

Parabolische PDEs

Wärmeleitungsgleichung

Wir betrachten (zunächst) die parabolische partielle Differentialgleichung

$$u_t(x, t) = u_{xx}(x, t), \quad x \in (x_l, x_r), \quad t \in [0, T].$$

Für $t = 0$ gelte die Anfangsbedingung

$$u(x, 0) = u_0(x), \quad x \in [x_l, x_r].$$

An den Intervallenden seien Dirichlet-Bedingungen vorgegeben

$$u(x_l, t) = u_l(t), \quad u(x_r, t) = u_r(t), \quad t \in [0, T].$$

Später erlauben wir folgende Modifikationen

- Inhomogenität des Materials $c_v u_t = (\kappa u_x)_x$,
- Wärmequellen $c_v u_t = (\kappa u_x)_x + r$,
- Neumann-Randbedingungen
 $-\kappa u_x(x_l, t) = q_l(t)$ bzw. $-\kappa u_x(x_r, t) = q_r(t)$.

Konvektion-Diffusion-Reaktion

Bemerkung

Die Differentialgleichung beschreibt auch Diffusionsprozesse.

Die Kombination mit Gleichungen erster Ordnung (Transportgleichung) ergibt die Konvektions-Diffusionsgleichung (Transportgleichung)

$$u_t(x, t) + au_x(x, t) = b + cu_{xx}(x, t).$$

Je nach Form von b können auch noch Reaktionen eine Rolle spielen. Ferner können mehrere Substanzen diffundieren (migrieren) $\rightsquigarrow$ Systeme gekoppelter Gleichungen.

Gleichgewicht

Dimension

Themenvorschau

Bemerkung

Neben der Lösung des Anfangs-Randwertproblems für die formulierten PDEs interessieren uns Gleichgewichtszustände, d.h. stationäre Lösungen

$$\begin{aligned} u(x, t) &= y(x), \\ u_t(x, t) &= 0. \end{aligned}$$

Diese erfüllen im räumlich eindimensionalen Fall die gewöhnliche Differentialgleichung (ODE)

$$ay'(x) = b + cy''(x).$$

Die Raumdimension wird in der Praxis meist eher zwei oder drei sein. Dann ist ∂_{xx} durch Δ zu ersetzen. Formal gibt es also die Variablen (x_1, x_2, x_3, t) , also vier. Dennoch sprechen wir von einem (räumlich) dreidimensionalen Problem.

Mit Gebiet Ω ist die Menge aller $(x_1, x_2, x_3) \in \mathbb{R}^3$ gemeint.

Der Rand $\Gamma = \partial\Omega$ ist der räumliche Rand.

Es werden Anfangs-Randwert-Probleme betrachtet.

Der Rand des Bereichs aller (x_1, x_2, x_3, t) spielt keine Rolle – nur der Boden des Zylinders bei $t = t_{ini}$.

(Keine Endbedingung)

Bemerkung

Die elliptischen Probleme in 2d (letztes Kapitel) sind gerade die Gleichgewichtsprobleme der parabolischen Evolutionsprobleme

$$u_t = \Delta u + r = 0 \quad \rightsquigarrow \quad -\Delta u = r$$

- explizite Methode
- Stabilitätsschranke
- implizite Methode
- Crank-Nicolson Schema
- Konsistenz, Stabilität, Konvergenz
- Gleichgewicht
- Dimension

Explizite FDM

Konventionen

Simplex Beispiel (analytisch)

Es liegt nahe die FDM direkt anzuwenden, also mittels finiter Differenzen Zeit- wie Ortsableitungen zu diskretisieren.

Im einfachsten Fall, also für $u_t = u_{xx}$, ergibt sich

$$\frac{u_j^{i+1} - u_j^i}{k} = \frac{u_{j-1}^i - 2u_j^i + u_{j+1}^i}{h^2}.$$

(4-Punkte 'Stern' – stencil)

Für Randpunkte ($j = 1$ und $j = n + 1$) werden die Dirichletbedingungen gesetzt, die erste Zeitschicht ($i = 1$, $t = t_{ini}$) wird aus den Anfangsbedingungen determiniert.

