



UNIVERSITÄT ZU LÜBECK



**MIC**

Institute of Mathematics and  
Image Computing

Bachelorarbeit

IMAGE DEBLURRING MIT  
REGULARISIERUNG

Jenny Stritzel

Thema ausgegeben von

Prof. Dr. Bernd Fischer

Institute of Mathematics and Image Computing

Arbeit betreut von

Ph.D. cand., Dipl-Inf. Björn Beuthien

Institute of Mathematics and Image Computing

Abgabe: 18.06.2012

## **Erklärung**

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, dass alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht sind und dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt wurde.

Lübeck, den 14. März 2013

Unterschrift

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Grundlagen</b>	<b>3</b>
2.1	Verwendetes Beispiel . . . . .	3
2.2	Singulärwertzerlegung . . . . .	3
2.3	Verfahren der konjugierten Gradienten . . . . .	7
<b>3</b>	<b>Regularisierungsverfahren</b>	<b>11</b>
3.1	TSVD Regularisierung . . . . .	11
3.2	Tikhonov Regularisierung . . . . .	13
3.3	Wahl des Regularisierungsparameters . . . . .	15
3.3.1	GCV-Verfahren . . . . .	16
3.3.2	L-Kurven Methode . . . . .	17
3.3.3	Vergleich von GCV und L-Kurven Methode . . . . .	19
3.4	Lanczos-Hybrid-Methode . . . . .	22
<b>4</b>	<b>HyBR</b>	<b>25</b>
<b>5</b>	<b>Vergleich der Verfahren</b>	<b>27</b>
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>30</b>

# 1 Einleitung

Image Deblurring ist der Prozess verrauschte, unscharfe Bilder so zu bearbeiten, dass die Originalbilder bestmöglich wiederhergestellt werden. Solche Prozesse findet man in vielen verschiedenen Bereichen, zum Beispiel in der Astronomie, medizinischen Bildgebung, im Militär, in der Überwachung, beim Scannen von Augen, in der Mikroskopie oder bei der Videokommunikation [9]. Aber auch bei der digitalen Fotografie im Alltag kann es zum Beispiel zu sogenannten „geblurrten“ Bildern kommen. Diese Bilder sehen dann verwischt und verwackelt aus.

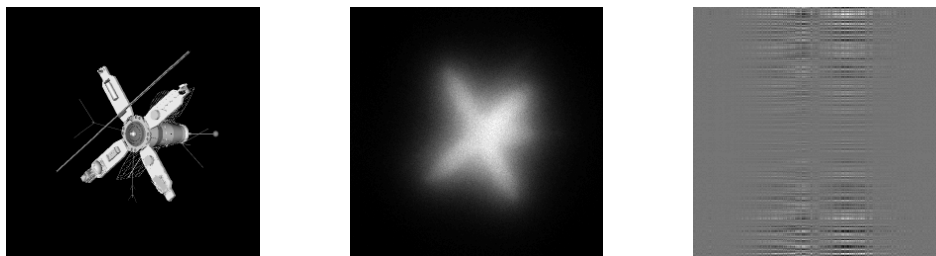


Abbildung 1: Im linken Bild ist das Original abgebildet. Das Original wird geblurt und mit einem Rauschen belegt, was im mittleren Bild zu sehen ist. Rechts sieht man dann den Versuch das Original mit Hilfe einer naiven Rückrechnung wiederherzustellen.

Typischerweise hat so ein Problem die Form

$$b = Ax + e \ .$$

Das geblurrte Bild  $b$  entsteht dadurch, dass das Originalbild  $x$  durch  $A$  geblurt und in der Regel noch ein Rauschen  $e$  dazu addiert wird. Wobei  $A$  eine sehr große und für gewöhnlich schlecht konditionierte Matrix ist.

Die Kunst besteht jetzt darin, das Originalbild  $x$  bestmöglich zu rekonstruieren. Wie man in Abbildung 1 sieht, erzielt man mit naiven Rekonstruktionsverfahren keine guten Ergebnisse. Das liegt vor allem daran, dass das Rauschen  $e$  nicht mit in die Berechnungen einbezogen wird, was für ein gutes Ergebnis aber zwingend notwendig ist.

Gegenüber der naiven Rekonstruktion stelle ich in meiner Arbeit zwei weitere Verfahren zur Bildrekonstruktion vor, zum Einen das Verfahren der konjugierten Gradienten und zum Anderen die Hybrid Bidiagonalization Regularization. Ich werde beide Verfahren herleiten und sie dann im weiteren Verlauf miteinander vergleichen.

Beide Verfahren nehmen sogenannte Regularisierungsverfahren, wie zum Beispiel die TSVD Regularisierung, die Tikhonov Regularisierung oder die Lanczos-Hybrid-Methode zur Behandlung von schlecht gestellten Problemen zu Hilfe. Hier spielt vor allem die geeignete Wahl eines Regularisierungsparameters eine wichtige Rolle, für die es wiederum verschiedene Verfahren gibt. Das GCV-Verfahren oder die L-Kurven Methode wären zwei dieser Verfahren, auf die wir hier näher eingehen und auch herleiten werden. Daraufhin versuche ich einen Vergleich zwischen dem GCV-Verfahren und der L-Kurven Methode zu ziehen, um herauszufinden, mit welcher Technik man anscheinend den besseren Regularisierungsparameter errechnen kann.

## 2 Grundlagen

### 2.1 Verwendetes Beispiel

In meiner Arbeit verwende ich das Beispiel „satellite“ aus den Restore Tools von James Nagy und Mitwirkenden [3, 4, 5]. Die Matlab-Daten beinhalten drei gegebene Matrizen aus dem  $\mathbb{R}^{256 \times 256}$ . Das Originalbild ist gegeben als  $x_{true}$ , die Blurringmatrix als  $PSF$ , was für point spread function steht und das geblurrte Bild als  $b$ .

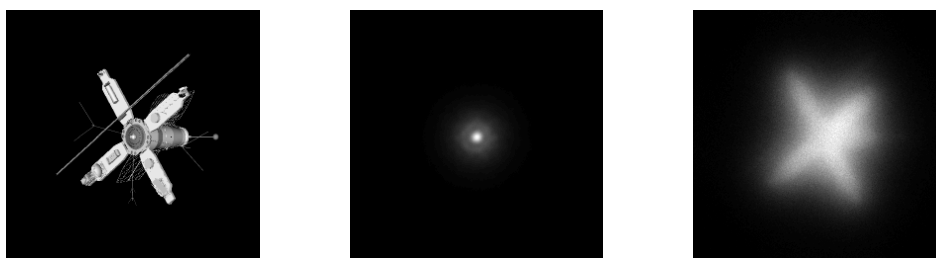


Abbildung 2: Im linken Bild ist das Original  $x_{true}$  abgebildet. Die Blurringmatrix  $PSF$  ist im mittleren Bild zu sehen. Rechts sieht man dann das geblurrte Bild  $b$ .

Außer diesen Beispieldaten verwende ich auch noch einige Methoden aus demselben Tool zur Berechnung entsprechender Beispielbilder. Direkt wird TikGCV, HyBR und CGLS benutzt. Diese Methoden verwenden intern aber noch einige weitere Methoden aus dem Tool, worauf ich nicht genauer eingehen werde, (siehe [4, 3]).

### 2.2 Singulärwertzerlegung

Im Englischen heißt die Singulärwertzerlegung Singular Value Decomposition und wird daher meistens kurz SVD genannt. Dies ist ein Verfahren zur Zerlegung beliebiger Matrizen aus dem  $\mathbb{R}^{n \times m}$ . Für nicht quadratische Matrizen ist diese Zerlegung besonders wichtig für das Invertieren, da man eine nicht invertierbare Matrix in invertierbare Matrizen zerlegt. Ich möchte hier aber nur auf die Zerlegung quadratischer Matrizen aus dem  $\mathbb{R}^{n \times n}$  eingehen, da wir im Weiteren auch nur diese verwenden werden. Bei quadratischen Matrizen wird diese Zerlegung vor allem bei schlecht konditionierten Matrizen angewendet. Diese Matrizen können zwar regulär, also invertierbar sein, jedoch kommt es beim Invertieren häufig zu schweren Rundungsfehlern, die diese Matrizen dann doch singulär erscheinen lassen. Auch hier schafft die

Singulärwertzerlegung Abhilfe, wie bei rechteckigen Matrizen.

Die Matrix  $A \in \mathbb{R}^{n \times n}$  wird hierbei in  $A = U\Sigma V^T$  zerlegt.  $U$  und  $V$  sind orthonormale Matrizen mit  $U, V \in \mathbb{R}^{n \times n}$ , die die Singulärvektoren von  $A$  enthalten. Es gilt hier die Beziehung  $U^T U = V^T V = E$ , wobei  $E$  die Einheitsmatrix ist. Die Singulärwerte von  $A$  stehen auf der Hauptdiagonalen von  $\Sigma$ , also  $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$  mit  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$ . Das Verhältnis  $\frac{\sigma_1}{\sigma_n}$  heißt auch Konditionszahl von  $A$ . Man kann  $A$  auch schreiben als  $A = \sum_{i=1}^n \sigma_i u_i v_i^T$ , mit den  $i$ -ten Spaltenvektoren  $u_i$  und  $v_i$  der dazugehörigen Matrizen  $U$  und  $V$ .

Die Singulärwerte sind die positiven Wurzeln der Eigenwerte der Matrix  $A^T A$ , aus diesem Grund kann die Matrix  $A$  mit Hilfe der Singulärwerte ähnlich gut beschrieben werden wie mit den Eigenwerten. Dies hat eine besondere Bedeutung für nicht quadratische Matrizen, da diese keine Eigenwerte besitzen, aber deren Eigenschaften dann durch die Singulärwerte beschrieben werden können. Allerdings ist die Singulärwertzerlegung für sehr große Matrizen auch sehr aufwendig.

Bei der Singulärwertzerlegung von schlecht konditionierten Matrizen kann man immer wieder beobachten, dass sich die Singulärwerte  $\sigma_i$  der Matrix  $A$  in der Nähe von Null häufen. Dies kann allerdings zu großen Problemen führen, da man bei der Lösung des linearen Gleichungssystems  $Ax = b$  durch die Singulärwerte  $\sigma_i$  dividieren muss:

$$x = A^{-1}b = \sum_{i=1}^n \frac{u_i^T b v_i}{\sigma_i}. \quad (1)$$

Dies bewirkt, dass ein beliebig kleines Rauschen in den Daten zu sehr großen Abweichungen, also zur Verstärkung des Rauschens führt. Deutlicher wird es, wenn wir vorherige Gleichung durch ein Rauschen  $e$  erweitern, damit wir mögliches Rauschen mit in die Rechnung einbeziehen können. So erhalten wir unser neues Gleichungssystem  $Ax + e = b$ . Die Lösung dieses Systems

$$x = A^{-1}(b - e) = \sum_{i=1}^n \frac{u_i^T b v_i}{\sigma_i} - \sum_{i=1}^n \frac{u_i^T e v_i}{\sigma_i} \quad (2)$$

verdeutlicht nochmal, dass durch das Dividieren durch kleine Singulärwerte  $\sigma_i$  auch sehr kleine Werte für  $e$  zu starkem Rauschen führen können, wie gut in Abbildung 3 zu erkennen ist.

