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 6 (Abgabe: Mo., 26.6.2017, 10h)

Auf diesem Blatt untersuchen Sie die geschachtelte Iteration und lösen ein nichtlineares Problem sowohl mit dem Newton-Mehrgitter-Verfahren als auch mit einem nichtlinearen Mehrgitter-Verfahren. Mit einem $\star$ gekennzeichnete Aufgaben sind als Zusatz zu verstehen und werden mit Bonus-Punkten bewertet.

Aufgabe 1 (Nested GMG, 8 Punkte)
Implementieren Sie eine Klasse `NestedGMG`, welche die geschachtelte Iteration ausführt.

(a) Lösen Sie mithilfe der geschachtelten Iteration das Poisson-Problem auf dem Einheitsquadrat $\Omega = (0,1) \times (0,1)$:

$$\begin{aligned} -\Delta u &= 4 \quad \text{auf } \Omega, \\ u &= 2 - x^2 - y^2 \quad \text{auf } \partial\Omega, \end{aligned} \tag{1}$$

Beginnen Sie mit einem Grobgitter Ω_0 bestehend aus 3×3 Knoten. Lösen Sie mit 1, 2, 3, 4, ... Verfeinerungsstufen (solange der Speicher reicht) und notieren Sie die Fehlernorm $\|e\|_h := \sqrt{h^d e^T e}$ der Lösung in den Gitterknoten. Verwenden Sie 1 GMG-Iteration (V-Zyklus, je 1x Vor-/Nachglätten mit Gauß-Seidel) pro Level. Kommentieren Sie.

(b) Lösen Sie nun dieselben Probleme mit einer klassischen GMG-Iteration (V-Zyklus, je 1x Vor-/Nachglätten mit Gauß-Seidel). Wie viele Iterationen benötigen Sie, um höchstens dieselbe Fehlernorm $\|e\|_h$ zu erhalten? Kommentieren Sie.

Aufgabe 2 (Newton-Verfahren, 6 Punkte)
Implementieren Sie eine Klasse `Newton`, die eine Newton-Iteration

$$\mathbf{x}_{k+1} = \mathbf{x}_k - J_d^{-1}(\mathbf{x}_k)\, d(\mathbf{x}_k) \tag{2}$$

ausführt. Der Klasse sollen ein Grid, eine Diskretisierung und ein linearer Löser übergeben werden, mit denen die linearen Systeme assembliert und dann gelöst werden. Nutzen Sie dazu die (neuen) Methoden `assemble_defect()` sowie `assemble_jacobian()` der Interface-Klasse `IAssemble`. Ermöglichen Sie optionalen Output über Defektnorm und deren Reduktion in jedem Schritt der Iteration.

Aufgabe 3 (⋆Nichtlineares Mehrgitterverfahren, 8 Bonus-Punkte)
Implementieren Sie ein nichlineares Mehrgitterverfahren in einer neuen Klasse `NonlinearGMG`, für die Sie ein Gerüst in der Datei »gmg_nl.h« finden. Nutzen Sie die *Nested Iteration* zum Aufbau geeigneter Lösungen auf den jeweiligen Grobgittern. Selbstverständlich dürfen Sie zusätzlich auch die Option zur Verwendung des *Full Approximation Scheme* bereitstellen.

Aufgabe 4 (Ein nichtlineares Problem, 6 Punkte + 6 Bonus-Punkte)
Gelöst werden soll das folgende nichtlineare Problem auf dem quadratischen Gebiet $\Omega = (-1, 1) \times (-1, 1)$:

$$\begin{aligned} -\Delta u &= -\frac{1+u^2}{u^3} \quad \text{auf } \Omega, \\ u(x,y) &= \sqrt{1+x^2+y^2} \quad \text{auf } \partial\Omega. \end{aligned} \tag{3}$$

(a) Geben Sie eine Lösung von (3) an.

(b) Lösen Sie das Problem mit dem Newton-GMG-Verfahren und untersuchen Sie den resultierenden Fehler für die Verfeinerungsstufen $1, 2, \ldots$ an. Benutzen Sie ein äquidistantes 4x4-Vertex-Gitter (nicht 3x3!) als Basis-Gitter und konstant 1 als Startlösung.
Hinweis: Hierzu ist es zunächst notwendig, eine eigene Diskretisierungsklasse für das Problem zur Verfügung zu stellen, welche die Methoden `assemble_defect()` und `assemble_jacobian()` von `IAssemble` implementiert. Dazu schreiben Sie am besten eine neue Klasse. Ein grobes Gerüst steht Ihnen dafür in »poisson_disc.h« zur Verfügung. Als Defekt $d(u)$ diskretisieren Sie

$$d(u) \coloneqq -\Delta u + \frac{1+u^2}{u^3} \tag{4}$$

und als Jacobian die Jacobi-Matrix der diskretisierten Defekt-Funktion.

⋆(c) Lösen Sie das Problem mit einem nichtlinearen GMG. Verwenden Sie einen nichtlinearen Jacobi-Glätter, der einen Newton-Schritt für jeden

Freiheitsgrad durchführt. Verwenden Sie einen Newton-Löser als Basislöser.
Untersuchen Sie den resultierenden Fehler bei wachsender Verfeinerung. Vergleichen Sie den Aufwand zur Lösung mit dem des Newton-GMG-Ansatzes. Vergleichen Sie auch *Full Approximation Scheme* und *Nested Iteration*.

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.