^m$. Wat stelt r voor? Vanaf welke waarde voor r wordt volgens de theorie het schema instabiel? Maak m.b.v. matlab een plotje van het stabiliteitsgebied van het expliciete 2-staps schema van Adams-Bashforth (vgl. p. 55 van het dictaat). Kun je uit dit plaatje zien voor welke r het schema (in)stabiel is? Onderzoek de foutvoortplanting in de buurt van deze kritische waarde voor r door afwisselend Δt en Δx langzaam te variëren. Hoe ziet (in het instabiele geval) de fout er in het algemeen uit?

We gaan nu kijken of het mogelijk is om de fout t.g.v. de snel-variërende componenten van de oplossing weg te werken, zodat we een grotere Δt kunnen gebruiken. Een voor de hand liggende manier om de fout 'weg te poetsen' is door gebruik te maken van een zgn. smoother. Dit is een operator S met als algemene eigenschap dat $S\vec{v}$ een gladdere functie is dan $\vec{v}$. Gebruik van een (enkele) smoother komt neer op toepassing van hetzelfde schema op de 'gladgemaakte' differentiaalvergelijking

$$\vec{u}' = \frac{1}{(\Delta x)^2} AS\vec{u} + \vec{f}. \tag{2.5}$$

$\mathbf{e}^m$. Een goedkope smoother is bijvoorbeeld, voor een geschikte $\varepsilon \in (0, 1]$, de volgende expliciete 'middelaar'. In de definitie worden de rijen van de matrix van S vastgelegd door de komponenten w_j van $\vec{w} = S\vec{v}$ te geven als functie van de komponenten v_j van $\vec{v}$:

$$w_j = (1-\varepsilon)v_j + \frac{\varepsilon}{4}\left(v_{j-1} + 2v_j + v_{j+1}\right) \qquad (2 \leq j \leq N-1). \tag{2.6}$$

Hoe zou je w_1 en w_N definiëren?

Begin met een situatie waarin Adams-Bashforth zonder smoothen (i.e. $\varepsilon = 0$) instabiel is. Probeer, door ε wat te verhogen, ervoor te zorgen dat je redelijk nauwkeurige benaderingen krijgt. Doe dit voor een aantal keuzes van N en Δt. Waarom ligt het voor de hand om ε zo klein mogelijk te kiezen?

De stabiliteit van het schema wordt bepaald door de eigenwaarden van AS. Merk op dat (6) te schrijven is als $w_j = v_j + \frac{\varepsilon}{4}\left(v_{j-1} - 2v_j + v_{j+1}\right)$, zodat $S = I + \frac{\varepsilon}{4}A$ en $AS = A + \frac{\varepsilon}{4}A^2$. De eigenwaarden van AS zijn dus $\eta_\ell + \frac{\varepsilon}{4}\eta_\ell^2$ voor $1 \leq l \leq N$. Deze eigenwaarden liggen in het bereik van de functie $x \to x + \frac{\varepsilon}{4}x^2$ met domein $[-4, 0]$. Bepaal dit bereik voor alle $\varepsilon \in (0, 1]$. Is het verstandig te werken met een $\varepsilon > 1$? Voor konsistentie is het van belang dat de kleinere eigenwaarden van AS lijken op die van A. Waarom?

Helaas kunnen we met bovenstaande expliciete smoother Δt niet veel groter nemen dan bij Adams-Bashforth zonder smoothing. Daarom gaan we nu een andere smoother bekijken.

$\mathbf{f}^m$. Een wat duurdere smoother is de volgende, waarbij $\vec{w} = S\vec{v}$ gedefinieerd wordt als de vector waarin het minimum wordt aangenomen van de functie

$$J(\vec{w}) = (1-\varepsilon)\|\vec{w} - \vec{v}\|^2 + \varepsilon\|A\vec{w}\|^2.$$

Ga na dat de matrix van S voor deze impliciete smoother gegeven wordt door

$$(2.7) \qquad S = \left(I + \frac{1}{\delta} A^T A\right)^{-1} \quad \text{met} \quad \delta \equiv \frac{1-\varepsilon}{\varepsilon}$$

(hint: differentieer J naar $\vec{w}$). Vergelijk deze impliciete smoother met de expliciete smoother (ga op dezelfde manier te werk). Laat zien dat je een grotere Δt kunt nemen dan bij de expliciete smoother uit **e**. Om dit te verklaren gaan we weer de eigenwaarden van AS onderzoeken; merk op dat $A^T = A$ en ga na dat de eigenwaarden van AS te schrijven zijn als $\varphi_\varepsilon(\eta_\ell)$, waarbij $\varphi_\varepsilon(x) = (1-\varepsilon)x/(1-\varepsilon+\varepsilon x^2)$. Plot nu, voor verschillende waarden van ε, de grafiek van φ_ε; neem als domein het interval $[-4, 0]$. Vergelijk, voor deze waarden van ε, het bereik van φ_ε met het bereik van de functie $x \to x + \frac{\varepsilon}{4}x^2$ uit **e**. Wat kun je hieruit concluderen?

g^m. De kwaliteit van een smoother wordt niet zozeer bepaald door de faktor waarmee de kritische waarde voor r feitelijk toeneemt, maar meer door de mogelijkheid om met een geschikt gekozen ε als functie van Δx een orde te winnen. In dit specifieke geval betekent een ordewinst dat de stabiliteit niet meer van r maar van

$$s \equiv \frac{\Delta t}{\Delta x}$$

afhangt, d.w.z. s moet kleiner dan een zekere konstante worden gekozen om stabiliteit te bewerkstelligen.

Onderzoek of je met ε als een zekere functie van Δx en de impliciete smoother een orde kunt winnen. Ga als volgt te werk. Begin met een keuze voor N, s en δ (δ juist voldoende groot om het schema stabiel te krijgen); neem verder T gelijk aan een geschikt veelvoud van Δt. Varieer nu N (en eventueel δ), maar houd s constant. Je kunt b.v. $\delta = c\Delta x$ of $\delta = c\Delta x^2$ nemen, voor een geschikte constante c. Welke keuzes voor s lijken zinvol (de constante c behorende bij δ kun je eventueel laten afhangen van de waarde s)?

h^m. Een duurdere methode die in de regel ook zonder smoothers prima werkt is meteen overgaan naar een impliciet schema voor de tijdintegratie van (5), bijv. het schema dat voortvloeit uit de trapeziumregel:

$$(2.8) \qquad \left(I - \frac{r}{2}A\right)\vec{u}_{n+1} = \left(I + \frac{r}{2}A\right)\vec{u}_n + \frac{\Delta t}{2}\left(\vec{f}_{n+1} + \vec{f}_n\right).$$

Deze methode wordt vaak de Crank-Nicholsonmethode genoemd. Verifieer dat in theorie voor geen enkele Δt en geen van de mogelijke eigenwaarden η_ℓ van A stabiliteitsproblemen te verwachten zijn. Toets dit in de praktijk door hele grote tijdstappen te nemen. Verklaar waarom het toch niet verstandig is om met deze methode hele grote tijdstappen te nemen. Vergelijk de efficiëntie van de Crank-Nicholsonmethode met die van Adams-Bashforth onder impliciet smoothen. Denk je dat het verschil maakt of de coëfficiënt van de homogene term in de differentiaalvergelijking konstant is (zoals in (4)) of variabel, d.w.z. een positieve (niet-konstante) functie a zoals in de meer algemene vergelijking

$$(2.9) \qquad \frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(a(x,t)\frac{\partial u}{\partial x}\right) + f(x,t)\,?$$