Programmierblatt 1.

Besprechungswoche: 01.11. – 05.11.2021

Gegeben sei ein Gaußsches Zufallsfeld[1] $a(x,\omega)$ mit $x \in D$ und $\omega \in \Omega$. Zu dessen effizienten numerischen Simulation trennt man die physikalische Variable x von der stochastischen Variable ω mit Hilfe der Karhunen-Loève-Entwicklung[2] von $a(x,\omega)$. Diese liefert die Darstellung

$$a(x,\omega) = \mathbb{E}_a(x) + \sum_{k=1}^{\infty} \sqrt{\lambda_k}\phi_k(x)\psi_k(\omega). \tag{1}$$

Hierbei ist $\mathbb{E}_a(x)$ das erwartete Zufallsfeld, $\{(\lambda_k,\phi_k)\}_{k=1}^{\infty}$ sind die Eigenpaare des Kovarianzoperators

$$(\mathcal{C}u)(x) = \int_D C(x,y)u(y)dy \tag{2}$$

und $\{\psi_k(\omega)\}_{k=1}^{\infty}$ sind i.i.d. normalverteilte Zufallsvariablen. Die Funktion $C : D\times D \longrightarrow \mathbb{R}$ wird als der Kovarianzkern bezeichnet. Im Folgenden wählen wir der Einfachheit halber $D = [0,1]$.

Ziel dieser Programmieraufgabe ist die numerische Approximation der Karhunen-Loève-Entwicklung. Dazu muss man die Summe in (1) geeignet abschneiden und die Eigenpaare des Kovarianzoperators berechnen. Dies machen wir mit dem sogenannten Nyström-Verfahren, bei welchem die Integration in (2) durch eine Quadraturformel angenähert wird. Speziell für die Mittelpunktregel gilt

$$\int_D C(x,y)u(y)dy \approx \frac{1}{n}\sum_{i=1}^{n} C(x,y_i)u(y_i), \quad y_i = \frac{i-1}{n} + \frac{1}{2n}.$$

Wir lösen dann das *kontinuierliche* Eigenwertproblem

$$\lambda\phi(x) = \int_D C(x,y)\phi(y)dy$$

in allen Punkten $x_i = \frac{i-1}{n} + \frac{1}{2n}$ und erhalten das *diskrete* Eigenwertproblem

$$\lambda^{(n)}\boldsymbol{\phi}^{(n)} = \mathbf{C}^{(n)}\boldsymbol{\phi}^{(n)}, \quad \boldsymbol{\phi} = [\phi(x_i)]_i^{(n)} \in \mathbb{R}^n, \quad \mathbf{C}^{(n)} = \frac{1}{n}\left[C(x_i,y_j)\right]_{i,j}. \tag{3}$$

Im Folgenden betrachten wir das diskrete Eigenwertproblem (3) für verschiedene Kovarianzkerne $C(x,y)$.

Aufgabe 1 (Exponentieller Kern).

In dieser Aufgabe betrachten wir den exponentiellen Kern und den zugehörigen Operator:

$$C_{\exp}(x,y) = \sigma\exp(-\sigma|x-y|), \quad (\mathcal{C}_{\exp}u)(x) = \int_{[0,1]} \exp(-|x-y|)u(y)\mathrm{d}y. \tag{4}$$

Dabei sei die Korrelationslänge $\sigma = 1$ gewählt. Die Matrix $\mathbf{C}_{\exp}^{(n)}$ sei durch (3) und den Kern $C_{\exp}$ definiert.

[1] Ein Gaußes Zufallsfeld liegt vor, wenn $a(x,\omega)$ für jedes $x \in D$ eine normalverteilte Zufallsgösse ist.

[2] Diese wird oft auch als POD (proper orthogonal decomposition), PCA (principle component analysis) oder SVD (singular value decomposition) bezeichnet

(a) Implementieren Sie eine Matlab-Funktion

```
C = expKernel(a,b,n),
```

welche für zwei Indexmengen `a` $\subset \{1,\dots,n\}$, `b` $\subset \{1,\dots,n\}$, die Einträge (i,j) der Matrix $\mathbf{C}_{\text{exp}}^{(n)}$ berechnet, wobei $(i,j) \in$ `a` $\times$ `b` gilt.

