Mehrgittermethoden

Eine Einführung

Sommersemester 2016 | Robert Speck

Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH

Worum geht es?

Das Thema:

- Iterative Löser dünn besetzter Gleichungssysteme, speziell:
- Idee, Umsetzung und Analyse von Mehrgitter-Algorithmen
- "Write your own multigrid code"

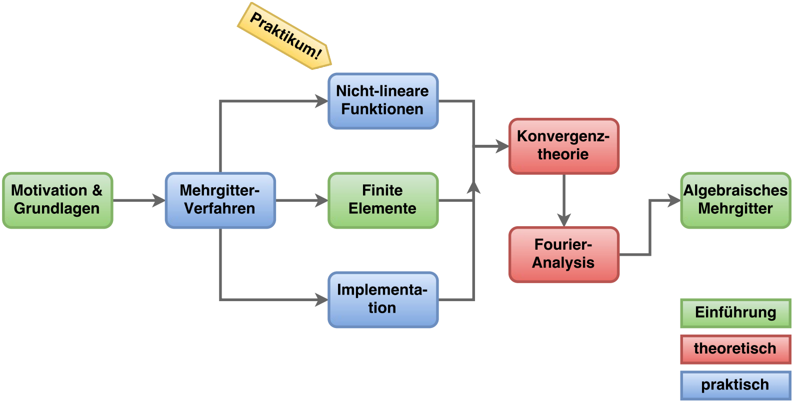
Was ist diese Veranstaltung?

- ein Einblick in die weite Welt von Mehrgitter-Algorithmen
- (vermutlich) nicht schwer
- (hoffentlich) angewandt

Was ist sie nicht?

- vollständig
- fehlerfrei
- klassisch...

Ein erster Überblick



Organisatorisches

- 3 Std. die Woche = Vorlesung + Übung + Praktikum
- Vermutlich Donnerstags, 09:00-12:00 oder 13:30-16:30 Uhr
- Die Übungen sind teils theoretisch, teils praktisch angelegt
- Es wird außerdem eine große Praxis-Aufgabe geben
- Wir nutzen und erweitern das `pyMG`-Framework:

  https://github.com/Parallel-in-Time/pyMG-2016

- Jeder braucht: eine lauffähige und erweiterbare Python 2.7 Installation + ein git Repository (GitHub, Bitbucket o.ä.)
- Mündliche Prüfung am Ende (ca. 30 Minuten)
- Zulassungsvoraussetzung: Erfolgreiches Bearbeiten der Übungsaufgaben + Teilnahme am Praktikum

Mehrgittermethoden

Teil I: Motivation

Sommersemester 2016 | Robert Speck

Die Problemstellung

Lösen von (zunächst) linearen Gleichungssystemen der Form

$$Au = f, \quad A \in \mathbb{R}^{N \times N}$$

wobei A dünn besetzt ist und im Allgemeinen von der Diskretisierung einer partiellen Differentialgleichung kommt, z.B. durch

- Finite Differenzen,
- Finite Elemente,
- Finite Volumen.

Direkte Löser

Aufgabe: Löse $Au = f$ so schnell wie möglich!

Ansatz: Direkte Löser, z.B. LR-Zerlegung oder Cholesky-Verfahren

Problem(e):

- Ohne Ausnutzung der Besetzungsstruktur brauchen wir $\mathcal{O}(N^2)$ Speicherplatz und $\mathcal{O}(N^3)$ Operationen
- Besetzungsstruktur ändert sich im Laufe der Rechnungen (fill-in)
- Bandbreite der Matrix kann sehr groß werden, z.B. in 3 Dimensionen oder Diskretisierungsverfahren höherer Ordnung
- Genauigkeit nicht kontrollierbar, Löser muss stets vollständig ausgeführt werden

Ein Modellproblem und seine Diskretisierung

Unser Modellproblem

Poisson- bzw. Helmholtz-Gleichung mit Dirichlet-Randbedingung

$$-\Delta u(x) + \sigma u(x) = f(x), \quad x \in \Omega, \sigma \geq 0$$
$$u(x) = 0, \quad x \in \partial\Omega$$

Ein Modellproblem und seine Diskretisierung

Unser Modellproblem

Poisson- bzw. Helmholtz-Gleichung mit Dirichlet-Randbedingung

$$-\Delta u(x) + \sigma u(x) = f(x), \quad x \in \Omega,\ \sigma \geq 0$$
$$u(x) = 0, \quad x \in \partial\Omega$$

Konkret wählen wir

- $\Omega = (0{,}1)$, d.h. $\partial\Omega = \{0{,}1\}$,
- $N - 1$ Freiheitsgrade für unsere Finite Differenzen,
- Diskretisierung Ω^h durch $x_j = jh$, $j = 1,\dots,N-1$ für $h = \frac{1}{N}$,

und erhalten so das System $Au = f$...

