Zentrum für
Technomathematik

Inverse Probleme
SS 2009
Prof. Peter Maaß
Kamil S. Kazimierski

http://www.math.uni-bremen.de/~kamilk/inverse_probleme_ss09/

Übungsblatt Nr. 6
Abgabe Freitag, 05.06.2009 vor der Vorlesung

Aufgabe 1: [Ein Kessel Buntes] 4 Punkte

a) Zeige, dass für die Itererierten des Landweberverfahrens $f_{n+1} = f_n - \omega A^*(Af_n - y^\delta)$ mit $0 < \omega < 2/\|A\|^2$

$$\|Af_{n+1} - y^\delta\|^2 < \|Af_n - y^\delta\|^2$$

gilt.

b) Zeige am Beispiel eines Operators $A : \mathbb{R}^2 \to \mathbb{R}^2$, dass cg-Verfahren nicht linear ist. Sind die TSVD, das Landweber-Verfahren und das Tikhonov-Verfahren linear?

Aufgabe 2, Programmierung: [Dekonvolution] 4 Punkte

In der Photographie hat man oft das Problem, dass aufgenommene Bilder unscharf sind, weil sich das aufgenommene Objekt bewegt (Bewegungsunschärfe) oder der Kamerafokus falsch eingestellt ist (Tiefenunschärfe). Der Prozess, der aus einem scharfen Bild ein unscharfes erzeugt, kann mathematisch durch eine *Konvolution* (Faltung)

$$g = Af = k * f, \quad g(y) = \int f(x)k(y-x)\,\mathrm{d}x$$

modelliert werden. Dabei ist f das scharfe Bild, g das unscharfe und k die Kernfunktion, die vom Typ der Unschärfe abhängt. Der Umkehrprozess, f aus $g = Af$ zu berechnen, wird Dekonvolution genannt und ist ein schlecht gestelltes inverses Problem. In dieser Aufgabe soll eine Dekonvolution durchgeführt werden. Ein Faltungskern k sowie die Operatoren A mit $Af = k * f$ und die Adjungierte A^* gegeben durch $A^*g = k(-\cdot) * g$ sind dabei bereits vorimplementiert. Du findest die Funktionen und Daten auf der Vorlesungs-Homepage.

Originalbild f

Degradiertes Bild g = Af

a) Implementiere das Landweberverfahren mit *unverrauschten* Daten aus `auge.gif`

$$f_{n+1} = f_n - \beta A^*(Af_n - g).$$

Benutze $\beta = 1$ und $f_0 = g$ als Startwert und führe 1000 Schritte durch. Plotten Sie die Werte $\|f_n\|$ sowie die Residuen $\|Af_n - g\|$ semilogarithmisch über n (Achtung: Bei Bildern `norm(f_n(:))` benutzen statt `norm(f_n)`!). Plotte außerdem f_n für $n = 5, 20, 50, 100, 300, 1000$. Beschreibe qualitativ den Verlauf der Kurven. Entspricht das Verhalten den theoretischen Vorhersagen?

b) Verrausche das Bild g mit punktweise normalverteiltem Rauschen mit Standardabweichung 0.4. Wiederholen Sie Teil a) mit den verrauschten Daten g^δ statt g. Welche Unterschiede ergeben sich? Entspricht auch hier das Verhalten den theoretischen Vorhersagen?

c) Wiederhole Teil b) mit dem Tikhonov-Gradientenabstiegsverfahren

$$f_{n+1} = f_n - \beta((A^*A + \alpha I)f_n - A^*g^\delta) = (1 - \beta\alpha)f_n - \beta A^*(Af_n - g^\delta)$$

anstelle des Landweberverfahrens. Experimentiere mit α und wähle von Hand einen sinnvollen Wert. Statt der Residuen plotte den Wert des Tikhonov-Funktionals $J_\alpha(f_n)$ semilogarithmisch über n. Beschreibe den Verlauf. Entspricht das Verhalten den theoretischen Vorhersagen?

Aufgabe 3, Programmierung: [Radon-Transformation] 4 Punkte

Die Radon-Transformation bildet ein Bild f ab auf Linienintegrale mit Abstand s zum Ursprung und im Winkel ϕ,

$$(Af)(\phi, s) = g(\phi, s) = \int_{\mathbb{R}} f(\omega s + \omega^\perp t)dt,$$

wobei $\omega = (\cos(\phi), \sin(\phi))$ und $\omega^\perp = (-\sin(\phi), \cos(\phi))$ ist. Die Radon-Transformation modelliert die Durchleuchtung eines Körpers von allen Seiten mit Röntgenstrahlen. Das Bild g, welches dabei entsteht, nennt man *Sinogramm* (s. Abbildung links).
Die Radon-Transformation sowie deren Adjungierte ist in den Programmen `aufgabe6_3_netz.m`, `aufgabe6_3_radon.cpp` und `aufgabe6_3_radon_adjoint.cpp` vorimplementiert. Die beiden letzten Dateien sind C++ Dateien und sind von Matlab aus nicht direkt verwendbar. Sie müssen erst kompiliert werden mit den Befehlen `mex -O aufgabe6_3_radon.cpp` und `mex -O aufgabe6_3_radon_adjoint.cpp`. Eventuell muss der Mex-Kompiler erst mit `mex -setup` konfiguriert werden (benutzen Sie den Lcc-Kompiler). Auf der Vorlesungs-Homepage findest du auch die Datei `aufgabe6_3_sinogramm.mat`, die das unten dargestellte verrauschte Sinogramm enthält. Schreibe einen Algorithmus, der das Originalbild rekonstruiert. Falls du einen iterativen Zugang wählst, kannst du als Abbruchkriterium das heuristische L-Kurven-Kriterium verwenden. Wähle dazu als Startwert $f^0 = 0$ und plotte $\log(\|f^n\|)$ gegen $\log(\|Af^n - g^\delta\|)$ (s. Abbildung rechts), wobei f^n die n-te Iterierte ist. Es entsteht eine Kurve in der Form eines L, die von rechts unten nach links oben durchlaufen wird. Brich ab, wenn die Iterierte die ecke des L passieren. In der Abbildung markiert der Pfeil diesen Punkt.

Sende deine Rekonstruktion als `.mat`-Datei an den Tutor (zum Speichern `save` verwenden).