(b) Nutzen Sie die Matlab-Funktion `eig` um die Eigenpaare der Matrix $\mathbf{C}_{\text{exp}}^{(n)}$ auszurechnen. Setzen Sie dabei $n = 2^8$.

Stellen Sie die ersten drei dominanten Eigenfunktionen, d.h. Eigenfunktionen zu den 3 betragsgrössten Eigenwerten, graphisch dar. Nutzen Sie dazu den Matlab-Befehl `plot`. Beachten Sie, dass die Funktion `eig` die berechneten Eigenwerte nicht sortiert.

(c) Die Eigenpaare zu dem Operator $\mathcal{C}_{\text{exp}}$ haben die Form

$$\lambda_k = \frac{2}{\gamma_k^2 + 1}, \quad \phi_k(x) = \frac{1}{\beta_k}(\sin(\gamma_k x) + \gamma_k \cos(\gamma_k x)).$$

Dabei sind γ_k die positiven Lösungen der Gleichung $(\gamma^2 - 1)\tan(\gamma) - 2\gamma = 0$. Die β_k werden so gewählt, dass $\|\phi_k\|_{L^2(D)} = 1$ gilt. Die Werte γ_k, β_k und λ_k für die ersten drei Eigenpaare sind in der Tabelle 1 angegeben.

k	γ_k	β_k	λ_k
1	1.30654237418881	1.53412075397455	0.738810809416452
2	3.67319440630425	2.87161602120694	0.138003775354263
3	6.58462004256417	4.81441694834056	0.045088487289781

Tabelle 1: Konstanten γ_k, β_k und Eigenwerte λ_k zu $\mathcal{C}_{\text{exp}}$.

Wir bezeichnen mit $\tilde{\phi}_k^{(n)}(x)$ die durch $\boldsymbol{\phi}^{(n)}$ mit $\|\boldsymbol{\phi}^{(n)}\|_2 = 1$ definierte Näherung an $\phi_k(x)$, wobei wir zwischen den Punkten x_i linear interpolieren. Weiter sei $\phi_k^{(n)}(x) = \tilde{\phi}_k^{(n)}(x)/\|\tilde{\phi}_k^{(n)}\|_{L^2(D)} = \sqrt{n}\tilde{\phi}_k^{(n)}(x)$.

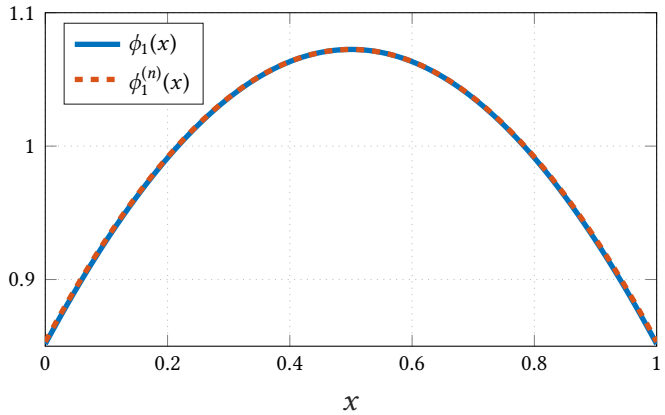

Abbildung 1: Funktionen $\phi_1(x)$ und $\phi_1^{(n)}(x)$ für $n = 2^8$.

Stellen Sie für $k = 1, 2, 3$ die Funktionen $\phi_k(x)$ sowie die durch die Funktion `eig` berechneten Näherungen graphisch dar (vgl. Abbildung 1). Wählen Sie $n = 2^8$.

(d) Untersuchen Sie die Konvergenz der ersten drei Eigenpaare in Abhängigkeit von n. Betrachten Sie dazu $n = 2^\ell$, $\ell = 6, 7, \dots, 11$.

