Projekt 6: Cholekyzerlegung schwachbesetzter Matrizen

Schwachbesetzte Matrizen zeichnen sich dadurch aus, dass “nur wenige” Einträge ungleich null sind. Es lohnt sich daher nicht, alle Einträge zu speichern, und es sind verschiedene effiziente Formate in Verwendung. Eines der meistverbreiteten ist das *CSC-Format* (*c*ompressed *s*parse *c*olumn). Folgende Datenstruktur repräsentiert dieses Format:

```
class SCSMatrix
{
  unsigned int    mDim;
  unsigned int  * mColStarts;
  unsigned int  * mEntryPos;
  double        * mEntries;
};
```

Dabei ist

- `mDim` die Anzahl der Spalten der Matrix
- `mColStarts` ein Array mit einem Eintrag mehr als Spalten vorhanden sind. Der letzte Eintrag enthält die Anzahl aller Nicht–Null–Elemente. Die Einträge $0, \dots, nDim-1$ geben den Index des ersten Spalteneintrags in `mEntryPos` und `mEntries` an.
- `mEntryPos` und `mEntries` enthalten Spalte für Spalte die Zeilenindizes bzw. Werte der nicht null Einträge. Die Zeilenindizes sind dabei aufsteigend.

Zum Beispiel hat die Matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0.5 & 3 & 0 \\ 0 & \pi & 0 & 7 & -5 \\ -11 & 0 & 0 & 2 & 0 \\ 0 & 0 & 10 & 0 & 0 \end{pmatrix} \tag{1}$$

die CSC-Darstellung

```
mDim       = 5;
mColStarts = {0, 2, 3, 5, 8, 10};
mEntryPos  = {0, 3, 2, 1, 4, 1, 2, 3, 0, 2};
mEntries   = {1, -11, 3.1415, 0.5, 10, 3, 7, 2, 4, -5};
```

Auf http://www.math.tuwien.ac.at/~melenk/teach/numerik_WS0708/projekte/index.html werden Codestücke zum Einlesen und Rausschreiben von Matrizen CSC-Format sowie Beispielmatrizen zur Verfügung gestellt. Insbesondere ist es bei symmetrischen Matrizen üblich, nur den unteren Teil der Matrix abzuspeichern.

1. Implementieren Sie die Matrix–Vektor–Multiplikation für Matrizen im CSC-Format. Nutzen Sie dabei dieses Format aus. Die Anzahl Operationen sollte proportional sein zur Anzahl Nicht–Null–Einträge der Matrix, unabhängig von der Dimension der Matrix.

Testen und dokumentieren Sie Ihren Code.

2. Der Rest des Projekt beschäftigt sich mit der Cholesky-Zerlegung von Matrizen. Es werden also nur noch symmetrische Matrizen betrachtet. Aus Effizienzgründen speichern Sie nur untere Dreiecksmatrizen und rechnen Sie auch nur mit solchen.

a) Modifizieren Sie den Code aus Aufgabe 1 so, dass er die Multiplikation für symmetrische Matrizen durchführt, bei denen nur das untere Dreieck gespeichert ist.

b) Sei A eine Matrix. Von einer Cholesky-Zerlegung der Matrix spricht man, wenn eine untere Dreiecksmatrix L existiert mit $A = LL^T$. Ist L bekannt, so kann man das Problem $Ax = y$ zu lösen darauf zurückführen die Gleichungen

$$Lz = y \qquad \text{und } L^T x = z \tag{2}$$

zu lösen (Vorwärts- und Rückwärtssubstitution).

Implementieren Sie die Vorwärts- und Rückwärtssubstitution für untere Dreiecksmatrizen im CSC-Format.

Testen und dokumentieren Sie Ihren Code. Überzeugen Sie sich davon, dass die Laufzeit der Methoden nicht von der Dimension sondern von der Anzahl Nicht–Null–Elemente abhängt.

3. Das Standardspaltenformat für symmetrische Matrizen speichert alle Einträge des unteren Dreiecks der Matrix nacheinander in einem Vektor. Implementieren Sie eine Konvertierungsroutine vom CSC in das Standardformat. Implementieren Sie die Vorwärts- und Rückwärtssubstitution für untere Dreiecksmatrizen im Standardformat. Testen und dokumentieren Sie Ihren Code.

4. Schwachbesetzte Matrizen haben nicht notwendigerweise schwachbesetzte Choleskyfaktoren. Beweisen Sie folgende Aussage über die Besetzungsstruktur: Sei A eine SPD–Matrix. Sei die untere Dreiecksmatrix L ihr Cholekyfaktor. Sei

$$J_i(A) := \min\{j : A_{ij} \neq 0\} \tag{3}$$

der erste Spaltenindex einer Zeile, für den A nicht null ist. Dann gilt

$$L_{ij} = 0 \quad \text{für } j < J_i(A). \tag{4}$$

In L bleiben also führende Nullen in Zeilen erhalten.

