

High-Performance Computing

Fallstudie PDEPack

Berechnungen für eine Kühlrippe

1 Ein einfaches Modell einer Kühlrippe

Wir betrachten eine Kühlrippe, die von einer Seite durch eine Wärmequelle aufgeheizt wird. Das könnte zum Beispiel der Prozessor einer CPU sein. Wir untersuchen die Wärmeleitung innerhalb dieses Bauteils. In Abbildung 1 ist das Modell grafisch dargestellt und in ein Koordinatensystem eingefasst. Wir werden meist davon ausgehen, dass das Modell sowohl in x - als auch in y -Richtung die Länge 1 hat, das macht die Beschreibung einfacher.

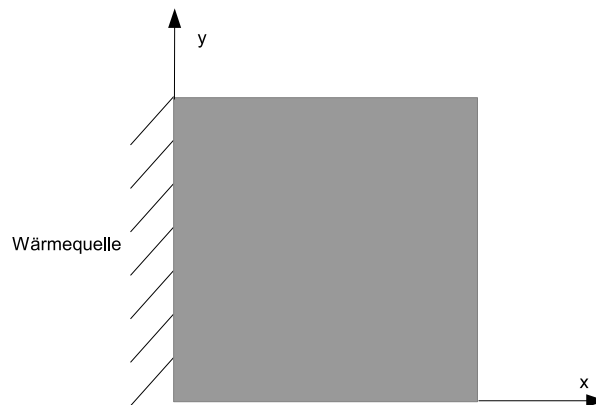


Abbildung 1: Ein einfaches Modell einer Kühlrippe

Wir hatten bereits die Drahtgleichung betrachtet. Damit hatten wir die Verformung eines eingespannten Drahtes durch das Einwirken einer Kraft berechnet. Wir hätten diese Berechnung auch als das Ergebnis der Verteilung der Wärme in einem Stab interpretieren können, der aus einem homogenen Material hergestellt ist.

2 Ein stationäres Modell

Als stationär bezeichnet man ein Modell, in dem die Zeit keinen Einfluss hat. Das bedeutet, dass wir eine Wärmequelle modellieren, die einen konstanten Wert in unsere Kühlrippe einbringt. Den instationären Fall, also eine Wärmequelle, deren Temperatur sich über die Zeit verändert betrachten wir im nächsten Abschnitt. Eine stationäre Temperaturverteilung kann man mit Hilfe einer elliptischen partiellen Differenzialgleichung, genauer eine Poisson-Gleichung, modellieren. Die Temperatur innerhalb unserer Kühlrippe beschreiben wir als Funktion u , die von x und y abhängt. Dann betrachten wir das Modell

$$\frac{\partial^2}{\partial x^2}u(x, y) + \frac{\partial^2}{\partial y^2}u(x, y) + f(x, y) = 0, 0 < x, y < 1 \quad (1)$$

$$u(0, y) = C, 0 \leq y \leq 1, u(x, 0) = u(x, 1) = 0, 0 < x < 1, u(1, y) = 0, 0 < y \leq 1. \quad (2)$$

Die Gleichungen in (2) beschreiben die Funktionswerte von u auf dem Rand des Gebiets der Kühlrippe. Der Wert 0 ist dabei willkürlich, wir könnten auch eine andere Umgebungstemperatur verwenden, aber der Wert 0 macht später die Gleichungen einfacher. Der Wert C am

linken Rand der Kühlrippe, bei $x = 0$, ist die Wärme, die die Wärmequelle in die Kühlrippe abgibt. Sie ist als konstant angenommen – wir hatten bereits festgehalten, dass wir einen stationären Fall betrachten.

