Prof. Dr. Karsten Urban
Kristina Steih SS 2012
16.05.2012

NumPDE II - Blatt 5

Ziel dieser Übung ist es, den Greedy-Algorithmus 7.1 zu implementieren. Dafür brauchen wir insbesondere einen (effizient berechenbaren) Fehlerschätzer. Wir wissen:

$$\|e_N(\mu)\|_X \le \Delta_N(\mu) := \frac{\|r_N(\cdot;\mu)\|_{X'}}{\alpha_{\mathrm{LB}}(\mu)}$$

mit Residuum $r_N(\cdot;\mu) : X^{\mathcal{N}} \to \mathbb{R}$, $r_N(v;\mu) = f(v;\mu) - a(u_N(\mu), v;\mu)$.

Vorbereitung 1:

(I) *Min-Θ Ansatz für $\alpha_{LB}(\mu)$:*

- Wieso verwendet man $\alpha_{LB}(\mu)$ und nicht $\alpha(\mu)$?
- Betrachten Sie die Funktion `getAlpha_LB` und Zeilen 38-43 im Skript `blatt5`. Vollziehen Sie nach, wie dort Formel (6.7) aus der Vorlesung umgesetzt wird. Wieso werden die Einträge `prob.fem.innProd.refalpha` bzw. `refthetas` angelegt und was steht dort drin?

(II) *Dual-Norm des Residuums:*
Der Riesz-Repräsentant $\hat{r}_N(\mu) \in X^{\mathcal{N}}$ von $r_N(\mu)$ erfüllt nach dem Satz von Riesz-Fréchet die Gleichung

$$(\hat{r}_N(\mu), v)_X = r_N(v;\mu) \qquad \forall\, v \in X^{\mathcal{N}}. \tag{1}$$

- Warum können wir $\|r_N(\cdot;\mu)\|_{X'}$ nicht direkt berechen?
- Zeigen Sie, dass $\|r_N(\cdot;\mu)\|_{X'} = \|\hat{r}_N(\mu)\|_X$ gilt.
- Was für ein Gleichungssystem muss für (1) also aufgestellt werden? Schreiben Sie dieses in Matrix-Vektor-Schreibweise auf. Wie wird insbesondere die rechte Seite berechnet?
- Wie wird dann (in Matrix-Vektor-Schreibweise) $\|r_N(\cdot;\mu)\|_{X'}$ berechnet?
- Vollziehen Sie nach, wie die obigen Schritte in der Funktion `getErrEst_direct` umgesetzt werden.

Aufgabe 1 *(Greedy-Algorithmus, erste ineffiziente Version)*

Implementieren Sie den Greedy-Algorithmus 7.1, um eine reduzierte Basis für das Heizblock-Problem zu konstruieren.

- Verwenden Sie z.B. $n_{\mathrm{train}} = 101$ logarithmisch gleichverteilte Zufallsvariablen für Ξ_{train}.
- Machen Sie sich das Leben leichter, indem Sie noch einen Anfangsparameter $\mu_0^* := \mu_{\min}$ übergeben und im Algorithmus zunächst noch den Initialisierungschritt $S_0^* := \{\mu_0^*\}$, $X_0^* := \mathrm{span}\{u(\mu_0^*)\}$ einfügen.
- Verwenden Sie hier noch den ineffizenten Fehlerschätzer `getErrEst_direct`.
- Orthonormalisieren Sie zur Stabilisierung mit Gram-Schmidt die Snapshots $u(\mu_n^*)$, $1 \le n \le N$, um die Basisfunktionen ξ_n, $1 \le n \le N$, zu erhalten.

Vorbereitung 2:

(III) *Offline-Online-Zerlegung :*
Wir nutzen nun die affine Zerlegung der Formen $f(v;\mu)$ und $a(u,v;\mu)$ aus. Mit einer RB-Basis $\{\xi_1,\ldots,\xi_N\}$ gilt $u_N(\mu)=\sum_{n=1}^N u_{n,N}(\mu)\xi_n$ und damit

$$r_N(v;\mu)=\sum_{q=1}^{Q_f}\theta_f^q(\mu)f^q(v)-\sum_{q=1}^{Q_a}\sum_{n=1}^{N}\theta_a^q(\mu)u_{n,N}(\mu)a^q(\xi_n,v).$$

Jetzt können wir die Riesz-Repräsentanten für die parameter-unabhängigen Formen berechnen, also

$$\begin{aligned}(\hat{f}_q,v)_X&=f^q(v) &&\forall\, v\in X^{\mathcal{N}},\ 1\le q\le Q_f,\\ (\hat{a}_{n,q},v)_X&=-a^q(\xi_n,v) &&\forall\, v\in X^{\mathcal{N}},\ 1\le q\le Q_a, 1\le n\le N.\end{aligned}$$

Damit gilt dann

$$\hat{r}_N(v;\mu)=\sum_{q=1}^{Q_f}\theta_f^q(\mu)\hat{f}_q+\sum_{q=1}^{Q_a}\sum_{n=1}^{N}\theta_a^q(\mu)u_{n,N}(\mu)\,\hat{a}_{n,q}. \tag{2}$$

- Aus was für einem Raum sind die $\hat{f}_q$, $\hat{a}_{n,q}$? Wie werden sie also in Matlab dargestellt?
- Wie kann nun also $\|r_N(\cdot;\mu)\|_{X'}$ in $\mathcal{O}((Q_f+NQ_a)^2)$ Operationen berechnet werden?

  Hinweis: Im Skript sind alle affinen Terme von $\hat{r}_N(\mu)$ als $\theta_r^q(\mu)$, v_r^q zusammengefasst. Machen Sie sich klar, wie der Vektor $\theta_r(\mu)$ und die Matrix $\underline{\underline{G}}$ genau aussehen, wenn man Zerlegung (2) hat.
- Zu welchem Zeitpunkt im Greedy-Algorithmus kann welcher Riesz-Repräsentant in der Zerlegung (2) berechnet werden?

Aufgabe 2 *(Greedy-Algorithmus, effiziente Version)*

Modifizieren Sie nun Ihre Lösung aus Aufgabe 1 und implementieren Sie einen Fehlerschätzer `getErrEst_decomp` auf der Basis von (2), um damit insgesamt einen effizienten Trainings-Algorithmus zu erhalten.

Aufgabe 3* *(Greedy-Algorithmus, Version wie im Skript)*

Modifizieren Sie nun Ihre Lösung aus Aufgabe 2, um wie im Skript eine Version ohne vorgegebenen Initial-Parameter μ_0^* zu erhalten. Im Wesentlichen müssen Sie dazu einen Fehlerschätzer für eine leere Basis ausrechnen können.