UNIVERSITÄT BASEL

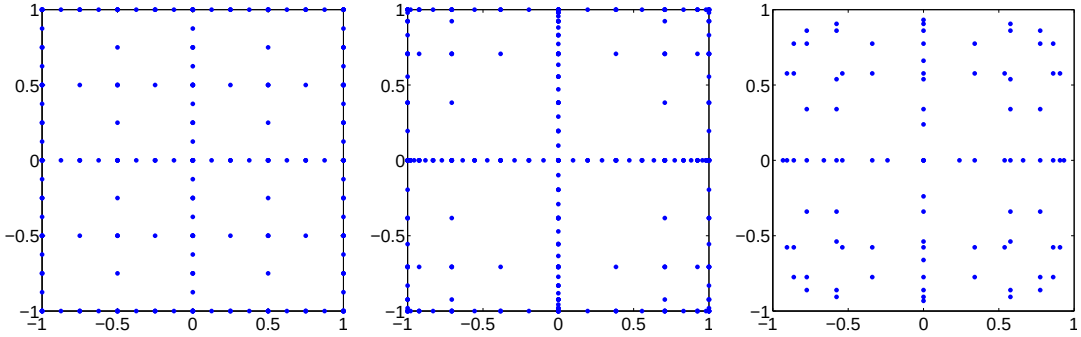
Numerik stochastischer Differentialgleichungen

Frühjahrsemester 2014

Prof. Dr. H. Harbrecht, M. Peters, M. Siebenmorgen

Projekt

Abgabe bis **15.8.2014**



Problemstellung

In diesem Projekt betrachten wir das elliptische, stochastische Diffusionsproblem

$$\left.\begin{aligned} -\operatorname{div}\big(\alpha(\mathbf{x},\mathbf{y})\nabla u(\mathbf{x},\mathbf{y})\big) &= f(\mathbf{x}) \quad \mathbf{x}\in D \\ u(\mathbf{x},\mathbf{y}) &= 0 \qquad \mathbf{x}\in\Gamma := \partial D \end{aligned}\right\} \quad \mathbf{y}\in\square \tag{1}$$

mit variablem *Diffusionskoeffizienten* $a(\cdot,\mathbf{y}) \in C(\overline{D})$ für fast alle $\mathbf{y} = [y_1,\dots,y_M]^\intercal \in \square := [-1,1]^M$. Für die Diskretisierung des Diffusionskoeffizienten nehmen wir an, dass dieser als orthogonale Entwicklung gemäss

$$\alpha(\mathbf{x},\mathbf{y}) = \overline{\alpha}(\mathbf{x}) + \sum_{k=1}^{M} \psi_k(\mathbf{x}) y_k$$

gegeben ist. Die Diskretisierung bezüglich der Ortsvariable $\mathbf{x} \in D$ soll mit linearen Finiten Elementen erfolgen.

Ist der Parameterbereich $\square$ mit dem M-dimensionalen (normierten) Lebesgue-Mass λ^M versehen, d.h. $2^{-M}\lambda^M(\square) = 1$, lassen sich der Erwartungswert und die Varianz der Lösung u von (1) als hochdimensionale Integrale schreiben:

$$\mathbb{E}[u](\mathbf{x}) = 2^{-M}\int_{\square} u(\mathbf{x},\mathbf{y})\,\mathrm{d}\mathbf{x} \quad \text{bzw.} \quad \mathbb{V}[u](\mathbf{x}) = 2^{-M}\int_{\square} u^2(\mathbf{x},\mathbf{y})\,\mathrm{d}\mathbf{x} - \big(\mathbb{E}[u](\mathbf{x})\big)^2.$$

Ziel dieses Projekts ist es, den Erwartungswert und die Varianz mit Hilfe der *Dünngitter-Quadratur* zu approximieren. Wir werden hierbei die zusammengesetzte Trapezregel, die Gauss-Legendre-Quadratur und die Clenshaw-Curtis-Quadratur zugrundelegen. Die zusammengesetzte Trapezregel unterscheidet sich von den anderen beiden Quadraturformeln dadurch, dass sie eine höhere Genauigkeit durch Verfeinerung erzielt (*h-Konvergenz*), während die anderen beiden Formeln eine höhere Genauigkeit durch Erhöhung des Polynomgrades erzielen (*p-Konvergenz*).