Wir ersetzen das Quadrat $[0, 1] \times [0, 1]$, die Kühlrippe, durch eine endliche Anzahl von Punkten. Dazu wählen wir eine Anzahl n von Punkten in x - bzw. y -Richtung und erhalten damit Punkte wie

$$x_i = i \cdot h, h = \frac{1}{n}, i = 0, \dots, n \text{ und } y_i = i \cdot h, h = \frac{1}{n}, i = 0, \dots, n$$

Wir suchen die Funktionswerte $u_{i,j} = u(x_i, y_j)$ an diesen Punkten. In Abbildung 2 wurde $n = 6$ verwendet. Die grün markierten Punkte sind Stellen, an denen wir auf Grund der Randbedingungen die Funktionswerte $u_{i,j}$ bereits kennen. Für die rot dargestellten Punkten in der Abbildung müssen wir die Differentialgleichung lösen und damit die Temperaturwerte bestimmen. In der Praxis werden wir auch nicht $n = 6$, sondern deutlich höhere Werte für n verwenden.

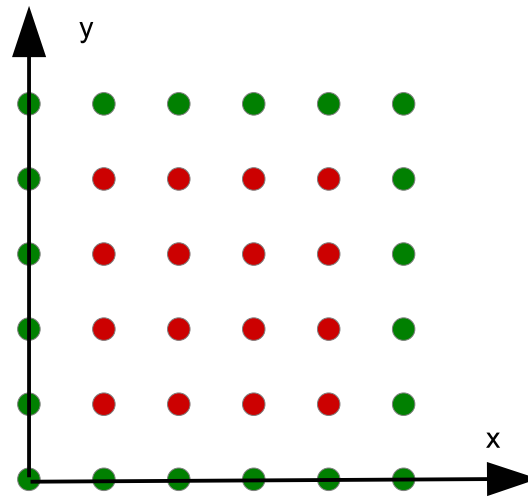


Abbildung 2: Die diskretisierte Kühlrippe

Wir müssen die partiellen Ableitungen in (1) durch eine diskrete Näherung ersetzen. Dazu verwenden wir Differenzenquotienten. Die erste partielle Ableitung in x -Richtung an einem Punkt $P = (x_i, y_j)$ können wir mit dem linken und rechten Nachbarn annähern:

$$\frac{\partial}{\partial x} u(x_i, y_j) \approx \frac{u(x_{i+1}, y_j) - u(x_{i-1}, y_j)}{2h}.$$

Wir könnten diesen Ansatz weiter verwenden, um die zweite Ableitung anzunähern. Genauso können wir die Ableitungen in y -Richtung annähern. In Abbildung 3 sind die Nachbarn, die wir für diese Näherung verwenden nochmals dargestellt. Man spricht häufig von den Nachbarn im „Westen“, im „Süden“, im „Osten“ und „Norden“, wie in der Abbildung dargestellt.

Setzen wir diese Differenzenquotienten in (1) ein und fassen zusammen, dann erhalten wir im Punkt P die Gleichung

$$4 \cdot u_P - u_N - u_W - u_S - u_E = -h^2 f_P$$

Dabei ist f_P der Funktionswert $f(x_i, y_j)$. Mit den Indices i und j schreiben wir dafür

$$4 \cdot u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1} = -h^2 f_{i,j}.$$

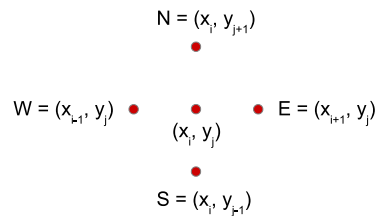


Abbildung 3: Nachbarn eines Punkts (x_i, y_j) für die Annäherung der zweiten partiellen Ableitung

Auch wenn die Unbekannten Werte x zwei Indices haben erkennen wir schnell, dass wir eine lineare Gleichung aufgestellt haben. Dies machen wir für alle i und j -Werte, für die wir die Funktionswerte suchen. Für $n = 6$ wie in Abbildung 2 erhalten wir so $5^2 = 25$ Gleichungen für genau so viele Unbekannte. Dabei benötigen wir auch die „grünen“ Funktionswerte, die wir dann auf die rechte Seite schreiben können. In Abbildung 4 sind die Punkte für $n = 6$ nochmals dargestellt.

