

## Projekt.

zu bearbeiten bis: **Freitag, 31.01.2020**

Muster sind überall in der Natur in verschiedenen Größenordnungen vorzufinden. Sehr reguläre und fast reguläre Muster findet man dabei auch, wenn man sich die Konzentrationen von Stoffen eines Prozesses anschaut, in welchem die Stoffe chemisch reagieren und diffundieren [4]. Ein Modell zur Beschreibung der Konzentrationen der Stoffe in einem spezifischen Prozess, der eine Variante des autokatalytischen Selkovschen Modells der Glykolyse [5] darstellt, ist dabei das Gray-Scott-Modell [1, 2]. Es ist durch das folgende System von Reaktionsdiffusionsgleichungen [3] gegeben:

$$\begin{aligned}\partial_t u &= a \Delta_{\mathbf{x}} u - cuv^2 + e(1 - u), \\ \partial_t v &= b \Delta_{\mathbf{x}} v + dw^2 - (e + f)v.\end{aligned}\tag{1}$$

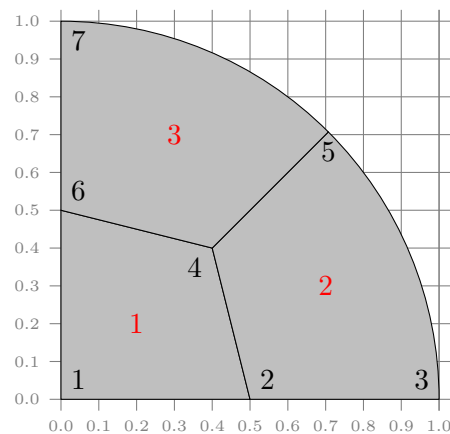
Das Ziel dieses Projekts ist die numerische Lösung dieses Systems auf Flächen und Oberflächen mit bilinearen Finiten-Elementen auf parametrischen Vierecksnetzen.

Als ersten Schritt werden nachfolgend parametrische Vierecksnetze und ihre Implementierung sowie die bilinearen Finite-Elemente auf diesen Netzen eingeführt. Danach wird die Diskretisierung der Gleichungen (1) auf den durch diese parametrischen Vierecksnetzen beschriebenen Gebieten diskutiert.

## Parametrischen Vierecksnetze

Die in den Programmierblättern der zugehörigen Vorlesung betrachteten Dreiecksnetze und auch die in der Vorlesung vorgestellten Vierecksnetze können im allgemeinen nur polygonal berandete Gebiete fehlerfrei darstellen. Wird also ein Gebiet mit nichtpolygonalem Rand durch solche Netze diskretisiert, so wird neben den üblichen Diskretisierungsfehlern auch ein Gebietsfehler eingebracht. Um nichtpolygonale berandete Gebiete in  $\mathbb{R}^2$  aber auch Oberflächen im  $\mathbb{R}^3$  als Gebiete exakt darstellen zu können, kann man auf parametrische Netze<sup>1</sup> ausweichen.

Um die Darstellung eines parametrischen Vierecksnetzes einzuführen, betrachten wir das folgende Beispiel einer Viertelkreisscheibe, wobei die Eckpunkte 1, 2 und 6 Neumann-Randpunkte und die Eckpunkte 3, 5 und 7 Dirichlet-Randpunkte sind:



<sup>1</sup>Obwohl wir uns hier auf Vierecksnetze beschränken, kann man im allgemeinen natürlich auch parametrische Dreiecksnetze u.ä. betrachten.

