

Practicum 1:

Numerical solution of the heat equation

Mattias Billast & Alexander Boucquey

28 november 2017

Het doel van deze taak is om de expliciete, Impliciete Euler en Crank-Nicolson methodes te implementeren en te vergelijken. Dit wordt bereikt m.b.v. 5 verschillende vragen.

De eerste drie vragen gaan over het oplossen van het een-dimensionaal, tijdsafhankelijk warmte probleem : $\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$ met $x \in [0, 1], t > 0$ en met gegeven randvoorwaarden: $u(0, t) = f_1(t), u(1, t) = f_2(t), u(x, 0) = g(x)$. Eerst moet de implementatie getest worden, vervolgens wordt de stabiliteitslimiet van de tijdstappen aangetoond en tot slot worden er verschillende grid waarden en tijdstappen gecombineerd om de consistentie te achterhalen.

Vraag 4 en 5 handelen telkens over niet-lineaire problemen die opgelost moeten worden met een aanpassing van de expliciete Euler methode gebruikt om de warmte vergelijking op te lossen.

1 Code tests

De geïmplementeerde algoritmes zijn op drie verschillende manieren getest.

Eerst is er onderling vergeleken. Dit resultaat is te zien in Figuur 4. De onderlinge verschillen zitten telkens in dezelfde orde. Het tijdsinterval hier is 0.065sec.

Vervolgens is er vergeleken met de standaard methodes van Matlab [2] om de differentiaalvergelijkingen op te lossen. Dit resultaat is terug te vinden in Figuren 1, 2 en 3. Het verschil met de geïmplementeerde matlabcode en onze methodes onderling (Figuur 4) is ook van dezelfde grootteorde (voor $t = 0.065\text{sec}$).

Tot slot is er vergeleken met de analytische oplossing [1] van de warmtevergelijking. Dit leverde volgend resultaat op:

$$u(x, t) = \sum_{n=1}^{\infty} B_n \sin(n\pi x) e^{-n^2 \pi^2 t} \quad (1)$$

met

$$B_n = 2 \int_0^1 \sin(n\pi x) f(x) dx \quad (2)$$

Voor een uniforme warmte verdeling over een staaf: $f(x) = u_0$, geeft dit de termen: $B_{2n} = 0, B_{2n-1} = \frac{4u_0}{(2n-1)\pi}$. Ingevuld in 1 geeft dit:

$$u(x, t) = \frac{4u_0}{\pi} \sum_{n=1}^{\infty} \frac{\sin((2n-1)\pi x)}{(2n-1)} e^{-(2n-1)^2 \pi^2 t} \quad (3)$$

Wanneer de eerste termen voor de serie worden uitgeschreven valt op dat (voor $n \rightarrow \infty$) de termen zeer snel zeer klein worden. De tweede term is reeds e^{-8} keer kleiner dan de eerste voor $t \geq \frac{1}{\pi^2}$. Een goede benadering wordt dus bereikt wanneer vergelijking al na de eerste term wordt afgebroken (voor voldoende grote t). Wanneer $x = \frac{1}{2}$ en $t = \frac{1}{\pi^2}$ worden gekozen, geeft dit de oplossing: $u(x, t) \approx \frac{4u_0}{\pi e} = 0.47u_0$, met $u_0 = u(x, 0)$. In Figuren 5, 6 en 7 is het resultaat te zien voor begin- en randvoorwaarden $u_0 = 1, f_1(t) = 0$ en $f_2(t) = 0$ in de laatste deelfiguur ($t = 0.101$).

