

6a. Iterative Verfahren: Nullstellen und Optima

Besser, schneller, höher, weiter!

Part-II - Reserve

Das Mehrgitterprinzip

Page 1 of 15

Konjugierte Richtungen

- Anstelle der Residuen oder negativen Gradienten $r^{(i)}$ suchen wir nun also nach alternativen Suchrichtungen $d^{(i)}$, $i = 0, 1, \dots$

$$x^{(i+1)} := x^{(i)} + \alpha_i d^{(i)}.$$

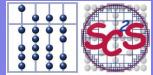
- Nehmen wir einmal kurz den obigen Idealfall an: Der aktuelle Fehler $e^{(i)}$ sei orthogonal zum von $d^{(0)}, \dots, d^{(i-1)}$ aufgespannten Teilraum. Dann ist die neue Korrektur $\alpha_i d^{(i)}$ sinnvollerweise so zu wählen, dass $d^{(i)}$ ebenfalls orthogonal zu $d^{(0)}, \dots, d^{(i-1)}$ und der neue Fehler $e^{(i+1)}$ zusätzlich orthogonal zu $d^{(i)}$ sind, also

$$0 = d^{(i)T} e^{(i+1)} = d^{(i)T} \left(e^{(i)} + \alpha_i d^{(i)} \right) \Rightarrow \alpha_i := - \frac{d^{(i)T} e^{(i)}}{d^{(i)T} d^{(i)}}.$$

- Selbst wenn uns jemand die neue Richtung $d^{(i)}$ schenkt, nützt uns das nichts, da wir $e^{(i)}$ nicht kennen. Deshalb ziehen wir uns wieder auf Residuen statt Fehler zurück und konstruieren **A-orthogonale** oder **konjugierte** Suchrichtungen statt orthogonaler.
- Zwei Vektoren $u \neq 0$ und $v \neq 0$ heißen dabei **A-orthogonal** oder **konjugiert**, falls $u^T A v = 0$. Die entsprechende Forderung an $e^{(i+1)}$ lautet nun

$$0 = d^{(i)T} A e^{(i+1)} = d^{(i)T} A \left(e^{(i)} + \alpha_i d^{(i)} \right) \Rightarrow \alpha_i := - \frac{d^{(i)T} A e^{(i)}}{d^{(i)T} A d^{(i)}} = \frac{d^{(i)T} r^{(i)}}{d^{(i)T} A d^{(i)}}.$$

- A wird also gewissermaßen in die übliche Orthogonalitätsdefinition „eingeschoben“ – im Falle der Identiät ($A = I$) ergibt sich keine Änderung.



Das Mehrgitterprinzip

Page 2 of 15

Konjugierte Richtungen (2)

- Die genannte Bedingung ist interessanterweise äquivalent zur Minimumsuche in Richtung $d^{(i)}$, wie wir sie beim steilsten Abstieg durchgeführt haben:

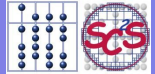
$$0 = \frac{\partial f}{\partial \alpha_i}(x^{(i+1)}) = f'(x^{(i+1)}) \frac{\partial x^{(i+1)}}{\partial \alpha_i} = -r^{(i+1)T} d^{(i)} = d^{(i)T} A e^{(i+1)}.$$

- Im Gegensatz zum zuvor Gesagten sind hier alle Ausdrücke bekannt. Somit erhalten wir folgenden vorläufigen Algorithmus – vorläufig deshalb, weil noch immer nicht klar ist, wie wir denn zu konjugierten Suchrichtungen $d^{(i)}$ kommen können:

- Man beginnt mit $d^{(0)} := r^{(0)} = b - Ax^{(0)}$ und iteriert für $i = 0, 1, \dots$:

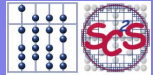
$$\begin{aligned}\alpha_i &:= \frac{d^{(i)T} r^{(i)}}{d^{(i)T} A d^{(i)}}, \\ x^{(i+1)} &:= x^{(i)} + \alpha_i d^{(i)}, \\ r^{(i+1)} &:= r^{(i)} - \alpha_i A d^{(i)}.\end{aligned}$$

- Ein Vergleich dieses Verfahrens mit dem steilsten Abstieg zeigt die enge Verwandtschaft dieser beiden Verfahren.