$x_{0,0}$	$x_{0,1}$	$x_{0,2}$	$x_{0,3}$	$x_{0,4}$	$x_{0,5}$	$x_{0,6}$
$x_{1,0}$	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$x_{1,4}$	$x_{1,5}$	$x_{6,6}$
$x_{2,0}$	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$	$x_{2,5}$	$x_{6,6}$
$x_{3,0}$	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$x_{3,4}$	$x_{3,5}$	$x_{6,6}$
$x_{4,0}$	$x_{4,1}$	$x_{4,2}$	$x_{4,3}$	$x_{4,4}$	$x_{4,5}$	$x_{6,6}$
$x_{5,0}$	$x_{5,1}$	$x_{5,2}$	$x_{5,3}$	$x_{5,4}$	$x_{5,5}$	$x_{6,6}$
$x_{6,0}$	$x_{6,1}$	$x_{6,2}$	$x_{6,3}$	$x_{6,4}$	$x_{6,5}$	$x_{6,6}$

Abbildung 4: Die Unbekannten in unserem Gitter

Wir nummerieren unsere Unbekannte $x_{i,j}$ für $1 \leq i, j \leq 5$ mit einem einzigen Index durch. Dazu beginnen wir in der „oberen linken“ Ecke in Abbildung 4 und gehen spaltenweise vor. Der Wert $x_{1,1}$ erhält also den Index 1, gefolgt vom Index 2, der zu $x_{2,1}$ gehört. Das gleiche führen wir für die Funktionswerte $f_{i,j}$ durch. Dadurch erhalten wir einen Vektor von Unbekannten, der für $n = 6$ in Abbildung 5 dargestellt ist. Dabei haben wir die Werte, die durch die Randbedingungen bekannt sind, die also in grün dargestellt werden, durch die Werte aus den Randbedingungen ersetzt.

C	0	0	0	0	0	0
C	x_1	x_6	x_{11}	x_{16}	x_{21}	0
C	x_2	x_7	x_{12}	x_{17}	x_{22}	0
C	x_3	x_8	x_{13}	x_{18}	x_{23}	0
C	x_4	x_9	x_{14}	x_{19}	x_{24}	0
C	x_5	x_{10}	x_{15}	x_{20}	x_{25}	0
C	0	0	0	0	0	0

Abbildung 5: Die Unbekannten in unserem Gitter, mit Randwerten und als Vektor

Jetzt schreiben wir einige dieser Gleichungen exemplarisch auf, und dabei bringen wir bekannte Werte (in grün) auf die rechte Seite. Für die Unbekannte x_1 erhalten wir die lineare Gleichung

$$4 \cdot x_1 - x_2 - x_6 = -h^2 \cdot f_1 + C.$$

Für x_2 erhalten wir

$$-x_1 + 4 \cdot x_2 - x_3 - x_7 = -h^2 \cdot f_2 + C.$$

Für die Unbekannte x_{14} erhalten wir die lineare Gleichung

$$-x_9 - x_{13} + 4 \cdot x_{14} - x_{15} - x_{19} = -h^2 \cdot f_{14}.$$

Entsprechend bringen wir am rechten Rand, für Unbekannte wir x_{21} oder x_{24} , die bekannten Werte wieder auf die rechte Seite. Dies ist Ihnen als Übung überlassen.

Tip:

Schreiben Sie für den Fall $n = 6$ *alle* linearen Gleichungen auf!

Insgesamt erhalten wir 25 Gleichungen für 25 Unbekannte, wobei die Koeffizientenmatrix nur sehr wenig Einträge hat, die ungleich Null sind. Wenn wir die beiden 5×5 -Matrizen

$$B = \begin{pmatrix} 4 & -1 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & -1 & 4 \end{pmatrix}, -I = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

eingeführen, dann können wir die Koeffizientenmatrix A blockweise durch

$$A = \begin{pmatrix} B & -I & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -I & B & -I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -I & B & -I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -I & B & -I \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -I & B \end{pmatrix}$$

schreiben. Dabei ist $\mathbf{0}$ eine 5×5 -Matrix, die nur Nullen enthält. Allgemein können wir dies genau so schreiben. B_n ist eine $n \times n$ tridiagonale Matrix mit 4 in der Diagonalen und -1 in den beiden Nebendiagonalen.

