

Praktikum Multicore-Programmierung

Aufgabenblatt 4: Teamprojekte

Abgabe: Protokoll und Quellcode 08.02.2016

Projekt 6: Mehrgittermethoden

Viele Problemstellung aus dem Bereich der Naturwissenschaften lassen sich mit Hilfe partieller Differentialgleichungen beschreiben und lösen. Dabei kommen im Allgemeinen Lösungsverfahren mit hohem Rechenaufwand zum Einsatz, wodurch besonders in diesem Bereich der Einsatz paralleler Hardware und parallelisierbarer Algorithmen notwendig ist.

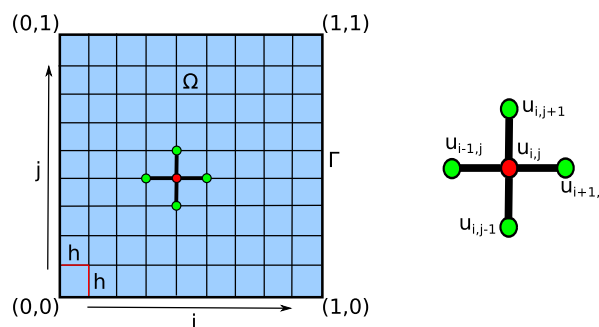
Ein Beispiel einer solchen Problemstellung ist die Beschreibung von Konzentrationen eines Stoffes innerhalb eines festgelegten Gebiets. Als grundlegende mathematische Beschreibung dient dabei die Poisson-Gleichung. Daneben ist für die korrekte Problemstellung zusätzlich die Angabe von Randbedingungen zwingend erforderlich. Für ein zweidimensionales Gebiet¹ Ω mit Rand Γ stellt sich das vollständige Problem wie folgt dar:

$$-\Delta u(x, y) = f(x, y) \quad (x, y) \in \Omega \subset \mathbb{R}^2 \quad (1)$$

$$u(x, y) = g(x, y) \quad (x, y) \in \Gamma. \quad (2)$$

Zur Vereinfachung werden hier nur dirichletsche² Randbedingungen betrachtet. Zusätzlich werden diese homogenisiert, es gilt also $g(x, y) = 0 \quad \forall (x, y) \in \Gamma$. Zuletzt soll $\Omega \cup \Gamma = [0, 1]^2$ gelten, der zu untersuchende Bereich beschränkt sich also auf das Einheitsquadrat.

Zur Lösung dieses Problems soll das zu betrachtende Gebiet zunächst diskretisiert werden. Dazu wird ein Diskretisierungsparameter h gewählt, mit welchem eine Beschreibung einer in jede Raumrichtung äquidistanten Diskretisierung möglich ist.



¹ Ω beschreibt dabei ein Gebiet nach mathematischer Definition, ist also offen, nichtleer und zusammenhängend.

²Dirichletsche Randbedingungen beschreiben Randbedingungen ohne Verwendung eines Differentials.

Es gilt also $(x_{i+1}, y_{j+1}) = (x_i + h, y_j + h)$. Die gesuchte Lösung u wird nun an jedem Gitterpunkt berechnet, wodurch eine Approximation an die analytische Lösung entsteht. Klar ist, dass die Genauigkeit der Lösung zunimmt, je kleiner h gewählt wird, daher wird h auch Feinheit genannt. Gleichzeitig steigt dabei jedoch der Rechenaufwand, weil mehr Gitterpunkte berechnet werden müssen.