Bemerkung

Für $t = t_{ini}$ (fast immer 0) und $x \in \{x_l, x_r\}$ schließen wir zunächst Konflikte aus, d.h.

$$u_0(x_l) = u_l(0), \quad u_0(x_r) = u_r(0).$$

Der obere Index bezeichne den Zähler der Zeitschichten.

Untere Indizes (einer, zwei, drei) zählen die Gitterpunkte in den Raumdimensionen.

Die Laufbereiche können variieren (Indizes von 0 bis n , bis $n - 1$, $n + 1$, Randpunkte mitgezählt oder nicht – je nach Aufgabenstellung ... und Programmierungsumgebung).

(Hier so, dass $h = (x_r - x_l)/n$)

Bemerkung

Die angegebene FDM-Formulierung wird als FTCS-Schema bezeichnet (forward time - centered space) und kann auf mehrere Dimensionen übertragen werden.

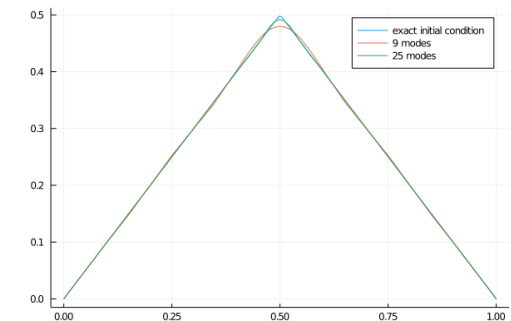


Abb. 14: Sinus-Reihe für Dreieckspuls

Example

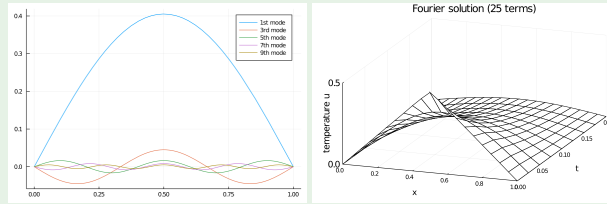


Abb. 15: Fourier-Lösung für gegebene Anfangsbedingung

Das Beispiel demonstriert das rapide Schrumpfen der höheren 'Moden'.

Das 1d-Modellproblem kann nach Diskretisierung in die Form

$$\frac{u^{i+1} - u^i}{k} = -A_h u^i \quad \rightsquigarrow \quad u^{i+1} = (I - kA_h)u^i$$

gebracht werden.

(Bei nichttrivialen Quellen/Randbedingungen sind entsprechende Modifikationen nötig.)

Die Spektrum von A_h reicht (fast) über das gesamte Intervall $h^{-2}(0, 4)$. (in 2d bis $8h^{-2}$, in 3d bis $12h^{-2}$).

(Übung: Gib EW und EV für Intervall, Rechteck, Quader an – besonders die größten!)

Folgerung: Wenn $4k/h^2 > 2$ kommt es für nichttriviale Anfangsbedingungen zu unkontrolliertem oszillatorischen Wachstum. Es ist die CFL-Bedingung zu beachten (je nach Dimension und Koeffizienten anpassen!):

$$k \leq \frac{h^2}{2}$$

```
% initial state
g=113; u0=@(x) exp(-g*(x-(xl+xr)/2).^2);
% boundary conditions
ul=@(t) 1.0*sin(2*pi*t); ur=@(t) 1.0*(1+tanh(10*(t-0.5)));
U(1,:)=u0(x);
% main loop
for i=1:m
    U(i+1,1)=ul(t(i+1));
    for j=2:n
        U(i+1,j)=U(i,j)+k/h^2*(U(i,j-1)-2*U(i,j)+U(i,j+1));
    end
    U(i+1,n+1)=ur(t(i+1));
end
```

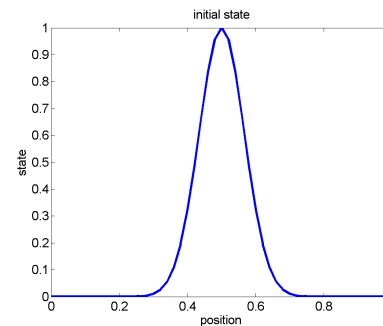
Bemerkung

In C, Python, Julia sind lediglich die Klammern eckig und gegebenenfalls die Schleifen und Deklarationen ...