In Abbildung 3 werden die Originaldaten in schwarz abgebildet. Für die Originaldaten wurde das Beispielbild des Satelliten in einen Spaltenvektor umgeschrieben. Dann wurde ein Teilvektor herausgenommen, so dass wir  $x \in \mathbb{R}^{150}$

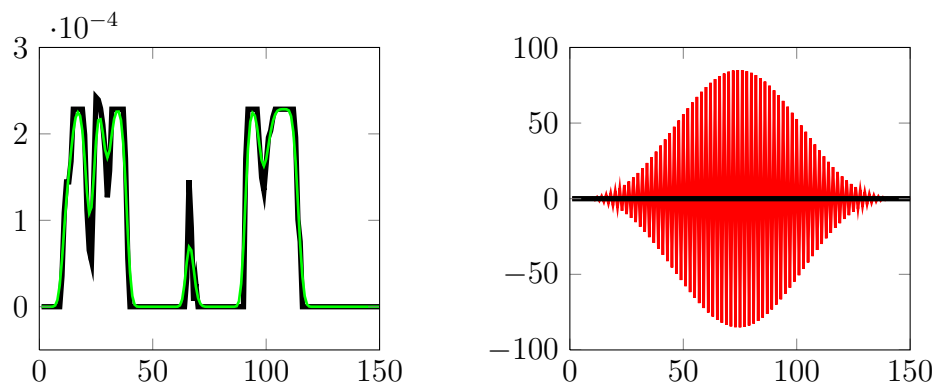


Abbildung 3: Im linken Bild sieht man einen Bildausschnitt  $x$  im Eindimensionalen (schwarz) und der gleiche Ausschnitt geblurt und mit einem Rauschen belegt,  $b$  (grün). Der Unterschied zwischen beiden Signalen ist nur minimal bei minimalem Rauschen der Größenordnung  $|10^{-10}|$ . Beim Versuch das Original  $x$  (schwarz) mit SVD wiederherzustellen, entsteht das Signal in rot auf der rechten Seite, welches durch sehr starkes Rauschen geprägt ist.

erhalten.  $A$  ist ein Gaußfilter in der Dimension  $\mathbb{R}^{150 \times 150}$ . Dieser Filter blurt das Originalbild und lässt es unscharf erscheinen. Im Eindimensionalen ist dies daran zu erkennen, dass die Ecken abgerundet, also weicher gemacht werden. Das Rauschen  $e$  ist ein Spaltenvektor im  $\mathbb{R}^{150}$ . Die einzelnen Einträge von  $e$  sind sehr klein und bewegen sich im Intervall von  $-5.2 \cdot 10^{-10}$  bis  $4.7 \cdot 10^{-10}$ . Das geblurte Bild  $b$  ist ebenfalls ein Spaltenvektor im  $\mathbb{R}^{150}$  und wird errechnet durch  $b = Ax + e$ .

Im linken Bild von Abbildung 3 kann man deutlich erkennen, dass die grüne Linie des gefilterten Bildes abgerundeter Ecken hat, im Gegensatz zu der schwarzen Linie des Originals. Der Unterschied beider Bilder ist durch das Filtern und das Rauschen aber dennoch sehr gering ausgefallen.

Im rechten Bild von Abbildung 3 stellt die rote Linie die durch Gleichung (1) wiederhergestellten Originaldaten (wieder schwarz) dar. Nun ist zu erkennen, dass die wiederhergestellten Daten enorm von den Originaldaten abweichen, obwohl das Rauschen  $e$  nur sehr gering ist. Die Umkehrung ist also numerisch instabil und oszilliert, bei uns vor allem im mittleren Teil, sehr stark. Die Amplitude der Originaldaten liegt bei ungefähr  $2.5 \cdot 10^{-4}$ , während die Amplitude der errechneten Daten bei ungefähr  $\pm 85$  liegt. Dies ist auf das Dividieren durch die Singulärwerte in Gleichung (1) zurückzuführen, welches dazu führt, dass ein kleines Rauschen zu sehr großen Problemen führen kann. Daher muss man das Rauschen bei der Rückrechnung mit einbeziehen, so wie



es in Gleichung (2) zu sehen ist.

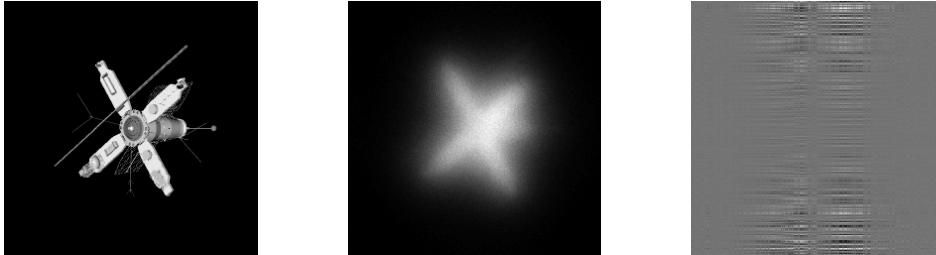


Abbildung 4: Im linken Bild ist das Original abgebildet. Das Original wird gefiltert und mit einem Rauschen belegt, was im mittleren Bild zu sehen ist. Im rechten Bild sieht man dann den Versuch das Original mit Hilfe von Gleichung (1) wiederherzustellen.

Im Zweidimensionalen verhält es sich genau wie im Eindimensionalen auch. Dieses Mal nehmen wir für  $x$  das komplette Beispielbild „satellite“ der Dimension  $\mathbb{R}^{256 \times 256}$  aus den Restore Tools. Dieses „unverfälschte“ Bild ist in Abbildung 4 ganz links zu sehen. Die Matrix  $A$  ist hier die PSF-Matrix auch aus dem „satellite“-Beispiel und hat dieselbe Größe wie  $x$ .

Im mittleren Bild von Abbildung 4 ist das geblurrte und verrauschte Bild  $b$  zu sehen, welches auch wieder durch  $b = Ax + e$  berechnet wird. Das Bild erscheint deutlich unschärfer und verwischter als das Original. Dennoch ist der Unterschied zwischen beiden Bildern eher gering. Im Gegensatz dazu, kann man das Originalbild (Abbildung 4 rechts) nicht mehr erahnen. Hier wurde wieder versucht das Original mit Hilfe von Gleichung (1) wiederherzustellen, was durch das Nichteinbeziehen des Rauschens völlig unmöglich ist. [2]

## 2.3 Verfahren der konjugierten Gradienten

Das Verfahren der konjugierten Gradienten (kurz CG-Verfahren, aus dem Englischen von conjugate gradients) ist das wohl effizienteste Iterationsverfahren zur Lösung linearer Gleichungssysteme. Hier wird die quadratische Funktion

$$\phi(x) = \frac{1}{2}x^T Ax - x^T b, \quad x, b \in \mathbb{R}^n, A \in \mathbb{R}^{n \times n} \quad (3)$$

betrachtet. Während  $x$  und  $b$  Vektoren sind, ist  $A$  eine symmetrische und positiv definite Matrix. Durch diese Eigenschaften der Matrix  $A$ , besitzt  $\phi(x)$  ein eindeutiges Minimum bei  $x = \hat{x}$ . Das Minimum erhält man durch ableiten und Null setzen von  $\phi(x)$ , dies ist zugleich auch die Lösung des linearen Gleichungssystems  $Ax = b$ .

Man kann durch die Eigenschaften von  $A$  auch das Skalarprodukt

$$\langle x, y \rangle_A = x^T Ay, \quad x, y \in \mathbb{R}^n \quad (4)$$

sowie die induzierte Norm

$$\|x\|_A = \sqrt{x^T Ax}, \quad x \in \mathbb{R}^n \text{ und } x^T Ax \geq 0, \quad (5)$$

definieren.

Nimmt man die Differenz der Funktionen, der errechneten Lösung  $x$  und der exakten Lösung  $\hat{x}$ ,

$$\phi(x) - \phi(\hat{x}) = \frac{1}{2}(x - \hat{x})^T A(x - \hat{x}) = \frac{1}{2}\|x - \hat{x}\|_A^2 \quad (6)$$

dann erhält man ein Fehlermaß für den Abstand dieser Lösungen, welches in jedem Iterationsschritt reduziert werden soll. Nun versucht man die Funktion  $\phi(x)$  aus (3) zu minimieren, indem zu jeder Iterierten  $x_k$  eine Suchrichtung  $d_k \neq 0$  bestimmt wird, um so die nächste Iterierte  $x_{k+1}$  mit

$$x_{k+1} = x_k + \alpha d_k \text{ und } \frac{1}{2}\|x_{k+1} - \hat{x}\|_A^2 < \frac{1}{2}\|x_k - \hat{x}\|_A^2 \quad (7)$$

zu berechnen. Im Weiteren folgt in Abhängigkeit von  $\alpha$  für die Funktion

$$\phi(x_{k+1}) = \phi(x_k) + \alpha d_k^T A x_k + \frac{1}{2}\alpha^2 d_k^T A d_k - \alpha d_k^T b.$$

Nach Differentiation dieser Gleichung nach  $\alpha$  und dem Null setzen erhält man die Schrittweite  $\alpha_k$  mit

$$\frac{d\phi}{d\alpha} = 0 \implies \alpha_k = \frac{r_k^T d_k}{d_k^T A d_k}, \quad \text{mit } r_k = b - A x_k \text{ und } d_k^T A d_k \neq 0. \quad (8)$$

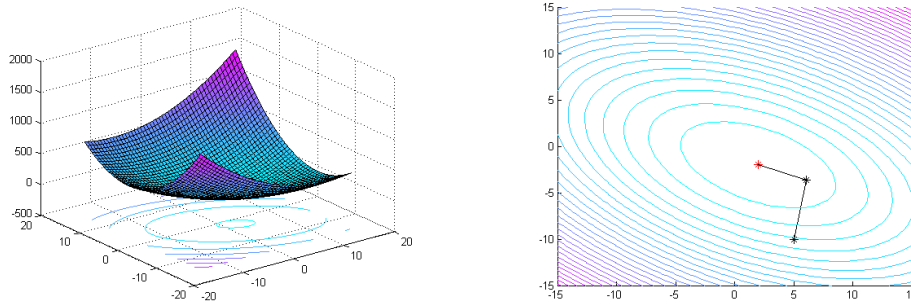


Abbildung 5: Links ist der Graph von  $\phi(x)$  abgebildet und rechts im Bild sind die Höhenlinien mit den beiden benötigten Iterationsschritten zu sehen. Der Mittelpunkt der Ellipsen (hier rot) ist das gesuchte Minimum  $\hat{x}$ .

