Humboldt Universität zu Berlin
Institut für Mathematik
JProf. Dr. A. Schröder
O. Schulz

22. April 2010

Iterative Verfahren für große Gleichungssysteme

1. Projekt

Als Modellproblem werde die Randwertaufgabe

$$\Omega = (0,1)^2, \quad \delta u(x,y) = f(x,y) \text{ für } (x,y) \in \Omega, \quad u(u,y) = \varphi(x,y) \text{ auf } \Gamma = \partial\Omega$$

betrachtet. Zur Diskretisierung der Differentialgleichung wählen Sie das in der Vorlesung vorgestellte finite Differenzenverfahren erster Ordnung mit der Maschenweite $h = 1/N$.

Aufgabe 1.1

Stellen Sie die Systemmatrix zum Modellproblem auf und implementieren Sie diese in Matlab. Ordnen Sie die zugehörigen Variablen der Gitterpunkte in lexikographischer Ordnung und Schachbrett-Ordnung an. Wie lautet die Funktion f für $u(x,y) := x^2 + y^2$ und $u(x,y) := \sin(\pi x)\sin(\pi y)$. Stellen Sie die zughörigen Vektoren für die rechte Seite auf. Lösen Sie das Gleichungssystem für $h = 1/32$ mit Hilfe des Gaußschen Algorithmus (Matlab-Routine) [1]

Aufgabe 1.2

Lösen das Gleichungssystem aus der vorherigen Aufgabe mit Hilfe des Gaußchen Algorithmen für Bandmatrizen. Bandmatrizen werden von Matlab nicht speziell unterstützt, aber es gibt entsprechende Erweiterungen im Netz. Verwenden Sie hierfür beispielsweise *gabamp*.

Aufgabe 1.3

Implementieren Sie das, in der Vorlesung vorgestellte, Gauß-Seidel-Iterationsverfahren für das Modellproblem und schreiben Sie für die numerische Analyse eine Routine, um den punktweisen Fehler

$$\epsilon_m := \epsilon(U^m) := \max\{U(i,j) - u(i*h, j*h) \mid 1 \leq i,j \leq N-1\}$$

zu den Beispielen aus 1 für die Iterationen U^m zu messen. Stellen Sie eine Tabelle auf, die für $N = 32$ und die Iterationsschritte $m = 0, 1, 2, 9, 10, 99, 100, 199, 299, 300$ die approximierte Lösung im Mittelpunkt von Ω, ϵ_m, $\epsilon_m/\epsilon_{m-1}$ beinhaltet.

Aufgabe 1.4

Implementieren das SOR-Verfahren für das Modellproblem und lösen hiermit die Gleichungssysteme aus 1. Wählen sie verschiedene Werte für $0 < \omega < 2$ und vergleichen Sie mit

[1] Beachten Sie, dass die Matlab-Routine *mldivide(A,B)* bzw. $A\backslash B$ von sich aus nicht auf die LU-Zerlegung zurückgreifen werden. Rufen Sie daher *lu()* manuell auf, um die Laufzeit des Gaußschen Eliminationsverfahrens zu messen. Laufzeiten können mittels *tic* und *toc* gemessen werden.

$\tilde{\omega} = 2/(1 + sin(\pi h))$ Vergleichen sie das Verfahren mit dem Gauß-Seidel-Verfahren anhand von ϵ_m, $\epsilon_m/\epsilon_{m-1}$.

Präsentation der Ergebnisse am Donnerstag, den 6.5.10