



Numerik von Gleichungssystemen 8. Übungsblatt

Aufgabe 8.0: Das ‘‘Apfelmännchen’’ zum Gradientenverfahren

Mit dem Aufkommen der Chaostheorie Ende der 80'ziger bzw. Anfang der 90'ziger Jahre erlangten bunte Bilder von fraktalen Mengen wie der *Mandelbrot-Menge* (im Volksmund ob seines Aussehens auch Apfelmännchen genannt) und verschiedener *Julia Mengen* viel Popularität. Die Bilder dieser Mengen stellen einen Ausschnitt der komplexen Ebene dar. Jedem Punkt wird ein Farbwert zugeordnet, welcher das Verhalten einer bestimmten komplexwertigen Zahlenfolge kodiert. Im Falle der Mandelbrot-Menge betrachtet man zu jedem Punkt $c \in \mathbb{C}$ der komplexen Zahlenebene die rekursiv definierte Folge

$$z_0 = 0, \quad z_{n+1} = z_n^2 + c.$$

Konvergiert die Folge, so wird ein schwarzer Punkt gesetzt, divergiert sie, so wird ein farbiger Punkt gesetzt, wobei sich die konkrete Farbwahl nach der Divergenzgeschwindigkeit richtet. In ähnlicher Weise kann man für jede Rekursion, welche auf \mathbb{C} bzw. \mathbb{R}^2 erklärt ist, Bilder erzeugen, die das Konvergenz- bzw. Divergenzverhalten der jeweiligen Folge widerspiegeln.

In dieser Aufgabe sollen derartige Konvergenzbilder für das Gradientenverfahren im \mathbb{R}^2 erstellt werden (welche zweifellos nicht so interessant aussehen werden wie die fraktalen Figuren der Mandelbrot- und Julia-Mengen, dennoch aufschlußreich genug sind, um sich der Mühe zu unterziehen). Dazu gebe man sich ein festes 2×2 Gleichungssystem vor. Jedem Punkt läßt sich nun eine rekursive Folge zuordnen – nämlich die durch das Gradientenverfahren definierte Folge – wobei der jeweilige Punkt als Startwert eingeht. Bekanntlich konvergiert das Gradientenverfahren für beliebige Startwerte aus \mathbb{R}^2 . Daher macht es nur Sinn, durch eine farbliche Markierung die Konvergenzgeschwindigkeit darzustellen. Um dies mit MATLAB umzusetzen, kann man folgendermaßen vorgehen:

- Erzeugung eines Gitters mittels `meshgrid`.
- Durlaufen aller Gitterpunkte mittels einer `for`-Schleife, dabei wird in jedem Gitterpunkt das Gradientenverfahren gestartet. Es bestehen nun zwei Möglichkeiten:
 - Man iteriert solange, bis eine bestimmte Fehlergrenze `TOL` unterschritten ist und speichert für den betreffenden Gitterpunkt die Anzahl der benötigten Iterationen. (Beachte: mittels der Konvergenzrate für das Gradientenverfahren läßt sich die Anzahl der benötigten Iterationen nach oben abschätzen.)
 - Man gibt eine feste Anzahl n an Iterationen vor (unabhängig von dem jeweiligen Gitterpunkt) und speichert den Fehler, welcher nach Durchführung der n Iterationen noch vorhanden ist.
- Mittels `surf` läßt sich die Anzahl der benötigten Iterationen bzw. der Fehler nach n Iterationen als Funktion des Startwertes graphisch darstellen. Es ist sinnvoll, das ‘‘Farbmuster’’ zusammen mit einigen elliptischen Niveaulinien der Funktion $f(x) = \frac{1}{2}\langle Ax, x \rangle - \langle b, x \rangle$ darzustellen.

Wie hängt das erhaltene Farbmuster von den Niveaulinien ab, welche Erwartung haben Sie? Welchen ‘‘Einfluß’’ hat das Farbmuster auf den Verlauf einer Iterationsfolge? Statt des Gradientenverfahrens lassen sich auch andere iterative Löser verwenden, bei denen der Startwert frei wählbar ist, wie z.B. das Jacobi-Verfahren.

Aufgabe 8.1: Nachtrag zur Vorlesung

Zeigen Sie, daß beim CG-Verfahren alternative Berechnungsmöglichkeiten für die Koeffizienten α_k, β_k bestehen; es gilt:

$$\alpha_k := -\frac{\langle g_k, d_k \rangle}{\langle Ad_k, d_k \rangle} \stackrel{!}{=} \frac{\langle g_k, g_k \rangle}{\langle Ad_k, d_k \rangle}, \quad \beta_k := \frac{\langle Ag_{k+1}, d_k \rangle}{\langle Ad_k, d_k \rangle} \stackrel{!}{=} \frac{\langle g_{k+1}, g_{k+1} \rangle}{\langle g_k, g_k \rangle}.$$

Warum ist es effizienter, die alternativen Berechnungsmöglichkeiten einzusetzen?

Aufgabe 8.2: Das Verfahren konjugierter Residuen (CR-Verfahren)

In der Vorlesung wurde das folgende Verfahren hergeleitet, welches das Residuum r_k der k 'ten Iteration x_k in der Standardnorm (euklidische Norm im \mathbb{R}^n) minimiert.