Ein Modellproblem und seine Diskretisierung

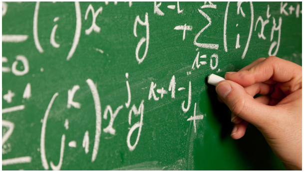
Unser Modellproblem

Poisson- bzw. Helmholtz-Gleichung mit Dirichlet-Randbedingung

$$-\Delta u(x) + \sigma u(x) = f(x), \quad x \in \Omega,\ \sigma \geq 0$$
$$u(x) = 0, \quad x \in \partial\Omega$$

Konkret wählen wir

- $\Omega = (0,1)$, d.h. $\partial\Omega = \{0,1\}$,
- $N - 1$ Freiheitsgrade für unsere Finite Differenzen,
- Diskretisierung Ω^h durch $x_j = jh$, $j = 1,\dots,N-1$ für $h = \frac{1}{N}$,



und erhalten so das System $Au = f$...

Jetzt: iterative Löser

Fragen:

- Können wir die Besetzungsstruktur bestmöglich ausnutzen?
- Können wir optimale Speichernutzung und Komplexität erreichen?
- Können wir den Aufwand von h unabhängig halten?
- ... und trotzdem individuelle Lösungen vermeiden?

Ziel: asymptotisch optimale iterative Löser, d.h.

- Nur $\mathcal{O}(N)$ Operationen
- Von h unabhängiger Aufwand

Ansatz: Iterationsvorschrift

$$v^{(0)} = u_0, \quad v^{(m+1)} = g(A,f,v^{(m)})$$

Jetzt: iterative Löser

Fragen:

- Können wir die Besetzungsstruktur bestmöglich ausnutzen?
- Können wir optimale Speichernutzung und Komplexität erreichen?
- Können wir den Aufwand von h unabhängig halten?
- ... und trotzdem individuelle Lösungen vermeiden?

Ziel: asymptotisch optimale iterative Löser, d.h.

- Nur $\mathcal{O}(N)$ Operationen
- Von h unabhängiger Aufwand

Ansatz: Iterationsvorschrift

$$v^{(0)} = u_0, \quad v^{(m+1)} = g(A, f, v^{(m)}) \to u$$

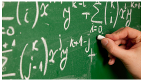
Mehrgittermethoden

Teil II: Grundlagen iterativer Löser

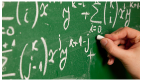
Sommersemester 2016 | Robert Speck

Notation

- Schreibweisen (ab sofort):
  - Kontinuierliche Lösung: $u(x)$
  - Lösungsvektor: $u = (u_j)_{j=1,\dots,M} = (u_1,\dots,u_M)^T$
  - Approximation: $v = (v_j)_{j=1,\dots,M} = (v_1,\dots v_M)^T \approx u$
  - $m$te Iterierte: $v^{(m)}$
  - Vektor auf dem Gitter Ω^h: u^h
- Fehler: $e = u - v$, $e^{(m)} = u - v^{(m)}$
- Residuum: $r = f - Av$, $r^{(m)} = f - Av^{(m)}$
- Standard-Vektornormen: $\|e\|_\infty$ und $\|e\|_2$
- Matrix-Normen für eine Matrix $B = (b_{ij})_{i,j=1,\dots,M}$:
  - Zeilensummennorm: $\|B\|_\infty = \max_i \sum_{j=1}^M |b_{ij}|$
  - Spektralnorm: $\|B\|_2 = \sqrt{\rho(B^T B)}$ mit $\rho(C) = \max|\lambda(C)|$



Einige iterative Löser



Jacobi-Verfahren

- Wähle $B = D$, d.h. die Diagonale von A muss invertiert werden
- Iterationmatrix: $R_\mathrm{J} = I - D^{-1}A = D^{-1}(L + U)$

Gewichtetes Jacobi-Verfahren

- Wähle $B = \omega D$ mit $\omega \in \mathbb{R}$
- Iterationmatrix: $R_\omega = I - \omega D^{-1}A = (1 - \omega)I + \omega R_\mathrm{J}$

Gauß-Seidel-Verfahren

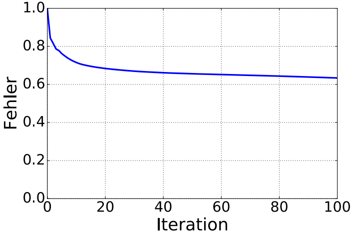
- Wähle $B = D - L$, d.h. der untere Teil von A muss invertiert werden
- Iterationmatrix: $R_\mathrm{GS} = I - (D - L)^{-1}A = (D - L)^{-1}U$

Und viele weitere, z.B. SOR, Red-Black-Gauß-Seidel, ...