Wir berechnen die Lösung unseres Randwertproblems durch Lösen des linearen Gleichungssystems mit dieser Koeffizientenmatrix. Die rechte Seite besteht aus den Funktionswerten f_i und den Randwerten, wie wir dies für x_1 bereits betrachtet hatten. Die Koeffizientenmatrix A ist eine „sparse matrix“, eine dünn besetzte Matrix. Selbst für sehr großes n liegen nur wenige Einträge ungleich Null vor. Wir kennen bereits Verfahren, die solche lineare Gleichungssysteme lösen können.

2.1 Aufgaben

Aufgabe 1

Implementieren Sie die Lösung des Randwertproblems wie

$$\frac{\partial^2}{\partial x^2} u(x, y) + \frac{\partial^2}{\partial y^2} u(x, y) - 2 = 0, 0 < x < 1 \quad (1)$$

$$u(0, y) = 1, 0 \leq y \leq 1, u(x, 0) = u(x, 1) = 0, 0 < x < 1, u(1, y) = 0, 0 < y \leq 1. \quad (2)$$

Implementieren Sie dabei Ihre Lösung so, dass die Randwerte für $u(0, y)$ in (2) und die Funktionswerte $f(x_i, y_j)$ aus einer Datei eingelesen werden können! Gehen Sie dabei davon aus, dass die Funktionswert für f konstant sind.

Verwenden Sie einen geeigneten Lösungsalgorithmus für das auftretende lineare Gleichungssystem und vermeiden Sie unbedingt unnötige Multiplikationen mit 0 oder Additionen von 0! Begründen Sie Ihre Auswahl des Löser für die linearen Gleichungssysteme, entweder durch theoretische Überlegungen oder durch Benchmarks.

Verwenden Sie mindestens drei verschiedene Werte für n , zum Beispiel $n = 10$, $n = 100$ und $n = 1\,000$. Für die Visualisierung Ihrer Lösung mit Hilfe von ParaView finden Sie im GitHub-Repo PDEPack Pipeline-Dateien und Funktionen!

Aufgabe 2

Da sowohl Jacobi- als Gauß-Seidel-Verfahren insbesondere für große Anzahlen von Unbekannten nicht schnell genug konvergieren hat man sogenannte *Successive Over-Relaxation*-Verfahren formuliert. Dazu führt man eine reelle Zahl $0 < \omega < 2$ ein und definiert ein iteratives Verfahren für die Lösung eines linearen Gleichungssystems $A \cdot \mathbf{x} = \mathbf{b}$:

$$\mathbf{x}_i^k = (1 - \omega)\mathbf{x}_i^{k-1} + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^k - \sum_{j=i+1}^n a_{ij}x_j^{k-1} \right), 1 \leq i \leq n.$$

Für $\omega = 1$ erhält man gerade das Gauß-Seidel-Verfahren zurück. Man weiß, dass für Werte ω zwischen 1 und 2 die Konvergenz des Verfahrens deutlich besser ist.

Für Koeffizientenmatrizen wie den Matrizen in Aufgabe 1 kann man optimale Werte für den Relaxationsparameter ω herleiten. Es gilt

$$\omega_{\text{optimal}} = \frac{2}{1 + \sqrt{1 - \left(\cos \left(\frac{\pi}{n+1} \right) \right)^2}}.$$

Dabei ist n die Dimension der Koeffizientenmatrix.

- Implementieren Sie eine Klasse für die Anwendung eines SOR-Verfahrens für die Lösung von linearen Gleichungssystemen!
- Verwenden Sie die Lösung aus a) für die Lösung des Randwertproblems aus Aufgabe 1! Verwenden Sie dabei den angegebenen optimalen Relaxationsparameter ω_{opt} !

Aufgabe 3

Führen Sie Benchmarks durch und vergleichen Sie die Performanz des SOR- und der anderen iterativen Verfahren für lineare Gleichungssystem in der Lösung der Randwertprobleme wie in Aufgabe 1!

