Marcell Kehmstedt
m.kehmstedt@tu-bs.de

Iterative Gleichungslöser

Viele praktisch bedeutsame Probleme in der Technik lassen sich auf das Lösen eines linearen Gleichungsproblems zurückführen, welches das jeweilige Problem in seiner Gesamtheit beschreibt. Zur Lösung eines solchen Problems bedarf es also folgender Schritte:

a) Aufstellen eines geeigneten Gleichungssystems $Ax = b$

b) Lösen des Gleichungssystems

Der zweite Schritt ist das eigentliche Berechnen einer Lösung. Im Laufe der Zeit hat die Mathematik eine ganze Reihe verschiedenartiger Verfahren zur Lösung von linearen Gleichungssystemen hervorgebracht. Diese lassen sich grob einteilen in **direkte** und **iterative** Verfahren.

Ein Beispiel für ein **direktes** Verfahren ist der **Gauß-Algorithmus**. Aus Gründen numerischer Stabilität (bedenke: Rundungsfehler in Eingabedaten und während der Berechnung) muss dieser zwingend in einer Variante mit Pivotierung verwendet werden.

Ein Beispiel für ein **iteratives** Verfahren ist das **Jacobi-Verfahren**.

Jacobi-Verfahren

Gegeben: Ein lineares Gleichungssystem mit n Unbekannten und n Gleichungen.

$$\begin{aligned} a_{11} \cdot x_1 + \cdots + a_{1n} \cdot x_n &= b_1 \\ a_{21} \cdot x_2 + \cdots + a_{2n} \cdot x_n &= b_2 \\ &\vdots \\ a_{n1} \cdot x_n + \cdots + a_{nn} \cdot x_n &= b_n \end{aligned}$$

Ansatz: Löse jede Gleichung einzeln nach je einem x_i auf.
Für die i-te Gleichung erhält man also:

$$x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} \cdot x_j - \sum_{j=i+1}^{n} a_{ij} \cdot x_j \right) = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} \cdot x_j \right), i = 1, \ldots, n$$

Das Problem, das sich hierbei jedoch ergibt, ist, dass für die Berechnung eines x_i jeweils alle anderen x_j $(j \neq i)$ benötigt werden. Die eigentliche Idee besteht nun darin eine Folge von Lösungen $(x^{(m+1)})$ zu berechnen, bei der auf der rechten Seite jeder Gleichung jeweils der vorhergegangene Lösungsvektor $x^{(m)}$ verwendet wird. Die Iterationsvorschrift lautet dann:

$$x_i^{(m+1)} := \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} \cdot x_j^{(m)} \right), i = 1, \ldots, n$$

Startlösung

Es ist klar, dass zur Berechnung der ersten Iteration $x^{(1)}$ ein Startvektor $x^{(0)}$ (die sog. Startlösung) vorgegeben werden muss. Die Wahl dieses Vektor hat maßgeblichen Einfluss auf die Konvergenzgeschwindigkeit des Verfahrens. Je näher die Startlösung an der eigentlichen Lösung des Gleichungssystems liegt, desto weniger Iterationen sind notwendig, um eine hinreichend genaue Lösung zu erhalten.

Hat man jedoch keinen Hinweis für eine gute Startlösung, so tut es in den meisten Fällen auch der Nullvektor.

Marcell Kehmstedt
m.kehmstedt@tu-bs.de

Residuum & Abbruchkriterien

Iterative Verfahren wie das hier vorgestellte Jacobi-Verfahren berechnen Näherungen an die eigentliche Lösungen basierend auf einer vorgegebenen Startlösung. Diese Näherungen werden durch die Iterationen von Schritt zu Schritt verbessert, um sich mehr und mehr der gesuchten Lösung anzunähern. Tatsächlich ist jedoch keine der Näherungen eine „Lösung“ im Sinne der Aufgabe. Denn für kein $x^{(m)}$, das das Verfahren liefert, gilt

$$Ax^{(m)} = b.$$

Während für den Fall der exakten Lösung $||Ax - b|| = 0$ gilt, so wird für Näherungen $x^{(m)}$ stets $||Ax^{(m)} - b|| > 0$ sein. $||Ax - b||$ bezeichnet man als das **Residuum**. Dabei handelt es sich um ein Maß für die Nichterfüllung des Gleichungssystems durch x.

Die wesentliche Frage lautet also: Nach wievielen Iterationen soll das Verfahren abbrechen?