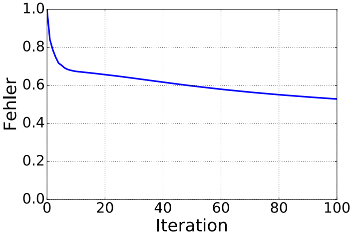
Ein Testproblem

Wir betrachten $-\Delta u = 0$ mit Lösung $u = 0$
Anfangswert für die Iterationen: $v_i^{(0)} = \mathrm{rand}(0{,}1)$

Jacobi mit $\omega = 2/3$



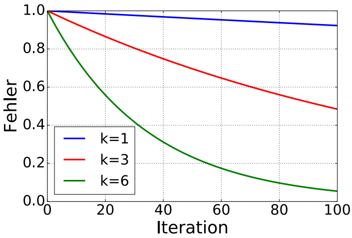
Gauß-Seidel



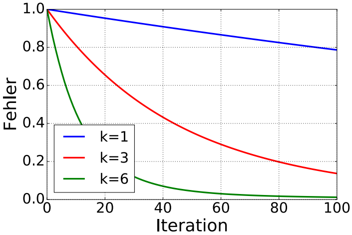
Ein Testproblem

Wir betrachten $-\Delta u = 0$ mit Lösung $u = 0$
Anfangswert für die Iterationen: $v_i^{(0)} = \sin(\pi k x_i)$

Jacobi mit $\omega = 2/3$



Gauß-Seidel



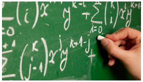
Konvergenz iterativer Verfahren

Satz

Ein Iterationsverfahren $v^{(m+1)} = Rv^{(m)} + c$ mit $R \in \mathbb{R}^{(N-1)\times(N-1)}$ konvergiert für alle $c \in \mathbb{R}^{N-1}$ und alle Startwerte $v^{(0)} \in \mathbb{R}^{(N-1)}$ genau dann, wenn $\rho(R) < 1$.

Es gilt:

- Spektralradius $\rho(R)$ wird asymptotischer Konvergenzfaktor genannt
- $-\log_{10}(\rho(R))$ heißt asymptotische Konvergenzrate
- Fehlerabschätzung:



$$\|e^{(m)}\| \leq \|R\|^m \|e^{(0)}\|$$

Terminologie

Die $N-1$ Vektoren

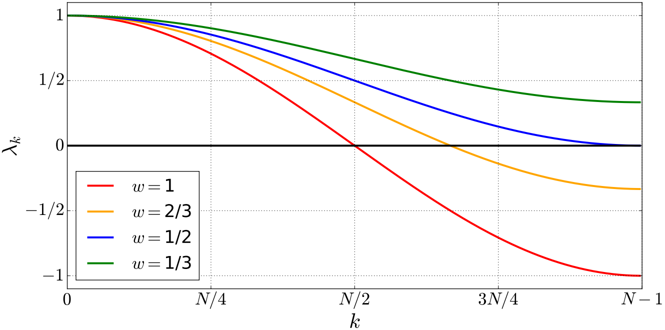
$$w_k = \left(\sin\left(\frac{jk\pi}{N}\right)\right)_{j=1,\ldots,N-1} \qquad \text{für } k = 1,\ldots,N-1$$

heißen Fourier-Moden. Heißt: Die Eigenvektoren von A (und R_ω) bestehen gerade aus diesen Moden!