Zur Konstruktion konjugierter Richtungen

- Somit bleibt einzig das Problem, konjugierte Richtungen zu konstruieren. Dies kann bspw. mittels einer **Gram-Schmidt A-Orthogonalisierung** erfolgen, die ganz analog zur bekannten Gram-Schmidt Orthogonalisierung abläuft (gehe aus von einer Basis $u^{(0)}, \dots, u^{(n-1)}$ und ziehe von $u^{(i)}$ alle nicht zu $u^{(0)}, \dots, u^{(i-1)}$ A-orthogonalen Anteile ab).
- Man kann zeigen, dass der resultierende Algorithmus bei Wahl der Basis aus Einheitsvektoren äquivalent zur Gauß-Elimination ist – der Kreis schließt sich also. Das war aber nicht der Zweck der Übung, es muss auch billiger und einer Iteration angemessen gehen.
- Wenn die Einheitsvektoren uns nicht helfen, wählen wir eben wieder die Residuen $r^{(i)}$ als zu konjugierende Basis. Eine Analyse dieser Vorgehensweise offenbart Überraschendes:
 - Die Residuen sind orthogonal: $r^{(i)T} r^{(j)} = 0$ für $i \neq j$.
 - $r^{(i+1)}$ ist *orthogonal* zu $d^{(0)}, \dots, d^{(i)}$ – somit ist der Fehler $e^{(i+1)}$ *konjugiert* zu $d^{(0)}, \dots, d^{(i-1)}$ (wie gefordert).
 - $r^{(i+1)}$ ist *konjugiert* zu $d^{(0)}, \dots, d^{(i-1)}$.
- Es stellt sich also nahezu alles von selbst ein, nur bezüglich $d^{(i)}$ ist noch eine Konjugation von $r^{(i+1)}$ erforderlich und explizit zu gewährleisten.
- Damit haben wir alle Ingredienzen für den cg-Algorithmus.



Das Mehrgitterprinzip

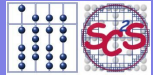
Page 4 of 15

Die Methode der konjugierten Gradienten (cg)

- Man beginnt mit $d^{(0)} := r^{(0)} = b - Ax^{(0)}$ und iteriert für $i = 0, 1, \dots$:

$$\begin{aligned}\alpha_i &:= \frac{r^{(i)T} r^{(i)}}{d^{(i)T} A d^{(i)}}, \\ x^{(i+1)} &:= x^{(i)} + \alpha_i d^{(i)}, \\ r^{(i+1)} &:= r^{(i)} - \alpha_i A d^{(i)}, \\ \beta_{i+1} &:= \frac{r^{(i+1)T} r^{(i+1)}}{r^{(i)T} r^{(i)}}, \\ d^{(i+1)} &:= r^{(i+1)} + \beta_{i+1} d^{(i)}.\end{aligned}$$

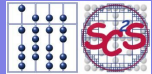
- Auch der cg-Algorithmus ist also sehr kompakt formulierbar und elegant, und dazuhin noch sehr effizient: Die Konstruktion der konjugierten Richtungen kostet pro Schritt lediglich die Berechnung einer Konstanten β_{i+1} sowie eine SAXPY-Operation (Multiplikation eines Vektors mit einem Skalar plus Vektoraddition) für $d^{(i+1)}$.
- Zwei Anmerkungen:
 - Erstens gilt $d^{(i)T} r^{(i)} = r^{(i)T} r^{(i)}$, weshalb man bei der Berechnung von α_i auf das d -Skalarprodukt verzichten kann.
 - Zweitens zeigt sich im Parameter β_{i+1} die Konjugation von $r^{(i+1)}$ bzgl. $d^{(i)}$.