Ein parametrisches Vierecksnetz lässt sich durch ein Reihe von Listen darstellen:

- Die *Elementenumerierungsliste*,  $\mathbf{C} \in \mathbb{N}^{4 \times n_c}$ , enthält in der  $j$ -ten Spalte die Indizes der globalen Enumerierung der vier Eckpunkte des  $j$ -ten Vierecks des Netzes. Dabei ist die Reihenfolge der Eckpunkte lexikographisch gewählt.
- Die *Elementpatchliste*,  $\mathbf{P} \in \mathbb{N}^{1 \times n_v}$ , enthält in der  $j$ -ten Spalte den Index der *Patchabbildung* des  $j$ -ten Vierecks des Netzes.
- Die *Elementreferenzliste*,  $\mathbf{K} \in \mathbb{N}^{1 \times n_r}$ , enthält in der  $j$ -ten Spalte den Index des *Referenzelementes* des  $j$ -ten Vierecks des Netzes.
- Die *Randbedingungsliste*,  $\mathbf{B} \in \mathbb{N}^{1 \times n_e}$ , enthält in der  $j$ -ten Spalte den Typ des  $j$ -ten Punktes in der globalen Enumerierung der Eckpunkte des Netzes; 1 steht dabei für einen inneren Punkt, 0 für einen Randpunkt mit Dirichlet-Randbedingung und 2 für eine Randpunkt mit Neumann-Randbedingung.
- Die *Patchkoordinatenliste*,  $\mathbf{V} \in \mathbb{R}^{2 \times n_k}$ , enthält in der  $j$ -ten Spalte die Koordinaten des  $j$ -ten Patchpunktes.
- Die *Referenzelementliste*,  $\mathbf{R} \in \mathbb{N}^{4 \times n_r}$ , enthält in der  $j$ -ten Spalte die Indizes der vier Eckpunkte in die Patchkoordinatenliste des  $j$ -ten Referenzelementes. Dabei ist die Reihenfolge der Eckpunkte lexikographisch gewählt.
- Die *Patchabbildungsliste*,  $M \in (\mathbb{R}^2 \rightarrow \mathbb{R}^d)^{1 \times n_p}$ , enthält in der  $j$ -ten Spalte die Abbildung des  $j$ -ten Patches.
- Die *Patch-Jacobi-Matrixliste*,  $J \in (\mathbb{R}^2 \rightarrow \mathbb{R}^{d \times 2})^{1 \times n_p}$ , enthält in der  $j$ -ten Spalte die Jacobi-Matrixabbildung der Abbildung des  $j$ -ten Patches.

Für die abgebildete Situation wählt man beispielsweise

$$\mathbf{C} = \begin{bmatrix} 1 & 2 & 6 \\ 2 & 3 & 4 \\ 6 & 4 & 7 \\ 4 & 5 & 5 \end{bmatrix}, \quad \mathbf{P} = [1 \quad 2 \quad 3] \quad \text{und} \quad \mathbf{K} = [1 \quad 1 \quad 1],$$

wobei weiter

$$\mathbf{B} = [2 \quad 2 \quad 2 \quad 1 \quad 0 \quad 2 \quad 2]$$

sowie

$$\mathbf{V} = \begin{bmatrix} -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix} \quad \text{und} \quad \mathbf{R} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

sind. Am einfachsten definiert man die erste Patchabbildung als die bilineare Abbildung

$$\begin{bmatrix} x \\ y \end{bmatrix} \mapsto \begin{bmatrix} (1+x)(0.225 - 0.025 \cdot y) \\ (1+y)(0.225 - 0.025 \cdot x) \end{bmatrix}.$$

Die zweite und dritte Patchabbildungen können dann passend dazu als

$$\begin{bmatrix} x \\ y \end{bmatrix} \mapsto \begin{bmatrix} (1-x)(0.225 - 0.025 \cdot y) + 0.5(1+x) \cos((1+y)\pi/8) \\ (1-x)(0.1 + 0.1 \cdot y) + 0.5(1+x) \sin((1+y)\pi/8) \end{bmatrix}$$

und

$$\begin{bmatrix} x \\ y \end{bmatrix} \mapsto \begin{bmatrix} (1-y)(0.1 + 0.1 \cdot x) + 0.5(1+y) \sin((1+x)\pi/8) \\ (1-y)(0.225 - 0.025 \cdot x) + 0.5(1+y) \cos((1+x)\pi/8) \end{bmatrix}$$

gebildet werden. Wichtig ist dabei, wie es hier der Fall ist, die Parametrisierungen auf den Patchrändern übereinstimmen.

