

Übungen zur Vorlesung
“Objektorientiertes Programmieren im Wissenschaftlichen Rechnen”

Dr. Olaf Ippisch, Ole Klein

Abgabe am 22. 06. 2014 in der Vorlesung

ÜBUNG 1 STL-CONTAINERTYPEN

Sie haben in der Vorlesung die verschiedenen Containertypen der STL besprochen. Jeder Typ bietet gewisse Garantien und ist für unterschiedliche Verwendungszwecke prädestiniert.

Sie haben eine Liste von Szenarien, welchen Containertyp würden Sie verwenden und warum?

1. Sie schreiben eine Finite-Differenzen-Diskretisierung der Laplace-Gleichung. Sie haben hierzu ein strukturiertes Gitter. Zu jedem Gitterknoten wird die Lösung an diesem Punkt gespeichert. Alle Knoten des Gitters sind konsekutiv durchnummeriert und die Anzahl ist a-priori bekannt.

Welcher Containertyp ist für die Speicherung der Knotenwerte gut geeignet?

2. Sie schreiben einen Quicksort. Um die rekursiven Funktionsaufrufe zu sparen möchten sie die Pivotelemente in einer Stack-Struktur speichern.

Welcher Containertyp ist gut geeignet um die Pivotelemente zu speichern und welcher um die Daten selbst zu speichern?

3. Bei der Implementierung direkter Löser für dünnbesetzte Matrizen ist es üblich, zunächst die Bandbreite der Matrix zu minimieren, um den Speicherbedarf der resultierenden LU-Zerlegung zu reduzieren. Eine Methode hierzu ist der Cuthill-McKee Algorithmus. Im Laufe dieses Algorithmus müssen Elemente in einem FIFO (first-in, first-out) zwischengespeichert werden. Der FIFO behält hierbei ursprüngliche Nummerierung der Elemente bei.

Welchen Containertyp würden Sie als FIFO einsetzen?

4. In Ihrem Finite-Differenzen-Programm wird die Lösung auf einem Teil des Gebietsrandes a-priori festgeschrieben. Knoten auf diesem Teil des Gebietsrandes (Dirichlet-Knoten) sind keine echten Freiheitsgrade und müssen beim Aufstellen der Matrix gesondert behandelt werden. Die Anzahl dieser Knoten ist klein im Vergleich zur Gesamtanzahl an Knoten.

Sie möchten die Liste der Dirichlet-Knoten, sowie deren Lösungswerte, dynamisch aus einer Konfigurationsdatei lesen und knotenweise zugreifbar machen. Bei der Lösung linearer PDEs ist meistens der Speicher der limitierende Faktor, da man sehr große Probleme lösen möchte.

In welchem Container würden Sie die Indizes der Dirichlet-Knoten und die Werte an diesen Knoten speichern?

12 Punkte

ÜBUNG 2 STL-ALGORITHMEN: VEKTORNORM

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

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

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 tun.