Am einzelnen Gitterpunkt in Ω erfolgt die Approximation der Lösung mit Hilfe der Finiten Differenzen. Dazu ist es nötig, einen Differenzenstern zu wählen. Es soll hier der 5-Punkte-Differenzenstern angesetzt werden, wie er in der Grafik oben zu sehen ist. Die punktuelle Approximation der 2. Ableitung an (x_i, y_j) kann so angegeben werden durch:

$$\frac{u(x_i - h, y_j) - 2u(x_i, y_j) + u(x_i + h, y_j)}{h^2} = \frac{\partial^2 u}{\partial x^2}(x_i, y_j) + \mathcal{O}(h^2) \quad (3)$$

$$\frac{u(x_i, y_j - h) - 2u(x_i, y_j) + u(x_i, y_j + h)}{h^2} = \frac{\partial^2 u}{\partial y^2}(x_i, y_j) + \mathcal{O}(h^2). \quad (4)$$

Zur Vereinfachung der Schreibweise gilt $u_{i,j} = u(x_i, y_j)$.

Werden diese beiden Ableitungen gemäß der Definition des Laplace-Operators³ addiert, ergibt sich eine Approximation von (1) durch:

$$-u_{i,j-1} - u_{i-1,j} + 4u_{i,j} - u_{i,j+1} - u_{i+1,j} = h^2 f(x_i, y_j). \quad (5)$$

Iteriert über alle $i, j \in [1, \frac{1}{h})^4$ ergibt sich so ein lineares Gleichungssystem der Form $Au = b$, beziehungsweise

$$\begin{pmatrix} T & -I & & & \\ -I & T & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & T & -I \\ & & & -I & T \end{pmatrix} \begin{pmatrix} u_{1,1} \\ \vdots \\ u_{n,n} \end{pmatrix} = h^2 \begin{pmatrix} f(x_1, y_1) \\ \vdots \\ f(x_n, y_n) \end{pmatrix} \quad \text{mit } T = \begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix} \quad (6)$$

und der Einheitsmatrix I . Die Dimension der einzelnen Matritzen und Vektoren hängt offensichtlich an der Wahl von h .

Es stehen verschiedene numerische Methoden zur Lösung dieses Gleichungssystems beziehungsweise der Gleichung (5) iteriert über alle i und j zur Verfügung. Es soll mit Hilfe der nachfolgenden Aufgabenstellungen die für den aktuellen Anwendungsfall schnellste Methode gefunden werden.

³Laplace-Operator: $\Delta u = \sum_{i=1}^d \frac{\partial^2 u}{\partial x_i^2}$

⁴Man beachte, dass nur über alle Gitterpunkte aus Ω iteriert werden muss, die Lösung auf Gitterpunkten in Γ ist durch die Randbedingungen bereits bekannt.

1.1 Jakobi- und Gauß-Seidel-Verfahren

Durch geeignete Zerlegung von A lassen sich Lösungsverfahren ermitteln. Nachfolgend findet sich eine Darstellung des so ermittelten Jakobi-Verfahrens

```

Wähle Startvektor  $u^0 \in \mathbb{R}^{(n^2)}$ 
for  $k = 0, 1, \dots$ 
  for  $j = 1, 2, \dots, n$ 
    for  $i = 1, 2, \dots, n$ 
       $u_{i,j}^{k+1} = \frac{1}{4}(u_{i,j-1}^k + u_{i-1,j}^k + u_{i,j+1}^k + u_{i+1,j}^k + h^2 f(x_i, y_j))$ 

```

sowie des sehr ähnlichen Gauß-Seidel-Verfahrens

```

Wähle Startvektor  $u^0 \in \mathbb{R}^{(n^2)}$ 
for  $k = 0, 1, \dots$ 
  for  $j = 1, 2, \dots, n$ 
    for  $i = 1, 2, \dots, n$ 
       $u_{i,j}^{k+1} = \frac{1}{4}(u_{i,j-1}^{k+1} + u_{i-1,j}^{k+1} + u_{i,j+1}^k + u_{i+1,j}^k + h^2 f(x_i, y_j))$ 