Bij de drie testen is gebleken dat het resultaat van onze numerieke oplossingen benaderend juist zijn. We kunnen er vanuit gaan dat het beoogde doel bereikt is.

```

1 function [x,u] = Explicit_Euler(tf,nx,nt,f1,f2,ic)
2     % with x in [0,1] and t in [0,tf]
3     % ic is initial condition u(x,t=0)
4     dx = 1/nx;
5     dt = tf/nt;
6     x = linspace(0,1,nx+1);    % x-point coordinates
7     mu = dt/(dx*dx);
8     j = 2:nx;                  % indices of interior points:2..nx
9                                % leftmost point has index 1
10                               % rightmost point has index nx+1
11     u = ic;
12     for n = 1:nt
13         u(j) = mu*u(j-1) + (1-2*mu)*u(j) + mu*u(j+1);
14         u(1) = f1(n*dt); u(nx+1) = f2(n*dt);
15     end
16 end

1 function [x,u] = Implicit_Euler(tf,nx,nt,f1,f2,ic)
2     % with x in [0,1] and t in [0,tf]
3     % ic is initial condition u(x,t=0)
4     dx = 1/nx;
5     dt = tf/nt;
6     x = linspace(0,1,nx+1);    % x-point coordinates
7     mu = dt/(dx*dx);
8     j = 2:nx;                  % indices of interior points:2..nx
9                                % leftmost point has index 1
10                               % rightmost point has index nx+1
11     a(1:nx-1) = 1+2*mu;
12     d(1:nx-2) = -mu;
13     A = diag(a) + diag(d,1) + diag(d,-1);
14     u = ic;
15     for n = 1:nt
16         u(1) = f1(n*dt); u(nx+1) = f2(n*dt);
17         b = u(j);
18         b(1) = u(2) + mu*u(1);
19         b(end) = u(nx) + mu*u(nx+1);
20         u(j) = A\b';
21     end
22 end

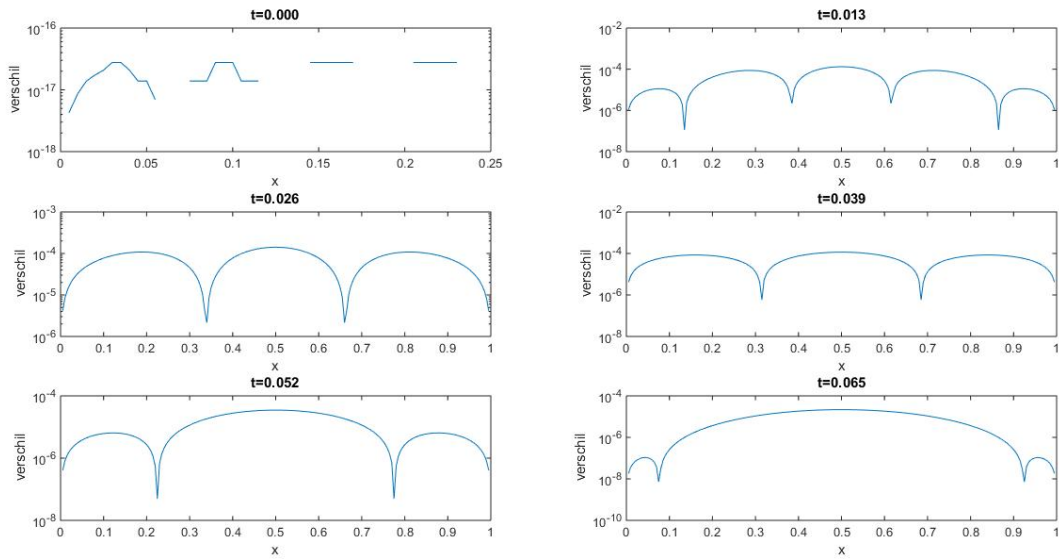
1 function [x,u] = Crank_Nicolson(tf,nx,nt,f1,f2,ic)
2     % with x in [0,1] and t in [0,tf]
3     % ic is initial condition u(x,t=0)
4     theta = 1/2;
5     dx = 1/nx;
6     dt = tf/nt;
7     x = linspace(0,1,nx+1);    % x-point coordinates
8     mu = dt/(dx*dx);
9     j = 2:nx;                  % indices of interior points:2..nx
10                                % leftmost point has index 1
11                               % rightmost point has index nx+1
12     a(1:nx-1) = 1+2*mu*theta;
13     d(1:nx-2) = -mu*theta;
14     A = diag(a) + diag(d,1) + diag(d,-1);
15     u = ic;
16     for n = 1:nt
17         un = u;
18         u(1) = f1(n*dt); u(nx+1) = f2(n*dt);
19         b = (1-2*(1-theta)*mu)*un(j) + (1-theta)*mu*(un(j-1)+un(j+1));
20         b(1) = b(1) + theta*mu*u(1);
21         b(end) = b(end) + theta*mu*u(nx+1);

```

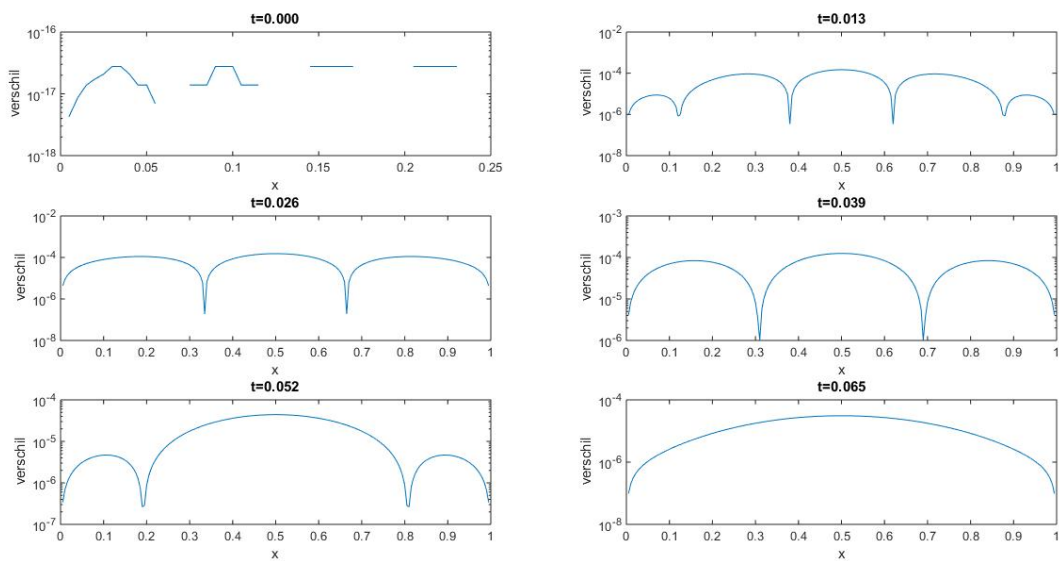
```

22     u(j) = A\b';
23     end
24 end

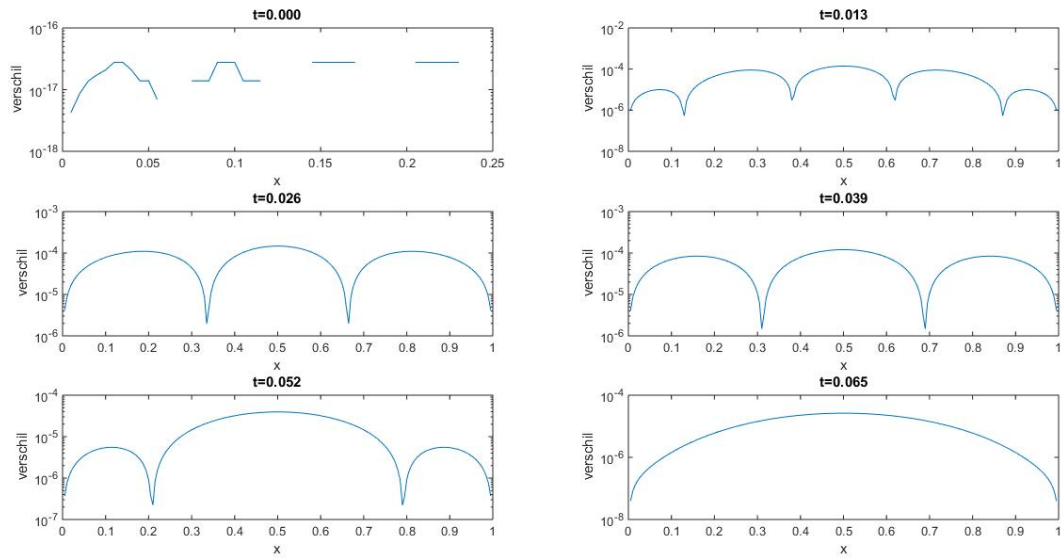
```



