UNIVERSITÄT BASEL

Numerik der partiellen Differentialgleichungen

Frühjahrssemester 2014
Prof. Dr. H. Harbrecht

Programmieraufgabe 2.

zu bearbeiten bis **Montag, 7.04.2014**.

In der vorangegangenen Programmieraufgabe wurde das Poisson-Problem

$$\begin{aligned} -\operatorname{div}\big(\alpha(\mathbf{x},\mathbf{y})\nabla u(\mathbf{x},\mathbf{y})\big) &= f(\mathbf{x}) \qquad && \mathbf{x}\in D,\ \mathbf{y}\in[-1,1]^M \\ u(\mathbf{x},\mathbf{y}) &= 0 && \mathbf{x}\in\partial D,\ \mathbf{y}\in[-1,1]^M \end{aligned}$$

mit variablem *Diffusionskoeffizienten* $a(\cdot,\mathbf{y})\in C(\overline{D})$ betrachtet. Ziel war es, mit Hilfe der *Monte-Carlo-Quadratur*, den Erwartunswert

$$\mathbb{E}[u](\mathbf{x}) = \int\limits_{[-1,1]^d} 2^{-d}u(\mathbf{x},\mathbf{y})\,\mathrm{d}\mathbf{x} \approx \frac{1}{N}\sum_{i=1}^{N} u(\mathbf{x},\boldsymbol{\xi}_i)$$

der Lösung für unabhängig uniform verteilte Zufallsvariablen $\boldsymbol{\xi}_i\in[-1,1]^M$ zu bestimmen. Mit dem *Gesetz der grossen Zahlen* kann man zeigen, dass die Konvergenzrate der Monte-Carlo-Quadratur, unabhängig von der Dimension M, im Mittel $\mathcal{O}(N^{-1/2})$ ist. Das Verfahren liefert demnach keine deterministische Konvergenzrate.

Möchte man Quadraturverfahren mit deterministischer Konvergenzrate verwenden, so ist die einfachste Möglichkeit die Verwendung einer *Quasi-Monte-Carlo-Quadratur*. Hierbei wird der Erwartungswert analog zur Monte-Carlo-Quadratur durch ein arithmetisches Mittel approximiert. Dabei werden die *Sampling-Punkte* $\boldsymbol{\xi}_i\in[-1,1]^M$ durch deterministische Folgen mit niedriger *Diskrepanz*, das bedeutet mit guten Gleichverteilungseigenschaften, gebildet. Weist die Funktion $u(\mathbf{x},\mathbf{y})$ bezüglich $\mathbf{y}$ hinreichend Glattheit auf, so kann man zeigen, dass die Quasi-Monte-Carlo-Quadratur mit einer deterministischen Rate $\mathcal{O}(N^{-1+\delta})$ für alle $\delta>0$ konvergiert.

Die einfachste Konstruktion einer Quasi-Monte-Carlo-Quadraturformel erfolgt vermittels der *Halton-Folge*. Diese generiert Punkte in $[0,1]^M$. Hierzu werden kompenentenweise *Van-der-Corput-Folgen* mit verschiedenen Primzahlbasen kombiniert. Sei $p\in\mathbb{N}$ eine Primzahl. Das i-te Folgeglied der Van-der-Corput-Folge zur Basis p berechnet sich aus der p-adischen Darstellung des Index

$$i = \sum_{k=0}^{j} d_k p^k \quad \text{mit} \quad d_k\in\{0,\dots,p-1\} \text{ für alle } k.$$

Der zugehörige Quadraturpunkt ξ_i ergibt sich dann als *Radikal-Inverse* (also Spiegelung am Nullpunkt) der Zahl i:

$$\xi_i = \sum_{k=0}^{j} d_k p^{-k-1}.$$

Die Folgeglieder lassen sich nun sukzessive gemäss des folgenden Algorithmus berechnen:

Input : Folgeglied ξ_{i-1}, Primzahlbasis p
Output : Folgeglied ξ_i
$z = 1 - \xi_{i-1}$
$v = 1/p$
while $z < v + \text{tol}$ **do**
  $v = v/p$
end
$\xi_i = \xi_{i-1} + (p+1) * v - 1$

Hierbei ist tol eine vorgegebene Toleranz für den Punktevergleich, beispielsweise $\text{tol} = 10^{-9}$. Die ersten Folgeglieder der Van-der-Corput-Folge zur Basis 2 lauten nun $\frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{5}{8}, \frac{3}{8}, \frac{7}{8}, \ldots$. Üblicherweise überspringt man am Anfang einige Folgeglieder, etwa die ersten 100, um die Verteilungseigenschaften zu verbessern. Bei der Implementierung können die einzelnen Folgeglieder jeweils überschrieben werden.

Aufgabe 1. Implementieren Sie die vektorwertige Funktion

`function Y = Halton_vector(Y),`

die zu einem gegebenen Punkt $\boldsymbol{\xi}_{i-1}$ der Halton-Folge bezüglich der ersten M Primzahlen den nächsten Punkt $\boldsymbol{\xi}_i$ der Halton-Folge berechnet.

Aufgabe 2. Approximieren Sie den Erwartungswert des Poisson-Problems vermittels $N = 10^4$ Samples jeweils mit der Monte-Carlo-Quadratur und der Quasi-Monte-Carlo-Quadratur. Hierbei seien $\psi_i(\mathbf{x}) = 2^{-i} \sin(i\pi x_1) \sin(i\pi x_2)$ und die rechte Seite $f(\mathbf{x}) = 2\pi^2 \sin(\pi x_1) \sin(\pi x_2)$ vorgegeben. Die Dimension des Parameterbereichs sei $M = 5$. Das Grobgitter auf dem Einheitsquadrat sollte fünfmal verfeinert werden (Level 5). Verwenden Sie die Lösung der Quasi-Monte-Carlo-Approximation als Referenz, um damit die Konvergenz der Monte-Carlo-Approximation zu untersuchen. Bestimmen Sie hierzu den relativen L^2-Fehler der Monte-Carlo-Lösungen mit $10, 100, 1000, 10000$ Samples und tragen diesen in einem `loglog`-Plot gegen die Anzahl der Samples auf.

Hinweis. Für die Lösung der Programmieraufgabe muss die FEM-Steifigkeitsmatrix sehr oft mit unterschiedlichen Koeffizienten aufgestellt werden. Daher ist hier eine effiziente Implementierung notwendig. Dabei geht man am besten wie folgt vor:

1. Bestimme simultan alle lokalen Elementmatrizen und speichere sie im Vektor **s**.
2. Bestimme den Vektor **i** aller zu den Einträgen in **s** gehörenden Zeilenindizes.
3. Bestimme den Vektor **j** aller zu den Einträgen in **s** gehörenden Spaltenindizes.
4. Setze `S=sparse(i,j,s)`. Hierbei wird verwendet, dass die `sparse`-Funktion bei in **i**, **j** doppelt vorkommende Einträge die zugehörigen Einträge aus **s** addiert.
5. Verwende in den Schritten 1.–4. keinesfalls Schleifen.