Eindimensionale Quadraturformeln

Zusammengesetzte Trapezregel

Sei $n = 2^{\ell-1} + 1$ und $h = 2^{-\ell+2}$ für ein $\ell \in \mathbb{N}_{>1}$. Wir setzen $\xi_k = (k-1)h - 1$ für $k = 1, \dots, n$ und definieren die zugehörigen Gewichte

$$w_k = \begin{cases} h, & \text{falls } 1 < k < n, \\ h/2, & \text{sonst.} \end{cases}$$

Für $\ell = 1$ setzen wir immer die Mittelpunktsregel $\xi_1 = 0$ und $w_1 = 2$.

Gauß-Legendre-Quadratur

Die Stützstellen ξ_k und Gewichte w_k zur Gauß-Quadraturformel auf dem Intervall I ergeben sich aus der *Dreitermrekursion*

$$\beta_{n+1} u_{n+1}(x) = (x - \alpha_n) u_n(x) - \beta_n u_{n-1}(x)$$

der zugehörigen Orthonormalpolynome. Denn sind $\{(\lambda_k, \mathbf{v}_k)\}_{k=1}^n$ die Eigenpaare der Jacobi-Matrix

$$\mathbf{J}_n = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{n-1} \\ & & \beta_{n-1} & \alpha_n \end{bmatrix},$$

so gilt

$$\xi_k = \lambda_k \quad \text{und} \quad w_k = v_{k,1} \int_I \rho(x)\,\mathrm{d}x.$$

Konkret ergibt sich für die *orthonormierten* Legendre-Polynome auf $I = [-1, 1]$ die Dreitermrekursion

$$u_{-1}(x) = 0, \quad u_0(x) = \frac{1}{\sqrt{2}}, \quad \frac{(n+1)u_{n+1}(x)}{\sqrt{4(n+1)^2 - 1}} = x u_n(x) - \frac{n u_{n-1}(x)}{\sqrt{4n^2 - 1}}, \quad n \geq 0.$$

Insbesondere sind in diesem Fall die Gewichte gegeben durch $w_k = 2v_{k,1}$. Bekanntermassen ist die Exaktheit der Gauss-Legendre-Quadratur mit n Punkten vom Grad $2n - 1$.

Clenshaw-Curtis-Quadratur

Die Stützstellen der Clenshaw-Curtis-Quadraturformeln sind gegeben durch die Extrema der Tschebyscheff-Polynome

$$T_{n-1}(x) = \cos\big((n-1)\arccos(x)\big)$$

zuzüglich der Randpunkte, das heisst

$$\xi_k = \cos\left(\frac{(k-1)\pi}{n-1}\right) \quad \text{für } k = 1, \dots, n.$$

Die Bestimmung der zugehörigen Gewichte ist etwas aufwendiger als bei der Gauß-Legendre-Quadratur. Für die Gewichte der inneren Stützstellen gilt mit $\theta_k = k\pi/(n-1)$ die Beziehung

$$w_k = \frac{1}{1 - \xi_k^2} \frac{2\sin(\theta_{k-1})}{n-1} \sum_{s=0}^{n-3} \sin\left(\frac{(s+1)(k-1)\pi}{n-1}\right) \lambda_s \quad \text{für } k = 2, 3, \dots, n-1.$$

Ferner ist das Gewicht der ersten Stützstelle gegeben durch

$$w_1 = \frac{1}{2(n-1)}\left(\gamma_0 + 2\sum_{s=1}^{n-2}\gamma_s + \gamma_{n-1}\right)$$

und das der letzten durch

$$w_n = \frac{1}{2(n-1)}\left(2\sum_{s=0}^{n-1}(-1)^j\gamma_s - \gamma_0 + (-1)^n\gamma_{n-1}\right).$$

Hierbei gilt

$$\gamma_s = \frac{1+\cos(\pi s)}{1-s^2} \quad \text{und } \lambda_s = \frac{2\cos(\pi s)+2}{-s^3-3s^2+s+3}.$$

Für eine Herleitung der Formeln sei auf [3] verwiesen. Die Formeln können in MATLAB effizient mit der *Diskreten Sinustransformation* `dst` implementiert werden.