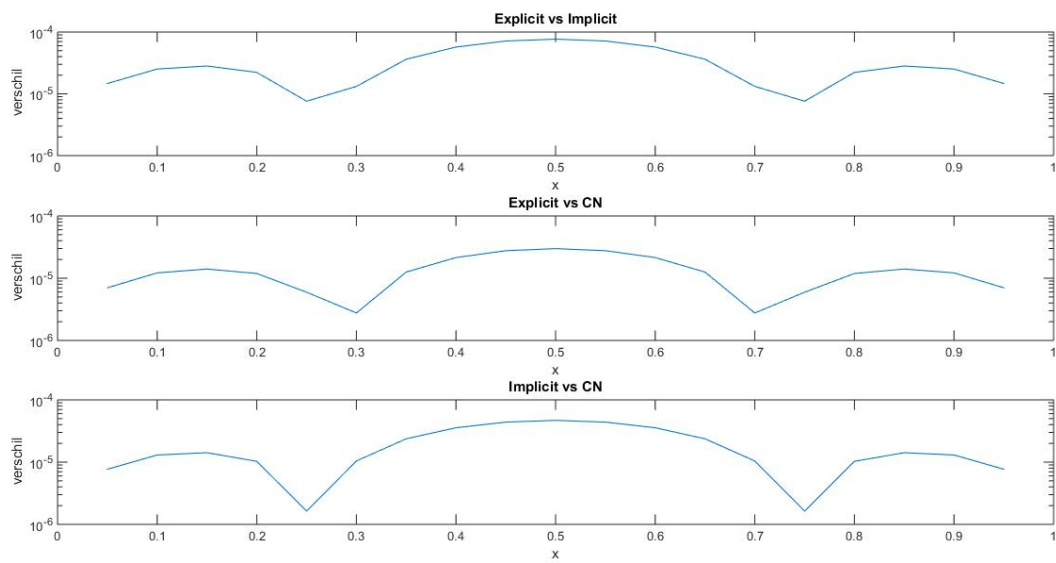
Figuur 1: Verschil expliciete Euler schema met MATLAB-solver voor warmtevergelijking met randvoorwaarden $u(0,t) = 0$, $u(1,t) = 0$ en $u(x,0) = -|x - \frac{1}{2}| + \frac{1}{2}$



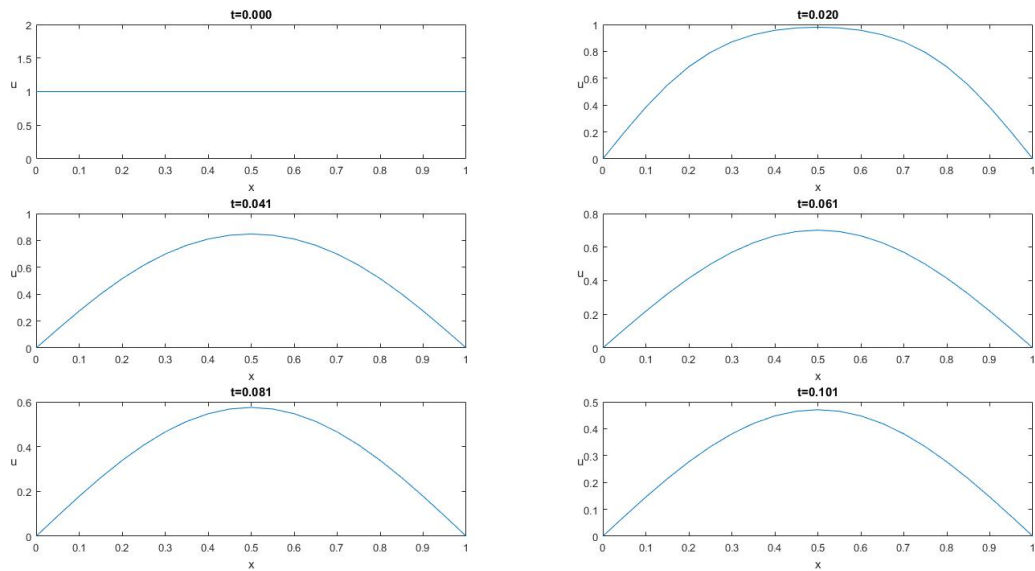
Figuur 2: Verschil impliciete Euler schema met MATLAB-solver voor warmtevergelijking met randvoorwaarden $u(0,t) = 0$, $u(1,t) = 0$ en $u(x,0) = -|x - \frac{1}{2}| + \frac{1}{2}$