## Bilineare Finite-Elemente auf parametrischen Vierecksnetze

Mit den parametrischen Netzen z handen, können wir nun passende Finite-Elemente betrachten. Hierbei geht man wie folgt vor: Betrachten wir das  $j$ -te Element des Vierecksnetzes, dann kennen wir dessen Patchabbildung,  $P_j$ , dessen Jacobi-Matrixabbildung,  $D P_j$ , die globale Numerierung seiner Eckpunkte,  $i_1^{(j)}$ ,  $i_2^{(j)}$ ,  $i_3$  und  $i_4^{(j)}$ , sowie dessen zugehöriges Referenzelement und somit die Patchkoordinaten der Eckpunkte,  $\mathbf{p}_1^{(j)}$ ,  $\mathbf{p}_2^{(j)}$ ,  $\mathbf{p}_3^{(j)}$  und  $\mathbf{p}_4^{(j)}$ .

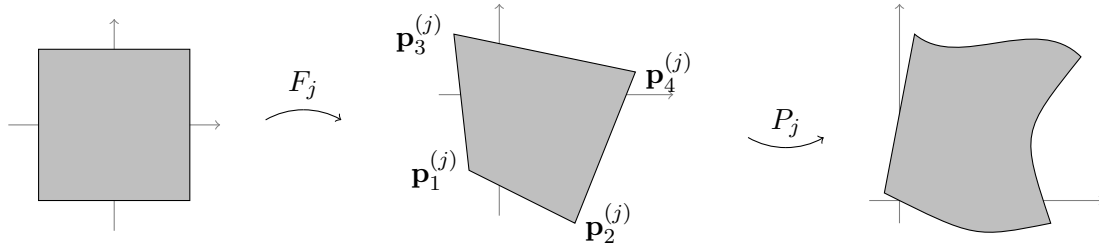
Wir betrachten weiter ein Referenzelement  $\square = [-1, 1]^2$ . Auf diesem können wir die vier Formfunktionen der bilinearen Finite-Elemente direkt angeben gemäss:

$$\begin{aligned} N_1(\xi_1, \xi_2) &= \frac{1 - \xi_1}{2} \frac{1 - \xi_2}{2}, & N_2(\xi_1, \xi_2) &= \frac{1 + \xi_1}{2} \frac{1 - \xi_2}{2}, \\ N_3(\xi_1, \xi_2) &= \frac{1 - \xi_1}{2} \frac{1 + \xi_2}{2}, & N_4(\xi_1, \xi_2) &= \frac{1 + \xi_1}{2} \frac{1 + \xi_2}{2}. \end{aligned}$$

Das Referenzelement können wir durch die bilineare Abbildung

$$F_j(\xi_1, \xi_2) = \sum_{k=1}^4 N_k(\xi_1, \xi_2) \mathbf{p}_k^{(j)}$$

auf das Referenzelement abbilden, welches dann durch die Abbildung  $P_j$  auf das eigentliche Element  $E_j$  abbildet.



Damit werden nun auf den Elementen folgende nodale Finite-Elemente-Funktionen betrachtet, die einen Punkt einfach auf das Referenzelement zurücktransformieren und dort die entsprechende Formfunktion auswerten,

$$f_i|_{E_j}(\mathbf{x}) = \begin{cases} N_s \circ F_j^{-1} \circ P_j^{-1}(\mathbf{x}), & \text{falls } i = i_s^{(j)}; \\ 0, & \text{sonst.} \end{cases}$$

Die lokalen Elementmatrizen für die Massen- und Steifigkeitsmatrix sind dadurch nun als

$$\mathbf{M}_j = [m_{r,s}^j]_{r,s=1}^4 \quad \text{und} \quad \mathbf{A}_j = [a_{r,s}^j]_{r,s=1}^4$$

gegeben, wobei

$$m_{r,s}^j = \int_{E_j} f_{i_s^{(j)}}(\mathbf{x}) f_{i_r^{(j)}}(\mathbf{x}) \, d\mathbf{x} = \int_{\square} N_s(\boldsymbol{\xi}) N_r(\boldsymbol{\xi}) \sqrt{\det T_j(\boldsymbol{\xi})} \, d\boldsymbol{\xi}$$