Jetzt muss noch die Suchrichtung  $d_k$  festgelegt werden. Für  $k = 0$  ist sie gleich dem negativen Gradienten von  $\phi$

$$d_0 = -\nabla\phi(x_0) = b - Ax_0 = r_0, \quad (9)$$

also gleich der Richtung des steilsten Abstieges, welcher wiederum mit dem Residuum übereinstimmt.

Dies ist aber nicht immer die beste Wahl, daher berechnet man alle weiteren Suchrichtungen auf andere Weise. Der Graph von  $\phi(x)$  zeigt ein elliptisches Paraboloid und die Höhenlinien von  $\phi(x)$  sind Ellipsen, wie in Abbildung 5 zu erkennen ist. Der gemeinsame Mittelpunkt dieser Ellipsen ist das gesuchte Minimum  $\hat{x}$ . Betrachtet man die Höhenlinien von  $\phi(x)$  jetzt in der  $\|\cdot\|_A$ -Geometrie, so beschreiben diese Kreise. Die erste Suchrichtung  $d_0$  ist die Tangente eines dieser Kreise. Deshalb soll die nächste Suchrichtung orthogonal zu der vorherigen sein, sodass sie direkt durch den Mittelpunkt geht und so zum Minimum  $\hat{x}$  führt. Daraus folgt, dass

$$\langle d_{k+1}, d_k \rangle_A \stackrel{!}{=} 0$$

gelten soll.

$$\begin{aligned} \langle d_{k+1}, d_k \rangle_A &\stackrel{(4)}{=} d_{k+1}^T A d_k \\ &\stackrel{(10)}{=} (r_{k+1} + \beta_k d_k)^T A d_k \\ &= r_{k+1}^T A d_k + \beta_k d_k^T A d_k \\ &\stackrel{(11)}{=} r_{k+1}^T A d_k - \frac{r_{k+1}^T A d_k}{d_k^T A d_k} d_k^T A d_k \\ &= r_{k+1}^T A d_k - r_{k+1}^T A d_k \\ &= 0 \end{aligned}$$

führt uns zu der neuen Suchrichtung  $d_{k+1}$ , die wir mit

$$d_{k+1} = r_{k+1} + \beta_k d_k \quad (10)$$

berechnen, wobei

$$\beta_k = -\frac{r_{k+1}^T A d_k}{d_k^T A d_k} . \quad (11)$$

Die Gleichungen (8) und (11) sind nur dann wohldefiniert, wenn  $d_{k+1} \neq 0$ . Dies tritt aber nur dann ein, wenn  $d_k$  und  $r_{k+1}$  linear abhängig sind, da  $d_k$  aber orthogonal zum Residuum  $r_k$  ist, ist das nur dann der Fall, wenn  $r_{k+1} = 0$  und somit die Lösung  $x = \hat{x}$  gefunden wurde.

Diese Suchrichtungen nennt man wegen der speziellen Orthogonalitätsbedingungen  $\langle d_{k+1}, d_k \rangle_A = 0$  zueinander A-konjugiert. Daher spricht man auch vom Verfahren der konjugierten Gradienten.

Für die Implementierung des CG-Verfahrens schreibt man die Gleichungen für  $\alpha$  und  $\beta$  noch in eine rechengünstigere Variante um. Bei der Berechnung von  $\alpha_k$  ersetzen wir den Zähler. Dadurch das

$$r_k^T d_k = r_k^T (r_k + \beta_{k-1} r_k^T d_{k-1}) = r_k^T r_k$$

gilt, können wir jetzt

$$\alpha_k = \frac{\|r_k\|_2^2}{d_k^T A d_k} \quad (12)$$

schreiben. Bei der Gleichung für  $\beta_k$  formulieren wir ebenfalls den Zähler um, indem wir für

$$r_{k+1}^T A d_k = \frac{1}{\alpha_k} (r_{k+1}^T r_k - r_{k+1}^T r_{k+1}) = -\frac{1}{\alpha_k} \|r_{k+1}\|_2^2 = -\frac{\|r_{k+1}\|_2^2}{\|r_k\|_2^2} d_k^T A d_k$$

schreiben. Dann erhalten wir

$$\beta_k = \frac{\|r_{k+1}\|_2^2}{\|r_k\|_2^2} \quad (13)$$

als neue Formel. So spart man sich einige Matrix-Vektor-Multiplikationen und muss nur auf die schon bekannten Residuen zurückgreifen. Der Rechenaufwand minimiert sich hierdurch.

Der Gesamtaufwand des CG-Verfahrens ist ähnlich dem von anderen Verfahren, wie beispielsweise für Gesamt- und Einzelschrittverfahren [1]. Jedoch wird dieser geringer, je weniger Einträge von  $A$  von Null verschieden sind. Die Lösung, also das Minimum  $\hat{x}$ , wird dabei immer in höchstens  $n$  Schritten gefunden. Im Beispiel aus Abbildung 5, mit der Dimension  $n = 2$ , sieht

man, dass das Minimum nach genau zwei Schritten gefunden wird und im Mittelpunkt der Ellipsen liegt. [1, 6]

Zusammengefasst sieht der Algorithmus des CG-Verfahrens dann so aus :

---

**Algorithm 1:** CG-Verfahren

---

**input** :  $A \in \mathbb{R}^{n \times n}$  symmetrisch, positiv definit,  $b \in \mathbb{R}^n$  und  $x_0 \in \mathbb{R}^n$   
beliebig

**output:** Minimum  $x$

**begin**

$$r_0 = b - Ax_0$$

$$d_0 = r_0$$

**for**  $k = 0, 1, 2, \dots$  **do**

$$\alpha_k = \frac{r_k^T r_k}{d_k^T A d_k}$$

$$x_{k+1} = x_k + \alpha_k d_k$$

$$r_{k+1} = b - Ax_{k+1}$$

$$\beta_k = -\frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$$

$$d_{k+1} = r_{k+1} + \beta_k d_k$$

**end**

**end**

---

### 3 Regularisierungsverfahren

Regularisierungsverfahren werden bei schlecht gestellten Problemen angewendet. Diese Probleme haben nach Definition entweder gar keine Lösung, unendlich viele Lösungen oder hängen nicht stetig von den Daten ab [8]. Tritt mindestens eine dieser Eigenschaften auf, dann ist das Problem schlecht gestellt und somit nicht mehr eindeutig lösbar. Die Regularisierungsverfahren haben die Aufgabe, solch ein schlecht gestelltes Problem in ein gut gestelltes zu überführen.

Kommen wir noch einmal auf unser Beispiel am Ende des letzten Abschnittes zurück. Auch hier haben wir es mit einem schlecht gestellten Problem zu tun. In der Regel ist das Rauschen eine unbekannte Größe, somit erhalten wir für Gleichung (2) mehr als eine Lösung, was eine Eigenschaft von schlecht gestellten Problemen ist. Um dieses Problem zu überwinden, verwendet man ein Regularisierungsverfahren, das eine stabile Approximation von  $x$  mit Hilfe eines Ersatzproblems erzeugt. Das heißt, dass eine einzelne, brauchbare und stabile, nicht wie in Abbildung 3 oszillierende, Lösung herausgefiltert wird.

Bei den Regularisierungsverfahren unterscheidet man direkte, iterative und hybride Methoden. Wir gehen hier insbesondere auf die direkten Verfahren ein, wie zum Beispiel die TSVD Regularisierung oder die Tikhonov Regularisierung. Desweiteren stellen wir noch eine hybride Methode, die Lanczos-Hybrid Methode vor.

Die Effektivität einer Regularisierungsmethode hängt strikt von der geeigneten Wahl des Regularisierungsparameters  $\lambda$  ab. Wählt man  $\lambda$  zu klein (z.B. TSVD) bzw. zu groß (z.B. Tikhonov), dann wird die Lösung zu glatt und man entfernt sich zu stark von den tatsächlichen Daten, obwohl das Problem stabil lösbar ist. Wählt man  $\lambda$  dagegen zu groß (z.B. TSVD) bzw. zu klein (z.B. Tikhonov), dann ist es genau andersherum, das Problem ist eher wenig stabil, dafür besteht aber sehr hohe Datentreue. Daher gibt es verschiedene Techniken dieses  $\lambda$  zu finden, zum Beispiel mit dem Diskrepanzprinzip, der L-Kurven Methode oder GCV. Wir werden hier aber nur auf die beiden letzteren näher eingehen. [2, 5, 7]

#### 3.1 TSVD Regularisierung

Die simpelste direkte Regularisierungsmethode ist die truncated SVD (TSVD) Regularisierung oder auch abgeschnittene Singulärwertzerlegung. Die schlecht konditionierte Matrix  $A$  kann, wie gezeigt, in  $A = U\Sigma V^T$  zerlegt werden. Die

Berechnung der Originaldaten erfolgt mit Gleichung (1). Bei dieser Berechnung kann es, wie schon erwähnt, zur Verstärkung des Rauschens kommen, in dem man durch sehr kleine Singulärwerte dividiert. Nun versucht man bei diesem Verfahren die sehr stark verrauschten, wiederhergestellten Daten herauszufiltern. Dies wird realisiert, in dem man bei Gleichung (1) den Teil mit den zu kleinen Singulärwerten herausnimmt. Der hintere Teil der Gleichung

$$x = \sum_{i=1}^{\lambda} \frac{u_i^T b v_i}{\sigma_i} + \sum_{i=\lambda+1}^n \frac{u_i^T b v_i}{\sigma_i}, \quad (14)$$

der von dem Rauschen dominiert wird, fällt somit also weg. Dann erhalten wir

$$x_{\lambda} = \sum_{i=1}^{\lambda} \delta_i \frac{u_i^T b v_i}{\sigma_i}, \quad \text{wobei } \delta_i = \begin{cases} 1 & , \text{ falls } i \leq \lambda \\ 0 & , \text{ falls } i > \lambda \end{cases} \text{ ist.} \quad (15)$$

$\lambda$  ist der sogenannte Regularisierungsparameter. Die in Gleichung (14) verwendete Matrix

$$A_{\lambda} = \sum_{i=1}^{\lambda} u_i \sigma_i v_i^T$$

hat jetzt nur noch den Rang  $\lambda \leq n$ . (15) ist die Lösung des Minimierungsproblems

$$\min_x \|A_{\lambda} x - b\|_2 \text{ in Abhängigkeit von } \min_x \|x\|_2.$$

Die TSVD Regularisierung ist zwar keine exakte Zerlegung der Originalmatrix  $A$  mehr, dennoch ist  $A_{\lambda}$  die bestmögliche Approximation unter allen Matrizen mit Rang  $\lambda$  bezüglich der verwendeten 2-Norm.