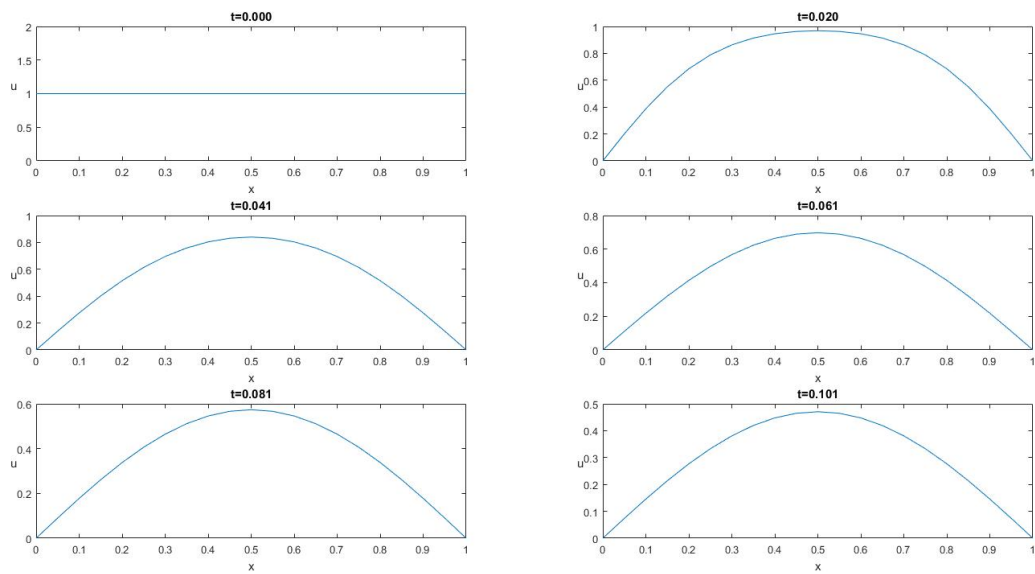
Figuur 3: Verschil Crank-Nicolson schema met MATLAB-solver voor warmtevergelijking met randvoorwaarden $u(0,t) = 0$, $u(1,t) = 0$ en $u(x,0) = -\left|x - \frac{1}{2}\right| + \frac{1}{2}$



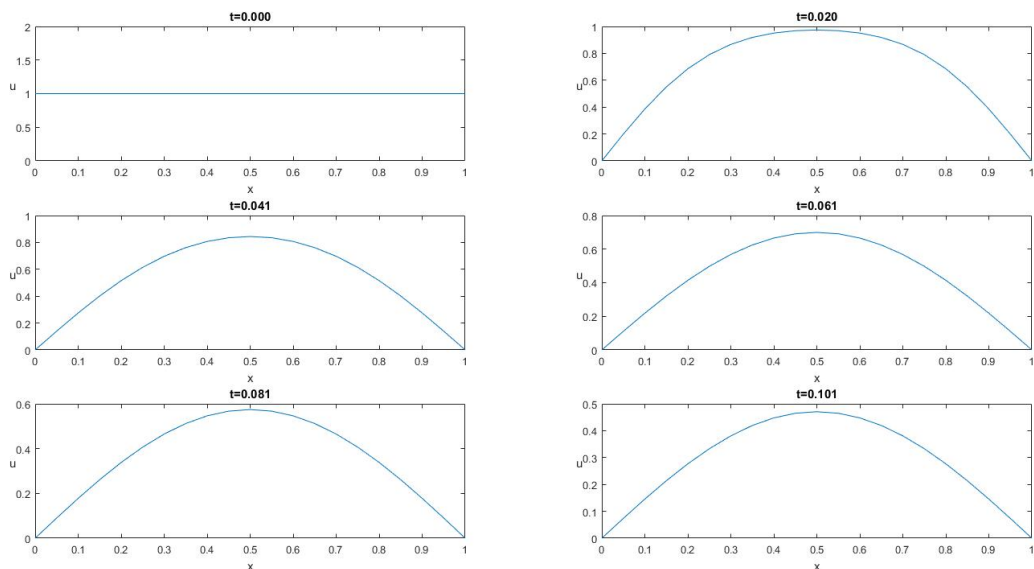
Figuur 4: De relatieve verschillen van de methodes. Voor de drie figuren geldt $t = 0.065$ sec.



