Prüfung Wissenschaftliches Rechnen

Prüfer: Dr. Fuhrmann
Beisitzer: Habe mir den Namen nicht gemerkt, vermutlich der gleiche wie bei den anderen Prüfungen in März/April 2019.

Note: 1,3

Zum Prüfer: Sehr freundlich, verzeiht bei der Notenvergabe auch, dass Dinge nicht gewusst wurden. Stellt relativ offene Fragen und lässt den Prüfling lange reden, es ist vermutlich eine sinnvolle Übung, das Referieren der wesentlichen Inhalte ein bisschen zu üben, man das die längste Zeit der Prüfung tut. Er fragt nach, wenn man wesentliche Dinge auslässt. Er sagt auch selber etwas zu den Inhalten. Er bohrt nicht ewig nach, wenn man etwas nicht weiß. Die Prüfungen scheinen häufiger dem Muster zu folgen, dass ein großer Teil in die Herleitung in die Herleitung von FEM fließt, kurze Fragen zu einer Verallgemeinerung folgen (bei mir nicht-linearität) und dann entweder C++ oder Paralleles Rechnen.

Zur Vorbereitung: Inhalte aus dem Skript gelernt, 2-3 Wochen

Inhalte
1. Frage: Wie ist die Wärmegleichung mit homogenen Dirichlet-Randbedingungen?
   - Gleichung, RB, zugehörige Räume vollständig aufgeschrieben - keine Rückfragen dazu

2. Frage: Wir wollen das diskritisieren? Wie gehen wir da ran?
   - habe Probleme mit starker Formulierung genannt.
   - habe schwache Formulierung aufgeschrieben.

Eingeschobene Frage: wo kommen die v's her?
   - v's und u's kommen aus H^1_0(Omega), da steckt auch die Boundary Condition schon drin
   - gesagt, dass Green zur Herleitung angewendet wird

Eingeschobene Frage: ist das lösbar?
   - Lax-Milgram mit Bedinung genannt
   - Lemma von Cea hinterhergeschoben

Ich fuhr fort:
   - ist aber immer noch unendlich dimensional
   - wähle endlich dimensionalen Unterraum --> wir haben endliche Basis, Problem kann auf LGS geführt werden
   - damit Matrix zum schnellen Lösen nachher Sparse wird, wählen wir Basis mit lokalen Trägern
   - habe kurz über Triangulierungen und Delaunay-Triangulierung gesprochen
   - Definition: finites Element

Eingeschobene Frage: wie sehen die globalen Basisfunktionen aus?
   - habe das Bild von der Pyramide, die aus dem Mesh rauskommt, aufgemalt

Fortgefahren:
   - Lagrange Finites Element definiert
   - Knoten global numerieren, globales Gleichungssystem, das war die Diskretisierung des Problems

Frage: wie löst man das?
   - a symmetrisch linear --> Matrix A symmetrisch, positiv definit --> CG-Methode

Frage: Wie ist bei der CG-Methode die Idee?
   - Herleitung über Steepest Descent, konnte genaue Vorschrift von CG-Methode nicht hinschreiben
   - Bei CG-Methode wird anders als beim normalen Steepest Descent in jede Richtung nur einmal gesucht, Richtungen sind a-orthogonal zueinander

- nach dem i.-ten Schritt beste Lösung aus Anfangswert + span {Richtungen 1 bis i}

Frage: Muss man sich alle Richtungen merken?

- Nein, aber wusste nicht mehr warum, habe nur darauf verwiesen, dass sich die Iterationsvorschrift jeweils nur auf die letzte Richtung bezieht

Dr. Fuhrmann erklärt: hat etwas mit Konstruktion über Gram-Schmidt zu tun.

Frage: Wie ist die Konvergenzrate?

- hingeschrieben (Wurzel(k).../ usw.)
- Was ist k in dem Ausdruck der Konvergenzrate (spektrale Konditionszahl, Formel hingeschrieben)

Dr. Fuhrtmann schreibt Koeffizient u in homogene Dirichlet Wärmegleichng, was jetzt?

- nicht mehr linear, müssen jetzt das Newton-Verfahren anwenden.
- grobe Idee mündlich gesagt. Konnte exakt Vorschrift nicht mehr aufschreiben

Dr. Fuhrmann schreibt die Vorschrift auf.

- Habe noch Dämpfung angesprochen, wie sich Dämpfung auf Konvergenz auswirkt

Neues Thema: Paralleles Rechnen

Frage: Beim Paralleles Rechnen gibt es welche Paradigmen?

