Michael Feischl
Dirk Praetorius

Sommersemester 2014
24. März 2014

Übungen zur Vorlesung
Einführung in Scientific Computing

Serie 2

Aufgabe 2.1. Es sei

$$x_i = a + i \cdot h, \quad i = 0, \ldots, N, \quad h = \frac{b-a}{N}$$

ein äquidistantes Gitter auf dem Intervall $[a, b]$, und $n \in \mathbb{N}$. Im Folgenden bezeichne $Q_{nN}^{[a,b]}$ die summierte n-Punkt Gauss-Legendre-Quadratur auf $[a, b]$ (d.h. Länge $n-1$ gemäß Vorlesung). Schreiben sie eine Funktion `hGauss`, welche als Parameter ein function-handle auf die zu integrierende Funktion f, die Intervallendpunkte a und b, die Ordnung n und die Anzahl der Teilintervalle N bekommt. Die Funktion soll den berechneten Integralwert sowie die Anzahl der Funktionsauswertungen zurückliefern. Berechnen Sie die Gauss-Punkte nur einmal auf dem Intervall $[-1, 1]$, und transformieren Sie diese entsprechend.

Aufgabe 2.2. Machen Sie nun numerische Experimente. Dazu betrachten Sie die beiden Funktionen $f_1(x) = \exp(x)$ und $f_2(x) = x^{0.1}$ auf dem Intervall $[0, 1]$. Sei Qf das Integral von f. Dann gilt (siehe Beweis von Satz 10)

$$|Q_{nN}^{[a,b]}(f) - Q(f)| \le Ch \sum_{i=0}^{N-1} \inf_{p \in \mathbb{P}_{2n-1}} \|f - p\|_{\infty,[x_i,x_{i+1}]}$$

- Welche maximale Konvergenzrate in N können Sie für fixe Ordnung n erwarten?
- Welche Konvergenzrate bedeutet das bezüglich der Anzahl der Funktionsauswertungen?
- Welche Konvergenzraten erwarten Sie für f_1 und f_2?
- Berechnen Sie für $n = 1, 2, 3$ und $N = 2^i$, $i = 0, \ldots, 9$, die Integrale von f_1 und f_2 mit der summierten Gauss-Regel, wobei N die Anzahl der Teilintervalle angibt und n die Ordnung. Plotten Sie für jeweils fixes n den absoluten Fehler doppelt logarithmisch gegen die Anzahl der Funktionsauswertungen.
- Welche Konvergenzraten stellen sich ein? Stimmt die experimentelle Beobachtung mit ihrer Erwartung überein?

Aufgabe 2.3. Berechnen Sie für $n = 1, \dots, 5$ die Integrale von f_1 und f_2 mit der einfachen Gauss-Regel aus der letzten Serie, wobei n die Ordnung angibt. Plotten Sie die absoluten Fehler wieder doppelt logarithmisch gegen die Anzahl der Funktionsauswertungen. Was beobachten Sie im Vergleich zur summierten Gauss-Regel?

Aufgabe 2.4. Schreiben Sie eine Funktion `hpGauss`, die eine summierte Gauss-Regel auf *beliebigen* Gittern mit *beliebiger* Polynomgradverteilung realisiert. Die Inputparameter sind ein function-handle auf die zu integrierende Funktion f, ein Vektor $\mathbf{x} = (a = \mathbf{x}_1, \dots, \mathbf{x}_N = b)$ und ein Vektor $\mathbf{p} = (\mathbf{p}_1, \dots, \mathbf{p}_{N-1})$. Dabei soll $\mathbf{x}$ das Gitter und $\mathbf{p}$ die Ordnung auf jedem Intervall sein. Die Funktion berechnet also

$$Q_{\mathbf{x},\mathbf{p}}^{[a,b]}(f) = \sum_{i=1}^{N-1} Q_{\mathbf{p}_i}^{[\mathbf{x}_i,\mathbf{x}_{i+1}]}(f),$$

wobei $Q_{\mathbf{p}_i}^{[\mathbf{x}_i,\mathbf{x}_{i+1}]}$ eine Gauss-Regel auf dem Intervall $[\mathbf{x}_i, \mathbf{x}_{i+1}]$ mit $\mathbf{p}_i$ Punkten ist. Die Funktion soll auch die Anzahl der Funktionsauswertungen von f zurückgeben. Verwenden Sie ihre Funktion `gauss`, um die Gaussknoten und Gewichte zu erzeugen.

Aufgabe 2.5. Berechnen Sie das Integral der Funktion $f(x) = x^{0.1}$ auf dem Intervall $[0, 1]$. Schreiben Sie eine Funktion `compareGauss`, in der Sie mit Hilfe von `hpGauss` das Integral von f durch folgende Methoden berechnen:

- p-Version: Für $p = 2^i$, $i = 1, \dots, 9$ durch Gauss-Regeln der Ordnung p auf dem Gitter $[0, 1]$.
- uniforme h-Version: Für $N = 2^i$, $i = 1, \dots, 9$ durch summierte Gauss-Regeln mit N Teilintervallen und der Ordnung 1.
- graduierte h-Version: Verwenden Sie dieselbe Methode wie zuvor, allerdings mit *graduierten* Gittern, d.h. erzeugen Sie ein uniformes Gitter und wenden Sie auf die Knotenpunkte die Funktion $x \mapsto x^3$ an.
- hp-Version: Sei $\sigma = 0.15$. Ein *geometrisches Gitter* ist ein Gitter der Form

  $$\mathbf{x}_1 = 0, \quad \mathbf{x}_i = \sigma^{N-i}, \quad i = 1, \dots, N.$$

  Für $N = 2^i$, $i = 1, \dots, 9$, erzeugen Sie ein geometrisches Gitter und eine Polynomgradverteilung, die konstant N ist, und rufen sie `hpGauss` mit diesen Parametern auf.

Plotten sie doppelt logarithmisch die Fehler der vier obigen Versionen gegen ihren jeweiligen Aufwand (also die Anzahl Funktionsauswertungen). Was beobachten Sie?