Figuur 5: De oplossing van de warmte vergelijking 1, m.b.v. het expliciete Euler schema met begin- en randvoorwaarden: $u_0 = 1$, $f_1(t) = 0$ en $f_2(t) = 0$.



Figuur 6: De oplossing van de warmte vergelijking 1, m.b.v. het impliciete Euler schema met begin- en randvoorwaarden: $u_0 = 1$, $f_1(t) = 0$ en $f_2(t) = 0$.



Figuur 7: De oplossing van de warmte vergelijking 1, m.b.v. het Crank Nicolson schema met begin- en randvoorwaarden: $u_0 = 1$, $f_1(t) = 0$ en $f_2(t) = 0$.

2 Stabiliteitslimiet

Gegeven de beginvoorwaarde: $f_1(t) = 0$, $f_2(t) = 1$, $g(x) = \sin(\frac{5\pi x}{2})$. Geef de numerieke oplossing voor $t = 0.05$ en $t = 0.5$, met $\Delta x = 1/20$ en m.b.v. het expliciete Euler schema. Het resultaat is te zien in Figuur 8, voor een stabiele tijdstap en in Figuur 9 voor een instabiele tijdstap. Hier is de oscillatie als gevolg van een te grote tijdstap te zien.

Een methode is instabiel wanneer $\mu > 1/2$, een methode is stabiel wanneer $\mu \leq 1/2$ is, met $\mu := \frac{\Delta t}{(\Delta x)^2}$. Bij figuur 8 is $\Delta x = \frac{1}{20}$ en $\Delta t = \frac{1}{401}$, dit levert een μ juist iets kleiner dan $1/2$, voldoende klein voor stabiliteit. Bij figuur 9 is $\Delta x = \frac{1}{20}$ en $\Delta t = \frac{1}{399}$, dit levert een μ die juist iets groter is dan $1/2$, onvoldoende klein voor stabiliteit.

De reden dat $\mu \leq 1/2$ moet zijn wordt duidelijk door de error op de oplossing beter te bekijken. Schrijf e als de error op U (de uitkomst). Dan is: $e_j^{n+1} = (1 - 2\mu)e_j^n + \mu e_{j+1}^n + \mu e_{j-1}^n - T_j^n \Delta t$. Doordat $\mu \leq \frac{1}{2}$ Zijn de coëfficiënten van de termen in e^n allemaal positief en samengeteld gelijk aan 1. Volgens de driehoeksongelijkheid en uit het feit dat de coëfficiënten rechts positief zijn, volgt dat de maximum error: $|e_j^{n+1}| \leq E^n + T \Delta t$. Verder geldt door gegeven beginvoorwaarde: $E^0 = 0$. Hieruit volgt: $E^n \leq nT \Delta t$. De fout gaat dus naar nul voor $\Delta t \rightarrow 0$ en $\mu \leq \frac{1}{2}$.

3 Afbreekfouten

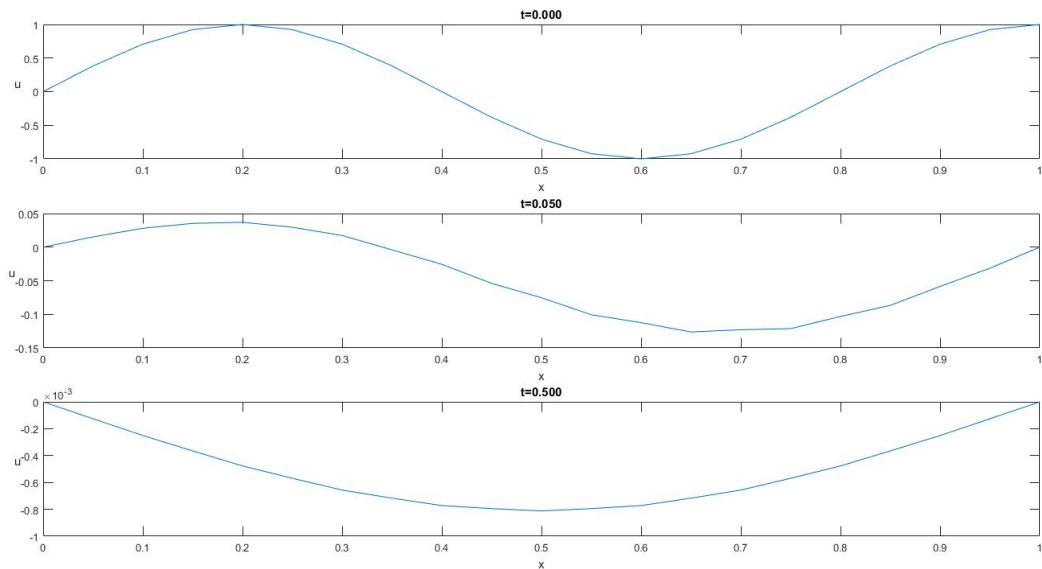
De afbreekfout voor het Impliciete Euler schema is

$$T_j^{n+\frac{1}{2}} = [-\frac{1}{2}\Delta t u_{xxt} - \frac{1}{12}(\Delta x)^2 u_{xxxx}] + \dots \sim \mathcal{O}((\Delta x)^2, \Delta t) \quad (4)$$

