Übungen zur Vorlesung

# Simulationswerkzeuge

Dr. S. Lang, D. Popović

Besprechung am 07. Juni 2009 in der Übung

---

## Übung 12 DUNE-pdelab: Stationäre Diffusions-Reaktions-Gleichung

n der letzten Übung haben wir die Poisson-Gleichung $-\triangle u(x) = f$ untersucht. Sie ist eine der einfachsten PDE's zweiter Ordnung und beschreibt Brownsche Bewegungen in einem homogenen, isotropen Medium. Eine Erweiterung der Poisson-Gleichung ist die stationäre Diffusions-Reaktions-Gleichung

$$\begin{aligned} -\triangle u(\mathbf{x}) + \alpha(\mathbf{x})u(\mathbf{x}) &= f(\mathbf{x}) \text{ in } \Omega, \\ u(\mathbf{x}) &= g(\mathbf{x}) \text{ auf } \Gamma_D \subseteq \partial\Omega, \\ -\nabla u(\mathbf{x}) \cdot \mathbf{n} &= j(\mathbf{x}) \text{ auf } \Gamma_N = \partial\Omega \setminus \Gamma_D \end{aligned}$$

mit Randbedingungen wie angegeben und einer skalarwertigen Funktion $\alpha(\mathbf{x})$.

In Ihrem Home-Verzeichnis ist wie gehabt das Verzeichnis `dune-1.2` vorhanden. Sie finden dort das Modul `dune-pdelab` mit dem Unterverzeichnis `dune/pdelab/test` mit der Implementierung des Problems in den Dateien

- `simw_diffusionreaction.cc`: Das Hauptporgramm, das ein Gitter und eine Finite Element-Map erstellt und die Funktion `diffusionreactionop()` aufruft,
- `simw_diffusionreactionfcts.hh`: Implementiert Parameterfunktionen für Quellen/Senken und die Boundaries,
- `simw_diffusionreactionop.hh`: Setzt die Komponenten zur Problemlösung auf (Gitterfunktionenräume, Instanziierung von Operatoren, Lösen der Matrix, Schreiben des `vtk`-files),
- `simw_diffusionreactionsolve.hh`: Enthält eine Klasse `DiffusionReaction`, die die FEM-Diskretisierung realisiert.

Während die letzten beiden Dateien die numerischen Methoden implementieren, interessiert uns besonders die zweite Datei mit den Parameter-Klassen für die Randbedingungen und Quellen. Das Programm können Sie in der Konsole mit

```
$make simw_diffusionreaction
$./simw_diffusionreaction
```

übersetzen und starten. Es generiert Output-Files mit der Endung `vtu`, die Sie wie gehabt mit *Paraview* visualisieren können.

Nachdem wir uns das Beispiel verdeutlich haben, wollen wir mit den Parameter-Klassen experimentieren. Versuchen Sie, Lösungen für folgende Probleme zu finden:

- Kein Neumann-Rand:

$$\begin{aligned} \Omega &:= (0,1)^2, \\ \partial\Omega_N &= \emptyset, \\ g(x,y) &= \begin{cases} 0 & \text{für } \neg\{x = 0 \ \&\& \ y > 0.25 \ \&\& \ y < 0.5\ \}, \\ 1 & \text{sonst}, \end{cases} \\ f(x,y) &\equiv 0, \\ \alpha(\mathbf{x}) &\equiv 10. \end{aligned}$$

Untersuchen Sie anschließend folgendes:

1. Schalten Sie den Reaktionsterm ab, und vergleichen Sie die Lösungen mit und ohne Reaktion.
2. Kompilieren Sie das Beispiel in $3d$, und visualisieren Sie die Lösung.
3. Benutzen Sie andere Finite Elemente Räume, ewta $Pk$ oder $Q2$ in zwei Raumdimensionen.
4. Versuchen Sie, analog zum Term $\alpha(\mathbf{x})$ einen räumlich konstanten Parameter $\beta(\mathbf{x})$ zu implementieren, der den Diffusionsterm skaliert. Betrachten Sie die Lösungen, die Sie für verschiedene Gewichtungen der Parameter erhalten. Wählen Sie zum Beispiel $\alpha : \beta = 10 : 1, 50 : 1, 200 : 1$.