Wir nennen:

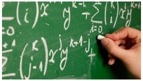
- Moden im unteren Spektrum, also mit $1 \leq k < \frac{N}{2}$, niederfrequente oder glatte Moden
- Moden im oberen Spektrum, also mit $\frac{N}{2} \leq k \leq N-1$, hochfrequente oder oszillierende Moden

Eigenwerte der Jacobi-Iterationsmatrix



Es gilt sogar:

- Je kleiner h, desto weniger lässt sich die glatteste Mode w_1 dämpfen
- Die Wahl $\omega = 2/3$ dämpft die hochfrequenten Moden optimal: der Glättungsfaktor beträgt hier 3



Die Glättungseigenschaft

Allgemein:

- Konvergenz von klassischen iterativen Verfahren ist oft erst sehr schnell, dann sehr langsam
- Verfahren dämpfen oszillierende Moden im Fehler schnell, glatte Moden aber nur langsam

Heißt: oszillierende Komponenten werden aus dem Fehler eliminiert, der Fehler wird glatt $\rightarrow$ Glättungseigenschaft

Idee: Darstellung des Fehler auf einem gröberen Gitter...

JÜLICH
FORSCHUNGSZENTRUM

Mehrgittermethoden

Teil III: Das Mehrgitter-Verfahren

Sommersemester 2016 | Robert Speck

Die Glättungseigenschaft

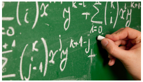
Zusammenfassung und Ausblick

Die spektrale Sicht:

- viele iterative Verfahren besitzen die Glättungseigenschaft, d.h. oszillierende Moden werden gut, glatte Moden schlecht gedämpft
- Verfeinerung des Gitters verstärkt diesen Effekt: Konvergenzfaktoren wie $1 - \mathcal{O}(h^2)$

Ansatz:

- Darstellung des Fehlers auf gröberem Gitter, um dort die glatten Moden besser zu dämpfen
- wichtige Beobachtung: glatte Moden auf Ω^h sehen weniger glatt aus auf Ω^{2h}
- was aber passiert mit oszillierende Moden...?



Die Glättungseigenschaft

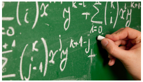
Zusammenfassung und Ausblick

Die spektrale Sicht:

- viele iterative Verfahren besitzen die Glättungseigenschaft, d.h. oszillierende Moden werden gut, glatte Moden schlecht gedämpft
- Verfeinerung des Gitters verstärkt diesen Effekt: Konvergenzfaktoren wie $1 - \mathcal{O}(h^2)$

Ansatz:

- Darstellung des Fehlers auf gröberem Gitter, um dort die glatten Moden besser zu dämpfen
- wichtige Beobachtung: glatte Moden auf Ω^h sehen weniger glatt aus auf Ω^{2h}
- was aber passiert mit oszillierende Moden...?

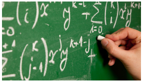


Ein erster Schritt

Das Grobgitter-Korrektur-Verfahren

Idee:

1. Glätte den Fehler auf Ω^h durch Iteration auf $Au = f$
2. Transferiere den Fehler zum nächst gröberen Level Ω^{2h}
3. Entferne die glatten Moden dort
4. Bringe das Ergebnis zurück auf Ω^h
5. Korrigiere die Lösung auf Ω^h



Genauer:

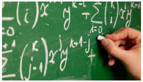
1. Relaxiere auf $A^h u^h = f^h$ und erhalte v^h
2. Bilde das Residuum $r^h = f^h - A^h u^h$ und restringiere zu r^{2h}
3. Löse $A^{2h} e^{2h} = r^{2h}$
4. Interpoliere den Fehler von e^{2h} zu e^h
5. Korrigiere die Lösung v^h durch e^h

Ein erster Schritt

Das Grobgitter-Korrektur-Verfahren

Idee:

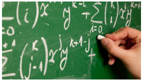
1. Glätte den Fehler auf Ω^h durch Iteration auf $Au = f$
2. Transferiere den Fehler zum nächst gröberen Level Ω^{2h}
3. Entferne die glatten Moden dort
4. Bringe das Ergebnis zurück auf Ω^h
5. Korrigiere die Lösung auf Ω^h



Genauer:

1. Relaxiere auf $A^h u^h = f^h$ und erhalte v^h
2. Bilde das Residuum $r^h = f^h - A^h u^h$ und restringiere zu r^{2h}
3. Löse $A^{2h} e^{2h} = r^{2h}$
4. Interpoliere den Fehler von e^{2h} zu e^h
5. Korrigiere die Lösung v^h durch e^h

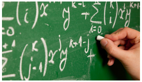
Fragen



- Wie transferiert man den Fehler von Ω^h zu Ω^{2h}?
  $\rightarrow$ wir brauchen Restriktionsoperatoren
- Wie transferiert man das Update von Ω^{2h} zu Ω^h?
  $\rightarrow$ wir brauchen Interpolations- oder Prolongationsoperatoren
- Wie ist A^{2h} definiert?
  $\rightarrow$ wir brauchen eine Darstellung des Problems auf Ω^{2h}
- Wie löst man die Residuumsgleichung auf Ω^{2h}?
  $\rightarrow$ lösen oder relaxieren von $A^{2h}e^{2h} = r^{2h}$?
- Wie bringt man die Iteration ins Rollen?
  $\rightarrow$ wir brauchen einen geeigneten Startwert