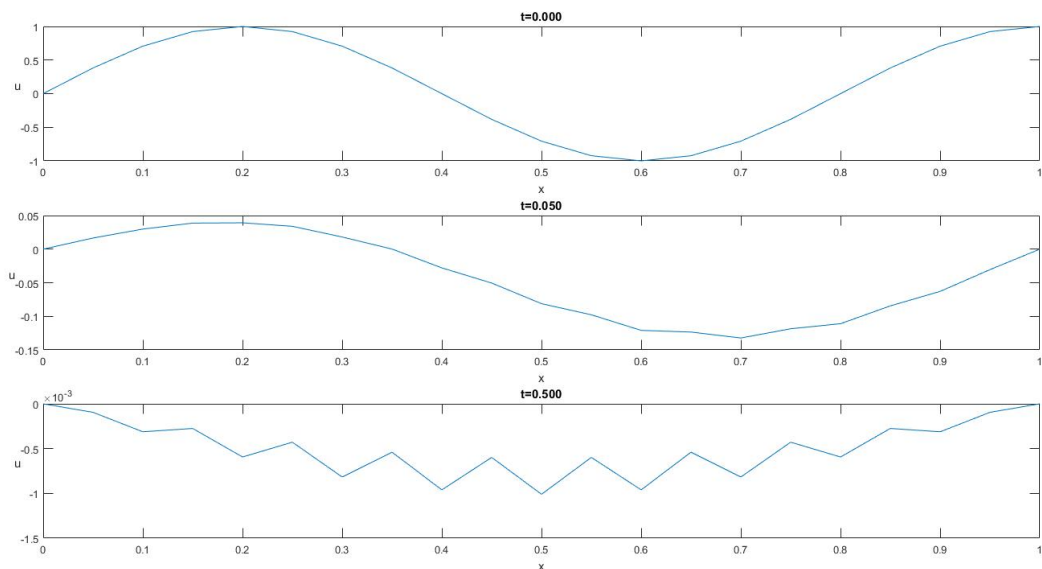
De afbreekfout voor het Crank-Nicolson schema is

$$T_j^{n+\frac{1}{2}} = -\frac{1}{12}[(\Delta x)^2 u_{xxxx} + (\Delta t)^2 u_{ttt}]_j^{n+\frac{1}{2}} + \dots \sim \mathcal{O}((\Delta x)^2, (\Delta t)^2) \quad (5)$$

Dit betekent dat als Δx en Δt van dezelfde orde grootte zijn, het Crank-Nicolson schema een kleinere afbreekfout zal hebben dan het Impliciete Euler schema. De fout verkleint met een factor 4 voor het Impliciete Euler schema als Δx halveert en Δt gedeeld wordt door 4. Daartegenover verkleint de fout met een factor 4 voor



Figuur 8: Numerieke oplossing van warmtevergelijking dat voldoet aan stabiliteitsvoorwaarde



Figuur 9: Numerieke oplossing van warmtevergelijking dat niet voldoet aan stabiliteitsvoorwaarde

het Crank-Nicolson schema als Δx en Δt halveert. Het laatste is voordeliger om het aantal tijdstappen te beperken. Dit is ook duidelijk te zien in Tabellen 1 en 2 waarin de maximale fout voor $(\Delta x, \Delta t)$ -combinaties staan.

Wanneer ofwel Δx ofwel Δt verkleint en de andere constant blijft, betekent dit niet persé een verkleining van de norm van de fout. Het kan immers zijn dat één van beide termen dominant wordt en de fout bepaalt, zodanig dat de andere term verkleinen enkel nog een verwaarloosbaar effect heeft op de norm van de fout.

$\Delta x \backslash \Delta t$	1/20	1/40	1/80	1/160	1/320	1/640
1/10	0,011419	0,0053564	0,0026875	0,0014557	0,00086698	0,00057958
1/20	0,011045	0,0050546	0,0024243	0,0012122	0,00063339	0,00035097
1/40	0,010952	0,0049801	0,0023595	0,0011522	0,00057589	0,00029471
1/80	0,010929	0,0049616	0,0023433	0,0011373	0,00056158	0,00028070

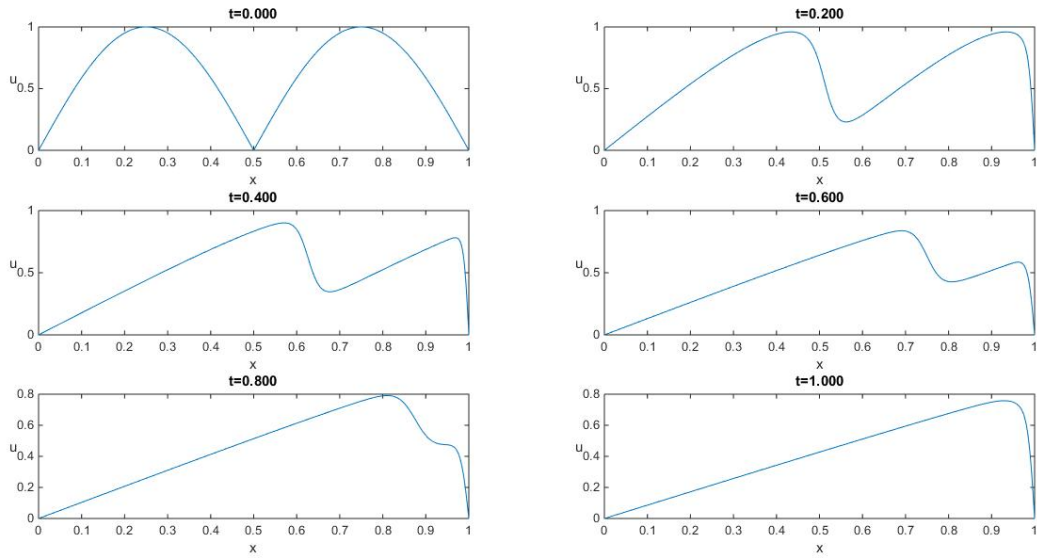
Tabel 1: Oneindig norm van de fout van de Impliciete Euler methode voor verschillende $(\Delta x, \Delta t)$ -combinaties