Stellen Sie in einem Plot $\log_2(|\lambda_k - \lambda_k^{(n)}|)$, und in einem weiteren Plot $\log_2(\mathrm{err}_k^{(n)})$ in Abhängigkeit von n dar (vgl. Abbildung 2). Dabei ist $\mathrm{err}_k^{(n)}$ eine Schätzung des $\|\cdot\|_{L^2(D)}$-Fehlers zwischen $\phi_k(x)$ und $\phi_k^{(n)}(x)$

$$\mathrm{err}_k^{(n)} = \frac{1}{\sqrt{n}} \sqrt{\sum_{i=1}^{n} (\phi_k(x_i) - \phi_k^{(n)}(x_i))^2}.$$

Diskutieren Sie die Konvergenzraten für die Eigenwerte und die Eigenvektoren.

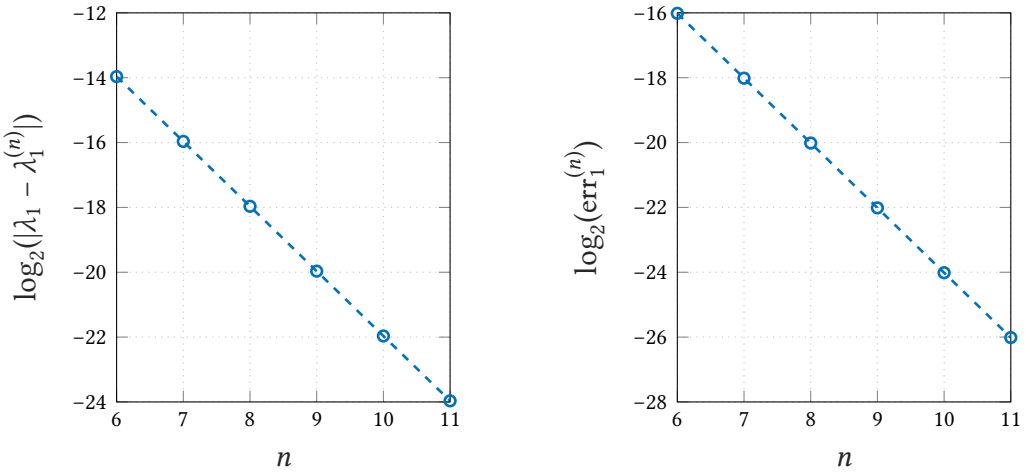

Abbildung 2: Konvergenz des ersten Eigenpaars für den exponentiellen Kern.

Aufgabe 2 (Niedrigrangapproximation).

Zur Lösung von dem Eigenwertproblem $\lambda \mathbf{u} = \mathbf{C}\mathbf{u}$ aus (3) wollen wir die *pivotisierte Cholesky-Zerlegung* $\mathbf{C} \approx \mathbf{L}\mathbf{L}^\mathsf{T}$ verwenden, welche gemäss dem folgenden Algorithmus bestimmt wird:

Algorithm pivotisierte Cholesky-Zerlegung

Input: $\mathbf{A} = \left[a_{i,j}\right]_{i,j=1}^{n} \in \mathbb{R}^{n\times n}$, Genauigkeit tol ≥ 0.

Output: $\mathbf{L} \in \mathbb{R}^{n\times k}$ mit $\mathbf{L}\mathbf{L}^\mathsf{T} \approx \mathbf{A}$, Permutationsvektor $\mathbf{p}$.