und

$$a_{r,s}^j = \int_{E_j} \langle \nabla f_{i_s^{(j)}}(\mathbf{x}), \nabla f_{i_r^{(j)}}(\mathbf{x}) \rangle \, d\mathbf{x} = \int_{\square} \langle \nabla N_s(\boldsymbol{\xi}), (T_j(\boldsymbol{\xi}))^{-1} \nabla N_r(\boldsymbol{\xi}) \rangle \sqrt{\det T_j(\boldsymbol{\xi})} \, d\boldsymbol{\xi}$$

mit  $T_j(\boldsymbol{\xi}) = \left( D P_j(F_j(\boldsymbol{\xi})) D F_j(\boldsymbol{\xi}) \right)^T D P_j(F_j(\boldsymbol{\xi})) D F_j(\boldsymbol{\xi})$  sind. Da  $T_j$  vom Vierecksnetz abhängt, werden die Elementintegrale über  $\square$  mit der Trapezregel approximiert. Dies hat auch den Vorteil, dass die so approximierte Matrix  $\mathbf{M}$  diagonal wird.<sup>2</sup>

<sup>2</sup>Das geeignete Approximieren der Massenmatrix mit einer diagonalen Matrix wird "mass lumping" genannt.

## Diskretisierung

Wir betrachten nun (1) für das durch ein Vierecksnetz gegebene Gebiet<sup>3</sup>  $D \subset \mathbb{R}^2$  und das Zeitintervall  $[0, T]$ . Das vollständige Modellproblem ist durch

$$\begin{aligned} \partial_t u &= a \Delta_{\mathbf{x}} u - cuv^2 + e(1 - u) && \text{in } D \times (0, T], \\ \partial_t v &= b \Delta_{\mathbf{x}} v + duv^2 - (e + f)v && \text{in } D \times (0, T], \\ \langle \nabla u, \mathbf{n} \rangle &= 0 && \text{auf } \partial D \times (0, T], \\ \langle \nabla v, \mathbf{n} \rangle &= 0 && \text{auf } \partial D \times (0, T], \\ u(\cdot, 0) &= u_0 && \text{in } D, \\ v(\cdot, 0) &= v_0 && \text{in } D \end{aligned}$$

gegeben. Die schwache Formulierung bezüglich den Gebietsvariablen lautet dann offensichtlich: Finde  $u, v: [0, T] \rightarrow H^1(D)$  mit  $u(0) = u_0$  und  $v(0) = v_0$ , so dass

$$\begin{aligned} (\partial_t u, \varphi)_{L^2} &= -a(\nabla_{\mathbf{x}} u, \nabla_{\mathbf{x}} \varphi)_{L^2} - c(uv^2, \varphi)_{L^2} + e((1 - u), \varphi)_{L^2} && \text{in } (0, T], \\ (\partial_t v, \psi)_{L^2} &= -b(\nabla_{\mathbf{x}} v, \nabla_{\mathbf{x}} \psi)_{L^2} + d(uv^2, \psi)_{L^2} - (e + f)(v, \psi)_{L^2} && \text{in } (0, T] \end{aligned}$$

für alle  $\varphi, \psi \in H^1(D)$  gilt. Durch Ersetzen des Raumes  $H^1(D)$  mit dem Raum der bilinearen Finiten-Elemente über dem parametrischen Vierecksnetz,  $V$ , und Betrachten der Koeffizientenvektoren der Finiten-Elemente-Funktionen erhält man

$$\begin{aligned} \mathbf{M} \partial_t \mathbf{u} &= -a\mathbf{A}\mathbf{u} - c\mathbf{N}(\mathbf{u}, \mathbf{v}) + e\mathbf{M}(\mathbf{1} - \mathbf{u}) && \text{in } (0, T], \\ \mathbf{M} \partial_t \mathbf{v} &= -b\mathbf{A}\mathbf{v} + d\mathbf{N}(\mathbf{u}, \mathbf{v}) - (e + f)\mathbf{M}\mathbf{v} && \text{in } (0, T], \\ \mathbf{u}(0) &= \mathbf{I}(u_0), \\ \mathbf{v}(0) &= \mathbf{I}(v_0), \end{aligned}$$

