

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 1 (Abgabe: Mo., 8.5.2016, 10h)

Auf dem ersten Aufgabenblatt werden Sie einige grundlegende Klassen implementieren, die im weiteren Verlauf des Praktikums benötigt werden. Eine erste Anwendung wird in der Diskretisierung einer Poisson-Gleichung bestehen.

Für alle zu implementierenden Klassen steht ein Code-Gerüst zur Verfügung, in das Sie Ihre Implementierung an den markierten Stellen einfügen können. Sie dürfen die Klassen auch anders als vorgeschlagen implementieren, sofern das vorgegebene Interface dabei erfüllt wird.

Aufgabe 1 (Vektor und Matrix, 4 Punkte)

(a) Implementieren Sie eine Klasse **Vector**, die mindestens das in der Datei »vector.h« vorgegebene und beschriebene Interface erfüllt.

Tipp: Sie können einige der Anforderungen erfüllen, indem Sie von der STL-Klasse `std::vector<double>` ableiten.

(b) Implementieren Sie eine Klasse **Matrix**, die mindestens das in der Datei »matrix.h« vorgegebene und beschriebene Interface erfüllt.

Aufgabe 2 (Strukturiertes Gitter, 6 Punkte)

Implementieren Sie eine Klasse **StructuredGrid**, die mindestens das in der Datei »structured_grid.h« vorgegebene und beschriebene Interface erfüllt.

Die Klasse soll ein strukturiertes, in jede Dimension (1, 2 oder 3) separat äquidistantes Vertexgitter beschreiben, das durch die unteren und oberen Ränder sowie durch die Anzahl der Vertices in jeder Dimension definiert ist. Die Vertices sollten am besten erst in x-Richtung, dann in y- und z-Richtung nummeriert sein. Dies ist nicht strikt erforderlich, macht Ihnen aber insb. in Aufgabe 3 das Leben einfach.

Aufgabe 3 (VTK-Exporter, 4 Punkte)

Implementieren Sie eine Klasse `VTKExporter`, die einen `Vector` ins VTK-Format exportiert, der für jeden Vertex eines `StructuredGrid` genau einen Wert enthält (und zwar in der im Grid definierten Ordnung).

- Beachten Sie hierzu in »file-format-vtk.pdf« insb. die Spezifikation für den Datensatztyp `STRUCTURED_POINTS` und die Beispieldatei »example.vtk«.
- Nutzen Sie das ASCII-Dateiformat.

Aufgabe 4 (Poisson-Diskretisierung, 6 Punkte)

Gegeben ist das Poisson-Problem auf dem d -dimensionalen Gebiet $\Omega = (0, 1)^d$:

$$\begin{aligned} -\Delta u &= f & \text{auf } \Omega, \\ u &= g & \text{auf } \Gamma_D \subset \partial\Omega. \end{aligned}$$

mit ggf. von der Raumkoordinaten \mathbf{x} abhängigen Funktionen f und g . Schreiben Sie eine Klasse `PoissonDisc`, die eine Finite-Differenzen-Diskretisierung für das Poisson-Problem (mit $d \in \{1, 2, 3\}$) durchführt, d.h. für ein gegebenes strukturiertes Gitter die Matrix und den Rechte-Seite-Vektor des entstehenden Gleichungssystems erzeugt.

Hinweis (zu allen Aufgaben): Sie können Ihre Implementierungen testen, indem Sie den Code in »main.cpp« (oder Teile davon) kompilieren und ausführen. Sollten Ihre Resultate von den vorgegebenen abweichen, enthält Ihr Code mit großer Wahrscheinlichkeit noch Fehler.

Zum Kompilieren nutzen Sie bitte – soweit möglich – `cmake`. Angenommen, Ihre Dateien liegen im Ordner »code«. Dann wechseln Sie in der Konsole/Terminal (Linux/Mac/Windows?) in diesen Ordner und geben Sie folgenden Befehl ein:

```
mkdir build && cd build && cmake .. && make
```

Wenn Sie alles vollständig und syntaktisch korrekt implementiert haben, sollte so in »code/bin« ein Executable mit dem Namen »execMain« entstehen, das Sie zum Testen ausführen können.

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.