setze $\mathbf{p} := \left[1, 2, \ldots, n\right]$, $\mathbf{d} := \left[a_{1,1}, a_{2,2}, \ldots, a_{n,n}\right]^\mathsf{T}$, trace $:= \|\mathbf{A}\|_{\mathrm{tr}}$, $k := 1$
while $k \leq n$ & trace $\geq$ tol **do**
  setze pivot $:= \mathrm{argmax}_{\ell\in\{k,k+1,\ldots,n\}} \mathbf{d}_{\mathbf{p}_\ell}$ (finde Pivotelement)
  vertausche $\mathbf{p}_{pivot} \leftrightarrow \mathbf{p}_k$
  setze $\mathbf{L}_{\mathbf{p}_k,k} := \sqrt{\mathbf{d}_{\mathbf{p}_k}}$ (update $\mathbf{L}$)
  setze $\mathbf{L}_{\mathbf{p}_{k+1:n},k} := \mathbf{A}_{\mathbf{p}_{k+1:n},\mathbf{p}_k} / \mathbf{L}_{\mathbf{p}_k,k}$
  update $\mathbf{L}_{\mathbf{p}_{k+1:n},k} := \mathbf{L}_{\mathbf{p}_{k+1:n},k} - \mathbf{L}_{\mathbf{p}_{k+1:n},1:k-1} \cdot \mathbf{L}^\mathsf{T}_{\mathbf{p}_k,1:k-1} / \mathbf{L}_{\mathbf{p}_k,k}$
  update $\mathbf{d}_{\mathbf{p}_{k:n}} := \mathbf{d}_{\mathbf{p}_{k:n}} - \left[\mathbf{L}^2_{\mathbf{p}_k,k}, \mathbf{L}^2_{\mathbf{p}_{k+1},k}, \ldots, \mathbf{L}^2_{\mathbf{p}_n,k}\right]^\mathsf{T}$ (update $\mathbf{d}$)
  update trace $:= \|\mathbf{d}_{\mathbf{p}_{k:n}}\|_1$ (berechne $\|\mathbf{A} - \mathbf{L}^{(k)}(\mathbf{L}^{(k)})^\mathsf{T}\|_{\mathrm{tr}}$)
  setze $k := k + 1$
end while

Beachten Sie, dass die Matrix $\mathbf{A}$ für den Algorithmus nicht explizit bekannt sein muss. Es genügt, eine Routine bereitzustellen, die jeweils die aktuell benötigten Matrixeinträge berechnet. Insbesondere bei grossen Matrizen ($n > 10^6$) reduziert sich dadurch der Speicheraufwand erheblich.

(a) Implementieren Sie die Matlab-Funktion

`[L, p] = choleskyLowrank(c,n,tol),`

welche die pivotisierte Cholesky-Zerlegung umsetzt. Der Parameter `c` sei hierbei ein `function handle`, der für ein Indexpaar (i, j) den zugehörigen Matrixeintrag zurückgibt. Insbesondere soll die Matrix $\mathbf{A}$ in dem Algorithmus nie explizit aufgestellt werden. Der Parameter n entspricht der Dimension der Matrix $\mathbf{A}$ und `tol` gibt die Abbruchsgenauigkeit vor. Es sollen abgesehen von der `while`-Schleife keine weiteren Schleifen verwenden werden.

(b) Testen Sie Ihre Implementierung. Gehen Sie dazu wie folgt vor:

```
[X, Y] = meshgrid(linspace(0, 1, 4001));
A = exp(-(X - Y).^2);                          (erzeuge Testmatrix)
[L, p] = choleskyLowrank(A, 4001, 1e-9);
B = L*L.'; norm(A(:) - B(:))                   (berechne ‖A − L^(k)(L^(k))^T‖_F)
```

Die berechnete Norm soll dabei von der Grössenordnung 10^{-10} sein.

(c) Betrachten Sie den Gaußschen Kovarianzkern

$$C_{\text{Gauß}}(x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-|x - y|^2}{2\sigma^2}\right)$$

mit $\sigma = 0.05$. Berechnen Sie die Eigenpaare der Matrix $\mathbf{C}^{(n)}_{\text{Gauß}}$, definiert durch (3) und den Kern $C_{\text{Gauß}}$, mit Hilfe der Niedrigrangapproximation. Wählen Sie die Parameter `tol`$=10^{-8}$ und $n = 2^{10}$.