3 Ein eindimensionales und instationäres Modell

Wir gehen jetzt davon aus, dass die Wärmeverteilung in unserer Kühlrippe nur von der x -Richtung abhängt und dass die Wärmequelle am linken Rand Temperaturwerte abgibt, die von der Zeit abhängen. Wir bezeichnen die Temperatur in unserer Kühlrippe an der Stelle x zum Zeitpunkt t als Funktion $u(x, t)$. Ist die Temperatur, die durch die Wärmequelle an der Stelle $x = 0$ eingeleitet wird durch die Funktion $g(t)$ gegeben, dann beschreiben wir dies durch die Randbedingung

$$u(0, t) = g(t).$$

Ein Beispiel für eine solche Randbedingung könnte eine periodische Einleitung sein:

$$u(0, t) = \sin(\pi t).$$

Zum Zeitpunkt $t = 0$, zu Beginn unserer Simulation, soll die Temperatur konstant sein, zum Beispiel Raumtemperatur. Wir vereinfachen dies und setzen

$$u(x, 0) = 0, 0 < x < 1.$$

Am rechten Rand soll unsere Kühlrippe wärme-isoliert sein. Dies drücken wir durch die Randbedingung

$$\frac{\partial}{\partial x} u(1, t) = 0, t > 0$$

aus. Wir könnten auch vorschreiben, dass die Temperatur am rechten Rand konstant ist, beispielsweise als

$$u(1, t) = 0, t > 0.$$

Die Wärmeleitung kann mit Hilfe einer partiellen Differentialgleichung beschrieben werden:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, 0 < x < 1, t = 0.$$

Insgesamt betrachten wir das eindimensionale parabolische Randwertproblem

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, 0 < x < 1, t = 0, \quad (1)$$

$$u(x, 0) = 0, 0 < x < 1, \quad (2)$$

$$u(0, t) = g(t), \frac{\partial}{\partial x} u(1, t) = 0, t > 0. \quad (3)$$

Wir könnten dabei die letzte Randbedingung in (3) durch

$$u(1, t) = 0, t > 0.$$

ersetzen.

Um eine numerische Lösung zu berechnen ersetzen wir die partiellen Ableitungen im Randwertproblem durch finite Differenzen und berechnen damit Werte $u(x_i, t_j)$ auf einem Gitter im (x, t) -Koordinatensystem. Dazu gibt es verschiedene Ansätze, die wir im Folgenden betrachten werden.

3.1 Die explizite Methode von Richardson

Um das Randwertproblem (1) - (3) numerisch zu lösen zerlegen wir das Intervall $(0, 1)$ in x -Richtung und das Zeitintervall $(0, \infty)$ in äquidistante Teile. Damit erhalten wir im (x, t) -Koordinatensystem ein Gitter wie in Abbildung 6. Das x -Intervall ist in der Abbildung durch 6 Punkte ersetzt worden. Dazu verwenden wir die Schrittweite h_x , im Bild wurde $h_x = \frac{1}{5}$ verwendet. Wir nummerieren die Punkte als x_0, \dots, x_5 durch. Allgemein, bei n Punkten, verwenden wir $h_x = \frac{1}{n}$ und wir erhalten $n + 1$ Punkte. Die grüne Farbe bedeutet, dass wir die Funktionswerte an diesen Punkten kennen, sie entsprechen den Werten

$$u(0, 0), u\left(\frac{1}{5}, 0\right), \dots, u(1, 0)$$

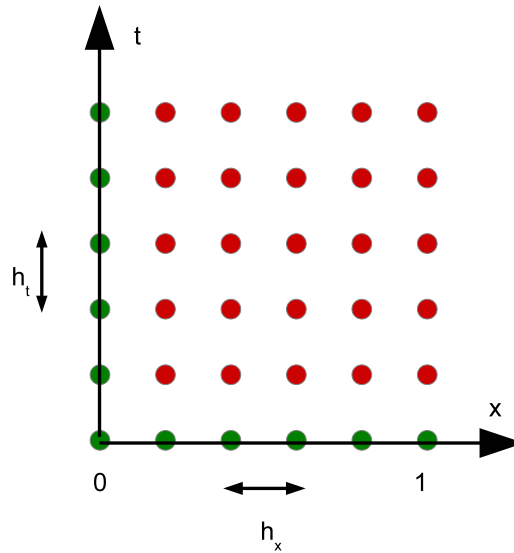


Abbildung 6: Das diskretisierte Wärmeleitungsproblem in der (x, t) -Ebene

zum Zeitpunkt $t = 0$. Die Gleichung (2) sagt aus, dass die Werte 0 sind. Für die Zeit verwenden wir eine andere Schrittweite, die wir h_t nennen. Auch hier kennen wir die Funktionswerte der Punkte auf der t -Achse, sie sind in (3) durch die Funktion $g(t)$ gegeben.