Wie bei allen anderen Regularisierungsverfahren auch, spielt die Wahl des Regularisierungsparameters eine sehr wichtige Rolle. Wird  $\lambda$  zu klein gewählt, dann ist die Lösung  $x_{\lambda}$  zu glatt und es werden zu viele Daten herausgefiltert. Dies hat den Verlust von Informationen zur Folge. Ist  $\lambda$  dagegen zu groß, dann ist die Lösung  $x_{\lambda}$  zu verrauscht, sodass immer noch ein Teil des Rauschens in den Daten verblieben ist. Es muss also ein passabler Kompromiss zwischen beiden Varianten gefunden werden. Wie man dies realisieren kann, wird im Kapitel 3.3 noch genauer gezeigt. Allerdings wird der gefilterte Teil immer auch eine gewisse Menge Daten enthalten, die dann verloren gehen. Andersherum enthält auch der andere Teil immer noch eine kleine Menge des Rauschens. [2, 7]

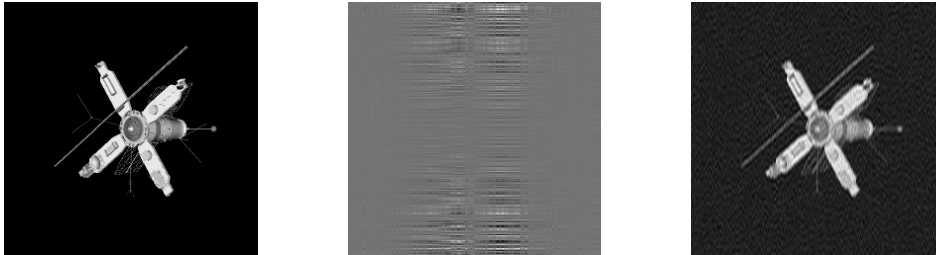


Abbildung 6: Links ist das Originalbild. In der Mitte sieht man das Ergebnis einer naiven Rückrechnung, wenn das Originalbild vorher gefiltert und mit einem Rauschen belegt wurde. Im rechten Bild wurde das geblurred Bild mit der TSVD-Regularisierung bestmöglich wiederhergestellt. Es ist sehr leicht zu erkennen, dass die Wiederherstellung des Originals sehr viel besser mit der TSVD-Regularisierung klappt, als komplett ohne Regularisierung. Die Ausgangswerte  $x$ ,  $A$  und  $e$  sind wie im zweidimensionalen Beispiel aus Abschnitt 2.2.

### 3.2 Tikhonov Regularisierung

Die Tikhonov Regularisierung ist das meist verbreitetste und auch bekannteste Regularisierungsverfahren für schlecht gestellte Probleme. Sie ist neben der TSVD eine weitere direkte Methode.

Wir nehmen unser lineares Gleichungssystem  $Ax + e = b$ , welches ein Rauschen  $e$  enthält und schreiben es in ein Minimierungsproblems um. Somit erhalten wir

$$\min_x \{ \|b - Ax\|_2^2 + \lambda^2 \|Lx\|_2^2 \} . \quad (16)$$

$L$  stellt den Regularisierungsoperator und  $\lambda$  den Regularisierungsparameter dar, wobei  $L \in \mathbb{R}^{n \times n}$  eine Matrix und  $\lambda \in \mathbb{R}$  ein Skalar ist. Für gewöhnlich erfüllt  $\lambda$  die Bedingung  $\sigma_n \leq \lambda \leq \sigma_1$ , mit  $\sigma_n$  als kleinsten und  $\sigma_1$  als größten Singulärwert von  $A$ . Der vordere Teil  $\|b - Ax\|_2^2$  stellt die Residuumsnorm dar und der hintere Teil  $\|Lx\|_2^2$  repräsentiert die  $(L^T L)$ -Norm der Lösung  $x$ . Bei der Standard-Tikhonov-Regularisierung ist  $L$  beispielsweise die  $n$ -te Einheitsmatrix, also  $L = E$  und somit ist

$$\min_x \{ \|b - Ax\|_2^2 + \lambda^2 \|Ex\|_2^2 \} , \quad (17)$$

das zu lösende Minimierungsproblem.

Wir wählen hier nun auch den Regularisierungsoperator  $L$  als  $n$ -te Einheitsmatrix. Seien  $u_i$  und  $v_i$  jeweils die  $i$ -te Spalte von  $U$  bzw.  $V$ , hervorgehend



aus der Singulärwertzerlegung von  $A$ , dann formen wir Gleichung (16)

$$\min_x \{\|b - Ax\|_2^2 + \lambda^2 \|Ex\|_2^2\} = \min_x \{b^T b - 2x^T A^T b + x^T A^T A x + \lambda^2 x^T E x\}$$

um, in dem wir sie nach  $x$  ableiten

$$\frac{d}{dx} \{b^T b - 2x^T A^T b + x^T A^T A x + \lambda^2 x^T E x\} = -A^T b + A^T A x + \lambda^2 E x$$

und dann Null setzen

$$0 = -A^T b + A^T A x + \lambda^2 E x .$$

Jetzt können wir diese Gleichung weiter umformen, um so die Gleichung zur Berechnung von  $x_\lambda$

$$\begin{aligned} 0 &= -A^T b + A^T A x + \lambda^2 E x \\ 0 &= -V \Sigma U^T b + V \Sigma U^T U \Sigma V^T x + \lambda^2 E x \\ 0 &= -V \Sigma U^T b + V \Sigma^2 V^T x + \lambda^2 E x \\ V \Sigma U^T b &= (V \Sigma^2 V^T + \lambda^2 E) x \\ x &= (V \Sigma^2 V^T + \lambda^2 E)^{-1} V \Sigma U^T b \\ x_\lambda &= \sum_{i=1}^n \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \frac{u_i^T b v_i}{\sigma_i} \end{aligned} \tag{18}$$

zu erhalten.

Den vorderen Term  $f_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \in [0, 1]$  nennt man Tikhonov-Filterfaktor. Wenn dieser Filterfaktor 1 ist, dann erhalten wir Gleichung (1) und somit ist  $x_\lambda$  die Lösung von  $\min_x \|b - Ax\|_2^2$ . [2, 5, 9]

### 3.3 Wahl des Regularisierungsparameters

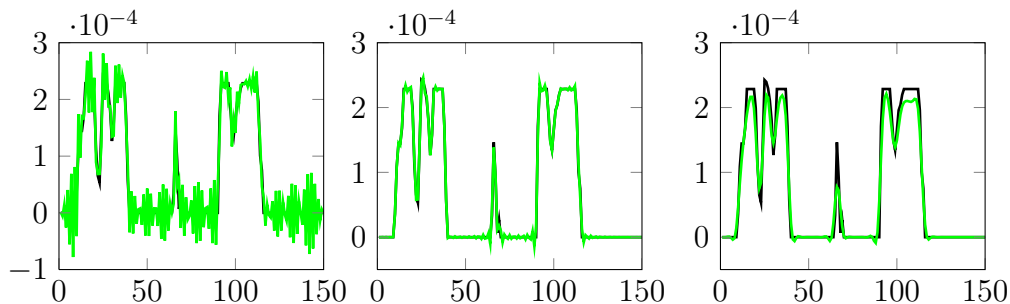


Abbildung 7: Originaldaten  $x$  (schwarze Linie) und die durch die Tikhonov Regularisierung errechneten Daten  $x_\lambda$  (grüne Linie): Im linken Bild ist  $\lambda$  zu klein gewählt, die Lösung ist zwar datentreu, aber wenig stabil. Im mittleren Bild ist  $\lambda$  nahezu optimal, die Originaldaten werden fast vollständig wiederhergestellt. Dagegen ist die Lösung im rechten Bild zu glatt und wenig datentreu, welches durch ein zu großes  $\lambda$  hervorgerufen wird.

Wie schon erwähnt, hängt die Effektivität jeder Regularisierungsmethode strikt von der geeigneten Wahl des Regularisierungsparameters  $\lambda$  ab.  $\lambda$  spielt eine sehr wichtige Rolle bei der Qualität der Lösung. Wählt man  $\lambda$  zu klein (TSVD) bzw. zu groß (Tikhonov), dann wird die Lösung zu glatt und man entfernt sich zu stark von den tatsächlichen Daten, obwohl das Problem stabil lösbar ist. Im rechten Bild von Abbildung 7 kann man dieses Problem erkennen. Hier wurde mit Tikhonov regularisiert und mit  $\lambda = 0.3$  ein zu großer Parameter gewählt. Man kann deutlich erkennen, dass die errechneten Daten  $x_\lambda$  (rote Linie) unter den Originaldaten  $x$  (schwarze Linie) verlaufen und somit noch zu weit von diesen entfernt sind. Außerdem sieht man auch, dass die rote Linie noch wesentlich glatter verläuft als die schwarze Linie der Originaldaten.

Wählt man  $\lambda$  dagegen zu groß (TSVD) bzw. zu klein (Tikhonov), dann ist es genau andersherum, das Problem ist eher wenig stabil, dafür besteht dann sehr hohe Datentreue. Dieses Problem kann man im linken Bild von Abbildung 7 erkennen. Auch hier wurde wieder mit Tikhonov regularisiert, aber diesmal wurde mit  $\lambda = 5 \cdot 10^{-8}$  ein deutlich zu kleiner Parameter gewählt. Die rote Linie der errechneten Daten  $x_\lambda$  liegt fast vollständig über der schwarzen Linie der Originaldaten  $x$ , was für die sehr hohe Datentreue spricht. Allerdings erkennt man vor allem in der zweiten Hälfte des Bildes deutlich, dass der Graph der  $x_\lambda$  stark oszilliert.