```

- Programmieren Sie eine serielle Version des Jakobi-Verfahrens. Gegeben sei dazu die rechte Seite von (1) mit $f(x, y) = 32(x(1 - x) + y(1 - y))$. Beachten Sie dabei auch die Randbedingungen aus (2). Verwenden Sie zur Überprüfung Ihres Programms auf Korrektheit die analytische Lösung $u(x, y) = 16x(1 - x)y(1 - y) \quad \forall (x, y) \in \Omega$.
- Implementieren Sie ebenfalls eine serielle Version des Gauß-Seidel-Verfahrens. Vergleichen Sie die gewonnenen Ergebnisse mit den Ergebnissen des Jakobi-Verfahrens. Was fällt beim Vergleich der Iterationsschrittzahlen sowie der Laufzeiten beider Verfahren über verschiedenen Verfeinerungen ($h = \frac{1}{2^l}$ mit $l = 1, 2, \dots, 6, \dots$) auf?
- Messen Sie mit Hilfe der in a) angegebenen analytischen Lösung für beide Verfahren den maximalen Approximationsfehler mit Hilfe der euklidischen Norm in jedem Iterationsschritt und für ein gewähltes h und stellen Sie diesen graphisch dar.
- Da hier innerhalb eines einzelnen Iterationsschritts keine Datenabhängigkeiten zu finden sind, lässt sich das Jakobi-Verfahren durch einfaches Aufteilen der Berechnungen

- auf verschiedene Prozessoren und Kommunizieren eventueller Überlappungen parallelisieren. Entwerfen Sie eine parallele Version des Verfahrens (wahlweise mit OpenMP oder Cuda/OpenCL) und verifizieren Sie Ihre Ergebnisse gegenüber der seriellen Variante.
- e) Im Gegensatz zum Jakobi-Verfahren zeigt das Gauß-Seidel-Verfahren Datenabhängigkeiten innerhalb der einzelnen Iterationsschritte. Daher eignet sich eine naive Herangehensweise nicht, um dieses Verfahren ausreichend zu parallelisieren. Ihre Aufgabe ist es, eine sinnvolle Methodik zu entwickeln, mit welcher sich dieses Verfahren parallelisieren lässt.
- f) Ermitteln Sie den Speedup für verschiedene Problemgrößen für beide Verfahren ($h = \frac{1}{2^l}$ mit $l = 1, 2, \dots, 6, \dots$). Beurteilen Sie damit auch die Qualität ihrer Parallelisierungen.

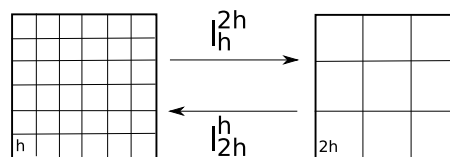
1.2 Mehrgitterverfahren

Beide in Aufgabe 1.1. vorgestellte Verfahren haben den Nachteil, dass sie zu langsam konvergieren, um gut im Rahmen großer Gleichungssysteme Verwendung zu finden. Dies hängt unter anderem damit zusammen, dass der Startwert unabhängig von der gesuchten Lösung gewählt werden muss, da im Allgemeinen kein Hinweis auf die Lage des Lösungsvektors gegeben ist. Es ist klar, dass ein Startwert, der näher an der gesuchten Lösung liegt, auch schneller zum gewünschten Ergebnis führt.

Eine Möglichkeit, dennoch einen zielgerichteten Startwert zu wählen, ist im Mehrgitterverfahren enthalten. Die Idee dahinter ist, dass auf einem Gitter mit wenig Gitterpunkten Gleichungssysteme deutlich schneller gelöst werden können, als auf feineren Gittern. Daher kann es sich lohnen, zunächst auf einem groben Gitter mit dem Lösen eines Problems zu beginnen, um eine besser gelegene „Vorlösung“ zu berechnen. Diese kann dann auf ein feineres Gitter übertragen werden, um die notwendige Genauigkeit zu erhalten. Schrittweise finden so mehrere Gitterverfeinerungen statt. Insgesamt kann dadurch ein lösungsnaher Startvektor gefunden und damit die gesuchte Lösung dadurch deutlich schneller ermittelt werden.