Das Mehrgitterprinzip

Diskussion des cg-Verfahrens

- Im Prinzip ist der cg-Algorithmus ein direkter Löser. Numerische Probleme (Rundungsfehler und Auslöschungen, die insb. die A -Orthogonalität der Suchrichtungen stören können) sowie die schiere Größe von n in realistischen Anwendungen sind jedoch dafür verantwortlich, dass er heute fast nur als iteratives Verfahren eingesetzt wird (n Iterationsschritte könnte kaum jemand abwarten).
- Auch beim cg-Verfahren ist – wie bei den Relaxationsverfahren – die Konvergenzgeschwindigkeit bei diskretisierten Differentialgleichungen dummerweise maschenweitenabhängig: je feiner die Auflösung, desto langsamer die Konvergenz.
- Beim cg-Verfahren heißt das Abhilfe schaffende Zauberwort **Vorkonditionierung**:
 - Statt $Ax = b$ löst man $M^{-1}Ax = M^{-1}b$, wobei der **Vorkonditionierer** M dafür sorgt, dass sich das Konvergenzverhalten entscheidend verbessert (die neue Systemmatrix ist jetzt ja $M^{-1}A$).
 - Ideal wäre natürlich die Wahl $M^{-1} := A^{-1}$, aber dann wäre die Berechnung des Vorkonditionierers teurer als die Lösung des Ausgangsproblems.
 - Die Kunst besteht somit darin, auf billige Weise einen wirkungsvollen Vorkonditionierer zu konstruieren. Hierbei spielt das im folgenden Abschnitt kurz angerissene Mehrgitterprinzip eine ganz entscheidende Rolle.



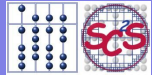
Das Mehrgitterprinzip

Page 6 of 15

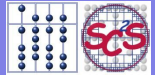
6a.1. Das Mehrgitterprinzip

Grundlegendes

- Die beiden vorigen Abschnitte haben gezeigt, dass sowohl bei Relaxationsverfahren als auch beim cg-Verfahren die Konvergenzgeschwindigkeit bei linearen Gleichungssystemen aus der Diskretisierung partieller Differentialgleichungen (dem wohl wichtigsten Fall) mit der Auflösung abnimmt: Je feiner diskretisiert wird, desto mehr Iterationsschritte sind zur Reduktion des Fehlers um einen vorgegebenen Faktor erforderlich.
- Hier hilft das **Mehrgitterprinzip**, eine der fruchtbarsten Entwicklungen in der jüngeren numerischen Mathematik. In der Literatur tauchen verschiedene Namen wie **Mehrgitter**, **Multigrid**, **Multilevel** oder **Multiresolution** auf – die konkreten Verfahren unterscheiden sich, das zugrunde liegende Prinzip ist dasselbe.
- Wir wollen dieses Prinzip zur Beschleunigung von Relaxationsverfahren einsetzen.
- Die Idee ist einfach, erfordert jedoch zumindest oberflächliche Kenntnisse aus der Fourier-Analysis. Man stelle sich den Fehler $e^{(i)}$ in einem Relaxationsverfahren im Sinne einer Fourier-Transformation zerlegt in Komponenten unterschiedlicher Frequenzen vor. Grob gesprochen ist das eine Reihenentwicklung, wobei jedoch nicht Monome x^k , sondern trigonometrische Terme $\sin(kx)$ und $\cos(kx)$ die Reihenglieder bilden; k gibt dabei die Frequenz an.



- Typischerweise reduzieren unsere Relaxationsverfahren die in Bezug auf das zugrunde liegende Gitter noch darstellbaren hochfrequenten Fehleranteile sehr schnell, wohingegen sie die niederfrequenten Anteile nur extrem langsam abbauen – sie **glätten** den Fehler (daher der Name).

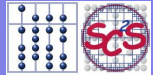


Das Mehrgitterprinzip

Page 8 of 15

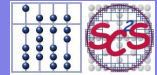
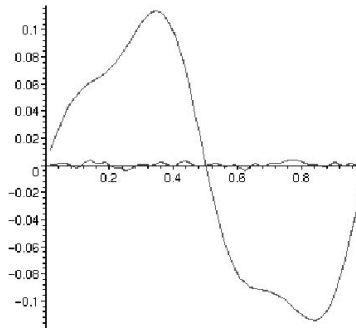
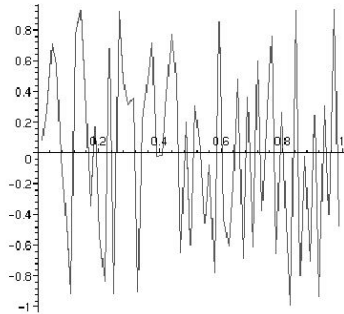
Fehlerreduktion bei der Relaxation