Im mittleren Bild ist  $\lambda = 0.0001$  nahezu optimal. Man kann das daran er-

kennen, dass die errechneten  $x_\lambda$  relativ genau auf den originalen  $x$  liegen.  
 Im 2D-Fall verhält es sich genau wie im 1D-Fall, den wir eben durch Abbil-

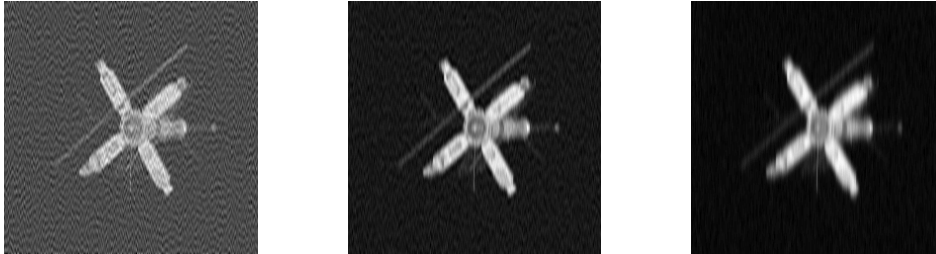


Abbildung 8: Wir haben wieder durch die Tikhonov Regularisierung erzeugte Daten. Im linken Bild ist  $\lambda$  zu klein gewählt, sodass ein wenig stabiles, aber datentreues Bild entsteht. Im mittleren Bild ist  $\lambda$  nahezu optimal und das Originalbild wird gut approximiert. Das rechte Bild ist stabil, dafür aber weniger datentreu, sodass es zur Unschärfe kommt.

dung 7 beschrieben haben. In Abbildung 8 sehen wir im linken Bild wieder ein durch die Tikhonov Regularisierung erstelltes Bild mit zu klein gewähltem  $\lambda = 0.08$ . Die hohe Datentreue erkennt man hier an der ausgeprägten Schärfe des Satelliten und die oszillierenden Daten daran, dass das Bild „verschneit“ erscheint. Das heißt, neben den dunklen Pixeln liegen immer helle und umgekehrt, da sich durch das Oszillieren die Farbe von hell auf dunkel und andersherum ändert. Im mittleren Bild hat  $\lambda = 0.8$  nahezu die richtige Größe, da das Originalbild gut approximiert wird. Im rechten Bild ist  $\lambda = 8$  dagegen wieder zu groß, sodass der Satellit eher als unscharf erscheint, was auf die Abweichung von den Originaldaten zurückzuführen ist. Dagegen ist das Bild an sich nicht „verschneit“ und daher eher stabil. [2, 5, 7]

### 3.3.1 GCV-Verfahren

Um ein geeignetes Mittel zwischen Datentreue und Stabilität zu generieren, benutzen wir zur Ermittlung von  $\lambda$  das GCV-Verfahren. GCV ist die Abkürzung von generalized cross validation.

Man benötigt keine Angaben über die Messfehler und der Hauptgedanke hierbei ist, dass man mit einem geeigneten Wert für  $\lambda$  fehlende Datenpunkte vorhersagen kann. Hierfür muss man die GCV-Funktion

$$G(\lambda) = \frac{n\|(E - AA_\lambda^\dagger)b\|_2^2}{\left(\text{Spur}(E - AA_\lambda^\dagger)\right)^2} \quad (19)$$

minimieren. Die Pseudoinverse von  $A$ ,  $A_\lambda^\dagger = (A^T A + \lambda^2 E)^{-1} A^T$ , liefert die regularisierte Lösung  $x_\lambda = A_\lambda^\dagger b$ . Ersetzt man die Matrix  $A$  wieder durch die Singulärwertzerlegung, dann erhält man

$$G^{Tikhonov}(\lambda) = \frac{n \sum_{i=1}^n \left( \frac{u_i^T b}{\sigma_i^2 + \lambda^2} \right)^2}{\left( \sum_{i=1}^n \frac{1}{\sigma_i^2 + \lambda^2} \right)^2} . \quad (20)$$

Hiermit kann man dann den Regularisierungsparameter für die Tikhonov Regularisierung berechnen. Die Berechnung des Regularisierungsparameters für die TSVD Regularisierung sieht etwas anders aus, da man  $A$  in Gleichung (19) hier nicht durch die vollständige Singulärwertzerlegung ersetzen kann. So entsteht

$$G^{TSVD}(\lambda) = \frac{\sum_{i=\lambda+1}^n (u_i^T b)^2}{(n - \lambda)^2} . \quad (21)$$

Dies ist eine sehr rechengünstige Approximation, vor allem für Standard - Minimierungsprobleme. Sie ist zudem auch sehr effektiv, falls die Singulärwertzerlegung von  $A$  ohne Probleme möglich ist. [5, 7, 9, 10]

### 3.3.2 L-Kurven Methode

Ein anderes Verfahren zur Bestimmung des Regularisierungsparameters  $\lambda$  ist die L-Kurven Methode. Die sogenannte L-Kurve ist ein Plot aller innerhalb eines Intervalls möglichen Parameter  $\lambda$ , bei dem die Norm der regularisierten Lösung  $\|Lx_\lambda\|_2$  gegen die korrespondierende Residuumsnorm  $\|Ax_\lambda - b\|_2$  geplottet wird. Es wird der Kompromiss zwischen der Minimierung dieser beiden Normen aufgezeigt. So wie der Name schon sagt, zeigt der Plot eine Kurve in L-Form (Abbildung 9, Mitte), wenn man logarithmierte Skalen verwendet. Der vertikale Teil der Kurve trennt sich deutlich vom horizontalen Teil ab, so dass eine „Ecke“ entsteht. Der vertikale Teil wird durch  $\log \|Lx_\lambda\|_2$  bestimmt, während der horizontale Teil von  $\log \|Ax_\lambda - b\|_2$  bestimmt wird. Der optimale Regularisierungsparameter  $\lambda$  liegt ganz in der Nähe der größten Krümmung der Kurve, also der „Ecke“.

In Abbildung 9 sieht man in der Mitte eine typische L-Kurve mit den Achsen eingeteilt wie eben erwähnt.  $L$  ist in dem Beispiel die Einheitsmatrix mit der Größe von  $x$ . Die Ausgangswerte  $x$ ,  $A$  und  $e$  sind wieder wie im zweidimensionalen Beispiel aus Abschnitt 2.2. Für die Wiederherstellung des Originals aus dem geblurrten und verrauschten Bild  $b$  wird die Tikhonov Regularisierung wie in Gleichung (18) verwendet.

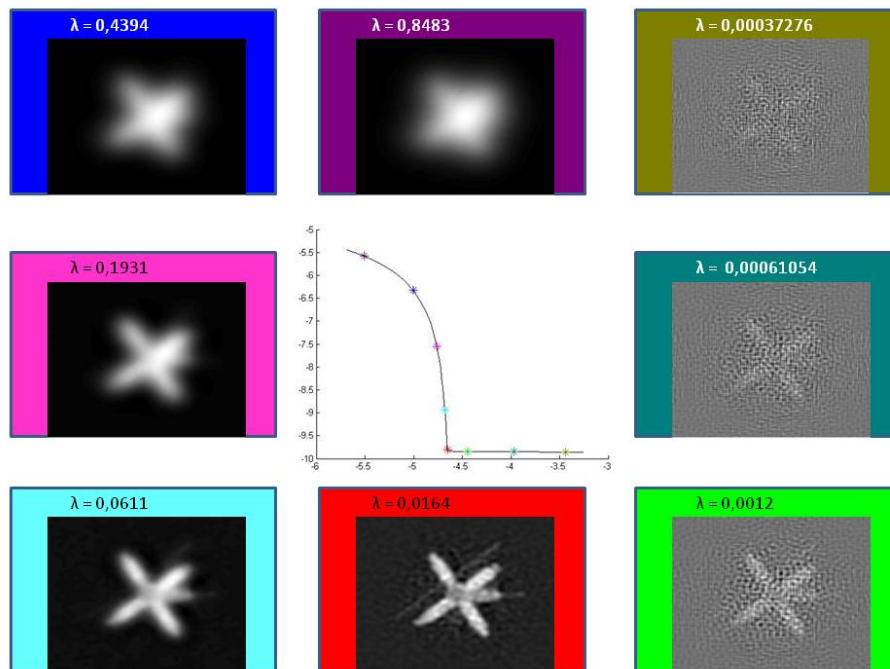


Abbildung 9: In der Mitte sieht man eine typische L-Kurve. Auf der x-Achse ist  $\log \|Ax_\lambda - b\|_2$  abgetragen und auf der y-Achse  $\log \|Lx_\lambda\|_2$ , wobei  $L$  die Einheitsmatrix ist. Die bunten Punkte auf der Kurve repräsentieren Werte für verschiedene Regularisierungsparameter  $\lambda \in [e^{-3.5}, 1]$ . Außen sind die jeweiligen Bilder angeordnet, die zu den gleichfarbigen Punkten auf der Kurve gehören. Wenn man rechts oben bei dem kleinsten Regularisierungsparameter startet und dann im Uhrzeigersinn weitergeht, dann kann man erkennen, wie die Bilder erst besser werden, sich also immer mehr dem Original annähern und dann wieder schlechter und damit unschärfer werden.

Zu jedem bunten Punkt auf der L-Kurve wird das dazugehörige Bild umliegend mit Angabe des Regularisierungsparameters  $\lambda \in [e^{-3.5}, 1]$  gezeigt. In der rechten oberen Ecke ist das Bild zu sehen, das zu dem hier kleinsten benutzten Parameter  $\lambda = 0.00037276$  gehört. Dieses  $\lambda$  ist noch deutlich zu klein und der Satellit ist kaum sichtbar, da das gesamte Bild noch deutlich von Rauschen überlagert ist.

Geht man dann im Uhrzeigersinn weiter, wird das  $\lambda$  größer und der Satellit ist immer deutlicher und besser zu erkennen. Mehr Details kommen zum Vorschein und die Lösung wird von Schritt zu Schritt stabiler. Die Lösung oszilliert weniger und dadurch wechseln sich helle und dunkle Pixel nicht mehr so schnell ab, bis der Hintergrund komplett schwarz ist. Wie schon erwähnt,

muss ein Mittelmaß zwischen Datentreue und Stabilität gefunden werden. Im Gegensatz zu dem ersten Bild rechts oben, ist bei dem Bild oben in der Mitte der Regularisierungsparameter  $\lambda = 0.8483$  zu groß. Diese Lösung ist wesentlich stabiler, was man am fast komplett schwarzen Hintergrund erkennen kann. Dagegen besteht keine so hohe Datentreue wie bei dem Bild rechts daneben. Der Satellit in der Mitte erscheint wesentlich verschwommener und unschärfer. Es können keine Details mehr wahrgenommen werden. Alle anderen Bilder zeigen die Abstufungen zwischen einem viel zu kleinen  $\lambda$ , was mit hoher Datentreue, aber geringer Stabilität einhergeht und einem viel zu großen  $\lambda$ , was für geringe Datentreue, aber hohe Stabilität steht. Das perfekte Mittelmaß bezeichnet das rote Bild unten in der Mitte mit einem verwendeten Regularisierungsparameter von  $\lambda = 0,0164$ . Dieser Parameter wurde gefunden, in dem die „Ecke“ der L-Kurve lokalisiert wurde, was hier nur mit Augenmaß geschah. Der Satellit ist hier am schärfsten und die meisten Details sind erkennbar, während die Lösung dennoch relativ stabil ist. [2, 5]

### 3.3.3 Vergleich von GCV und L-Kurven Methode

Was uns jetzt natürlich noch interessiert, ist ein Vergleich der beiden Verfahren. Liefert GCV oder die L-Kurven Methode das bessere Ergebnis? Ich möchte hier versuchen eine Antwort zu finden. Dafür nehmen wir wieder unser bekanntes Beispiel des Satelliten.