Fragen und Antworten

- Wie transferiert man den Fehler von Ω^h zu Ω^{2h}?
  $\rightarrow$ Full-weigthed Restriction ✓
- Wie transferiert man das Update von Ω^{2h} zu Ω^h?
  $\rightarrow$ Lineare Interpolation ✓
- Wie ist A^{2h} definiert?
  $\rightarrow$ wir brauchen eine Darstellung des Problems auf Ω^{2h}



- Wie löst man die Residuumsgleichung auf Ω^{2h}?
  $\rightarrow$ lösen oder relaxieren von $A^{2h}e^{2h} = r^{2h}$?
- Wie bringt man die Iteration ins Rollen?
  $\rightarrow$ wir brauchen einen geeigneten Startwert

Fragen und Antworten

- Wie transferiert man den Fehler von Ω^h zu Ω^{2h}?
  $\rightarrow$ Full-weigthed Restriction ✓
- Wie transferiert man das Update von Ω^{2h} zu Ω^h?
  $\rightarrow$ Lineare Interpolation ✓
- Wie ist A^{2h} definiert?
  $\rightarrow$ Galerkin-Bedingung ✓
- Wie löst man die Residuumsgleichung auf Ω^{2h}?
  $\rightarrow$ lösen oder relaxieren von $A^{2h}e^{2h} = r^{2h}$?
- Wie bringt man die Iteration ins Rollen?
  $\rightarrow$ wir brauchen einen geeigneten Startwert

Das Zweigitter-Verfahren

$v^h \leftarrow \mathrm{TG}(v^h, f^h, \nu_1, \nu_2)$

1. Relaxiere ν_1 mal auf $A^h u^h = f^h$ mit Startvektor v^h
2. Berechne das Residuum $r^h = f^h - A^h u^h$ und bilde $r^{2h} = I_h^{2h} r^h$
3. Löse $A^{2h} e^{2h} = r^{2h}$
4. Bilde $e^h = I_{2h}^h e^{2h}$ und korrigiere $v^h \leftarrow v^h + e^h$
5. Relaxiere ν_2 mal auf $A^h u^h = f^h$ mit Startvektor v^h

Fragen und Antworten

- Wie transferiert man den Fehler von Ω^h zu Ω^{2h}?
  $\rightarrow$ Full-weigthed Restriction ✓
- Wie transferiert man das Update von Ω^{2h} zu Ω^h?
  $\rightarrow$ Lineare Interpolation ✓
- Wie ist A^{2h} definiert?
  $\rightarrow$ Galerkin-Bedingung ✓
- Wie löst man die Residuumsgleichung auf Ω^{2h}?
  $\rightarrow$ lösen oder relaxieren von $A^{2h}e^{2h} = r^{2h}$?
- Wie bringt man die Iteration ins Rollen?
  $\rightarrow$ wir brauchen einen geeigneten Startwert

Mehrgitter-Verfahren

Der V-Zyklus

$v^h \leftarrow \mathrm{VMG}(v^h, f^h, L, \nu_1, \nu_2)$

- Relaxiere ν_1 mal auf $A^h u^h = f^h$ mit Startvektor v^h
- Berechne $f^{2h} = I_h^{2h} r^h$
- Relaxiere ν_1 mal auf $A^{2h} u^{2h} = f^{2h}$ mit Startvektor $v^{2h} = 0$
- Berechne $f^{4h} = I_{2h}^{4h} r^{2h}$
- ...Löse $A^{2^L h} u^{2^L h} = f^{2^L h}$ ...
- Korrigiere $v^{2h} \leftarrow v^{2h} + I_{4h}^{2h} v^{4h}$
- Relaxiere ν_2 mal auf $A^{2h} u^{2h} = f^{2h}$ mit Startvektor v^{2h}
- Korrigiere $v^h \leftarrow v^h + I_{2h}^h v^{2h}$
- Relaxiere ν_2 mal auf $A^h u^h = f^h$ mit Startvektor v^h

Mehrgitter-Verfahren

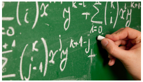
Der V-Zyklus (rekursive Definition)