$\Delta x \backslash \Delta t$	1/20	1/40	1/80	1/160	1/320	1/640
1/10	0,00042502	0,00011458	0,00025121	0,000285474	0,00029404	0,00029618
1/20	0,00063983	0,00010687	2,8129e-05	6,1989e-05	7,04609e-05	7,2579e-05
1/40	0,00069265	0,00016136	2,6757e-05	7,0006e-06	1,5446e-05	1,7558e-05
1/80	0,00070580	0,00017492	4,0425e-05	6,6918e-06	1,7481e-06	3,8586e-06

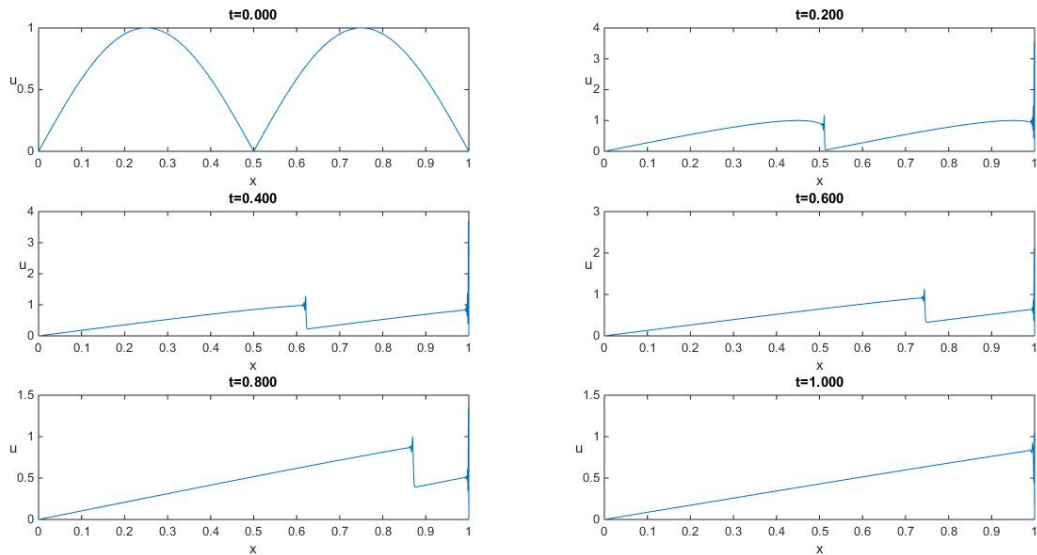
Tabel 2: Oneindig norm van de fout van de Crank-Nicolson methode voor verschillende $(\Delta x, \Delta t)$ -combinaties

4 Visceuze Burgersvergelijking

Gevraagd is om het niet-lineaire probleem, de Viscous Burgers equation op te lossen: $\frac{\partial u}{\partial t} = b \frac{\partial^2 u}{\partial x^2} - u \frac{\partial u}{\partial x}$.



Figuur 10: Numerieke oplossing van de Viscous Burgers equation voor $b = 0.005$, met behulp van expliciet schema.



Figuur 11: Numerieke oplossing van de Viscous Burgers equation voor $b = 0.0001$, met behulp van expliciet schema.

De extra term bij de differentiaalvergelijking is een niet-lineaire transportterm en zorgt er voor dat de grafiek meer en meer naar rechts verschuift, zoals te zien is in Figuur 10. Aangezien de transportterm gelijk is aan $-u \frac{\partial u}{\partial x}$ gaan de delen waar $u(x,t)$ groter is sneller naar rechts verschuiven. Dit zorgt voor de scheefgetrokken structuur van de grafiek. De term $b \frac{\partial^2 u}{\partial x^2}$ bepaalt de viscositeit van het systeem waardoor $u(x,t)$ uiteindelijk na verloop van tijd zal convergeren naar de constante randvoorwaarden. Om een juiste oplossing te bekomen, zijn kleine waarden voor $\Delta x (= \frac{1}{1000})$ en $\Delta t (= \frac{1}{1500000})$ genomen.