Der Vorteil der Clenshaw-Curtis-Quadratur ist, dass die Quadraturpunkte geschachtelt sind, was in hohen Dimensionen M zu wesentlich weniger Quadraturpunkten führt. Dies geht allerdings zu Lasten der polynomialen Exaktheit. So ist die Clenshaw-Curtis-Quadratur mit n Punkten nur exakt vom Grad $n-1$.

Dünngitterquadratur

Im Folgenden bezeichne Q_ℓ den Quadraturoperator einer beliebigen Quadraturformel mit n_ℓ Punkten. Dies bedeutet

$$Q_\ell \colon C([-1,1]) \to \mathbb{R}, \quad Q_\ell f = \sum_{k=1}^{n_\ell} w_k f(\xi_k). \tag{2}$$

Dabei gilt für die zusammengesetzte Trapezregel und für die Clenshaw-Curtis-Quadraturformel $n_\ell = 2^{\ell-1}+1$ für $\ell \geq 2$ und $n_\ell = 1$ für $\ell = 1$. Für die Gauß-Legendre-Quadratur gilt hingegen stets $n_\ell = \ell$.

Via Tensorproduktbildung lässt sich nun einfach eine Quadraturformel für Funktionen in $C(\square)$ konstruieren gemäss

$$\mathbf{Q}_\ell \colon C(\square) \to \mathbb{R}, \quad \mathbf{Q}_\ell f = (Q_\ell \otimes \cdots \otimes Q_\ell) f = \sum_{k_1=1}^{n_\ell} \cdots \sum_{k_M=1}^{n_\ell} w_{k_1} \cdots w_{k_M} f(\xi_{k_1}, \ldots, \xi_{k_M}).$$

Führen wir nun die *Differenz-Quadraturformeln*

$$\Delta_\ell f := (Q_\ell - Q_{\ell-1}) f \quad \text{mit } Q_0 = 0$$

ein, so lässt sich die Tensorprodukt-Quadraturformel schreiben als

$$\mathbf{Q}_\ell f = \sum_{\|\mathbf{k}\|_\infty \leq \ell} (\Delta_{k_1} \otimes \cdots \otimes \Delta_{k_M}) f$$

mit dem Multiindex $\mathbf{k} \in \mathbb{N}_{\geq 1}^M$. Die zugehörige *Dünngitter-Quadraturformel* ergibt sich nun durch Ausdünnung der Tensorproduktformel:

$$\mathbf{Q}_\ell^{\mathrm{SG}} f = \sum_{\|\mathbf{k}\|_1 \leq \ell+M-1} (\Delta_{k_1} \otimes \cdots \otimes \Delta_{k_M}) f. \tag{3}$$

Implementierung

Für die Implementierung der Dünngitter-Quadraturformel (3) wird eine sinnvolle Numerierung der Multiindizes benötigt. Hierzu sortieren wir die Formel (3) gemäss

$$\mathbf{Q}_\ell^{\mathrm{SG}} f = \sum_{s=1}^{\ell} \sum_{\|\mathbf{k}\|_1 = s+M-1} (\Delta_{k_1} \otimes \cdots \otimes \Delta_{k_M}) f.$$

Eine Auflistung aller Multiindizes mit $\|\mathbf{k}\|_1 = s + M - 1$ liefert nun der *Tröpchenalgorithmus*, siehe [2]:

Input : Dimension M und Level s
Output : Matrix $\mathbf{K}$ mit allen zulässigen Indizes $\mathbf{k} = [k_1, \ldots, k_M]$
setze $\mathbf{k} = [1, \ldots, 1]$;
setze $j = 1$;
while $k_M \leq s$ **do**
  if $\|\mathbf{k}\|_1 > s + M - 1$ **then**
    setze $k_j = 1$;
    setze $j = j + 1$;
  else
    speichere $\mathbf{k}$ in $\mathbf{K}$;
    setze $j = 1$;
  end
  setze $k_j = k_j + 1$;
end

Die effiziente Implementierung der Dünngitterquadratur hängt massgeblich von der Anzahl der benötigten Funktionsauswertungen ab. Im Falle der nicht geschachtelten Gauss-Legendre-Quadratur kann die *Kombinationstechnik* verwendet werden, um die Quadraturformel (3) auszuwerten. Hierzu werden nur die eindimensionalen Quadraturformeln (2) benötigt, die entsprechend der folgenden Formel kombiniert werden:

$$\mathbf{Q}_\ell^{\mathrm{SG}} f = \sum_{\ell \leq \|\mathbf{k}\|_1 \leq \ell+M-1} (-1)^{\ell+M-\|\mathbf{k}\|_1-1} \binom{M-1}{\|\mathbf{k}\|_1 - 1} (Q_{k_1} \otimes \cdots \otimes Q_{k_M}) f. \tag{4}$$

Bei dieser Formel wird nur der Funktionswert im Nullpunkt mehrfach berechnet. Für die geschachtelten Quadraturformeln ist die Auswertung etwas komplizierter. In diesem Fall gilt, vgl. [1],

$$\mathbf{Q}_\ell^{\mathrm{SG}} f = \sum_{\|\mathbf{k}\|_1 \leq \ell+M-1} \sum_{j_1 = n_{k_1} - m_{k_1} + 1}^{n_{k_1}} \cdots \sum_{j_M = n_{k_M} - m_{k_M} + 1}^{n_{k_M}} w_{\mathbf{k},\mathbf{j}} f(\boldsymbol{\xi}_{\mathbf{k},\mathbf{j}}). \tag{5}$$

Die Zahlen m_ℓ entsprechen hierbei der Anzahl der neuen Quadraturpunkte auf Level ℓ, das heisst $m_\ell = n_\ell - n_{\ell-1}$, wobei wir von einer hierarchischen Numerierung der Quadraturpunkte ausgehen. Ferner ist $\boldsymbol{\xi}_{k,j}$ der Punkt mit Index j aus der Quadraturformel auf Level k. Die zugehörigen Quadraturgewichte lauten dann

$$w_{\mathbf{k},\mathbf{j}} = \sum_{\|\mathbf{k}+\mathbf{q}\|_1 \leq \ell+2M-1} v_{k_1+q_1, j_1} \cdots v_{k_M+q_M, j_M} \tag{6}$$

mit $\mathbf{q} \in \mathbb{N}_{\geq 1}^M$ und

$$v_{(k+q),j} := \begin{cases} w_{k,j}, & \text{falls } q = 1, \\ w_{k+q-1,j} - w_{k+q-2,j}, & \text{sonst.} \end{cases}$$

Aufgabe 1. Implementieren Sie die eindimensionalen Quadraturformeln. Schreiben Sie hierzu die MATLAB-Funktionen

```
function [Q] = Trapeziodal(l),
function [Q] = GaussLegendre(l),
function [Q] = ClenshawCurtis(l),
```

die jeweils alle oben eingeführten Quadraturformeln auf allen Leveln $1, \ldots, \ell$ aufstellen und in dem `cell`-Array **Q** speichern.