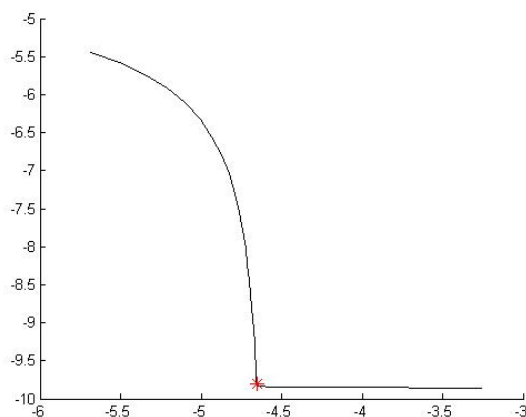


Abbildung 10: Hier ist die L-Kurve zu sehen mit  $\lambda$ -Werten zwischen  $3.16 \cdot 10^{-4}$  und 1. Bei der stärksten Krümmung liegt der gesuchte  $\lambda$ -Wert, dieser beträgt hier 0.0164 und ist mit einem roten Stern gekennzeichnet.

Zur Bestimmung des besten Wertes der L-Kurven Methode nehmen wir eine

L-Kurve wie im Abschnitt zuvor erklärt, bei der sich der bestmögliche Wert für  $\lambda$  nahe der größten Krümmung befindet. Wir werden diesen Punkt so gut wie möglich grafisch bestimmen.

Hierfür erstellen wir uns aus dem Beispiel eine L-Kurve für  $\lambda$ -Werte von  $\approx 3.16 \cdot 10^{-4} - 1$ , wie in Abbildung 10 zu sehen ist. Der Punkt an der Stelle mit der stärksten Krümmung ist rot markiert und liegt bei  $\lambda = 0.0164$ .

Beim GCV-Verfahren ermitteln wir den  $\lambda$ -Wert nicht grafisch, sondern rechnerisch. Hier nehmen wir die Funktion  $[x, \lambda] = Tikhonov(A, b)$ , ebenfalls aus den RestoreTools, zur Hilfe. Für die Eingabe benötigen wir das geblurrte Bild  $b$  und die Blurringmatrix in der Form  $A = psfMatrix(PSF)$ . Ausgegeben wird uns dann nicht nur der Regularisierungsparameter  $\lambda$ , sondern auch das mit der Tikhonov Regularisierung rekonstruierte Bild  $x$ , welches wir für den Vergleich noch brauchen werden. Der so ermittelte Regularisierungsparameter beträgt  $\lambda = 0.0099$ .

Schon bei dem Regularisierungsparameter unterscheidet sich die L-Kurven Methode von dem GCV-Verfahren. Nun wissen wir aber allein von der Präsenz dieser Werte nicht, welcher von beiden der bessere ist. Daher wollen wir nun die mit der Tikhonov Regularisierung rekonstruierten Bilder vergleichen. Bei dem GCV-Verfahren wurde das Bild gleich mitberechnet und dies tun wir für die L-Kurven Methode auf gleiche Weise. Dies machen wir auch mit der Funktion  $x = Tikhonov(A, b, 0.0164)$ , nur geben wir hier den Regularisierungsparameter mit vor, sodass innerhalb der Methode kein Parameter mehr berechnet wird.

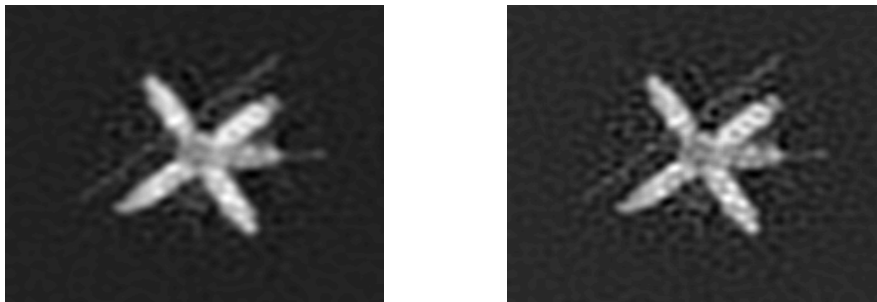


Abbildung 11: Auf dem linken Bild ist das rekonstruierte Bild mit Hilfe der L-Kurven Methode dargestellt. Das rechte Bild zeigt die Rekonstruktion mit Hilfe des GCV-Verfahrens. Beide Bilder sind deutlich besser als das geblurrte Bild  $b$ .

Wenn man jetzt beide Bilder vergleicht, dann sieht man schnell, dass beide

Rekonstruktionen deutlich besser sind als das geblurrte Bild  $b$ . Jedoch ist ein großer Unterschied zwischen beiden Bildern nicht zu erkennen. Das linke Bild von Abbildung 11 gehört zu der L-Kurven Methode. Der Satellit in der Mitte ist sehr deutlich zu erkennen, aber noch etwas unscharf und man kann nicht alle Details erkennen. Dafür ist der Hintergrund aber relativ stabil schwarz. Die Rekonstruktion mit Hilfe des GCV-Verfahrens sieht man im rechten Bild von Abbildung 11. Auch hier ist der Satellit in der Mitte wieder deutlich zu erkennen. Allerdings ist dieser hier schärfer abgebildet und man kann einige Details deutlicher erkennen als im linken Bild. Jedoch ist der Hintergrund, vor allem um den Satelliten, eher instabil und sieht sehr verpixelt aus und nicht durchgehend schwarz.

Beide Bilder sind also nicht perfekt und haben ihre Schwachpunkte. Deswegen muss man abwägen, was wichtiger ist. Daher ist das rechte Bild, dass mit Hilfe des GCV-Verfahrens hergestellt wurde in meinen Augen besser als das andere. Das wichtigste auf dem Bild ist der Satellit und daher kann man die Qualität des Hintergrundes gerne zugunsten der Detailtreue vernachlässigen. Genauso wäre es mit anderen Bildmotiven auch, der Vordergrund ist in der Regel am wichtigsten.

Jetzt möchte ich aber nicht im Allgemeinen darauf schließen, dass das GCV-Verfahren immer besser als die L-Kurven Methode ist. Bei anderen Beispielen könnte es auch andersherum sein. Zumal wir hier auch berücksichtigen müssen, dass wir die L-Kurven Methode nur grafisch gelöst haben. Fehler können hier also schnell passieren und wir können nicht hundertprozentig sicher sein, dass auch der beste Regularisierungsparameter gewählt wurde, auch wenn ich versucht habe sehr sorgfältig zu sein. Daher kann man davon ausgehen, dass beide Methoden ungefähr gleich gut sind und ähnlich gute Ergebnisse erzielen.



### 3.4 Lanczos-Hybrid-Methode

Die Lanczos-Hybrid-Methode ist, wie der Name schon sagt, eine hybride Regularisierungsmethode. Die hybriden Regularisierungsmethoden bestehen aus zwei Teilen, der äußeren und der inneren Regularisierung. Ist die Dimension der Matrix  $A$  zu groß, ist die Singulärwertzerlegung mit einem zu großen Rechenaufwand verbunden. Durch die Lanczos-Hybrid-Methode soll dies aber dennoch möglich gemacht werden, in dem das Gesamtproblem auf einen Teilraum mit deutlich kleinerer Dimension projiziert wird. Diese Projektion gehört zu der äußeren Regularisierung. [7]

Diese Methode kann effektiv das semi-konvergente Verhalten stabilisieren, welches charakteristisch für iterative Methoden im Zusammenhang mit schlecht gestellten Problemen ist. Um solche großen, schlecht gestellten, inversen Probleme effizient zu lösen, werden iterative Verfahren (wie zum Beispiel die Lanczos-Bidiagonalisierung) mit direkten Regularisierungsverfahren (wie beispielsweise die Tikhonov Regularisierung) kombiniert und für die projizierten Probleme eingesetzt. Das Benutzen einer direkten Regularisierungsmethode gehört dann zu der inneren Regularisierung.

Bei der Lanczos-Hybrid-Methode verwendet man für die äußere Regularisierung die Lanczos-Bidiagonalisierung. Hierbei wird die Matrix  $A$  aus dem gegebenen Minimierungsproblem  $Ax = b$ , ähnlich wie bei der Singulärwertzerlegung, in ein Produkt aus drei Matrizen zerlegt. Die Matrix  $A \in \mathbb{R}^{n \times n}$  wird in jeder Iteration ( $k = 1, 2, \dots, n$ ) in

$$A = W_{k+1} B_k Y_k^T \text{ mit } W_{k+1} \in \mathbb{R}^{n \times k+1}, Y_k \in \mathbb{R}^{n \times k} \text{ und } B_k \in \mathbb{R}^{k+1 \times k}$$

zerlegt.  $B_k$  bezeichnet eine Bidiagonalmatrix der Form

$$B_k = \begin{bmatrix} \alpha_1 & & & & & \\ \beta_2 & \alpha_2 & & & & \\ & \ddots & \ddots & & & \\ & & & \beta_k & \alpha_k & \\ & & & & & \beta_{k+1} \end{bmatrix},$$

mit  $\beta = \|b\|$ . Weiterhin soll in jeder Iteration

$$A^T W_k = Y_k B_k^T + \alpha_{k+1} y_{k+1} e_{k+1}^T$$

gelten.  $y_{k+1} \in \mathbb{R}^n$  ist die  $(k+1)$ -te Spalte von  $Y_{k+1}$ ,  $e_{k+1}^T$  ist der  $(k+1)$ -te Einheitsvektor und  $\alpha$  der  $(k+1)$ -te Diagonaleintrag der Matrix  $B_{k+1}$ .  $W_k$  und  $Y_k$  besitzen orthonormale Spalten, sodass  $W W^T = E$ , sowie  $Y Y^T = E$  gilt und die erste Spalte von  $W_k$  ist gegeben durch  $b/\|b\|$ .