In Figuur 11 is de oplossing van dezelfde vergelijking te zien met dezelfde Δx en Δt maar een kleinere b . Er is duidelijk te zien dat $u(x,t)$ minder snel daalt door de kleinere b want als $b = 0$ is het systeem niet-visceus en conservatief. Verder valt het op dat er schokken ontstaan waar $\frac{\partial u}{\partial x}$ groot wordt.

```

1 function [x,u] = Viscous_Burgers(tf,nx,nt,f1,f2,ic,b)
2     % with x in [0,1] and t in [0,tf]
3     dx = 1/nx;
4     dt = tf/nt;
5     x = linspace(0,1,nx+1);           % x-point coordinates
6     mu = dt/(dx*dx);
7     j = 2:nx;                         % indices of interior points:2..nx
8                                     % leftmost point has index 1
9                                     % rightmost point has index nx+1
10    u = ic;
11    for n = 1:nt
12        u(j) = (mu*u(j-1) - 2*mu*u(j) + mu*u(j+1)).*b(j) - (u(j)*dt/(2*dx)).*(u(j)
13            +1)-u(j-1))+u(j);
14        u(1) = f1(n*dt); u(nx+1) = f2(n*dt);
15    end
end

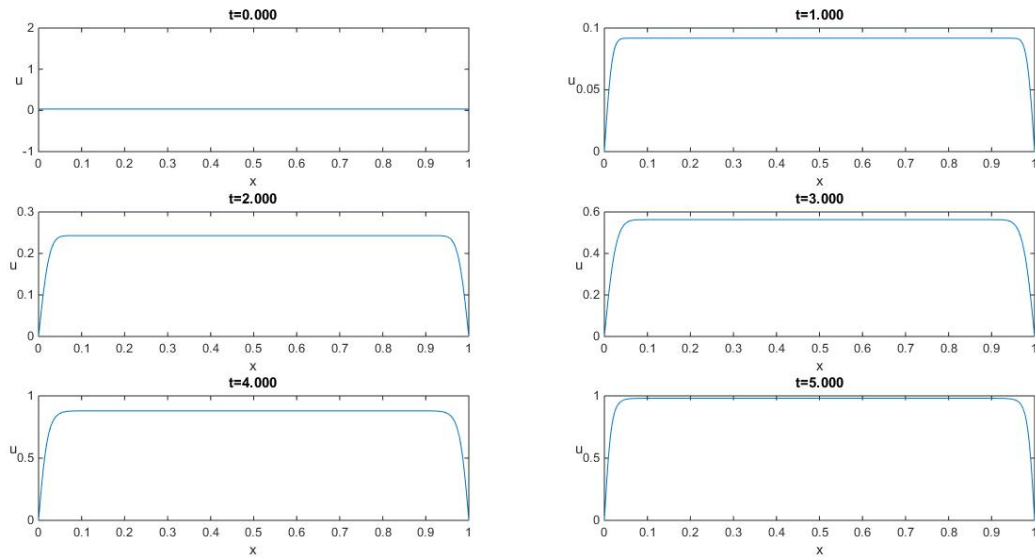
```

5 Een diffusie reactie

Gevraagd wordt om het niet-lineaire probleem, phase-field equation, op te lossen. $\frac{\partial u}{\partial t} = b \frac{\partial^2 u}{\partial x^2} - (u^3 - u)$ met $b = 10^{-4}$.

De oplossing van deze vergelijking is geïllustreerd in Figuur 12. Aangezien b zeer klein is, zal de term $-(u^3 - u)$ de grootste impact op de oplossing hebben, tenzij $\frac{\partial^2 u}{\partial x^2}$ zeer groot is. Op figuur 12 is te zien dat enkel aan de randen $\frac{\partial^2 u}{\partial x^2}$ de oplossing niet verschillend is van 0. De term $-(u^3 - u)$ is positief voor $u < 1$ en negatief voor

$u > 1$. Dus $u(x,t)$ zal overal buiten aan de randen convergeren naar 1. Om zeker een juiste convergentie te hebben, zijn kleine waarden voor $\Delta x (= \frac{1}{1000})$ en $\Delta t (= \frac{1}{1500000})$ genomen.



Figuur 12: Numerieke oplossing van de fase-veld vergelijking, met behulp van expliciete Euler.

```

1 function [x,u] = Phase_Field_Equation(tf,nx,nt,f1,f2,ic,b)
2     % with x in [0,1] and t in [0,tf]
3     dx = 1/nx;
4     dt = tf/nt;
5     x = linspace(0,1,nx+1); % x-point coordinates
6     mu = dt/(dx*dx);
7     j = 2:nx; % indices of interior points:2..nx
8               % leftmost point has index 1
9               % rightmost point has index nx+1
10    u = ic;
11    for n = 1:nt
12        u(j) = (mu*u(j-1) - 2*mu*u(j) + mu*u(j+1)).*b(j) - (u(j)*dt).*((u(j)).^2 - 1)
13            + u(j);
14        u(1) = f1(n*dt); u(nx+1) = f2(n*dt);
15    end
end

```

Referenties

- [1] Matthew J. Hancock, The 1-D Heat Equation, Fall 2006, url: <https://ocw.mit.edu/courses/mathematics/18-303-linear-partial-differential-equations-fall-2006/lecture-notes/heateqni.pdf>
- [2] function pdepe, Solve initial-boundary value problems for parabolic-elliptic PDEs in 1-D, url: <https://nl.mathworks.com/help/matlab/ref/pdepe.html>