Verwenden wir die Bedingungen an die Ableitung am rechten Rand wie in (3), dann kennen wir die Funktionswerte nicht. Würden wir hier auch die Funktionswerte vorschreiben färben wir die Punkte grün.

Insgesamt wählen wir also Punkte

$$x_i = i \cdot h_x, h_x = \frac{1}{n}, i = 0, 1, \dots, n$$

und in t -Richtung

$$t_j = j \cdot h_t, j = 0, 1, 2, \dots$$

Wir bekommen damit die Punkte $P(x_i, t_j)$ und die Funktionswerte $u_{ij} = u(x_i, t_j)$. Für die partielle Ableitung nach der Zeit in (1) verwenden wir einen vorwärts gerichteten Differenzenquotienten:

$$\frac{\partial}{\partial t} u(x_i, t_j) \approx \frac{u_{i,j+1} - u_{i,j}}{h_t}. \quad (4)$$

Die zweiten Ableitungen auf der rechten Seite unserer Gleichung nähern wir mit einem symmetrischen Differenzenquotienten an:

$$\frac{\partial^2}{\partial x^2} u(x_i, t_j) \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2}. \quad (5)$$

Setzen wir die Gleichungen (4) und (5) gleich, dann erhalten wir eine diskrete Variante unserer Differenzialgleichung (1). Wir multiplizieren mit h_t und erhalten dann

$$u_{i,j+1} - u_{i,j} = \frac{h_t}{h_x^2} (u_{i+1,j} - 2u_{i,j} + u_{i-1,j}). \quad (6)$$

Die Gleichung (6) schauen wir uns genau an. Die Werte mit dem Index j sind Werte aus dem vorhergegangenen Zeitschritt. Für $j = 1$ sind dies bis auf den Wert $u_{i,1}$ dann nur Werte, die

in Abbildung 6 grün gefärbt sind – wir kennen sie durch die Randbedingungen. Das können wir aber auch für andere Werte von j – wenigstens so lange, wenn wir die Vorgängerwerte bereits berechnet haben. Wir setzen, damit im Folgenden die Gleichungen etwas übersichtlicher werden, den Parameter r als

$$r = \frac{h_t}{h_x^2},$$

lösen (6) nach $u_{i,j+1}$ auf und fassen noch zusammen:

$$u_{i,j+1} = ru_{i-1,j} + (1 - 2r)u_{i,j} + ru_{i+1,j}, i = 1, \dots, n-1, j = 0, 1, \dots \quad (7)$$

Die Randbedingungen gehen in (7) ein, bei der Berechnung von $u_{1,j+1}$ gilt

$$u_{0,j} = g(j \cdot h_t).$$

Die Randbedingung in (3) mit der partiellen Ableitung muss man separat behandeln, dafür verwenden wir

$$u_{n,j+1} = 2ru_{n-1,j} + (1 - 2r)u_{n,j}, j = 0, 1, \dots$$

Wenn wir einen Funktionswert am rechten Rand vorschreiben, dann diesen Wert einfach für $u_{n-1,j}$ in die Gleichung mit ein.

Dieses Verfahren in (7) wird *explizites Verfahren von Richardson* ([Sch86]) genannt. Explizit, denn wir benötigen für die Durchführung keine Lösung einer Gleichung, sondern gehen einfach zeilenweise vor und können alle Werte sukzessive berechnen.