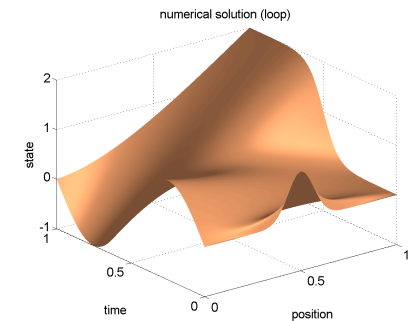
Julia-Code

```
using Plots;
xl=0.0; xr=1.0;
# initial state
g=113;
u0(x)=exp(-g*(x-(xl+xr)/2)^2);
# boundary conditions
ul(t)=1.0*sin(2*pi*t);
ur(t)=1.0*(1+tanh(10*(t-0.5)));
m=1000; k=1.0/m; t=0.0:k:1.0;
n=20; h=(xr-xl)/n; x=xl:h:xr;
U=NaN*zeros(m+1,n+1);
U[1,:]=u0(x);
# main loop
for i=1:m
    U[i+1,1]=ul(t[i+1]);
    for j=2:n
        U[i+1,j]=U[i,j]+k/h^2*(U[i,j-1]-2*U[i,j]+U[i,j+1]);
    end
    U[i+1,n+1]=ur(t[i+1]);
end
surface(x,t[1:10:m],U[1:10:m,:],colorbar=false,st=:wireframe,
        xlabel="x",ylabel="t")
```

Anfangsbedingung

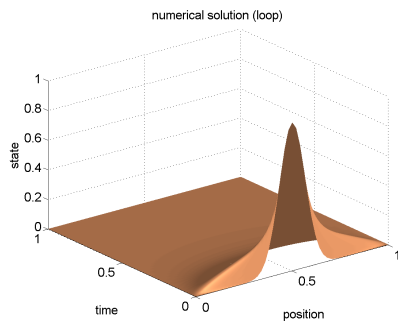


Lösung



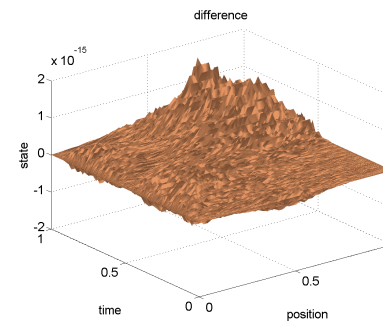
Lösung – homogene RB

Offenbar ist der Einfluss der Anfangsbedingung nur transient. Wir demonstrieren dies durch Einfrieren von Null-Randbedingungen.



Differenz

Nutzt man für den Zeitschritt eine Matrix-Vektor-Multiplikation, ergibt sich bis auf minimales Rauschen dieselbe Lösung:



Differenzenmatrix

Im 1d-Fall lautet die Systemmatrix $A_h = -h^{-2}D$ mit

$$D = \begin{pmatrix} -2 & 1 & 0 & \dots & \dots \\ 1 & -2 & 1 & 0 & \dots \\ \vdots & & & & \\ \dots & 0 & 1 & -2 & 1 \\ \dots & 0 & 1 & -2 \end{pmatrix}$$

Die entsprechenden räumlichen Diskretisierungsmatrizen im 2d-Fall haben die bekannte Struktur, Vorzeichenverteilung und Dominanzeigenschaften. Dimension bei der obigen Wahl der Implementierung $(n-1) \times (n-1)$.

Wir setzen Dirichlet-Bedingungen und $r = 0$ voraus.
Die Ortsableitung wird mittels der Gitterwerte im neuen Zeitschritt berechnet – welche noch unbekannt sind.
Es ergibt sich die Beziehung

$$u^{i+1} = u^i + \frac{k}{h^2} D u^{i+1},$$

und nach Auflösung

$$u^{i+1} = \left(I - \frac{k}{h^2} D \right)^{-1} u^i.$$

(Bitte sinnvoll implementieren!)

Bemerkung

LGS nur für die inneren Gitterwerte.

Aufgabe: Eigenschaften der Matrix untersuchen (Wdh.).