- SIMD, MIMD, kurz erläutert
- SIMD eignet sich für Rechnen auf GPUs, z.B. eine Transformation auf jeden Pixel/jedes Polygon
- wie kann man MIMD programmieren? Habe auf openmp verwiesen (#pragma omp sections mit verschiedenen Instruktionen)

Frage: Wie ist das bei OpenMP mit dem Speicher und Threads organisiert?

- Mehrere Threads, teilen sich einen Adressraum, kommunizieren über geteilte Variablen im selben Adressraum, man muss aufpassen, dass sich keine Race Conditions ergeben, mit Mutexes oder Reduktionen

Frage: Wie kommunizieren verschiedene Prozesse ohne gemeinsamen Adressraum?

- Nur grob gesagt: mit MPI, sie tauschen Nachrichten untereinander aus, grob erläutert, nichts genaues mehr dazu aufgeschrieben
- Zeit abgelaufen.

Prüfungsprotokoll Mathematik

Fach: Wissenschaftl. Rechnen (Sci. Comp.) ☒ Bachelor
Studiengang: Technomathematik ☐ Master

(Sonstiges bitte von Hand eintragen.) — Anrechnung später im Master

Prüfer/in: Jürgen Fuhrmann
Beisitzer/in: René Kehl
Datum: 25.04.2018
Note: 2,3
Prüfungsdauer: 32 Minuten
Anzahl der Kandidaten: 1

Vorbereitungszeit: 5 Tage à 5 Std.
Literatur: Skript, „Numerik für Ingenieure, Physiker und Informatiker" – Bärwolff

Beurteilung der Prüfung und des/r Prüfers/in:

- entspannte Prüfung, viel freies Reden
- Herr Fuhrmann und auch René helfen weiter, wenn es klemmt

Fragen:

Einstiegsfrage: Wir wollen die Wärmeleitungsgleichung mit homogenen Dirichlet-Randbed. ~~lösen~~ mithilfe der FE-Methode lösen – Wie macht man das?

$$-\nabla \cdot \nabla \lambda u = f \text{ auf } \Omega, \quad u = 0 \text{ auf } \partial\Omega$$

→ schwache Formulierung herleiten (Multiplikation mit Testfkt, Integration, Green's Theorem)

$$-\int_\Omega (\nabla \cdot \nabla \lambda u) v \, dx = \int_\Omega f v \, dx$$

Green

$$\Rightarrow \underbrace{\int_\Omega \lambda \nabla u \nabla v \, dx}_{=: a(u,v)} = \underbrace{\int_\Omega f v \, dx}_{=: f(v)}$$

~~FEs~~ → Diskretisierung: $V_h \subset V$ endlichdimensionaler Unterraum
suche $u_h \in V_h$: $a(u_h, v_h) = f(v_h) \quad \forall v_h \in V_h$

Frage: Was wissen wir über die Lösbarkeit dieses Problems?
→ Lax-Milgram: V Hilbertraum, a selbstadj., koerzitive Bilinearform $\Rightarrow$ eindeutige Lösbarkeit

aus der Lösbark. des kontinuierl. Problems $\Rightarrow$ Lösbarkeit des diskreten Problems

weiter mit Galerkin-Verfahren: Basis $\{\phi_1, \dots, \phi_n\}$ von V_h

$$\Rightarrow u_h = \sum_{j=1}^{n} u_j \phi_j$$

$$\Rightarrow \text{löse } \sum_{j=1}^{n} a(\phi_j, \phi_i) u_j = f(\phi_i) \Leftrightarrow AU = F$$

- Wähle Basis so, dass möglichst oft gilt
$a(\phi_i, \phi_j) = 0$ $\Rightarrow$ Matrix A dünnbesetzt

Übergang zur Definition eines Finiten Elements
$\{K, P, \Sigma\}$ mit $K \subset \mathbb{R}^d$ mit Lipschitz-Rand, nichtleeres Inneres

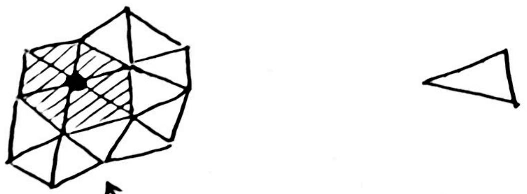
P endlichdimensionaler Vektorraum von Funktionen $p: K \to \mathbb{R}$
- in unserem Fall Polynome -

$\Sigma = \{\sigma_1, \ldots, \sigma_s\} \subseteq \mathcal{L}(P, \mathbb{R})$ lokale Freiheitsgrade, sodass
$P \to \mathbb{R}^s$, $p \mapsto (\sigma_1(p), \ldots \sigma_s(p))$ bijektiv ist, also

Σ basis von $\mathcal{L}(P, \mathbb{R})$

→ gut erklärt anhand von Lagrange Finiten Elementen unter:
www.uni-muenster.de/AMM/num/Vorlesungen/WissenschaftlichesRechnen_WS1213/Dateien/FEM-intro.pdf

Bild: vom einzelnen Dreieck zur gesamten Triangul.