Das Verfahren hat allerdings Nachteile, die sie nach der Implementierung sofort erkennen werden. Wählen wir zu große Werte für h_t in der Zeit, dann erhalten wir schnell physikalisch vollkommen unsinnige Ergebnisse. Analysiert man das Verfahren, was wir hier nicht tun wollen, dann kann man die Stabilitätsbedingung

$$r \leq \frac{1}{2} \Leftrightarrow h_t \leq \frac{1}{2}h_x^2$$

beweisen. Werte für die Schrittweite h_t , die diese Bedingung verletzen führen zu sinnlosen Ergebnissen. Das bedeutet insbesondere, dass wir bei einer sehr feinen Auflösung in x -Richtung nicht beliebig große Schritt in t verwenden können.

3.2 Die implizite Methode von Crank-Nicholson

Die Stabilitätsbedingung für das explizite Verfahren von Richardson bedeutet, dass wir eventuell sehr kleine Schritte in t verwenden müssen. Das kann dazu führen, dass wir trotz des expliziten Charakters des Verfahrens einen großen Rechenaufwand erhalten.

Mit Hilfe von leicht veränderten Näherungen der partiellen Ableitungen in unserer partiellen Differentialgleichung, auf die wir hier nicht eingehen werden, kann man eine andere diskrete Näherung der Gleichung herleiten:

$$-ru_{i-1,j+1} + (2 + 2r)u_{i,j+1} - ru_{i+1,j+1} = ru_{i-1,j} + (2 - 2r)u_{i,j} + ru_{i+1,j}. \quad (8)$$

Auf der linken Seite von (8) sehen jetzt Werte für den Zeitschritt $(j+1)h_t$. Beim Verfahren von Richardson stand hier genau ein Wert, und wir konnten danach auflösen. Auf der rechten Seite von (8) finden wir Werte aus dem vorhergegangenen Zeitschritt – wenn wir wieder

sukzessive mit $j = 1$ beginnen stehen dann auf der rechten Seite die Randbedingungen für $t = 0$. Auch die Randbedingungen am linken Rand können wir neu formulieren:

$$(2 + 2r)u_{1,j+1} + ru_{2,j+1} = (2 - 2r)u_{1,j} + r(g(jh_t) + g((j+1)h_t)).$$

Rechts, für die Bedingung, die die partielle Ableitung enthält, erhalten wir

$$-2ru_{n-1,j+1} + (2 + 2r)u_{n,j+1} = 2ru_{n-1,j} + (2 - 2r)u_{n,j}.$$

Was haben wir dadurch gewonnen? Wir können auf jeden Fall nicht nach einem der Werte mit dem Index $j + 1$ auflösen, aber wenn wir die linke Seite genauer ansehen erkennen wir, dass jeder dieser Zeilen eine lineare Gleichung ist. Für die Gleichungen aus (8) sind $u_{i-1,j+1}$, $u_{i,j+1}$ und $u_{i+1,j+1}$ die Unbekannten; die Werte auf der rechten Seite definieren die rechte Seite eines linearen Gleichungssystems mit einer tridiagonalen Koeffizientenmatrix. Auf den Nebendiagonalen steht $-r$, und auf der Diagonalen der Wert $2 + 2r$. Wir verwenden also für jeden Zeitschritt eines der Verfahren zur Lösung eines tridiagonalen linearen Gleichungssystems und können wieder von Zeitschritt zu Zeitschritt gehen.

Dieses Verfahren wird *implizites Verfahren von Crank-Nicholson* ([Sch86]) genannt. Implizit, denn wir lösen lineare Gleichungssysteme für die Berechnung eines Zeitschritts.

Der große Vorteil dieses Verfahrens ist, dass es *keine* Einschränkungen an die Wahl von h_x und h_t gibt – wir können beide Schrittweiten frei wählen und erhalten immer stabile Ergebnisse!

3.3 Aufgaben

Aufgabe 4

Implementieren Sie das explizite Verfahren von Richardson für die Lösung eines Randwertproblems wie

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, 0 < x < 1, t = 0, \quad (1)$$

$$u(x, 0) = 0, 0 < x < 1, \quad (2)$$

$$u(0, t) = \sin(\pi t), \frac{\partial}{\partial x} u(1, t) = 0, t > 0. \quad (3)$$

Implementieren Sie dabei Ihre Lösung so, dass die Randwerte für $u(0, t)$ in (2) und (3) aus einer Datei eingelesen werden können!