wobei  $\mathbf{A}$  und  $\mathbf{M}$  die vorher erläuterten Approximationen der Steifigkeits- und Massenmatrix sind.  $\mathbf{I}(g)$  bezeichnet die Punktinterpolation der Funktion  $g$  in den Finite-Elemente-Raum und  $\mathbf{N}(\mathbf{u}, \mathbf{v})$  ist wie folgt gegeben:

$$[\mathbf{N}(\mathbf{u}, \mathbf{v})]_i := \int_D \left( \sum_j [\mathbf{u}]_j f_j \right) \left( \sum_j [\mathbf{v}]_j f_j \right)^2 f_i \, dx.$$

Ersetzt man die Integration in  $\mathbf{N}$  mit der Summe der Elementanteile, wo für jeder Elementanteil die Trapezregel benützt wird, so erhält man die Approximation

$$\mathbf{N}(\mathbf{u}, \mathbf{v}) \approx \mathbf{M}[\mathbf{u}\mathbf{v}^2],$$

wobei die Multiplikation und Quadrierung in  $[\mathbf{u}\mathbf{v}^2]$  eintragsweise zu geschehen hat. Damit erhalten wir die gebietsdiskretisierte Gleichung

$$\begin{aligned} \mathbf{M} \partial_t \mathbf{u} &= -a\mathbf{A}\mathbf{u} - c\mathbf{M}[\mathbf{u}\mathbf{v}^2] + e\mathbf{M}(\mathbf{1} - \mathbf{u}) && \text{in } (0, T], \\ \mathbf{M} \partial_t \mathbf{v} &= -b\mathbf{A}\mathbf{v} + d\mathbf{M}[\mathbf{u}\mathbf{v}^2] - (e + f)\mathbf{M}\mathbf{v} && \text{in } (0, T], \\ \mathbf{u}(0) &= \mathbf{I}(u_0), \\ \mathbf{v}(0) &= \mathbf{I}(v_0), \end{aligned}$$

welche die Form eines Systems zweier gekoppelten, gewöhnlichen Differentialgleichungen annimmt. Diese kann nun in der Zeit mit dem expliziten Euler-Verfahren diskretisiert werden.

<sup>3</sup>Man kann ebenso eine Oberfläche  $D \subset \mathbb{R}^3$  betrachten. Man ersetzt dann den Laplace-Operator mit dem Laplace-Beltrami-Operator, wobei die betrachteten Gradienten dann zu Oberflächengradienten werden. Mit diesen Anpassungen bleiben alle anderen Aussagen gleich.

## Aufgaben

**Aufgabe 1.** Schreiben Sie eine Funktion

```
function m = mesh_generate_diskseg(),
```

welche das parametrische Vierecksnetz des vorher betrachteten Beispiels der Viertelkreisscheibe zurückgibt. Dabei soll  $m$  hier und auch nachfolgend jeweils ein Struct sein, welches alle Listen in Feldern abgespeichert hat; die *Patchabbildungsliste* und die *Patch-Jacobi-Matrixliste* sind dabei als Cell-Arrays von Function-Handles zu speichern.

**Aufgabe 2.** Schreiben Sie eine Funktion

```
function mesh_plot(m, subdiv),
```

die das Vierecksnetz,  $m$ , visualisiert und die Randknoten farblich markiert, wobei für Dirchlet- und Neumann-Randbedingungen je ein eigenes Markersymbol verwendet wird. Die Kanten der Element sollen dabei als die interpolativen Polygonzüge mit jeweils  $2^{\text{subdiv}}$  Liniensegmente gezeichnet werden.

**Aufgabe 3.** Schreiben Sie eine Funktion

```
function m = mesh_refine(m),
```

welche das parametrische Vierecksnetz verfeinert und zurückgibt.