5. In der Vorlesung wurde die Choleskyzerlung in der Variante (VORL)

for $k = 1, \ldots, n$
  $L_{kk} = (A_{kk} - \sum_{j=1}^{k-1} L_{kj}^2)^{1/2}$
  for $i = k+1, \ldots, n$
    $L_{ik} = (a_{ik} - \sum_{j=1}^{k-1} L_{ij} L_{kj})/L_{kk}$
  end
end

angegeben. Zeigen Sie, dass diese äquivalent ist zu der folgenden Variante (PROJ), die nur auf Spalten agiert und A mit seinem Choleskyfaktor überschreibt:

for $k = 1, \ldots, n$
  $A(k,k) = \sqrt{(A(k,k)}$
  $A(k+1:n,k) = A(k+1:n,k)/A(k,k)$
  for $j = k+1, \ldots, n$
    $A(j:n,j) = A(j:n,j) - A(j:n,k)A(j,k)$
  end
end

6. Wählen Sie eine der oben genannten Variationen und implementieren Sie die Choleskyzerlegung für vollbesetzte Matrizen im Standardspaltenformat. Dokumentieren Sie Ihren Code und Ihre Tests.

7. Implementieren Sie die Cholskyzerlegung für CSC-Matrizen ausgehend von der Variante (PROJ) aus Aufgabe 5. Gehen Sie dazu in drei Schritten vor:

a) Implementieren Sie eine Funktion, die zur Eingabe der Matrix `A` im CSC-Format einen "leeren" Choleskyfaktor `L` erzeugt, der eine Besetzungstruktur gemäß Aufgabe 4 aufweist. Die Werte der existierenden Einträge werden mit 0 initialisiert.

b) Kopieren Sie `A` nach `L`.

c) Berechnen Sie die Einträge von `L` gemäß Variante (PROJ).

Dokumentieren Sie Ihren Code und Ihre Tests.

8. Laden Sie sich die Beispielmatrizen herunter.

a) Die Komplexität Ihrer Implementierung soll untersucht werden. Dazu liegen zwei Serien von Matrizen vor:

- `tridiagonal_n.mtx`, wobei `n` die Dimension angibt. Es handelt sich um Tridiagonalmatrizen.
- `poisson_n.mtx`, wobei `n` die Dimension angibt. Es handelt sich dabei um Matrizen, die aus der Diskretisierung des Poissonproblems in 2D entstehen. Es sind Bandmatrizen, bei denen die Bandbreite sich wie $\sqrt{n}$ verhält.

Mit der Datei `mmread.m` können Sie die Matrizen auch in Matlab einlesen. Der `spy`-Befehl visualisiert die Besetzungsstruktur. Die von Ihnen generierten Choleskyfaktoren lassen sich auch auf diese Weise anzeigen.

```
> A = mmread("poisson_256.mtx");
> spy(A);
```

Schreiben Sie Programme (für vollbesetzte und schwachbesetzte Arithmetik), die diese Matrizen einlesen, choleskyzerlegen und ein Gleichungssystem lösen. Notieren Sie die Dimension (und bei den schwachbesetzen auch die Anzahl der Nichtnulleinträge sowie die Anzahl der vorhergesagten Nichtnulleinträge). Messen Sie die Zeit für das Zerlegen und das Lösen. Welche Komplexitäten erwarten Sie aufgrund der Struktur der Matrizen? Welche beobachten Sie? Dokumentieren Sie Ihre Ergebnisse mit geeigneten Matlabplots.

b) In (5) sehen Sie die Besetzungsstruktur der Matrizen `arrowhead_backwards_n.mtx`. Welche Besetzungsstruktur haben die entsprechenden Choleskyfaktoren? Wie sehen die Laufzeiten Ihres schwachbesetzten Choleskyzerlegers aus? Erklären Sie.

$$B = \begin{pmatrix} * & * & * & * & * & * \\ * & * & & & & \\ * & & * & & & \\ * & & & * & & \\ * & & & & * & \\ * & & & & & * \end{pmatrix} \tag{5}$$

In (6) sehen Sie die Besetzungsstruktur der Matrizen `arrowhead_n.mtx`. Welche Besetzungsstruktur haben die entsprechenden Choleskyfaktoren? Wie sehen die Laufzeiten

Ihres schwachbesetzten Choleskyzerlegers aus?

$$A = \begin{pmatrix} * & & & & & * \\ & * & & & & * \\ & & * & & & * \\ & & & * & & * \\ & & & & * & * \\ * & * & * & * & * & * \end{pmatrix} \tag{6}$$

Überzeugen Sie sich davon, dass man aus dem eine Format durch Umnumerieren der Unbekannten das andere erzeugen kann, also dass es eine Permutationsmatrix P gibt mit $A = P^{-1}BP$