Nachfolgend zwei Möglichkeiten:

a) Wenn das Residuum eine vorgegebene Toleranzschwelle unterschreitet: $||Ax^{(m)} - b|| < \epsilon_{\text{tol}}$

b) Wenn sich die Näherungen nur noch geringfügig ändern: $||x^{(m+1)} - x^{(m)}|| < \delta_{\text{tol}}$

Konvergenz

Eine Frage, die bisher noch völlig unbeantwortet blieb, ist die Frage nach der Konverganz des Verfahrens. Bisher wurde unterstellt, dass das Verfahren eine Näherungslösung von Iteration zu Iteration schrittweise verbessert, so dass sich diese der gesuchten Lösung beliebig annähert. Dies ist die Idee, die den iterativen Verfahren zugrunde liegt. Die Frage ist jedoch, ob die konstruierte Folge überhaupt einen Grenzwert hat. Und tatsächlich kann z.B. mit Hilfe des Banach'schen Fixpunktsatzes die Konvergenz des hier vorgestellten Jacobi-Verfahrens gezeigt werden. Die Konvergenz hängt dabei von der Systematrix A des Gleichungssystems ab. So konvergiert das Verfahren, wenn A **strikt diagonaldominant** ist.

Eine $(n \times n)$-Matrix $A = (a_{ij})$ heißt **strikt diagonaldominant** genau dann, wenn:

$$\sum_{j \neq i} |a_{ij}| < |a_{ii}|, i = 1, \ldots, n$$

Parallelisierbarkeit

Da die Komponenten einer neuen Näherung $x_i^{(m+1)}$ komplett unabhängig voneinander berechnet werden können, ist das Jacobi-Verfahren sehr gut parallelisierbar. Zur Berechnung von $x_i^{(m+1)}$ muss der vollständige Vektor $x^{(m)}$ aus der vorhergegangenen Iteration vorliegen.

Anwendung: Wärmeleitung

Als nächstes soll ein aus der Praxis motiviertes Problem behandelt werden. Es soll Wärmeleitung auf einem abgeschlossenen Gebiet betrachtet werden. Zur Vereinfachung soll das Gebiet als rechteckig angenommen werden.

Wärmeleitung in einem hier festen Medium ist ein Diffusionsprozess, bei dem sich Wärme über das gesamte Medium ausbreitet und Temperaturunterschiede auszugleichen tendiert. Ein solches Medium kann z.B. ein Stück Metallblech sein. Physikalisch basiert Wärmeleitung auf dem **Fourierschen Gesetz** (1882), das die übertragene Wärmeleistung beschreibt. Mathematisch wird ein solcher Diffusionsprozess durch eine partielle Differentialgleichung beschrieben.

Marcell Kehmstedt
m.kehmstedt@tu-bs.de

Der Einfachheit halber soll sich hier auf einen homogenen und stationären (d.h. zeitunabhängigen) Speziallfall, die Laplace-Gleichung beschränkt werden. Gesucht wird eine zweifach stetig differenzierbare, skalare Funktion Φ auf einem Gebiet $\Omega \subset \mathbb{R}^n$, die die Laplace-Gleichung

$$\Delta\Phi = 0$$

erfüllt.

Hierbei bezeichnet Δ den Laplace-Operator, welcher in kartesischen Koordinaten geschrieben werden kann als:

$$\Delta = \sum_{k=1}^{n} \frac{\partial^2}{\partial x_k^2}$$

Für $n = 2$ Dimensionen (Raumrichtungen) ergibt sich der wichtige Spezialfall:

$$\Delta = \sum_{k=1}^{2} \frac{\partial^2}{\partial x_k^2} = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2}$$

Dirichlet-Problem

Da es nun eine ganze Reihe an Funktionen gibt, die die Laplace-Gleichung erfüllen, müssen zusätzliche Bedingungen an gesuchte Funktion Φ gestellt werden, damit die Lösung eindeutig wird. Das hier betrachtete Gebiet ist endlich. Insofern gibt es einen Rand $\partial\Omega$, auf dem das Verhalten der Funktion vorgegeben werden muss. Eine Möglichkeit ist es, die Werte der Funktion Φ auf dem Rand fest vorzugeben. Dies wird als **Dirichlet-Randbedingung** bezeichnet und soll auch hier verwendet werden.

Diskretisierung

Das Wärmeleitungsproblem in seiner mathematischen Formulierung sucht nach einer zweifach stetig differenzierbaren Funktion Φ. Allerdings können mit dem Computer nur endlich viele Punkte der Funktion erfasst und diese somit nur auf endlichen vielen Punkten berechnet werden. Es ist daher nötig, die partiellen zweiten Ableitungen $\frac{\partial^2}{\partial x_k^2}$ aus der Laplace-Gleichung zu diskretisieren. Da an dieser Stelle jedoch keine vollständige mathematische Herleitung gegeben werden soll, wird im Nachfolgenden nur die wesentliche Vorgehensweise skizziert.