Auf Fall $r \neq 0$ erweitern.

Es sei $\sigma \in [0, 1]$.

Wir untersuchen nun

$$u^{i+1} = u^i + \frac{k}{h^2} D \left(\sigma u^{i+1} + (1 - \sigma) u^i \right).$$

Von besonderem Interesse ist der Fall $\sigma = 1/2$.

Die Fehlerordnung in der x -Ableitung ist $O(h^2)$.

Der Fehler bei der Zeitableitung ist für $\sigma = 0.5$ der Ordnung $O(k^2)$, sonst nur $O(k)$.

Wir unterteilen den Zeitschritt in zwei Halbschritte, führen einen implizit aus, den zweiten explizit.

In Matrixnotation ergibt sich

$$u^{i+1} = \left(I - \frac{k}{2h^2} D \right)^{-1} \left(I + \frac{k}{2h^2} D \right) u^i.$$

Man beachte, dass die Inverse nicht berechnet wird, sondern ein (tridiagonales) Gleichungssystem gelöst wird.

Das Verfahren ist zweiter Ordnung in Orts- wie Zeitrichtung.

Die Schrittweitenbeschränke ist zweimal größer als beim expliziten Verfahren: $k \leq h^2$.

Inhomogenität

$$u^{i+1} = \left(I - \frac{k}{2h^2} D \right)^{-1} \left(\left(I + \frac{k}{2h^2} D \right) u^i + k(r^i + r^{i+1})/2 \right).$$

Konvergenz

Alle angegebenen Verfahren sind nach Konstruktion konsistent.
Um Konvergenz zu beweisen, ist also lediglich noch ihre Stabilität nachzuweisen.

Wir leiten eine Abschätzung der Norm der Lösung durch die Normen der Eingangsdaten her.

Theorem (Stabilitätsabschätzung)

Für die Lösung der Gleichung $u_t = u_{xx} + r$, $x \in (0, 1)$ mit der Anfangsbedingung $u(\cdot, 0) = u_0$ nach dem gewichteten Verfahren mit $\sigma \in [0, 1]$, $1 - 2(1 - \sigma)k/h^2 \geq 0$, gilt:

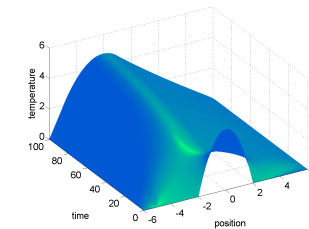
$$\|u^i\|_{\infty} \leq \|u_0\|_{\infty} + k \sum_{l=0}^i \|r^l\|_{\infty}.$$

Beweis: Man benutze, dass $-D$ eine M-Matrix ist. (ÜA)

Test (eindimensional)

Beispiel

```
c=0.1; kappa=0.03; a=kappa/c;
x_left=-6; x_right=6; n=351;
x=linspace(x_left,x_right,n); h=x(2)-x(1);
% initial temperature distribution
u0=max(0,4-x.*x);
```



Test (zweidimensional)

Als Test lösen wir das folgende Beispiel und stellen den Zustand zu einem ausgewählten Zeitpunkt dar.

Beispiel

$$u_t = \Delta u(x, y) + r \quad \text{für } x \in (x_l, x_r), y \in (y_l, y_r).$$

Daten: $x_l = y_l = 0$, $x_r = 0.9$, $y_r = 1.3$, $r=12$.

Auf dem Rand seien homogene Dirichlet-Bedingungen gegeben.

Lösung des zweidimensionalen Tests

Die Lösung hängt von der Zeit ab – wir präsentieren hier einen Frame der Animation:

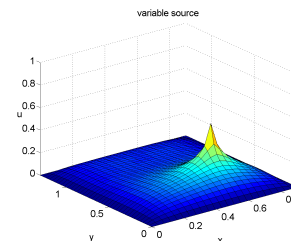


Abb. 16: Momentaufnahme

Der Hotspot (heating spot) sticht ins Auge.

Semi-Diskretisierung

Alternativ können wir nur die Ortsableitung diskretisieren, und so statt einer PDE ein System von ODEs erster Ordnung aufstellen, welches anschließend mit einem geeigneten Solver zu lösen ist.