Nun soll  $\min_x \{\|b - Ax\|_2^2 + \lambda^2 \|x\|_2^2\}$  (aus Gleichung (17)) approximiert werden durch das Minimierungsproblem

$$\begin{aligned} \min_x \{\|b - Ax\|_2^2 + \lambda^2 \|x\|_2^2\} &= \min_f \{\|W_k^T b - B_k f\|_2^2 + \lambda^2 \|f\|_2^2\} \\ &= \min_f \{\|\beta e_1 - B_k f\|_2^2 + \lambda^2 \|f\|_2^2\}. \end{aligned} \quad (22)$$

Als approximierte Lösung erhält man hier  $x_k = Y_k f$ .

Aufgrund dessen, dass wir jetzt ein Minimierungsproblem mit deutlich kleinerer Dimension haben, können jetzt wieder Verfahren angewandt werden, die auf der Singulärwertzerlegung basieren. So benötigt jede Iteration der Lanczos-Bidiagonalisierung die Lösung des Minimierungsproblems (22), welches die Bidiagonalmatrix  $B_k$  enthält. Hierfür können wir nun die Tikhonov Regularisierung benutzen und gehen damit zu der inneren Regularisierung über:

Zum Lösen des Minimierungsproblems (22) müssen wir die Gleichung nach  $f$  ableiten

$$\frac{d}{df} \{\beta^2 e_1^T e_1 - 2\beta f^T B_k^T e_1^T + f^T B_k^T B_k f + \lambda^2 f^T f\} = -\beta B_k^T e_1^T + B_k^T B_k f + \lambda^2 f,$$

dann Null setzen

$$0 = -\beta B_k^T e_1^T + B_k^T B_k f + \lambda^2 f$$

und anschließend nach  $f$  umstellen. Wir erhalten

$$f_\lambda = \beta B_{k,\lambda}^\dagger e_1 \quad (23)$$

als Lösung des Minimierungsproblems. Wobei  $B_{k,\lambda}^\dagger = (B_k^T B_k + \lambda^2 E)^{-1} B_k^T$  ist. Für die ersten Iterationen, also für kleine  $k$ , nähern sich die Singulärwerte von  $B_k$  den kleinsten und größten Singulärwerten von  $A$  an. Wenn die Matrix  $A$  des Originalproblems schlecht konditioniert ist, dann ist auch  $B_k$  schlecht konditioniert, das heißt, die Konditionszahlen von  $A$  und  $B_k$  sind deutlich größer als 1. Da die Dimension von  $B_k$  im Vergleich zu  $A$  klein ist, können wir nun wieder Methoden zum Lösen von  $f_\lambda$  und zum Finden von  $\lambda$  verwenden, die auf der Singulärwertzerlegung basieren.

Für die Wahl eines geeigneten Regularisierungsparameters  $\lambda$  verwenden wir hier das GCV-Verfahren. Um ein geeignetes  $\lambda$  auszurechnen, müssen wir die GCV-Funktion

$$G_{B_k, \beta e_1}(\lambda) = \frac{k \| (E - B_k B_{k,\lambda}^\dagger) \beta e_1 \|_2^2}{\left( \text{Spur}(E - B_k B_{k,\lambda}^\dagger) \right)^2} \quad (24)$$

minimieren. Diese wollen wir aber, genau wie in Abschnitt 3.3.1, zuerst einmal mit Hilfe der Singulärwertzerlegung von  $B_k$  etwas umformen. Nun ist allerdings  $B_k \in \mathbb{R}^{k+1 \times k}$  nicht quadratisch. Aber auch dies soll kein Problem darstellen, denn wie schon in Abschnitt 2.2 erwähnt, kann die Singulärwertzerlegung auch bei nicht quadratischen Matrizen verwendet werden. Diese sieht dann etwas anders aus.  $B_k$  lässt sich zerlegen in

$$B_k = P_k \begin{bmatrix} \Delta_k \\ 0^T \end{bmatrix} Q_k^T \quad (25)$$

mit  $P_k \in \mathbb{R}^{k+1 \times k+1}$ ,  $\Delta_k \in \mathbb{R}^{k \times k}$ ,  $0^T \in \mathbb{R}^{1 \times k}$  und  $Q_k^T \in \mathbb{R}^{k \times k}$ .

Jetzt können wir die GCV-Funktion schreiben als

$$G_{B_k, \beta e_1}(\lambda) = \frac{k\beta^2 \left( \sum_{i=1}^k \left( \frac{\lambda^2}{\delta_i^2 + \lambda^2} [P_k^T e_1]_i \right)^2 + \left( [P_k^T e_1]_{k+1} \right)^2 \right)}{\left( 1 + \sum_{i=1}^k \frac{\lambda^2}{\delta_i^2 + \lambda^2} \right)^2}, \quad (26)$$

wobei  $[P_k^T e_1]_i$  die  $i$ -te Komponente des Vektors  $P_k^T e_1$  und  $\delta_i$  das  $i$ -te Diagonalelement der Matrix  $\Delta_k$  ist. Die Minimierung dieser Funktion ergibt dann das gesuchte  $\lambda$ . [3, 5]

## 4 HyBR

Die Hybrid Bidiagonalization Regularization, kurz HyBR, ist ein Iterationsverfahren zur Lösung linearer Gleichungssysteme  $Ax (+e) = b$ . HyBR besteht an und für sich, wie andere hybride Methoden auch, aus zwei Teilen. Allerdings bleibt es nicht bei einer äußeren und einer inneren Regularisierung, sondern man wiederholt diesen Vorgang iterativ. Dadurch erhält man  $k$  äußere und  $k$  innere Regularisierungsmethoden.

Bei HyBR wird die Lanczos-Hybrid-Methode  $k$ -mal iterativ gelöst. Jeder Iterationsschritt erfolgt wie in Kapitel 3.4 beschrieben.

### 1. Iterationsschritt:

Projektion des Ausgangsproblems auf ein Problem mit kleinerer Dimension  
 $\min_x \{\|b - Ax\|_2^2 + \lambda^2 \|x\|_2^2\} = \min_f \{\|\beta e_1 - B_k f\|_2^2 + \lambda^2 \|f\|_2^2\}$

### 2. Iterationsschritt:

Wahl des Regularisierungsparameters  $\lambda$  mit GCV

$$G_{B_k, \beta e_1}(\lambda) = \frac{k\beta^2 \left( \sum_{i=1}^k \left( \frac{\lambda^2}{\delta_i^2 + \lambda^2} [P_k^T e_1]_i \right)^2 + ([P_k^T e_1]_{k+1})^2 \right)}{\left( 1 + \sum_{i=1}^k \frac{\lambda^2}{\delta_i^2 + \lambda^2} \right)^2}$$

### 3. Iterationsschritt:

Lösen des Minimierungsproblems mit Tikhonov Regularisierung

$$f_\lambda = \beta (B_k^T B_k + \lambda^2 E)^{-1} B_k^T e_1$$

### 4. Iterationsschritt:

Lösung des Ausgangsproblems

$$x_k = Y_k f_\lambda$$

In jeder der  $k$  Iterationen werden alle beschriebenen Schritte erneut durchgeführt. So wird die Suche nach  $\lambda$  in jedem Schritt verfeinert und das regularisierte Bild  $x_k$  nähert sich immer mehr dem Originalbild an. [3]

Wir betrachten nun wieder unser Beispiel des Satelliten und verwenden die Funktion  $HyBR(A, b)$ . Hierfür wandeln wir die gegebene Blurringmatrix  $PSF$  mit Hilfe von  $A = psfMatrix(PSF)$  in die Form um, die wir für diese Funktion brauchen, siehe [4]. Intern wird die Tikhonov Regularisierung in Verbindung mit dem Weighted GCV verwendet, siehe auch [5]. In Abbildung 12 ist links oben die Blurringmatrix  $PSF$  abgebildet und rechts daneben das geblurrte Bild  $b$ . Darunter sieht man dann im Vergleich das Originalbild und das durch HyBR wiederhergestellte Bild. Dieses Bild ist vielleicht noch nicht optimal, aber deutlich besser als das geblurrte Bild darüber. Verwendet man jetzt zusätzlich zum HyBR-Verfahren zum Beispiel noch eine

Vorkonditionierung, dann kann man das Ergebnis nochmals verbessern [4]. Aber hier in unserem Fall ist so eine deutlich sichtbare Verbesserung erst einmal ausreichend.

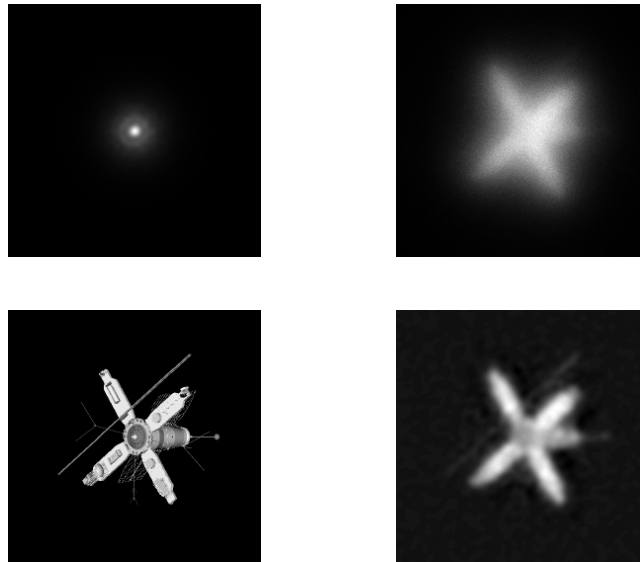


Abbildung 12: Links oben ist die Blurringmatrix  $PSF$  abgebildet, mit der das geblurrte Bild  $b = PSF \cdot x + e$  berechnet wird, welches rechts oben zu sehen ist. Die beiden unteren Bilder zeigen den Vergleich zwischen dem Originalbild links und dem mit HyBR rekonstruierten Bild rechts. Man erkennt eine deutliche Verbesserung zu dem geblurrten Bild  $b$ , aber immer noch ein schlechtes Bild im Vergleich zum Original.

## 5 Vergleich der Verfahren

Zuletzt wollen wir noch die beiden Iterationsverfahren, das CG-Verfahren und HyBR, vergleichen. Hierfür nehmen wir wie auch zuvor das Satellitenbeispiel aus den Restore Tools und außerdem die Methoden *HyBR* und *CGLS*. Um einen einigermaßen ordentlichen Vergleich zu erreichen, müssen die Grundverfahren sowohl für HyBR, als auch für das CG-Verfahren dieselben sein. Das bedeutet, wir müssen zum Einen dasselbe Regularisierungsverfahren und zum Anderen dasselbe Verfahren zur Wahl des Regularisierungsparameters verwenden. Daher benutzen wir bei beiden Verfahren die Tikhonov Regularisierung in Verbindung mit dem GCV-Verfahren.

