Michael Feischl
Dirk Praetorius

Sommersemester 2014
31. März 2014

Übungen zur Vorlesung
Einführung in Scientific Computing

Serie 3

Aufgabe 3.1. Sei $\mathcal{T} = \{T_1, \dots, T_{N-1}\}$ mit Knoten $\mathcal{K} = \{a = x_1, \dots, x_N = b\}$ und Elementen $T_j = [x_j, x_{j+1}]$ ein Gitter auf $[a, b]$. Sei $\mathcal{S}^1(\mathcal{T}) = \{u \in C(a, b) \mid u|_T \in \mathcal{P}_1(T)\ \forall T \in \mathcal{T}\}$ der Raum der stückweise affinen, global stetigen Funktionen. Sei $\Pi^{\mathcal{T}} : L^2(a, b) \to S^1(\mathcal{T})$ die L^2-Orthogonalprojektion.
Schreiben Sie eine Funktion `L2projection`, die $\Pi^{\mathcal{T}}$ realisiert. Die Inputparameter sind ein function-handle auf die zu approximierende Funktion und ein Vektor $\mathtt{x} = (a = x_1, \dots, x_N = b)$ der das Gitter darstellt. Die Ausgabeparameter sind ein Vektor, der die Knotenwerte von $\Pi^{\mathcal{T}} u$ enthält, sowie die *Energie* von $\Pi^{\mathcal{T}} u$.

1. Überlegen Sie sich zuerst, welche Konvergenzordnung Sie für $\|u - \Pi^{\mathcal{T}} u\|_{L^2(a,b)}$ maximal erreichen können.

2. Schreiben Sie die Funktion `L2projection`. Gehen Sie dazu element-basiert vor, wie in der Vorlesung angegeben. Warum wählt man als Basis die Hutfunktionen? Die *Massenmatrix* `A` können Sie analytisch berechnen (siehe Vorlesung), den *Lastvektor* `b` jedoch nicht. Verwenden Sie dazu auf jedem Element eine Gaussregel mit n Punkten.

3. Die Funktion soll auch die *Energie* von $\Pi^{\mathcal{T}} u$ berechnen, die durch $\|\Pi^{\mathcal{T}} u\|^2_{L^2(a,b)}$ definiert ist. Überlegen Sie sich, wie Sie die Energie nur mit Hilfe des Knotenvektors und der Massenmatrix bzw. des Lastvektors berechnen können.

4. Der Energiefehler $\|u - \Pi^{\mathcal{T}} u\|_{L^2(a,b)}$ ist von Interesse. Angenommen, Sie kennen von der Funktion u die $L^2(a, b)$-Norm. Wie können Sie den Energiefehler bestimmen?. Wie kann man in Experimenten $\|u - \Pi^{\mathcal{T}}_a u\|_{L^2(a,b)}$ berechnen, wenn man die L^2-Norm von u nicht kennt?

5. Machen Sie Experimente mit Gittern unterschiedlicher Gitterweite. Verwenden Sie wieder die Funktionen $x^{1/10}$ und e^x. Welche Konvergenzordnungen stellen sich für den Energiefehler ein? Wie kann man die maximale Konvergenzordnung für die Funktion $x^{1/10}$ erhalten?

Aufgabe 3.2. Im Skriptum wird folgender (Pseudo-)Code zum Aufbau der *Massenmatrix* `A` von Aufgabe 3.1 vorgeschlagen

```
A = sparse(N,N);
for j = 1:size(elements,1)
      nodes = elements(j,:);
      A(nodes,nodes) = A(nodes,nodes) + [2 1; 1 2]/6*hTj;
end
```

1. Welches asympthotische Verhalten $t(N) = \mathcal{O}(N^\alpha)$ erwarten Sie für die zum Aufstellen von `A` benötigte Rechenzeit.

2. Plotten Sie die benötigte Rechenzeit zum Aufstellen der Matrix `A`, sodass der Aufwand $t(N) = \mathcal{O}(N^\alpha)$ sichtbar ist.

3. Wie erklären Sie sich den Unterschied zwischen Erwartung und Beobachtung (Hinweis: MATLAB speichert *sparse* Matrizen intern im CCS-Format)? Ist ihr eigener Code (der natürlich nicht vom Skriptum abgeschrieben wurde) besser?

4. Verbessern Sie den Code so, dass die Zeitmessung mit dem erwarteten Aufwand übereinstimmt (Hinweis: `help sparse`). Visualisieren Sie den Aufwand $t(N) = \mathcal{O}(N^\alpha)$ des verbesserten Codes im selben Plot.

Aufgabe 3.3. Betrachten Sie das Skalarprodukt

$$a(u,v) := \int_a^b u'(x)v'(x)dx + \left(\int_a^b u(x)dx\right)\left(\int_a^b v(x)dx\right).$$

Es sei $\Pi_a^{\mathcal{T}} : C^1[a,b] \to S^1(\mathcal{T})$ die Orthogonalprojektion bezüglich dieses Skalarprodukts. Dabei seien $\mathcal{T}$ und $\mathcal{S}^1(\mathcal{T})$ wie in Aufgabe 3.1.

1. Sei $\|\cdot\|_a^2 := a(\cdot,\cdot)$ die induzierte Norm. Welche Konvergenzordnung erwarten Sie für $\|u - \Pi_a^{\mathcal{T}} u\|_a$?

2. Schreiben Sie eine Funktion `solve_neumann`, die $\Pi_a^{\mathcal{T}}$ realisiert. Inputparameter sind wieder ein Vektor **x** wie oben und ein function-handle auf u. Rückgabewert ist wieder der Knotenvektor von $\Pi_a^{\mathcal{T}} u$ und die Energie $\|\Pi_a^{\mathcal{T}} u\|_a^2$. Berechnen Sie die Energie wie in Aufgabe 1. Gehen Sie wieder element-basiert vor, um die *Steifigkeitsmatrix* $\mathtt{A}_{ij} = \int \varphi_j'(x)\varphi_i'(x)dx$ zu berechnen. Wie können Sie das analytisch machen? Die Stabilisierung $\mathtt{C}_{ij} = \int \varphi_j(x)dx \int \varphi_i(x)dx$ ist eine Rang-1 Matrix, da Sie als Produkt von zwei Vektoren geschrieben werden kann. Nutzen Sie das in der Implementierung aus.

3. Betrachten Sie die Approximation der Funktion $\cos(x)$ auf dem Intervall $(0, 2\pi)$. Machen Sie Experimente mit verschiedenen Gittern und plotten Sie den Energiefehler.