Die Startwerte ergeben sich aus der Evaluierung von u_0 auf dem Ortsgitter.

Für $u_t = u_{xx}$, ergibt sich

$$\dot{u}_j(t) = \frac{u_{j-1}(t) - 2u_j(t) + u_{j+1}(t)}{h^2}.$$

für alle inneren Punkte, $x_j = x_l + (j - 1)h$, $j = 2, 3, \dots, n - 1$.

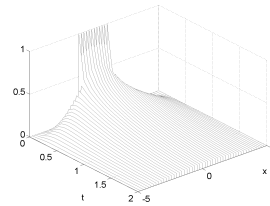
Für Randpunkte x_l und x_r ($j = 1$ bzw. $j = n + 1$) werden die Dirichlet-Bedingungen gesetzt, d.h. $u_1(t) = u_l(t)$, $u_{n+1}(t) = u_r(t)$.

Für $t=0$ setzen wir $u_j(0) = u_0(x_l + (j - 1)h)$.

Bemerkung

Man spricht auch von der Linienmethode (MoL – method of lines).

- Diffusion, Konvektion, Produktion
- Behandlung der Ränder
- ODE-Solver
- Numerische Experimente



Lösung der parabolischen PDE

$$u_t = u_{xx}$$

mit stückweise konstanten Anfangsbedingungen und homogenen Dirichlet-Randbedingungen, gelöst mit Semi-Diskretisierungsmethode (51 Gitterpunkte in x-Richtung).

```
a=0.0; b=2.0*pi; h=pi/12; n=25;
xgrid=range(a,stop=b,length=n);
p=[0.2,0.01];
U0=collect(0.0*xgrid);
ul(t)=12.0*t/(100.0+t);
ulprime(t)=0.12/(1.0+0.01*t)^2;
r(x)=abs(x-1.5)<=0.5 ? 0.4 : 0.0;
tf=1000.0;
function rhs!(Up,U,p,t)
    Up[1]=ulprime(t); # derivative of left bdy condition
    for j=2:n-1
        Up[j]=(p[2]*(U[j-1]-2*U[j]+U[j+1])/h^2+r(xgrid[j]))/p[1];
    end
    Up[n]=p[2]*(2*U[n-1]-2*U[n])/h^2/p[1];
    return;
end;
dynprob=ODEProblem(rhs!,U0,(0.0,tf),p);
dynsol=solve(dynprob);
```

Lösung (julia)

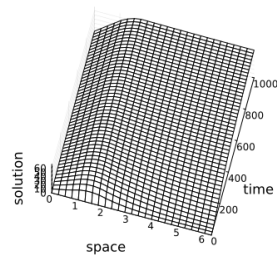


Abb. 17: Lösung mit Liniemethode

Inverse Probleme

Man betrachte das ARWP für die Wärmeleitungsgleichung in $\mathbb{R}^1$ mit zwei verschiedenen Anfangsbedingungen.

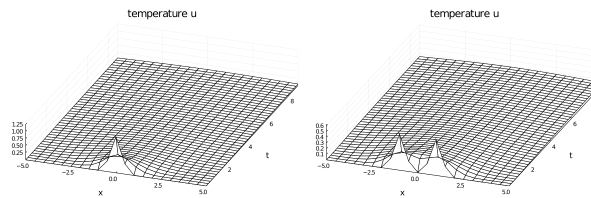


Abb. 18: Vergleich zweier Lösungen

Es zeigt sich, dass die Unterschiede rapide verschwinden.

Momentaufnahmen

Die Momentaufnahmen bei $t = 0.25$ und $t = 2.5$ zeigen, dass die Details der Anfangsverteilung schnell verschmieren, was zählt, ist die Gesamtenergie.

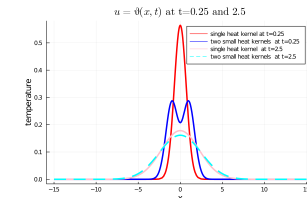


Abb. 19: Vergleich

Bemerkung

Die Rückwärtsverfolgung in der Zeit, das heißt die Lösung des ARWP für $u_t = -\Delta u$, ist ill-posed.

Hyperbolische PDE