- Wir betrachten dieses Phänomen an einem sehr einfachen Beispiel einer eindimensionalen Laplace-Gleichung (also nichts anderes als $u''(x) = 0$), die auf einem äquidistanten Gitter der Maschenweite $h = 2^{-6}$ mit einem Finite-Differenzen-Ansatz (einer der verarbeitetsten Diskretisierungsmethoden für PDE) gelöst werden soll:
 - Links ist der zufällig gewählte Startfehler $e^{(0)}$ aufgetragen.
 - Rechts ist der Fehler nach 100 Iterationsschritten (sehr viel für solch ein popeliges Beispiel) einer gedämpften Jacobi-Iteration mit Dämpfungsparameter $\alpha = 0.5$ zu sehen.
 - Offensichtlich wird der Fehler geglättet: Die hoch oszillierenden Anteile sind nahezu verschwunden, noch da sind die niederfrequenten Komponenten. Der im Folgenden vorgestellte Mehrgitteralgorithmus erledigt dagegen alle Frequenzen nach nur zwei (!) Schritten, wie im Bild rechts anhand der kaum sichtbaren, um Null oszillierenden Kurve zu sehen ist).



Das Mehrgitterprinzip

Page 9 of 15

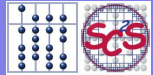


Das Mehrgitterprinzip

Das Prinzip der Grobgitterkorrektur

- **Konsequenz:**

- Offensichtlich hat ein (zu) feines Gitter Probleme mit der Beseitigung langwelliger Fehlerkomponenten. Die hochfrequenten bewältigt es dagegen (natürlich nur, solange sie auf dem Gitter überhaupt darstellbar sind – memento Shannon!).
 - Da die Begriffe 'langwellig' etc. relativ sind, liegt es nahe, zur Reduktion der niederfrequenten Anteile auf ein gröberes Gitter zu wechseln. Dies ist zudem eine ökonomische Entscheidung, da grobe Gitter mit weniger Rechenaufwand verbunden sind.
 - Rekursives Fortschalten dieses **Zweigitteransatzes** (die auch für das grobe Gitter noch zu langwelligen Komponenten werden auf einem noch gröberen Gitter erledigt etc.) führt uns zu **Mehrgitterverfahren**.
- Die zunächst nahe liegende Vorgehensweise, eine gegebene PDE separat auf unterschiedlich fein aufgelösten Gittern zu diskretisieren und anschließend die Gleichungssysteme per Relaxation zu lösen, führt nicht weiter: In jeder „Lösung“ wurde nur ein Teil der Arbeit getan, wie soll man die Information zusammenbringen?
 - Besser sind **Korrektur-Schemata**, bei denen die Berechnung auf dem gröberen Gitter als Korrektur der Feingitterlösung betrachtet wird.
 - Aus Gründen der Einfachheit beschränken wir uns auf den Fall zweier regulärer Gitter.



Der Algorithmus der Grobgitterkorrektur

- Gegeben seien
 - ein feines Gitter Ω_f und ein grobes Gitter Ω_g mit Maschenweite h_f bzw. h_g (Vergrößerung bedeutet i.d.R. den Übergang zu doppelter Maschenweite, also $h_g = 2h_f$),
 - eine Näherungslösung x_f auf Ω_f ,
 - Matrizen A_f, A_g und rechte Seiten b_f, b_g , die die Diskretisierung der PDE auf den jeweiligen Gittern beschreiben.
- Damit können wir folgenden Algorithmus zur **Grobgitterkorrektur** formulieren:

glätte die aktuelle Näherung x_f ;

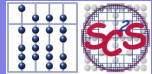
bilde das Residuum $r_f := b_f - A_f x_f$;

restringiere r_f auf das grobe Gitter Ω_g ;

ermittle eine Lösung e_g zu $A_g e_g = -r_g$;

prolongiere e_g auf das feine Gitter Ω_f ;

subtrahiere die resultierende Korrektur von x_f (evtl. nochmals glätten);



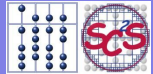
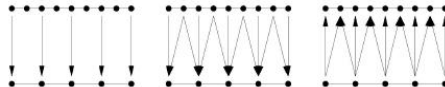
Die Komponenten der Grobgitterkorrektur

