Übungen zur Vorlesung

Modellierung und Simulation in den Neurowissenschaften

http://conan.iwr.uni-heidelberg.de/teaching/numsimneuro_ss2013

Dr. Stefan Lang

Abgabe: 05. Juni 2013 in der Übung

Übung 9 Stationäre und instationäre Diffusion in 1D und 2D (10 Punkte)

Eine nicht nur in den Neurowissenschaften sehr wichtige Gleichung ist die (stationäre oder instationäre) Diffusions-Gleichung, die Sie in der Vorlesung zum Beispiel beim extrazellulären Potential kennengelernt haben. Die stationäre Diffusions-Gleichung spielt zum Beispiel in der Wärmeleitung oder allgemeinen Elektrostatik eine große Rolle. Im Allgemeinen existieren keine analytischen Lösungen. Daher werden wir in dieser Übung zwei Diffusions-Probleme numerisch mit Octave lösen, in dieser Aufgabe zunächst in 1D.

Die stationäre Diffusions-Gleichung (Poisson-Gleichung) lautet für eine unbekannte Größe $u(x)$ auf einem Gebiet Ω in 1D, 2D oder 3D:

$$\begin{aligned} -\nabla g_a(x)\nabla u(x) &= f(x) && \text{auf } \Omega, \\ \nabla u(x) &= u_N(x) && \text{auf } \partial\Omega_N, \\ u &= u_D(x) && \text{auf } \partial\Omega_D. \end{aligned}$$

Hierbei ist g_a ein ortsabhängiger Diffusionskoeffizient. Ein Teil $\partial\Omega_N$ des Randes des Gebiets trägt Neumann-Randbedingungen u_N, der Teil $\partial\Omega_D$ hingegen Dirichlet-Randbedingungen u_D. Für Neumann-Randbedingungen wird der Fluß von u über den Rand vorgegeben (u kann sich auf diesem Rand beliebig einstellen), während bei Dirichlet-Randbedingungen der Wert von u direkt vorgegeben wird, und u damit auf dem Rand keine Unbekannte mehr darstellt. Meist sind Ω_D und Ω_N disjunkt und es gilt $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$. In vielen Fällen ist der Diffusionskoeffizient g_a konstant.

Teilaufgabe (a)

Wir betrachten im ersten Teil der Aufgabe die eindimensionale, stationäre Diffusion mit reinen Dirichlet-Randbedingungen (d.h. $\partial\Omega_N = \emptyset$) auf dem Gebiet $\Omega = (0, l)$:

$$\begin{aligned} -\partial_x^2 u(x) &= f(x) && \text{auf } \Omega = (0, l), \\ u &= 0 && \text{für } x = 0, \\ u &= u_D && \text{für } x = l. \end{aligned} \tag{0.1}$$

Anschaulich bedeutet diese Gleichung, daß eine (zweimal stetig differenzierbare) Funktion u in Ω gesucht ist, deren zweite Ableitung durch eine gegebene Quell- oder Senken-Funktion $f(x)$ in Ω vorgegeben ist, und die am Rand die Dirichlet-Randbedingungen efüllt.

Um diese Gleichung numerisch zu lösen, unterteilen wir das Gebiet Ω in N äquidistante Teilstücke, siehe Abbildung 0.1. Es gibt $N - 1$ innere Knoten und somit mit den zwei Randknoten insgesamt $N + 1$ Stützstellen für u. Die Gitterknoten seien mit x_i, $i = 0 \ldots N + 1$, bezeichnet. Soll Gleichung

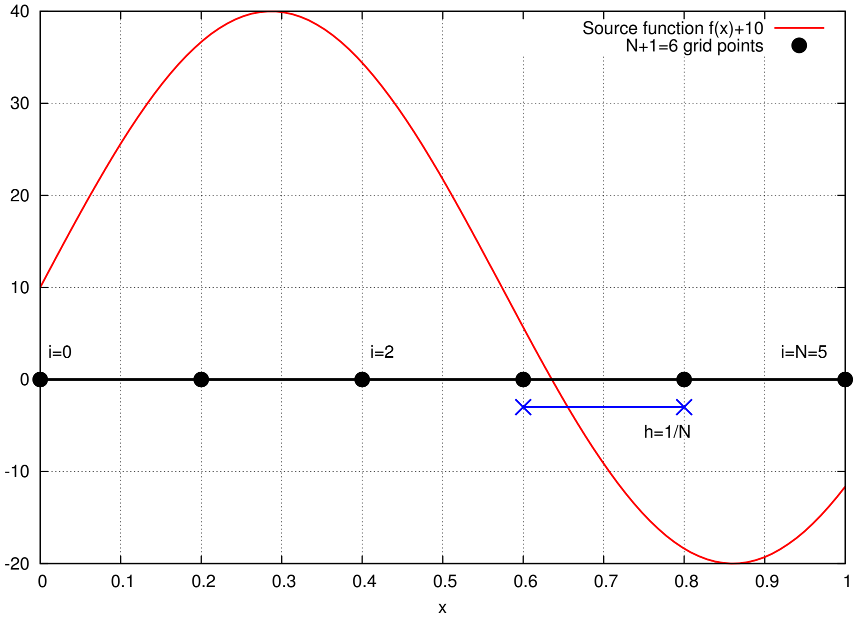

Abbildung 0.1: Gitter für die numerische Lösung der Poisson-Gleichung in 1D.