Numerisch gesehen ist es jedoch sinnvoller, nicht direkt auf dem sehr groben Gitter zu beginnen, sondern erst einige wenige Schritte auf einem feineren Gitter zu berechnen. Dadurch entsteht ein Algorithmus, der zunächst vom geforderten feinen Gitter zu einem sehr groben Gitter aufsteigt und anschließend wieder zu dem sehr feinen Gitter zurückkehrt.

Insgesamt treten hier zwei Problemstellungen auf. Zunächst gilt es ein Lösungsverfahren zu wählen, welches auf den gewählten Gittern iterieren kann. Zudem müssen die notwendigen Gitterübergänge betrachtet werden. Das erste Problem kann durch Verfahren gelöst werden, wie sie in Aufgabe 1.1. vorgestellt wurden. Das zweite Problem kann in Richtung des feineren Gitters durch lineare Interpolation und in Richtung des groben Gitters durch Restriktion gelöst werden. Hierzu werden die Operatoren I_{2h}^h (Interpolation) beziehungsweise I_h^{2h} (Restriktion) eingeführt, die ein gegebenes Gitter in ein Gitter mit doppelter Feinheit überführen und umgekehrt:



Insgesamt ergibt sich daraus der folgende rekursive Algorithmus für das Mehrgitterverfahren:

```
Algorithmus ( $MV^h(v^h, f^h, \alpha)$ ):  
if (größtes Gitter erreicht) then  
    Löse  $A^h u^h = f^h$   
else  
    Iteriere  $z_1$ -mal mit  $A^h u^h = f^h$ , erhalte Näherung  $v^h$   
    Berechne  $f^{2h} = I_h^{2h}(f^h - A^h v^h)$   
    Setze  $v^{2h} = 0$   
    Führe  $\alpha$ -mal ( $MV^{2h}(v^{2h}, f^{2h}, \alpha)$ ) aus, erhalte neues  $v^{2h}$   
    Berechne  $v^h = v^h + I_{2h}^h v^{2h}$   
    Iteriere  $z_2$ -mal mit  $A^h u^h = f^h$ 
```

- a) Warum ist es aus Sicht der Numerik nicht sinnvoll sofort auf einem groben Gitter zu beginnen, sondern zunächst einige Schritte auf einem feinen Gitter durchzuführen? Welchen Zusammenhang gibt es dabei zur Fehleroszillation?
- b) Implementieren Sie eine serielle Version des Mehrgitterverfahrens. Verwenden Sie dabei die Operatoren aus den Einführungsfolien sowie $\alpha = 1$ und mindestens drei Gitterfeinheiten. Als Lösungsverfahren soll das Gauß-Seidel-Verfahren zum Einsatz kommen. Achten Sie auf die Korrektheit ihrer Ergebnisse. Welche Werte sind für z_1 und z_2 sinnvoll?
- c) Zeigen Sie experimentell, dass eine Wahl von $\alpha > 1$ sinnvoller sein kann. Welche Begründung kann für das beobachtete Verhalten gefunden werden?
- d) Parallelisieren Sie das Verfahren in möglichst hohem Maße (wahlweise mit OpenMP oder Cuda/OpenCL). Verwenden Sie dazu $\alpha = 2$ sowie mindestens drei Gitterfeinheiten. Verifizieren Sie die Korrektheit ihrer Implementation.
- e) Ermitteln Sie den Speedup und die Efficiency gegenüber der seriellen Implementation des Mehrgitterverfahrens sowie des parallelen Gauß-Seidel-Verfahrens. Beurteilen Sie damit auch die Qualität ihrer Parallelisierung.
- f) Entscheiden Sie auf Grundlage der Ergebnisse aus 1.1. und 1.2., welches der vorgestellten Verfahren (Jakobi, Gauß-Seidel, Mehrgitter) Sie für geeigneter halten um Gleichungssysteme großer Dimension zu lösen. Begründen Sie ihre Wahl und testen Sie diese auf Skalierbarkeit.