*Hinweis. Ein Vorgehen ähnlich wie bei den Dreiecksnetzen auf Programmierblatt 1 ist möglich; dabei kommen beim Verfeinern von Vierecken nebst den Mittelpunkten der Kanten auch die Mittelpunkte der Vierecke als neue Eckpunkte hinzu. Effektiv hat man gleich zwei Elementlisten – die Elementenumerierungsliste und die Referenzelementliste – mit unterschiedlichen Elementen, welche man separat aber miteinander konsistent verfeinern kann.*

**Aufgabe 4.** Schreiben Sie eine Funktion

```
function m = mesh_generate_disk(),
```

welche ein parametrisches Vierecksnetz für die Einheitskreisscheibe zurückgibt.

*Hinweis. Man erhält ein mögliches Netz für die Einheitskreisscheibe, indem man Rotationen des Netzes der Viertelkreisscheibe von 0, 90, 180 und 270 Grad geeignet “zusammenklebt”. Die Punkte am Rand sollen dabei Neumann-Randbedingungen erfüllen.*

**Aufgabe 5.** Schreiben Sie eine Funktion

```
function m = mesh_generate_torus(),
```

welche ein parametrisches Vierecksnetz für die Torusoberfläche, welche von dem Rotationsstorus mit den Radien  $R = 1$  und  $r = 0.5$  stammt, zurückgibt.

**Aufgabe 6.** Schreiben Sie eine Funktion

```
function [A, M] = matrices_assemble(m),
```

welche die Steifigkeits- und die Massenmatrix für das übergebene, parametrische Vierecksnetz als Sparse-Matrizen zurückgibt.

**Aufgabe 7.** Schreiben Sie eine Funktion

```
function g = compute_nodal_interpolation(gf, m),
```

welche die nodale Interpolation der, durch das Function-Handle `gf` beschriebenen Funktion berechnet.

**Aufgabe 8.** Schreiben Sie eine Funktion

```
function mesh_function_plot(f, m),
```

welche die durch `f` beschriebene Finite-Elemente-Funktion zeichnet. Dabei sollen die parametrischen Viereckselemente durch geradkantige Vierecke approximiert werden.

**Aufgabe 9.** Schreiben Sie ein Skript `main9`, welches das Poisson-Problem

$$\begin{aligned} -\Delta u(\mathbf{x}) &= 4 \quad \text{für } \mathbf{x} \in \Omega, \\ u(\mathbf{x}) &= 0 \quad \text{für } \mathbf{x} \in \Gamma_D, \\ \langle \nabla u(\mathbf{x}), \mathbf{n}(\mathbf{x}) \rangle &= 0 \quad \text{für } \mathbf{x} \in \Gamma_N \end{aligned}$$

auf den Vierecksnetzen löst, die man durch ein- bis sechsmaliges Verfeinern des vorgestellten, groben Vierecksnetz der Viertelkreisscheibe erhält. Benutzen Sie die Randbedingungen des Netzes zur Definition der Partition des Randes in  $\Gamma_D$  und  $\Gamma_N$ . Das resultierende, lineare Gleichungssystem dürfen Sie mit dem Backslash-Solver lösen.

Für jede Verfeinerungsstufe berechnen Sie den  $L^2$ - und  $H^1$ -Fehler der numerischen Lösung  $u$  gegenüber der nodalen Approximation der exakten Lösung

$$u\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = 1 - x^2 - y^2.$$

Anschliessend erzeugen Sie einen `semilogy`-Plot, in dem Sie die Anzahl Verfeinerungen gegen den Fehler auftragen.

Sind die jeweiligen Konvergenzraten in Übereinstimmung mit den theoretischen Aussagen?

**Aufgabe 10.** Schreiben Sie eine Funktion

```
function [uh, vh, ts, gs]...  
= gray_scott(a, b, c, d, e, f, m, u0f, v0f, dt, T),
```

welche das Gray-Scott-Modell numerisch auf dem Vierecksnetz `m` mit dem expliziten Euler-Verfahren auf dem Zeitintervall 0 bis `T` mit der Schrittweite `dt` löst. Die Anfangswerte werden dabei als Function-Handles `u0f` und `v0f` übergeben. `a`, `b`, `c`, `d`, `e`, `f` sind die Parameter des Gray-Scott-Modells.