Die Näherungen sollen dabei bis $t = 5$ berechnet werden. Verwenden Sie die Schrittweiten $h_x = 0,1$ und

a) $h_t = 0,002$,

b) $h_t = 0,005$,

c) $h_t = 0,01$.

Für die Visualisierung Ihrer Lösung mit Hilfe von ParaView finden Sie im GitHub-Repo PDEPack Pipeline-Dateien und Funktionen!

Aufgabe 5

Implementieren Sie das implizite Verfahren von Crank-Nicholson für die Lösung eines Randwertproblems wie

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, 0 < x < 1, t = 0, \quad (1)$$

$$u(x, 0) = 0, 0 < x < 1, \quad (2)$$

$$u(0, t) = \sin(\pi t), \frac{\partial}{\partial x} u(1, t) = 0, t > 0. \quad (3)$$

Implementieren Sie dabei Ihre Lösung so, dass die Randwerte für $u(0, t)$ in (2) und (3) aus einer Datei eingelesen werden können! Verwenden Sie für die Lösung der auftretenden tridiagonalen linearen Gleichungssysteme eines der bisher in der Lehrveranstaltung eingeführten Verfahren!

Die Näherungen sollen dabei bis $t = 5$ berechnet werden. Verwenden Sie die Schrittweiten $h_x = 0,1$ und

a) $h_t = 0,01,$

b) $h_t = 0,1,$

Verwenden Sie Ihre Lösung aus Aufgabe 4 und berechnen Sie die Lösungen für die vorgegebenen Schrittweiten auch mit dem expliziten Verfahren von Richardson! Führen Sie Benchmarks für Ihre Implementierungen durch und diskutieren Sie Ihre Ergebnisse! Begründen Sie Ihre Entscheidung für den Löser der linearen Gleichungssysteme, entweder mit Hilfe von theoretischen Überlegungen oder auf der Basis von Benchmarks!

Für die Visualisierung Ihrer Lösung mit Hilfe von ParaView finden Sie im GitHub-Repo PDEPack Pipeline-Dateien und Funktionen!

Aufgabe 6

Wie in den Aufgaben zuvor betrachten wir das Randwertproblem

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, 0 < x < 1, t = 0, \quad (1)$$

$$u(x, 0) = 0, 0 < x < 1, \quad (2)$$

$$u(0, t) = \sin(\pi t), \frac{\partial}{\partial x} u(1, t) = 0, t > 0. \quad (3)$$

Die Näherungen sollen dabei bis $t = 4$ berechnet werden, mit einer Schrittweite $h_x = 0,05$. Verwenden Sie sowohl das explizite Verfahren von Richardson als auch das implizite Verfahren von Crank-Nicholson!

Führen Sie Benchmarks durch und stellen Sie sicher, dass die Ergebnisse für $t = 4$ so gut wie möglich übereinstimmen! Diskutieren Sie Ihre Ergebnisse!

Literatur und Quellen

Die Idee, ein einfaches Modell einer „Kühlrippe“ zu betrachten stammt aus [BM04]. Die Konfiguration des Modells wurde leicht angepasst. Mehr zur numerischen Behandlung von partiellen Differentialgleichungen finden Sie in [BM04, Eij14, Sch86] oder in [PFTV02].

Literatur

- [BM04] BOLLHÖFER, MATTHIAS und MEHRMANN, VOLKER: *Numerische Mathematik – Eine projektorientierte Einführung für Ingenieure, Mathematiker und Naturwissenschaftler*. Vieweg, 2004.
- [Eij14] EIJKHOUT, VICTOR: *Introduction to High Performance Computing*. lulu.com, 2014. Distributed under a Creative Commons Attribution 3.0 Unported (CC BY 3.0) license.
- [PFTV02] PRESS, WILLIAM, FLANNERY, BRIAN, TEUKOLSKY, SAUL und VETTERLING, WILLIAM: *Numerical Recipes in C++ – The Art of Scientific Computation*. Cambridge University Press, 2002.
- [Sch86] SCHWARZ, HANS RUDOLF: *Numerische Mathematik*. Teubner Verlag, 1986.