Goethe-Center for Scientific Computing (G-CSC)
Goethe-Universität Frankfurt am Main

Modellierung und Simulation II

(Praktikum SIM2, SoSe 2017)

M. Breit, Dr. A. Vogel

Blatt 9 (Abgabe: Mo., 17.07.2017, 10h)

Auf dem letzten Blatt implementieren Sie eine effiziente Speicherstruktur für dünn besetzte Matrizen.

Aufgabe 1 (CRS-Matrix, 12 Punkte)
Schreiben Sie eine Klasse `SparseMatrix`, die das *Compressed-Row-Storage*-Format umsetzt. Zur Vereinfachung dürfen Sie ein leicht adaptiertes Format benutzen, das wie folgt spezifiziert ist:
Für jede Zeile steht die gleiche Anzahl A an speicherbaren Nicht-Null-Einträgen zur Verfügung. D.h., gespeichert wird ein Array (`std::vector`) »colInds«, das für jede Zeile A Spalten-Indices von Nicht-Null-Einträgen in aufsteigender Sortierung speichert. Hat eine Zeile weniger als A solcher Einträge, wird das Array mit `(std::size_t) -1` aufgefüllt. Entsprechend wird ein Array (`std::vector`) »values« gespeichert, das für jede Zeile A Nicht-Null-Einträge in der in »colInds« vorgegebenen Reihenfolge speichert. Hat eine Zeile weniger als A Nicht-Null-Einträge, wird mit 0 aufgefüllt.

Beispiel Die Matrix

$$\begin{pmatrix} 3.1 & 0 & 0 & 4.1 & 5.9 \\ 0 & 2.6 & 0 & 0 & 0 \\ 5.3 & 0 & 5.8 & 0 & 0 \\ 0 & 0 & 0 & 9.7 & 0 \\ 0 & 9.3 & 0 & 0 & 2.3 \end{pmatrix} \tag{1}$$

wird in diesem Format für $A = 3$ dargestellt durch die beiden Vektoren

$$\text{colInds} = (0, 3, 4;\ 2, -1, -1;\ 0, 2, -1;\ 3, -1, -1;\ 1, 4, -1), \tag{2}$$

$$\text{values} = (3.1, 4.1, 5.9;\ 2.6, 0, 0;\ 5.3, 5.8, 0;\ 9.7, 0, 0;\ 9.3, 2.3, 0). \tag{3}$$

Durch diese Änderung (für die der Preis nicht-optimalen Speicherbedarfs gezahlt wird) entfällt das dritte Array des CRS-Formats, welches Pointer zu den ersten Einträgen jeder Zeile im colInds-Array enthält.

Aufgabe 2 (Anwendung der CRS-Matrix, 8 Punkte)
Adaptieren Sie Ihre Löser-Strukturen, damit Sie ein Poissonproblem sowohl mit der bisherigen (voll besetzten) Matrix-Struktur als auch mit der CRS-Matrix lösen können (mind.: CG + Jacobi, besser: Iterative Solver / CG, GMG, (symm.) Gauß-Seidel, LU).
Vergleichen Sie die beiden Ansätze am Beispiel der 2d-Poissongleichung auf dem Einheitsquadrat $\Omega = (0,1) \times (0,1)$ und der Problemspezifikation

$$\begin{aligned} -\Delta u &= 4 \quad \text{auf } \Omega, \\ u &= 2 - x^2 - y^2 \quad \text{auf } \partial\Omega. \end{aligned} \tag{4}$$

Wie entwickelt sich die Laufzeit mit zunehmender Verfeinerung? Wie viele Verfeinerungen sind möglich (Speicher!)?

Abgabe: Senden Sie Ihren Code sowie sonstige Antworten als Text, PDF oder Scans bitte per E-Mail an `practical.sim2@gcsc.uni-frankfurt.de`. An diese Adresse können Sie sich auch bei Fragen zu den Aufgaben wenden.