Nutzen Sie aus, dass die Eigenwerte von $\mathbf{C}^{(n)}_{\text{Gauß}} \approx \mathbf{L}\mathbf{L}^\mathsf{T}$ dieselben sind, wie die von $\mathbf{L}^\mathsf{T}\mathbf{L}$. Sind $\{(\lambda_k, \tilde{\mathbf{v}}_k)\}_{k=1}^M$ die Eigenpaare von $\mathbf{L}^\mathsf{T}\mathbf{L}$, so lösen $\{(\lambda_k, \mathbf{L}\tilde{\mathbf{v}}_k)\}_{k=1}^M$ das ursprüngliche Eigenwertproblem. Bestimmen Sie die Eigenwerte und Eigenvektoren $\tilde{\mathbf{v}}_k$ des kleineren Eigenwertproblems für $\mathbf{L}^\mathsf{T}\mathbf{L}$ sowie die Eigenvektoren $\mathbf{v}_k$ von $\mathbf{L}^\mathsf{T}\mathbf{L}$ mit der Matlab-Funktion `eig`.

Stellen Sie für $k = 1, 2, 3$ die Vektoren $\mathbf{v}_k$ und $\mathbf{L}\tilde{\mathbf{v}}_k$ graphisch dar (vgl. Abbildung 1).

(d) Untersuchen Sie für jede der Abbruchsgenauigkeiten `tol` $= 10^{-2}, 10^{-3}, 10^{-4}$ die Konvergenz der ersten drei Eigenwerte in Abhängigkeit von n. Betrachten Sie dazu jeweils $n = 2^\ell$, $\ell = 4, 5, \ldots, 10$.

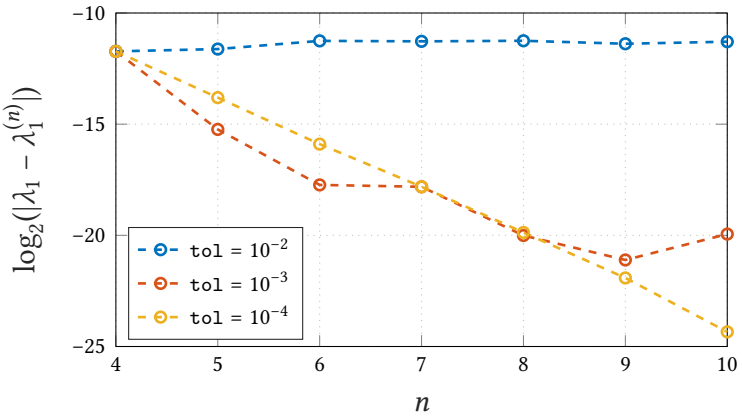

Abbildung 3: Konvergenz von $\lambda_1^{(n)}$ bei verschiedenen Abbruchsgenauigkeiten `tol`.

Nehmen Sie als Referenzen die von der Funktion `eig` berechneten Eigenwerte λ_k der Matrix $\mathbf{C}^{(n)}_{\text{Gauß}}$ für $n = 2^{13}$.

Stellen Sie in einem Plot die Grössen $\log_2(|\lambda_k - \lambda_k^{(n)}|)$ in Abhängigkeit von n für verschiedene Werte von `tol` dar (vgl. Abbildung 3). Diskutieren Sie die Konvergenz der Eigenwerte.

Aufgabe 3 (Rayleigh-Quotient-Iteration).

In dieser Aufgabe betrachten wir den exponentiellen Kern (4) mit $\sigma = 0.001$. Die Matrix $\mathbf{C}_{\text{exp}}^{(n)}$ sei entsprechend zu der Aufgabe 1 definiert. Wir nutzen `tol` $= 10^{-9}$ für die Abbruchsgenauigkeiten der pivotisierten Cholesky-Zerlegung, sowie $n = 2^{12}$ für die Quadratur.

(a) Schreiben Sie eine Funktion

```
[Z, Mu] = RayleighQuotientIter(A,Z0,Mu0),
```

welche die Rayleigh-Quotient-Iteration aus der Vorlesung implementiert. Die Rückgabewerte `Z` und `Mu` sind Näherungen an das erste Eigenpaar. `A` ist eine Matrix. `Z0` und `Mu0` sind die Startnäherungen für den Algorithmus.