Beginnen wir nun mit dem CG-Verfahren aus Abschnitt 2.3. Dort minimieren wir die Norm  $\|Ax - b\|_2^2$ . Da wir nun aber auch Regularisierung in die Rechnung mit einfließen lassen wollen, benötigen wir das Minimierungsproblem  $\min_x \{\|b - Ax\|_2^2 + \lambda^2 \|Lx\|_2^2\}$  aus Gleichung (16).  $L$  ist hierbei die Einheitsmatrix und der Regularisierungsparameter  $\lambda$  wird wie erwähnt mit dem GCV-Verfahren berechnet. Mit diesem Wissen verwenden wir die Matlabmethode *CGLS*. Das Programm endet nach circa 92 Sekunden bei der 311. Iteration und bringt das Rekonstruktionsbild aus Abbildung 13 (linke Seite) hervor. Das Bild ist zwar nicht perfekt, jedoch ist der Satellit wieder deutlich zu erkennen und etliche Details können wahrgenommen werden. Alles in allem ist also eine sehr gute Approximation des Originals entstanden.

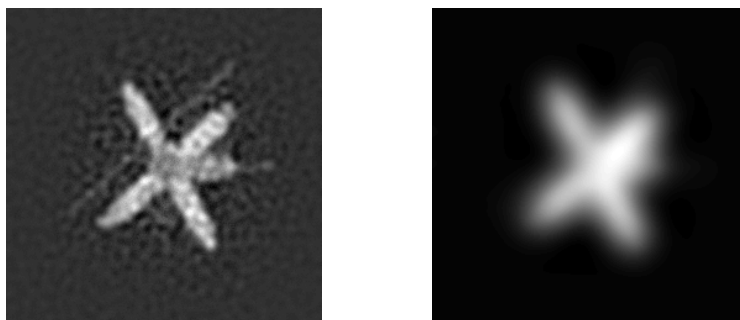


Abbildung 13: Links ist das, mit Hilfe des CG-Verfahrens, rekonstruierte Bild dargestellt. Viele Details können hier wieder erkannt werden. Im Gegensatz dazu ist das rechte Bild, das mit HyBR erzeugt wurde, sehr viel schlechter. Hier ist der Satellit immer noch sehr verschwommen zu sehen.

Kommen wir nun zum HyBR. Wir verwenden die Methode *HyBR* wie im

vorherigen Abschnitt. Jedoch geben wir dieses Mal noch zusätzlich vor, dass die Tikhonov Regularisierung und das GCV-Verfahren verwendet werden soll. Tikhonov wird zwar als Standard schon verwendet, jedoch in Verbindung mit dem Weighted GCV, wofür wir später noch den Grund sehen werden. [11] Nach Berücksichtigung dieser Forderungen erhalten wir Abbildung 13 (rechte Seite) nach circa 87 Minuten, aber nur 9 Iterationsschritten. HyBR braucht nicht nur mehr Zeit für die Rekonstruktion, das rekonstruierte Bild ist auch noch deutlich schlechter als das des CG-Verfahrens. Der Hintergrund ist zwar komplett schwarz, also ist die Lösung stabil, aber der Satellit ist verschwommen und keine Details sind erkennbar.

Für unser Problem unter Verwendung der Tikhonov Regularisierung und dem GCV-Verfahren ist das CG-Verfahren deutlich besser als HyBR, sowohl in der Qualität der Rekonstruktion, als auch in zeitlicher Hinsicht. Der Grund dafür, dass HyBR hier eine so schlechte Rekonstruktion liefert, ist die Benutzung von GCV. Denn wie in [11] beschrieben, neigt das GCV-Verfahren dazu, das Bild zu glatt zu machen, was man auch in Abbildung 13 deutlich erkennen kann. Deswegen wird in Zusammenhang mit HyBR in der Regel das Weighted GCV-Verfahren angewendet. Mehr zum Weighted GCV siehe [5, 9, 11]. Dann ergibt sich eine Rekonstruktion wie in Abbildung 12, rechts unten.

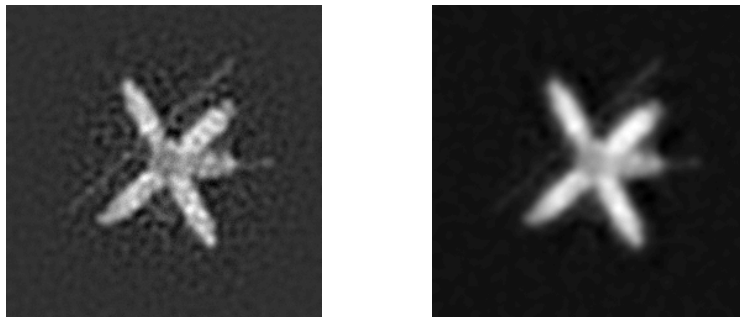


Abbildung 14: Links ist abermals das durch das CG-Verfahren rekonstruierte Bild sehen. Rechts ist das mit HyBR rekonstruierte Bild dargestellt, welches hier nicht wie zuvor mit GCV, sondern mit dem Weighted GCV hergestellt wurde.

In Abbildung 14 sieht man jetzt den Vergleich der mit CG errechneten Rekonstruktion wie zuvor und der Rekonstruktion mit HyBR in Verwendung des Weighted GCV. Das rechte Bild, das mit HyBR erstellt wurde, ist deutlich besser als das aus Abbildung 13, aber dennoch immer noch nicht so gut wie das linke Bild des CG-Verfahrens. Rechts sind deutlich weniger Details

zu sehen als links, aber auch die Rekonstruktionszeit hat sich mit circa 34,5 Sekunden enorm verbessert.

Alles in allem können wir also mit Recht behaupten, dass für dieses Problem und unter diesen Bedingungen das CG-Verfahren das bessere Iterationsverfahren ist.



## 6 Zusammenfassung und Ausblick

Es wurde gezeigt, wie man an das Problem von geblurrten Bildern herangeht. Regularisierungsverfahren spielen hier eine ganz große Rolle. Ohne Regularisierung konnte kein halbwegs gutes Ergebnis erzielt werden. Von den vorgestellten Verfahren, die TSVD Regularisierung und die Tikhonov Regularisierung, ist in der Praxis vor allem die Tikhonov Regularisierung sehr beliebt und wird meistens verwendet. Jedoch ist kein Regularisierungsverfahren effizient ohne einen geeigneten Regularisierungsparameter. Dieser kann auch mit verschiedenen Verfahren berechnet werden. Vorgestellt wurden hier insbesondere das GCV-Verfahren und die L-Kurven Methode, welche beide ungefähr gleich gut sind. Desweiteren gibt es, wie erwähnt, auch noch das Weighted GCV, das den Nachteil des einfachen GCV-Verfahrens verbessern soll, das dazu neigt die Bilder zu überglätten.

Die Regularisierungsverfahren werden innerhalb von Iterationsverfahren verwendet, die das lineare Gleichungssystem  $Ax (+ e) = b$  lösen sollen. In der Arbeit wurde das CG-Verfahren und HyBR vorgestellt. Beide Verfahren wurden hergeleitet und später dann auch miteinander verglichen. Im Weiteren könnte man die Verfahren noch mit anderen Regularisierungsverfahren kombinieren und testen, ob dies geeignete Kombinationen für gute Ergebnisse sind. Zum Beispiel könnte man das CG-Verfahren zusammen mit dem Weighted GCV testen.

Natürlich gibt es noch viele weitere Iterations- und Regularisierungsverfahren. Man müsste schon alle möglichen Kombinationen aus beiden Arten testen, um festzustellen was zur bestmöglichen Bildrekonstruktion beitragen kann.

Außerdem ist die Art des Problems sehr wichtig. Ein Iterationsverfahren kann bei dem einen Problem ein sehr gutes Ergebnis erzielen, aber dafür bei einem anderen ein sehr schlechtes. Bei einer sehr großen Matrix  $A$  beispielsweise ist es nicht mehr möglich eine Singulärwertzerlegung durchzuführen. In diesem Fall kann man solche Verfahren nicht mehr anwenden, die auf der Singulärwertzerlegung beruhen. Die Tikhonov Regularisierung und TSVD könnte man also nicht mehr verwenden bei diesem Problem. Die Lanczos-Hybrid-Methode jedoch ist speziell für solche Probleme geeignet. Das hat zur Folge, dass hier dann auch HyBR ein großen Vorteil gegenüber dem CG-Verfahren hätte.

Ein weiteres großes Thema in Zusammenhang mit diesen Problemen ist die Vorkonditionierung. Mit Hilfe einer Vorkonditionierung soll die Qualität der Lösung nochmals verbessert werden. Man kann jedes Iterationsverfahren zusätzlich mit einer Vorkonditionierung verwenden. Hierbei muss im Vorfeld ein guter Vorkonditionierer ermittelt werden, der die Inverse von  $A$

bestmöglich approximieren soll.

Man sieht also, dass in diesem Themenbereich noch sehr viel offen ist. Dies war also nur ein sehr kleiner Einblick in diese Thematik.

## Literatur

- [1] Martin Hanke-Bourgeois: Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens, *B.G.Teubner Verlag, Wiesbaden, 2002*
- [2] Per Christian Hansen: Regularization Tools, A Matlab Package for Analysis and Solution of Discrete Ill-Posed Problems, *March 2008*
- [3] James G. Nagy: Accelerating Convergence of Iterative Image Restoration Algorithms, *TR-2007-020*
- [4] James G. Nagy, Katrina Palmer und Lisa Perrone: Iterative Methods for Image Deblurring: A Matlab Object Oriented Approach, *2003*
- [5] Julianne Chung, James G. Nagy und Dianne P. O’Leary: A Weighted GCV Method for Lanczos Hybrid Regularization *TR-2007-004-A*
- [6] Jonathan Richard Shewchuk: An Introduction to the Conjugate Gradient Method Without the Agonizing Pain, *1994*
- [7] I. Hnetynkova, M. Plesinger und Z.Strakos: Ill-Posed Inverse Problems in Image Processing, *SNA ’11, Januar 24-28*
- [8] A. K. Louis: Inverse und schlecht gestellte Probleme, *Teubner, Stuttgart, 1998*
- [9] Julianne Chung: Numerical Approaches for Large-Scale Ill-Posed Inverse Problems, *Ph. D. Dissertation, Emory University, 2009*
- [10] Nir Y. Krakauer and Tapio Schneider, James T. Randerson, Seth C. Olsen: Linear inversion methods and generalized cross-validation, *California, 2004*
- [11] Julianne Chung, James Nagy: HyBR: A Method for Solving Ill-Posed Inverse Problems in Image Processing, *Emory University Atlanta, GA*