$v^h \leftarrow \mathrm{VMGr}(v^h, f^h, h, \nu_1, \nu_2)$

1. Wenn Ω^h das gröbste Level ist, dann löse dort, ansonsten:
   1. Relaxiere ν_1 mal auf $A^h u^h = f^h$ mit Startvektor v^h
   2. Berechne $f^{2h} = I_h^{2h} r^h$
   3. Rekursion: $v^{2h} \leftarrow \mathrm{VMGr}(0, f^{2h}, 2h, \nu_1, \nu_2)$
2. Korrigiere $v^h \leftarrow v^h + I_{2h}^h v^{2h}$
3. Relaxiere ν_2 mal auf $A^h u^h = f^h$ mit Startvektor v^h

Bezeichnung: $V(\nu_1,\nu_2)$-Zyklus

Und es gibt noch mehr solcher Zyklen, z.B. $W(\nu_1,\nu_2)$-Zyklen



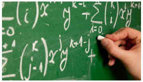
Fragen und Antworten

- Wie transferiert man den Fehler von Ω^h zu Ω^{2h}?
  $\rightarrow$ Full-weigthed Restriction ✓
- Wie transferiert man das Update von Ω^{2h} zu Ω^h?
  $\rightarrow$ Lineare Interpolation ✓
- Wie ist A^{2h} definiert?
  $\rightarrow$ Galerkin-Bedingung ✓
- Wie löst man die Residuumsgleichung auf Ω^{2h}?
  $\rightarrow$ Rekursion ✓
- Wie bringt man die Iteration ins Rollen?
  $\rightarrow$ wir brauchen einen geeigneten Startwert

Der Full Multigrid Ansatz

Idee:

- Nutze geschachtelte Iterationen ausgehend vom gröbsten Level
- Kopple dies auf jedem Level mit V-Zyklen



Rekursive Definition:

$v^h \leftarrow \mathrm{FMG}(f^h,\ h,\ \nu_0,\ \nu_1,\ \nu_2)$

1. Wenn Ω^h das gröbste Level ist, löse dort, ansonsten:
   1. Berechne $f^{2h} = I_h^{2h} f^h$
   2. Rekursion: $v^{2h} \leftarrow \mathrm{FMG}(f^{2h},\ 2h,\ \nu_0,\ \nu_1,\ \nu_2)$
2. Korrigiere $v^h \leftarrow I_{2h}^h v^{2h}$
3. Mache ν_0 $V(\nu_1,\nu_2)$-Zyklen: $v^{2h} \leftarrow \mathrm{VMG}(v^h,\ f^h,\ h,\ \nu_1,\ \nu_2)$

Fragen und Antworten

- Wie transferiert man den Fehler von Ω^h zu Ω^{2h}?
  $\rightarrow$ Full-weigthed Restriction ✓
- Wie transferiert man das Update von Ω^{2h} zu Ω^h?
  $\rightarrow$ Lineare Interpolation ✓
- Wie ist A^{2h} definiert?
  $\rightarrow$ Galerkin-Bedingung ✓
- Wie löst man die Residuumsgleichung auf Ω^{2h}?
  $\rightarrow$ Rekursion ✓
- Wie bringt man die Iteration ins Rollen?
  $\rightarrow$ Full-Multigrid-Ansatz ✓

Zusammenfassung

Die zentralen Ideen:

1. Glättungseigenschaft iterativer Verfahren eliminieren die oszillierenden Komponenten aus dem Fehler
2. Glatte Moden auf feinen Gittern verhalten sich wie oszillierende Moden auf groben Gittern
3. Residuumsgleichung erlaubt die Verwendung der Glätter für den Fehler auf groben Gittern
4. Lineare Interpolation und Full-Weighting transportieren Informationen zwischen den Gittern
5. Der FMG-Ansatz sorgt für geeignete Anfangswerte für anschließende V-Zyklen

Jetzt: Erweiterungen

Zusammenfassung

Die zentralen Ideen:

1. Glättungseigenschaft iterativer Verfahren eliminieren die oszillierenden Komponenten aus dem Fehler
2. Glatte Moden auf feinen Gittern verhalten sich wie oszillierende Moden auf groben Gittern
3. Residuumsgleichung erlaubt die Verwendung der Glätter für den Fehler auf groben Gittern
4. Lineare Interpolation und Full-Weighting transportieren Informationen zwischen den Gittern
5. Der FMG-Ansatz sorgt für geeignete Anfangswerte für anschließende V-Zyklen

Jetzt: Erweiterungen