- Zu den einzelnen Schritten:
 - das **Vorglätten**: glättet den Fehler (d.h., reduziert die hochfrequenten Fehleranteile); typischerweise ein paar gedämpfte Jacobi- oder Gauß-Seidel-Schritte;
 - die **Restriktion**: einfache Übernahme der Werte an den Grobgitterpunkten (**Injektion**) oder geeigneter Mittelungsprozess über Werte in benachbarten Feingitterpunkten (**full weighting**);
 - die **Grobgittergleichung**: auf dem feinen Gitter gelöst, würde uns die Korrekturgleichung in einem Schritt ans Ziel führen:

$$A_f(x_f - e_f) = Ax = b_f;$$

auf dem groben Gitter (also $A_g e_g = -r_g$) ergibt sich nur eine Korrektur; die Grobgittergleichung kann direkt (falls Ω_g schon grob genug ist), mittels Relaxation oder rekursiv mittels erneuter Grobgitterkorrektur gelöst werden (dann sind wir beim echten Mehrgitterverfahren);

- die **Prolongation**: die berechnete Grobgitterkorrektur e_g muss zurück auf Ω_f interpoliert werden (z.B. direkte Übernahme des Werts in den Grobgitterpunkten und Mittelung in den Feingitterpunkten);
- das **Nachglätten**: manchmal sind ein paar Relaxationsschritte auch *nach* dem Ausflug auf's grobe Gitter ratsam.



Das Mehrgitterprinzip

Page 13 of 15

Der Mehrgitteralgorithmus

- Wird die Grobgittergleichung rekursiv per erneuter Grobgitterkorrektur gelöst, gelangt man vom Zweigitter- zum Mehrgitterverfahren mit einer Hierarchie von Gittern $\Omega_l, k = L, \dots, 1$, hier als **V-Zyklus**:

glätte die aktuelle Näherung x_l ;

bilde das Residuum $r_l := b_l - A_l x_l$;

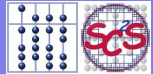
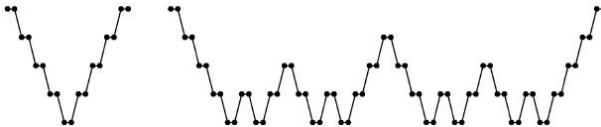
restringiere r_l auf das nächst größere Gitter Ω_{l-1} ;

ermittle eine Lösung e_{l-1} zu $A_{l-1} e_{l-1} = -r_{l-1}$ mittels Grobgitterkorrektur;

prolongiere e_{l-1} auf das nächst feinere Gitter Ω_l ;

subtrahiere die resultierende Korrektur von x_l (evtl. nochmals glätten);

- Auf dem größten Gitter Ω_1 kann oft direkt gelöst werden (wenige Unbekannte). Je nach Verschachtelung der Rekursion ergeben sich verschiedene algorithmische Varianten.

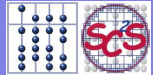


Das Mehrgitterprinzip

Page 14 of 15

Analyse des V-Zyklus'

- Was gewinnen wir durch den Einsatz von Mehrgitteralgorithmen im Vergleich zu einfachen Relaxationen?
 - Erstens wird der Gesamtaufwand vom feinsten Gitter dominiert. Wenn C die Anzahl von Rechenoperationen für einen Glättungsschritt auf Ω_L bezeichnet, dann fallen $C/2$ bzw. $C/4$ bzw. $C/8$ Operationen auf Ω_{L-1} an im 1 D, 2 D oder 3 D Fall. Entsprechendes gilt für Restriktion und Prolongation. Die Gesamtkosten eines V-Zyklus sind damit von derselben Größenordnung wie die eines Relaxationsschritts auf dem feinsten auftretenden Gitter (geometrische Reihe!), also kC mit kleinem k . Dasselbe gilt für den Speicheraufwand. In diesem Sinne kostet ein Mehrgitterschritt also nicht mehr als ein simpler Relaxationsschritt.
 - Bzgl. der Konvergenzgeschwindigkeit gilt in vielen Fällen, dass der Spektralradius ρ der Mehrgitter-Iterationsmatrix nicht von der Anzahl n der Unbekannten und somit nicht von der Feinheit der Diskretisierung abhängt. Reduktionsraten von 0.2 oder kleiner pro Iterationsschritt sind keine Seltenheit. Die Zahl der erforderlichen V-Zyklen bleibt somit i.d.R. im einstelligen Bereich.
- Diese beiden Eigenschaften machen Mehrgitterverfahren zu optimalen iterativen Verfahren. Deshalb wird das Mehrgitterprinzip heute sehr allgemein und in zahlreichen Varianten bzw. Verfeinerungen des hier vorgestellten Grundverfahrens eingesetzt.



Das Mehrgitterprinzip

Page 15 of 15