(0.1) numerisch gelöst werden, sind an den Gitterpunkten x_i Approximationen u_i des exakten, in der Regel aber unbekannten Funktionswerts $u(x_i)$ gesucht. Dazu wird die zweite Ableitung ∂_{xx} an einem inneren Gitterpunkten x_i wie folgt über den links- (∂^-) und rechtsseitigen (∂^+) Differenzenquotienten definiert:

$$\begin{aligned}
\partial^- u_i &:= \tfrac{1}{h}\left(u_i - u_{i-1}\right), \\
\partial^+ u_i &:= \tfrac{1}{h}\left(u_{i+1} - u_i\right), \\
\partial_x^2 u_i &= \partial^+\left(\partial^- u_i\right) = \tfrac{1}{h^2}\left(u_{i+1} - 2\,u_i + u_{i-1}\right),
\end{aligned}$$

mit der Gitterweite $h = x_{i+1} - x_i = x_i - x_{i-1}$. Diese Approximation der zweiten Ableitung läuft auf ein diskretes Gleichungssystem an inneren Gitterknoten hinaus, das die Unbekannten u_i an den inneren Gitterpunkten x_i, $i = 1 \ldots N-1$, eindeutig bestimmt:

$$-\frac{1}{h^2}\begin{pmatrix}
-2 & 1 & & & & & & \\
1 & -2 & 1 & & & & & \\
 & \ddots & \ddots & \ddots & & & & \\
 & & \ddots & \ddots & \ddots & & & \\
 & & & 1 & -2 & 1 & & \\
 & & & & \ddots & \ddots & \ddots & \\
 & & & & & 1 & -2 & 1 \\
 & & & & & & 1 & -2
\end{pmatrix}_{(N-1)\times(N-1)} \cdot \begin{pmatrix} u_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ u_{N-1} \end{pmatrix} = \begin{pmatrix} f_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f_{N-1} + \frac{1}{h^2} u_D \end{pmatrix}. \tag{0.2}$$

Auf der rechten Seite der Gleichung stehen die Funktionsauswertungen $f_i = f(x_i)$ an den inneren Gitterpunkten. Man beachte, daß die letzte Gleichung, die zum letzten inneren Punkt x_{N-1} neben der rechten Intervallgrenze $x_N = l$ korrespondiert, einen anderen Term auf der rechten Seite hat. Dieser kommt von der Dirichlet-Randbedingungen am rechten Terminal (überlegen Sie sich, wieso). Am linken Randpunkt gilt $u = 0$, und der zusätzliche Beitrag tritt nicht auf. In Kurz-Schreibweise kann man den Vektor der Unbekannten an den inneren Gitterpunkten $\vec{u}_h = \{u_1, \ldots, u_N\}^T$ und in gleicher Weise den Vektor $\vec{f}_h$ der Funktionswerte an den Gitterpunkten definieren und erhält

$$A \cdot \vec{u}_h = \vec{f}_h, \tag{0.3}$$

wobei A die obige dünnbesetzte Matrix ist.

Aufgabe

Es sei nun die Quellfunktion $f(x) = 30\ \sin(\sqrt{30}x)$ gegeben. Für diesen Fall existiert eine analytische Lösung der Gleichung (0.1), die man leicht durch Nachrechnen verifizieren kann:

$$u(x) = \sin(\sqrt{30}x) - x \cdot (\sin(\sqrt{30}) - 1).$$

Wir werden zusätzlich das Gleichungssystem (0.2) numerisch lösen (das liefert die durch die u_i bzw. $\vec{u}_h$ definierte lineare Approximation der Funktion u), und können an Hand der analytischen Lösung u die Güte der numerischen Approximation u_h quantifizieren.

Aufgaben

1. Implementieren Sie mit Octave das numerische Gleichungs-System (0.2) mit der gegebenen Quellfunktion $f(x)$. Exportieren Sie einen Plot der Lösung u über x.
2. Testen Sie das Verhalten des maximalen Fehlers an den vorhandenen Gitterpunkten x_i,

   $$e(x) := max_i \left| u(x_i) - u_h(x_i) \right|,$$

   in Abhängigkeit der Gitterweite $h = 2^{-j}$ für $j = 3, \ldots, 14$. Erstellen Sie einen doppelt-logarithmischen Plot des Fehlers e über der Gitterweite h. Es sollte sich eine Gerade einstellen, die Steigung ist die Konsistenz-Ordnung des Verfahrens.
3. Wechseln Sie das Vorzeichen von f und setzen Sie den rechten Dirichlet-Randwert bei $x = l$ auf $-u_D$. Wie ändert sich die Lösung u_h?