`uh` und `vh` sollen die numerischen Lösungen am Endzeitpunkt sein, `ts` der Zeilenvektor der Zeitpunkte und `gs` ein Zeilenvektor der Maximumnorm von  $\partial_t u$  zu jedem Zeitpunkt in `ts`.

**Aufgabe 11.** Schreiben Sie ein Skript `main10`, dass auf dem fünffach verfeinerten Kreisscheiben Vierecksnetz das Gray-Scott-Modell mit obiger Funktion löst und dessen Rückgabe Variablen graphisch darstellt. Dabei seien die Anfangswerte durch

$$u_0(\mathbf{x}) = \begin{cases} 0.5, & \text{falls } \|\mathbf{x} - \mathbf{p}\| < 0.3; \\ 1, & \text{sonst,} \end{cases} \quad \text{und} \quad v_0(\mathbf{x}) = \begin{cases} 0.25, & \text{falls } \|\mathbf{x} - \mathbf{p}\| < 0.3; \\ 0, & \text{sonst,} \end{cases}$$

gegeben, und der Endzeitpunkt als  $T = 4$  und die Zeitschrittweite als  $10^{-4}$  gesetzt. Als Modellparameter wählen Sie:

Situation	$\mathbf{p}$	$a$	$b$	$c$	$d$	$e$	$f$
1	$[0.3 \ 0.2]^T$	0.2	0.1	1000	1000	31	55
2	$[0.3 \ 0.3]^T$	0.2	0.1	1000	1000	28	55
3	$[0.5 \ 0.2]^T$	0.2	0.1	10000	10000	280	550

Was passiert, wenn Sie die Zeitschrittweite vierteln oder vervierfachen?

**Aufgabe 12.** Wiederholen Sie Aufgabe 11, wobei Sie als Vierecksnetz die sechsfach verfeinerte Torusoberfläche benutzen. Dabei seien die Anfangswerte gegeben durch

$$u_0(\mathbf{x}) = \begin{cases} 0.5, & \text{falls } \|\mathbf{x} - \mathbf{p}\| < 1.6; \\ 1, & \text{sonst,} \end{cases} \quad \text{und} \quad v_0(\mathbf{x}) = \begin{cases} 0.25, & \text{falls } \|\mathbf{x} - \mathbf{p}\| < 1.6; \\ 0, & \text{sonst,} \end{cases}$$

und die Modellparameter durch

Situation	$\mathbf{p}$	$a$	$b$	$c$	$d$	$e$	$f$
1	$[1 \ 0 \ 0.5]^T$	0.2	0.1	1000	1000	31	55
2	$[1 \ 0 \ 0.5]^T$	0.2	0.1	1000	1000	28	55

**Aufgabe 13** (Freiwillig). Schreiben Sie eine Funktion, die ein parametrisches Vierecksnetz für ein Gebiet Ihrer Wahl (z.B. Einheitsphäre, Ellipse, Pacman etc.) zurückgibt und wiederholen Sie Aufgabe 11 mit diesem Vierecksnetz. 😊

## Literatur

- [1] P. Gray and S. K. Scott. Autocatalytic reactions in the isothermal, continuous stirred tank reactor: Isolates and other forms of multistability. *Chemical Engineering Science*, 38(1):29–43, 1983.
- [2] P. Gray and S. K. Scott. Sustained oscillations and other exotic patterns of behavior in isothermal reactions. *The Journal of Physical Chemistry*, 89(1):22–32, 1985.
- [3] J. E. Pearson. Complex patterns in a simple system. *Science*, 261(5118):189–192, 1993.
- [4] I. Prigogine and G. Nicolis. *Self-Organisation in Nonequilibrium Systems: Towards A Dynamics of Complexity*, pages 3–12. Springer, Netherlands, 1985.
- [5] E. E. Sel'kov. Self-oscillations in glycolysis 1. A simple kinetic model. *European Journal of Biochemistry*, 4(1):79–86, 1968.