Übungen zur Vorlesung

"Objektorientiertes Programmieren im Wissenschaftlichen Rechnen"

Dr. O. Ippisch, Dr. C. Engwer

Abgabe am 29. 06. 2010 in der Vorlesung

Übung 1 Eigenwerte einer Matrix

In der Numerik ist es immer wieder nötig Eigenwerte einer Matrix zu wissen. Ein besonders einfaches Verfahren zur Berechnung des größten Eigenwertes einer Matrix ist die Potenzmethode nach v. Mises. Es handelt sich hierbei um ein iteratives Verfahren.

Gegeben eine quadratische Matrix $A \in \mathbb{C}^{n\times n}$ mit Eigenwerten $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$. Wir wählen einen Startvektor $z^0 \in \mathbb{C}^n$ mit $Az^0 \neq 0$. In jedem Iterationsschritt wird einfach die aktuelle Iterierte normiert und dann die Matrix A auf den normierten Vektor angewandt:

$$z^{(k+1)} = A \cdot z^{(k)} \qquad \text{für} \quad k = 0, 1, 2, \ldots$$

Der Eigenwert λ_1 ergibt sich dann als

$$\lambda_1 = \lim_{k\to\infty} \frac{z^{(k+1)}}{z^{(k)}}$$

und $\int k \to \infty z^k$ ist der dazugehörige Eigenvektor.

1. Welche Mathematischen Operationen werden hier verwendet.
2. Welche Teile sollte man als eigene Komponente programmieren, um sie in unterschiedlichen Berechnungen einsetzen zu können?
3. Programmieren Sie die Komponenten und daraus das Potenzverfahren. Verwenden Sie dazu das abstrakte Iteratorinterface.
4. Iterieren Sie, bis die Änderung in λ_1 kleiner als 10^{-8} sind.
5. Testen Sie das Verfahren mit der Matrix

$$\begin{pmatrix} 3 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 3 \end{pmatrix}$$

Der Eigenwert ist 5. Was ist der dazugehörige Eigenvektor?

6 Punkte

Übung 2 Abstrakte Interfaces & Strukturausnutzung

Betrachten Sie folgende Matrix Struktur. Diese ist bekannt als 3-Punkt-Stern, sie ergibt sich z.B. aus der Finite-Differenzen-Diskretisierung der eindimensionalen Wärmeleitungsgleichung. Es handelt sich um eine symmetrische, dünnbesetzte Matrix. Jede Zeile entspricht einem Gitterknoten. Die Spalten enthalten für jeden Nachbarknoten eine -1. Der Wert der Hauptdiagonale entspricht der Anzahl an Nachbarknoten.

$$\begin{pmatrix} 1 & -1 & 0 & & & \dots & 0 \\ -1 & 2 & -1 & \ddots & & & \vdots \\ 0 & -1 & 2 & -1 & \ddots & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & \ddots & -1 & 2 & -1 & 0 \\ \vdots & & & \ddots & -1 & 2 & -1 \\ 0 & \dots & & & 0 & -1 & 1 \end{pmatrix}$$

Es gibt Entsprechungen in 2D (5-Punkt-Stern) und 3D (7-Punkt-Stern).

Wir haben auf den vorangegangenen Zetteln eine abstrakte Schnittstelle für allgemeine Matrizen implementiert. Wir wollen dieses Iterator basierte Interface beibehalten.

Welche Struktureigenschaften der Matrix können Sie bei der Implementierung einer optimierten Matrix-Implementierung ausnutzen? Wie macht sich die Nutzung dieser Eigenschaften bemerkbar (reduzierter Speicherverbrauch, schnellere Programmlaufzeit)?

Lassen Sie ihrer Phantasie freien Lauf ;-)

6 Punkte

ÜBUNG 3 STL ALGORITHMEN: VEKTOR NORM

Gegeben ein Vektor $x \in \mathbb{R}^n$, kennen wir verschiedene Vektornormen, z.B.:

$$||x||_2 = \sqrt{x \cdot x}, \qquad ||x||_1 = \sum_{i=0}^{n-1} |x_i|, \qquad ||x||_\infty = \max_{i=0}^{n-1} |x_i|$$

Der Vektor x sei als STL-Container gespeichert.

1. Implementieren Sie die obigen Vektornormen unter Verwendung der STL Algorithmen.
   - Welche Algorithmen können hier eingesetzt werden, um Programmieraufwand zu sparen?
   - Würde sich Ihre Wahl des verwendeten Algorithmus ändern, wenn der Container modifiziert werden darf? Wenn ja wie?
2. Schreiben Sie die Algorithmen so um, dass diese möglichst viel Code teilen.
   - Identifizieren Sie die Gemeinsamkeiten der verschiedenen Normberechnungen.
   - Entwerfen Sie ein Interface um diese Gemeinsamkeiten auszunutzen.
   - Welche Art von Austauschbarkeit würden Sie für die sich unterscheidenden Teile wählen und warum? (Template-Spezialisierung, statischer Polymorphismus, dynamischer Polymorphismus)

8 Punkte

Wie immer gilt: Kommentieren Sie Ihr Programm. Erklären Sie was Sie tuen.