Start: $x_0 \in \mathbb{R}^n$, $r_0 = Ax_0 - b$, $w_0 = -r_0$

Für $k = 1, 2, \dots$ bis maximal n :

- $\eta_{k-1} = \frac{\langle Ar_{k-1}, r_{k-1} \rangle}{\langle Aw_{k-1}, Aw_{k-1} \rangle}$
- $x_k = x_{k-1} + \eta_{k-1} w_{k-1}$
- $r_k = r_{k-1} + \eta_{k-1} A w_{k-1}$
- $\nu_{k-1} = \frac{\langle Ar_k, r_k \rangle}{\langle Ar_{k-1}, r_{k-1} \rangle}$
- $w_k = -r_k + \nu_{k-1} w_{k-1}$

- a) Implementieren Sie den Algorithmus, achten Sie darauf, die Anzahl der Rechenoperationen wie z.B. Matrix-Vektor Multiplikationen möglichst gering zu halten.
- b) Warum ergeben sich paarweise A -konjugierte Residuen. Prüfen Sie dies numerisch. Vergleichen Sie das CR- und mit dem CG-Verfahren.
- c) Wie wirkt sich die Verteilung der Eigenwerte von A auf das Konvergenzverhalten des CR- bzw. CG-Verfahrens aus? Wenden Sie dazu das Verfahren auf symmetrische, positiv definite Matrizen an, welche vergleichsweise wenige Eigenwerte (dafür mit relativ hoher Multiplizität) besitzen.

Aufgabe 8.3: Polynom-basierte Herleitung des CG-Verfahrens

Um den polynomialen Zugang, welcher in der Vorlesung anhand der Herleitung des Verfahrens konjugierter Residuen vorgestellt wurde, zu verinnerlichen, soll hier das CG-Verfahren noch einmal alternativ hergeleitet werden.

Abstrakt gesehen ist das CG-Verfahren dadurch ausgezeichnet, daß die k 'te Iteration x_k den A -Abstand (d.h. bzgl. der $\|\cdot\|_A$ -Norm) zur Lösung \bar{x} in dem verschobenen Krylovraum (affin-linearer Teilraum) $x_0 + \mathcal{K}_k(A, r_0) = x_0 + \text{span}\{r_0, Ar_0, \dots, A^{k-1}r_0\}$ mit $r_0 := b - Ax_0$ minimiert.

- 0) Zeigen Sie, daß die k 'te Iterierte x_k bzw. der zugehörige Fehler $e_k = x_k - \bar{x}$ auch durch folgende Gleichung charakterisiert ist: $\| \underbrace{x_k - \bar{x}}_{e_k} \|_A = \min_{p \in \mathcal{P}_k, p(0)=1} \| p(A) \underbrace{(x_0 - \bar{x})}_{e_0} \|_A$
- a) Es sei $\langle \cdot, \cdot \rangle$ ein Skalarprodukt auf dem Vektorraum \mathcal{P}_k aller Polynome vom Grad kleiner gleich k . Ein Polynom K_k in zwei Variablen t, x mit $K_k(t, \cdot) \in \mathcal{P}_k$ und $K_k(\cdot, x) \in \mathcal{P}_k$ heißt *Kernpolynom*, falls gilt:

$$\forall p \in \mathcal{P}_k : \langle K_k(t, \cdot), p \rangle = p(t)$$

Wie läßt sich das Kernpolynom K_k aus einer orthonormierten Basis in \mathcal{P}_k erzeugen?

- b) Zeigen Sie, daß sich mittels der Kernpolynome folgende Minimierungsaufgabe konkret lösen läßt: Gesucht ist $\tilde{p} \in \mathcal{P}_k$ mit

$$i) \tilde{p}(x_0) = 1, \quad ii) \forall q \in \mathcal{P}_k, q(t_0) = 1 : \langle \tilde{p}, \tilde{p} \rangle \leq \langle q, q \rangle$$

Aufgrund welcher allgemeinen Argumente ist die Existenz (und Eindeutigkeit) einer Lösung zu diesem Minimierungsproblem gesichert?

- c) Generell gilt, daß die Polynome $x \mapsto K_k(t, x)$ und $x \mapsto K_\ell(t, x)$ nicht orthogonal sind für $k \neq \ell$. Angenommen das Skalarprodukt ist in dem folgenden Sinne mit der Polynommultiplikation verträglich, d.h. für zwei beliebige Polynome p, q gelte

$$\langle p, q \rangle = \langle 1, pq \rangle.$$

Geben Sie unter dieser Bedingung ein Skalarprodukt an, bezüglich dessen die Kernpolynome $x \mapsto K_k(x_0, x)$, $k \in \mathbb{N}_0$ paarweise orthogonal sind. **Hinweis:** Das Skalarprodukt darf von x_0 abhängen.

- d) Wenden Sie die allgemeinen Resultate aus a), b) und c) an und bestimmen sie mittels geeigneter Kernpolynome dasjenige Polynom $p_k \in \mathcal{P}_k$ mit $p_k(0) = 1$, welches $\|p(A)e_0\|_A$ unter der Nebenbedingung $p(0) = 1$ minimiert. Führen Sie dazu zunächst ein geeignetes Skalarprodukt ein.
- e) Leiten Sie den Algorithmus des CG-Verfahrens konkret her unter Rückgriff auf den Prototyp Algorithmus aus der Vorlesung.