(b) Bestimmen Sie mit der Rayleigh-Iteration das dominierende Eigenpaar von $\mathbf{C}_{\text{exp}}^{(n)}$. Gehen Sie dabei wie in der Aufgabe 2 (c) vor. Berechnen Sie eine Niedrigrangapproximation $\mathbf{L}\mathbf{L}^\top$ von $\mathbf{C}_{\text{exp}}^{(n)}$ und nutzen Sie anschliessend die Matrix $\mathbf{L}^\top\mathbf{L}$ für die Rayleigh-Iteration, wobei Sie als Startwerte `Z0` $= e_1$ und `Mu0` $= 1$ wählen.

Vergleichen Sie die Ergebnisse (μ, z) der Rayleigh-Iteration mit dem durch die Funktion `eig` bestimmten dominanten Eigenpaar (λ, v) von $\mathbf{L}^\top\mathbf{L}$. Berechnen Sie dazu $|\mu - \lambda|$ und $\|z - v\|_2$.

(c) Erweitern Sie die Funktion `RayleighQuotientIter` aus Teil (a) so, dass die ersten drei dominanten Eigenpaare der Matrix `A` approximiert werden. In diesem Fall ist `Z` $= [z_1, z_2, z_3]$ eine Matrix der Näherungen an die ersten drei Eigenvektoren. `Mu` $= (\mu_1, \mu_2, \mu_3)$ ist ein Vektor der Approximationen an die Eigenwerte $\lambda_1, \lambda_2, \lambda_3$.

Bestimmen Sie mit der modifizierten Funktion `RayleighQuotientIter` die ersten drei Eigenpaare von $\mathbf{L}^\top\mathbf{L}$. Vergleichen Sie die Ergebnisse zu den durch die Funktion `eig` bestimmten Eigenpaaren. Wählen Sie `Z0` $= [e_1, e_2, e_3]$ und `Mu0` $= (1, 1, 1)$ als Startwerte.

Hinweis. *Nutzen Sie das Gram-Schmidtsche Orthogonalisierungsverfahren.*

(d) Wir bezeichnen mit $\mu_k^{(l)}$ die in der l-ten Iteration des Rayleigh-Verfahrens berechnete Näherung an den Eigenwert λ_k. Stellen Sie für die Matrix $\mathbf{C}_{\text{exp}}^{(n)}$ und $k = 1, 2, 3$, den Fehler $|\lambda_k - \mu_k^{(\ell)}|$ in Abhängigkeit von der Iterationsnummer ℓ graphisch dar (vgl. Abbildung 4).

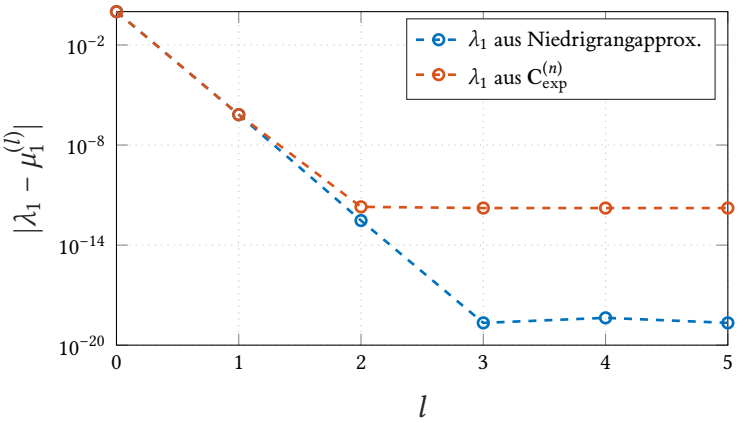

Abbildung 4: Konvergenz von $\mu_1^{(\ell)}$ gegen λ_1 über die Iterationen ℓ des Rayleigh-Verfahrens.

Die Werte μ_k sollen mit Hilfe der Niedrigrangapproximation berechnet werden. Bestimmen Sie die Referenzwerte λ_k durch die Funktion `eig`. Betrachten Sie dazu sowohl die Niedrigrangapproximation als auch die exakte Matrix $\mathbf{C}_{\text{exp}}^{(n)}$.

Diskutieren Sie die Konvergenz.