Aufgabe 2. Schreiben Sie eine MATLAB-Funktion

```
function [K,flag] = droplet(dim,lmax),
```

die den Tröpfchenalgorithmus numerisch umsetzt. Hierbei ist es hinsichtlich der Kombinationstechnik sinnvoll, direkt die Variable `flag` mit auszugeben, die angibt, ob $(-1)^{\ell+M-\|\mathbf{k}\|_1-1} = 1$ gilt, vgl. (4). Weiterhin benötigen wir noch die Funktion

```
function [Kq] = q_droplet(dim,lmax,k,K),
```

die den Multiindex **q** zur Berechnung der Gewichte gemäss (6) bestimmt. Wie man sich überlegt, ergeben sich für **q** dieselben Möglichkeiten wie für **k** (setze $\mathbf{k} = \mathbf{1}$). Die zulässigen **q** erfüllen somit genau $\|\mathbf{k}_i + \mathbf{k}\|_1 \leq \ell + 2M - 1$.

Aufgabe 3. Implementieren Sie schliesslich die Funktionen

```
function [val] = CombinationQuadrature (f,l,Q),
function [val] = SparseGridQuadrature (f,l,Q),
```

welche die Dünngitterquadratur basierend auf der Kombinationstechnik (4) beziehungsweise die Dünngitterquadratur für geschachtelte Punkte aus Formel (5) numerisch umsetzen. Die Funktionen erhalten als Argumente einen `function handle` f, ein Level ℓ und die gewünschten eindimensionalen Quadraturformeln **Q** bis zum Level ℓ.

Beispiele

Testen Sie Ihre Implementierung anhand der folgenden Beispiele.

Beispiel 1. Gegeben sei das Integral

$$\int_{[0,1]^M} (1+1/M)^M \prod_{k=1}^{M} x_k^{1/M} \, \mathrm{d}\mathbf{x},$$

dessen exakte Wert 1 ist. Transformieren Sie das Integral auf das Gebiet $\square$ und approximieren Sie den Wert mit Hilfe der implementierten Dünngitterquadratur für $M = 5$ und $\ell = 1, 2, \ldots, 7$. Als Referenz sollen dabei die numerischen Resultate aus [1] herangezogen werden. Ein Mass für den Aufwand ist hierbei die Anzahl der benötigten Funktionsauswertungen. Plotten Sie den Fehler der einzelnen Quadraturformeln in einen gemeinsamen `loglog`-Plot gegen die Anzahl der benötigten Funktionsauswertungen.

Aufgabe 2. Es sei $D = [0,1]^2$. Approximieren Sie den Erwartungswert und die Varianz des Poisson-Problems (1) durch $N = 10^5$ Samples mit der Quasi-Monte-Carlo-Quadratur. Hierbei seien $\psi_k(\mathbf{x}) = 2^{-k} \sin(\pi k x_1) \sin(\pi k x_2)$ und die rechte Seite $f(\mathbf{x}) = 1$ vorgegeben. Ferner setzen wir $\overline{\alpha}(\mathbf{x}) \equiv 1$. Die stochastische Dimension des Parameterbereichs sei $M = 8$. Das räumliche Grobgitter auf dem Einheitsquadrat soll 6 mal verfeinert werden (Level 6). Verwenden Sie die Lösung der Quasi-Monte-Carlo-Approximation als

Referenz (evtl. in einer Datei abspeichern). Bestimmen Sie nun approximativ den Erwartungswert und die Varianz der Lösung von (1) mit der implementierten Dünngitterquadratur. Die benötigten Funktionsauswertungen entsprechen dabei jeweils der Lösung einer partiellen Differentialgleichung. Visualisieren Sie die L^2-Fehler des Erwartungswerts und der Varianz gegen die Anzahl der gelösten Differentialgleichungen bezüglich der Quasi-Monte-Carlo-Lösung in einen `loglog`-Plot für $\ell = 1, \ldots, 7$.

Literatur

[1] Thomas Gerstner and Michael Griebel. Numerical integration using sparse grids. *Numer. Algorithms*, 18(3–4):209–232, 1998.

[2] Astrid Meyer (geb. Fischer). *Diplomarbeit: Interpolation von vektorwertigen Funktionen mit adaptiven dünnen Gittern.* Institut für Numerische Simulation, Universität Bonn, 2007.

[3] Alvise Sommariva. Fast construction of Fejér and Clenshaw-Curtis rules for general weight functions. *Comput. Math. Appl.*, 65(4):682–693, 2013.