Die Ableitung einer Funktion u an der Stelle x (der sog. *Differentialquotient*) ist definiert als der Grenzwert des Differenzenquotienten

$$u'(x) = \lim_{h \to 0} \frac{u(x+h) - u(x)}{h}.$$

Verzichtet man jedoch auf die Grenzwertbildung, so erhält man die diskrete Ableitung mit einer endlichen Schrittweite h und einem Fehler in derselben Größenordnung:

$$u'(x) = \frac{u(x+h) - u(x)}{h} + \mathcal{O}(h)$$

Eine Formel für die diskrete 2. Ableitung lautet:

$$u''(x) = \frac{u(x-h) - 2u(x) + u(x+h)}{h^2} + \mathcal{O}(h^2)$$

Hat man es ferner mit partiellen Ableitungen zu tun, so ist die Schrittweite h zu ersetzen durch die Schrittweite in die jeweilige Richtung. Beim Einsetzen in die Laplace-Gleichung für $n = 2$ Raumrichtungen ergibt sich schließlich die Formel:

$$\left(\frac{-u(x-dx,y) + 2u(x,y) - u(x+dx,y)}{dx^2}\right) + \left(\frac{-u(x,y-dy) + 2u(x,y) - u(x,y+dy)}{dy^2}\right) = 0$$

Das Schema auf der linken Seite wird als **5-Punkt-Differenzenstern** bezeichnet.

Marcell Kehmstedt
m.kehmstedt@tu-bs.de

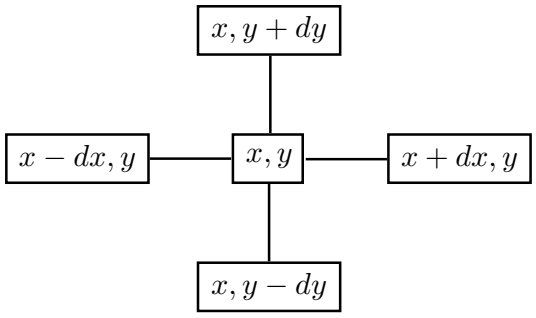

Abbildung 1: 5-Punkt-Differenzenstern

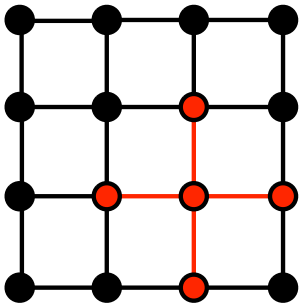

Abbildung 2: Differenzenstern im Gitter

Zweidimensionales Randwertproblem

Das zweidimensionale Gebiet $\Omega = [a, b] \times [c, d]$, auf dem eine Näherung für die Funktion Φ berechnet werden soll, wird diskretisiert und durch ein regelmäßiges Gitter von Punkten repräsentiert. Der Abstand zwischen zwei Gitterpunkten in Richtung x betrage dx und der Abstand zweier Gitterpunkte in Richtung y betrage dy. Die Funktion Φ soll durch die Lösungen $u_{x,y} := u(x, y)$ in den Punkten $\{(x, y) | x \in [a, b], y \in [c, d]\}$ approximiert werden.

