

Übungen zur Vorlesung
Paralleles Höchstleistungsrechnen
Dr. S. Lang, D. Popović

Abgabe: 08. Dezember 2009 in der Übung

Übung 15 Gauss-Seidel-Verfahren mit dünnbesetzten Matrizen (10 Punkte)

In den Übungen haben Sie das allgemeine Gauss-Seidel-Verfahren als iterativen Löser für Gleichungssysteme kennengelernt. In jedem Schritt m ist dabei folgendes System zu berechnen:

$$\mathbf{Ax} = \mathbf{b} \iff \left(\sum_{j=1}^{i-1} a_{ij} x_j^{m+1} \right) + a_{ii} x_i^{m+1} = b_i - \sum_{j=i+1}^n a_{ij} x_j^m.$$

Zur Herleitung des Verfahrens haben wir dabei die Aufspaltung von A in Diagonal-, linke untere und obere rechte Dreiecksmatrix verwendet, $A = D + L + R$. Genaueres entnehmen Sie bitte dem zusätzlich zur Übung ausgeteiltem Material.

- (a) Betrachten Sie obige Iterationsvorschrift. Weshalb ist das Verfahren nicht trivial parallelisierbar? (2 Punkte)

Die Parallelität des Verfahrens erhöht sich für dünnbesetzte Matrizen, die sich aus manchen physikalischen Problemstellungen ergeben. Ein Beispiel hierfür ist das Laplace-Problem $\Delta u = f$ des ersten Übungsblattes und des zu den Übungen ausgeteilten Zusatzmaterials. Für ein $2D$ -Feld mit $n_x \times n_y$ Punkten erhält man aus diesem, falls der Lastvektor b und damit der Quellterm f nicht identisch 0 ist, die Vorschrift

$$A_{Laplace} \mathbf{u} = \mathbf{b} \iff u_i^{m+1} = -\frac{1}{4} (b_i - u_{i+1}^m + u_{i-1}^{m+1} + u_{i+n_x}^m + u_{i-n_x}^{m+1}).$$

Um eine noch bessere Parallelität des Gauss-Seidel-Verfahrens zu erhalten, kann man die schon besprochene Schachbrett-Musterung verwenden. Dabei wird das Feld in rote und schwarze Punkte disjunkt so aufgeteilt, dass schwarze Punkte nur rote Nachbarn haben und umgekehrt. Die Vektoren \mathbf{b} und \mathbf{x} lassen sich dadurch in zwei Anteile $\mathbf{x}_r, \mathbf{x}_s$ bzw. $\mathbf{b}_r, \mathbf{b}_s$ aufteilen. Die Länge der Vektoren ist gerade die Anzahl n_s der schwarzen bzw. n_r der roten Punkte. Das Gleichungssystem in der Schachbrett-Musterung hat nun die Gestalt

$$\tilde{\mathbf{A}} \mathbf{x} = \begin{pmatrix} D_R & F \\ E & D_S \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x}_r \\ \mathbf{x}_s \end{pmatrix} = \begin{pmatrix} \mathbf{b}_r \\ \mathbf{b}_s \end{pmatrix}$$

wobei die $n_r \times n_r$ -Matrix D_r die Kopplung eines Punktes mit sich selber repräsentiert (d.h. auf der Diagonalen steht die 4, entsprechend für D_s). Die $n_s \times n_r$ -Matrix E drückt in Ihrer i .ten Zeile die Verbindung des i .ten schwarzen Punktes \mathbf{x}_i mit seinen 4 roten Nachbarn $\mathbf{x}_{i+1}, \mathbf{x}_{i-1}, \mathbf{x}_{i+n}$ und \mathbf{x}_{i-n} aus. Entsprechendes gilt für die $n_r \times n_s$ -Matrix F . Auch das Gleichungssystem in der Schachbrett-Musterung kann man in der üblichen Gauss-Seidel-Manier aufteilen, in dem man setzt:

$$\tilde{D} = \begin{pmatrix} D_R & 0 \\ 0 & D_S \end{pmatrix}, \quad \tilde{L} = \begin{pmatrix} 0 & 0 \\ -E & 0 \end{pmatrix}, \quad \tilde{R} = \begin{pmatrix} 0 & -F \\ 0 & 0 \end{pmatrix},$$

- (b) Leiten Sie nun an Hand dieser Matrizen mittels der Aufspaltung $\tilde{\mathbf{A}} = \tilde{D} - \tilde{L} - \tilde{R}$ das Gauss-Seidel-Verfahren in Schachbrett-Musterung her (In der Form $u_i^{m+1} = \dots$). (6 Punkte)
- (c) Wieviele Komponenten des Verfahrens können Sie nun maximal parallel berechnen? (2 Punkte)

Übung 16 MPI-paralleles Gauss-Seidel-Verfahren

(10 Punkte)

Implementieren Sie ein mit MPI parallelisiertes Gauss-Seidel-Verfahren mit Schachbrett-Musterung für dünnbesetzte Matrizen, wie in der vorherigen Aufgabe hergeleitet. Benutzen Sie die bekannte Matrix des zweidimensionalen Laplace-Problems. Die sequentielle Form haben wir in den Zusatzmaterialien zur Übung besprochen. Sie können folgende Annahmen treffen:

- $n \times n$ -Feld $\Rightarrow \mathbf{b}, \mathbf{x} \in \mathbb{R}^n, A \in \mathbb{R}^{n^2 \times n^2}$.
- $f \equiv 0$, d.h. es gibt keine Quellen oder Senken.
- 1D-Aufteilung des Feldes.
- Hier erlaubte Annahme zur Vereinfachung: $n \bmod p = 0$. Besser ist es aber natürlich, eine allgemeine Variante zu verwenden.

Falls Sie an einem Gebietsrand sind, setzen Sie die Dirichlet-Randbedingung $u = 2$, d.h. für diesen Fall müssen Sie die Nachbarpunkte nicht auswerten.

Bei der Aufteilung des Problems in Streifen kann jeder Prozessor „seinen“ Anteil der Matrix speichern. Den Vektor $\mathbf{x} = \mathbf{x}_r + \mathbf{x}_s$ müssen Sie allerdings mit MPI auf die lokalen Speicher der Prozessoren verteilen. Nach jeder Rechnung muss jeder Prozessor demnach seinen Vektor-Anteil allen anderen Prozessoren mitteilen. Sie benötigen wie für das *Jacobi*-Verfahren eine Abbruchbedingung für die Iteration. Sie können wiederum die Maximums-Norm des Vektors \mathbf{x} verwenden und eine beliebige Toleranz vorgeben. Sie sollten auch eine maximale Anzahl an Iterationen vorgeben, nach denen das Verfahren im Falle der Divergenz abbricht.

Setzen Sie einen beliebigen Startvektor für die Iteration. Die Problemgröße können Sie variabel einstellen, aber die Daten für das sequentielle Programm sollten nicht in den Cache eines Rechners passen, zum Beispiel $n = 1000$. Messen Sie die Ausführungszeiten für die sequentielle Variante (einfach mit einem Prozessor rechnen) und für die parallele Variante mit mehreren Prozessoren und berechnen Sie den Speed-Up. Zur Zeitmessung verwenden Sie am Besten `MPI_Wtime()`. Falls möglich, erstellen Sie ein Diagramm *Zeit-Anzahl Prozessoren* für eine feste Problemgröße.