Hinweise

- Zum Lösen des diskreten Gleichungssystems (0.2) in Octave können Sie den \-Operator verwenden (`u = A \ f`).
- Die System-Matrix A ist dünnbesetzt. Es bietet sich daher an, geeignete Datenstrukturen (z.B. Sparse Matrices) zu verwenden. Der benötigte Befehl zum Konstruieren von Sparse Matrices lautet `spdiags`. Hinweise hierzu finden Sie z.B. im GNU Octave Handbuch, Kapitel 22.4 oder durch die Built-in Hilfe mit `help spdiags`.

Teilaufgabe (b)

Im zweiten Teil der Aufgabe betrachten wir eine instationäre Diffusion in 1D ohne Quelle ($f \equiv 0$) und mit Dirichlet-Randbedingungen:

$$\begin{aligned} \partial_t u(x,t) - \partial_x^2 u(x,t) &= 0 && \text{auf } \Omega = (0,l), \\ u(0,t) &= 0 && \text{(linke Dirichlet-Randbedingung)}, \\ u(l,t) &= 0 && \text{(rechte Dirichlet-Randbedingung)}, \\ u(x,0) &= \sin(3\pi x) && \text{(Anfangsbedingung)}. \end{aligned} \tag{0.4}$$

Bei dieser Gleichung handelt es sich um eine (parabolische) partielle Differentialgleichung auf dem Raum-Zeit-Gebiet $\Omega \times [0,T]$. Sie wird auch die Wärmeleitungsgleichung genannt und ist der Prototyp parabolischer Differentialgleichungen. Wie in Teilaufgabe (a) existiert auch hier eine analytische Lösung,

$$u(x,t) = \exp\left(-9\pi^2 t\right) \cdot \sin(3\pi x).$$

Wir werden diese Gleichung numerisch mit Octave lösen. Da die Quelle f verschwindet, ist dies zum Glück einfach möglich. Angenommen, das Gitter sei wie im stationären Fall zerlegt (Abbildung 0.1), und der Diffusions-Anteil ∂_x^2 ist wie in Teilaufgabe (a) durch die Matrix A aus (0.3) diskretisiert. Dann ist an den inneren Gitterpunkten x_i folgendes Gleichungssystem zu lösen:

$$\partial_t \vec{u}_h(t) - A\vec{u}_h(t) = 0,$$

wobei $\vec{u}_h(t)$ der Vektor der nun zeitabhängigen Unbekannten an den inneren Gitterknoten ist. Man kann nun, wie wir es von den gewöhnlichen Differentialgleichungen kennen, die Zeit-Ableitung an einem Zeitpunkt t_n durch einen Differenzenquotienten approximieren und ein Forward Euler-Zeitschrittverfahren anwenden. Dies liefert mit der Zeitschrittweite $\kappa = t_n - t_{n-1}$ folgende Vorschrift für die gesuchten $\vec{u}_h^n$ zu einem festen Zeitpunkt t_n:

$$\vec{u}_h^n - \vec{u}_h^{n-1} = \kappa \cdot A\vec{u}_h^{n-1} \Longleftrightarrow \vec{u}_h^n = (1 + \kappa \cdot A)\,\vec{u}_h^{n-1}. \tag{0.5}$$

Achtung: n ist hier kein Exponent, sondern der Zeitschritt-Index.

Aufgaben

1. Implementieren Sie mit Octave das numerische Gleichungs-System (0.5), Sie können dazu die Matrix aus Teilaufgabe (a) verwenden. Exportieren Sie einen Plot der Lösung für die Parameter Simulationsdauer $T = 0.1$, $\kappa = 1e-4$ und $N = 19$. Zeichnen Sie zum Vergleich auch die exakte Lösung ein.
2. Die Zahl $\beta = \kappa/h^2$ bestimmt wesentlich die Stabilität des numerischen Verfahrens (0.5). Um dies zu untersuchen, führen Sie Simulationen mit den Parametern $(N, \kappa) = (29, 1e-4)$, $(29, 2e-4), (29, 8e-4)$ und $(29, 1e-3)$ aus, berechnen Sie β und exportieren Sie jeweils einen Plot der numerischen sowie analytischen Lösung zum Zeitpunkt $T = 0.1$. Wie verhält sich die numerische Lösung qualitativ?
3. *Freiwilliger Zusatz:* Die Rechnung ist offenbar für $\beta \approx 0.5$ stabil. Falls Sie $N = 254$ setzen, wie muss dann κ gewählt werden (berücksichtigen Sie, daß T durch κ teilbar sein sollte)? Testen Sie Ihre Vorhersage praktisch!

Hinweis

1. In dem gegebenen sehr einfachen Fall kann die Vorschrift (0.5) auch gänzlich ohne Matrizen umgesetzt werden: Man iteriert zu jedem Zeitpunkt t_n über alle inneren Gitterpunkte x_i und berechnet den unbekannten Wert u_i^n durch Bilden von

   $$u_i^n = u_i^{n-1} + \tfrac{\kappa}{h^2} \cdot \left(u_{i-1}^{n-1} - 2\,u_i^{n-1} + u_{i+1}^{n-1}\right).$$

   Auf der rechten Seite stehen nur bekannte Werte des vorherigen Zeitpunkts t_{n-1}.