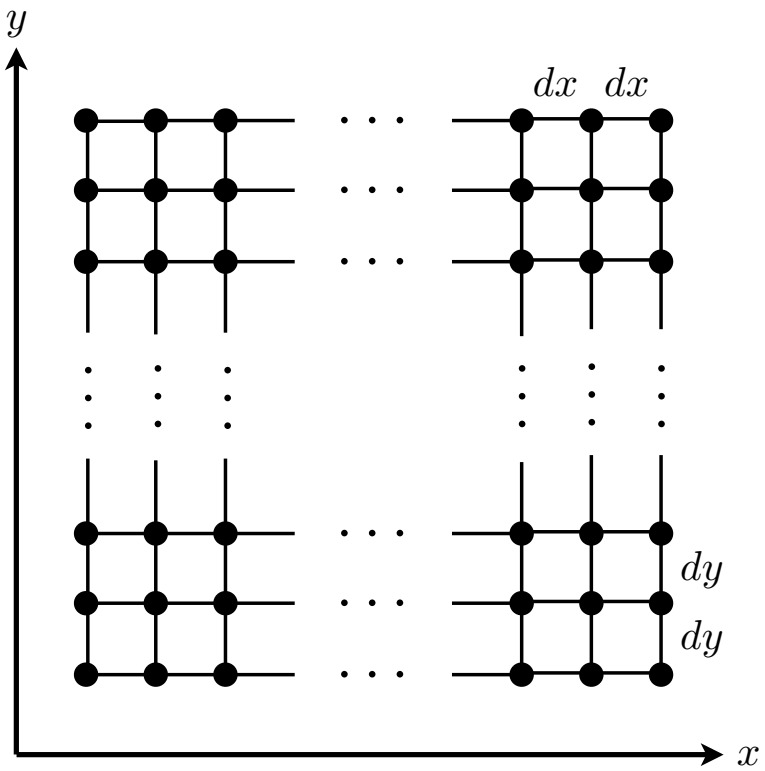

Abbildung 3: Zweidimensionales Gitter

Durch Einsetzen dieser Punkte in die Laplace-Gleichung entsteht ein lineares Gleichungssystem mit genauso vielen Gleichungen wie Unbekannten. Die Unbekannten sind die Lösungen $u_{x,y}$ in den Punkten (x, y). Dieses Gleichungssystem kann z.B. mit einem iterativen Lösungsverfahren (wie dem zuvor behandelten Jacobi-Verfahren) oder durch Berechnung des Differenzensterns auf den einzelnen Gitterpunkten gelöst werden.

Marcell Kehmstedt
m.kehmstedt@tu-bs.de

Aufgaben

In beiden folgenden Aufgaben soll es um das zuvor vorgestellte Modellproblem der Wärmeleitung auf einem rechteckigen Gebiet gehen. In beiden Aufgaben soll eine Näherungslösung berechnet werden. Dazu muss das Problem zunächst aufgestellt werden, bevor es gelöst werden kann. Für beide Aufgaben wird Ihnen eine Referenzimplementierung zur Verfügung gestellt. Laden Sie sich die Datei `heatconduction.tar.gz` herunter und entpacken Sie diese in ein beliebiges Verzeichnis auf dem Cluster. Machen Sie sich zunächst mit dem Inhalt vertraut und lassen Sie die Beispielprogramme bauen und ablaufen. Die Beispiele können Sie bauen, indem Sie `make examples` ausführen. Die erstellten Programme namens „heat“ finden Sie anschließend in den Unterordnern *exercise1* und *exercise2*. Während der Ausführung speichern die Programme eine Reihe von Bildern im jeweiligen Unterordner *images*. Wenn Sie anschließend in diesen Ordner wechseln und `make video` ausführen, können Sie aus den Einzelbildern ein Video erstellen. Das dazu verwendete Programm `ffmpeg` wird im Ordner *utilities* bereits vorkompiliert mitgeliefert.

Aufgabe 1 Jacobi-Verfahren

Parallelisieren Sie das Jacobi-Verfahren mit MPI und verwenden Sie Ihre Implementierung zur Lösung des Modellproblems. Die Matrix A und den Rechteseitevektor b, die zusammen das Problem beschreiben, können Sie sich von der Methode `setupProblem()` aus der Datei `exercise1/example/problem.c` erzeugen lassen. Verwenden Sie $M = 100$ Punkte in y-Richtung und $N = 100$ Punkte in x-Richtung und berechnen Sie 1.000 Iterationen jeweils mit $P = 2, 4, 8, 16, 32$ Prozessen. Bestimmen Sie den Speedup Ihrer parallelen Implementierung gegenüber der Referenzimplementierung. Erzeugen Sie darüberhinaus ein Video, das den Verlauf der Berechnung veranschaulicht.

(15 Punkte)

Aufgabe 2 Gebietszerlegung

Parallelisieren Sie die Implementierung aus der Datei `exercise2/example/program.c`. Zerlegen Sie das Gebiet in Teilgebiete entsprechend der Anzahl der verwendeten Prozesse. Berechnen Sie in jedem Prozess eine Iteration auf dem jeweiligen Teilgebiet und tauschen Sie die neue Näherung an den Rändern mit den Prozessen der benachbarten Gebiete aus. Verwenden ein Gebiet mit $M = 1024$ und $N = 1024$ Punkten und berechnen Sie 10.000 Iterationen wieder jeweils mit $P = 2, 4, 8, 16, 32$ Prozessen. Bestimmen Sie abermals den Speedup und erzeugen Sie ein Video, das den Verlauf der Berechnung zeigt.

(15 Punkte)