- Einführung von globalen Koordinaten

→ zusammensetzen der globalen Steifigkeitsmatrix aus lokalen Matrizen

Träger der Formfunktion im markierten Punkt?

Frage: Welche Eigenschaften hat die Steifigkeitsmatrix? (bei $P = \mathbb{P}_1$)
- symmetrisch positiv definit

Jetzt also Gleichungssystem $Ax = b$ lösen.

Frage: Welche Möglichkeiten gibt es?
- direkte und indirekte Verfahren, z.B. Cholesky-Zerlegung und CG-Verfahren

→ Empfehlung: indirektes Verfahren ⇒ konvergiert schneller, oft einfacher zu implementieren

- CG für $n \times n$ Matrix konvergiert spätestens nach n Schritten
Idee von CG erläutern: steilster Abstieg, A adj. Suchrichtungen

Zwischenfrage: Für $n = 1.000.000$ dauert das auch sehr lange - was kann man tun? (Genaue Def. der Konvergenzrate)

$$\|e_i\|_A \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^i \|e_0\|_A \qquad \kappa = \frac{\lambda_{max}(A)}{\lambda_{min}(A)}$$

⇒ Konvergenzrate abh. von der Konditionszahl von A

⇒ Lösung für schnellere Konvergenzrate = Präkonditionierung

...plikation von A mit etwas, das die Konditionszahl senkt

~~$EE^T = A$~~ Cholesky-Zerlegung, M spektral äquivalent zu A

$M = EE^T$

mit $\kappa(M^{-1}A) \ll \kappa(A)$

→ Löse GS mit CG mit

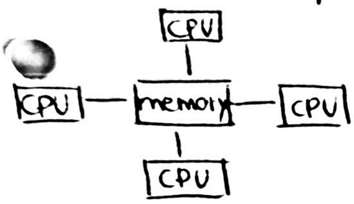
$$E^{-1} A E^{T} \tilde{x} = E^{-1} b$$

Frage: Was tun wir, wenn A nicht symmetr. ist?
- z.B. Gauss-Seidel-Verfahren

Themensprung:

Parallel Paradigms ⇒ SIMD, MIMD

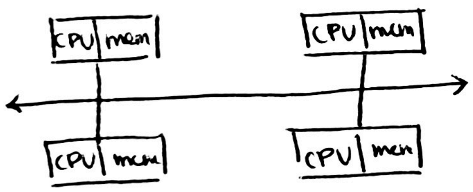
shared memory MIMD



- Threads: - wie in C++ nutzen?

  std::threads, mutex, OpenMP

was ist der Unterschied zwischen OpenMP and OpenMPI?
OpenMPI verwaltet auch Datenübergabe bei distributed memory



- wie verhindert man unberechtigten Zugriff von Threads auf Daten bei OpenMP?

  - #pragma omp critical

Prüfungsprotokoll Mathematik

Fach: Scientific computing
Studiengang: Scientific computing

☐ Vordiplom
☐ Zwischenprüfung
☐ Diplom

☐ Bachelor
☒ Master

Prüfer/in: Fuhrmann
Datum: 14/3/2018
Prüfungsdauer: 20'-25'

Beisitzer/in: Rene Kehl
Note: 1.0
Anzahl der Kandidaten:

Vorbereitungszeit:
Literatur:

Beurteilung der Prüfung und des/r Prüfers/in:

Both of them were very nice and relaxed.

Fragen:

- Heat equation with homogenous Dirichlet boundary condition and how to solve it with P_1 finite element
  - explained Galerkin ~~Res~~ and mentioned Sobolev spaces
  - asked about Lax-Milgram and Cea's Lemma
  - said some stuff about the simplex and the basis functions a bit about going from local to global, interpolation

I said everything too fast so he asked

- Jacobi method
  - just said the basics. Precondition and converging
- Parallel programming
  - The two examples
  - shared /distributed memory
  - OpenMP, MPI
    - threads, processes, deadlock, mutex
- asked what happens if two threads access simultaneously the same data. I said "I don't know", "Correct" he answered.

Prüfungsprotokoll Mathematik

Fach: Scientific Computing
Studiengang: Scientific Computing

☐ Bachelor
☒ Master

(Sonstiges bitte von Hand eintragen.)

Prüfer/in: Jürgen FUHRMANN
Datum: 5.3.2018
Prüfungsdauer: 28 min

Beisitzer/in: Olivier Sète
Note: 1,0
Anzahl der Kandidaten: 1

Vorbereitungszeit: 2 Wochen
Literatur: Vorlesungsfolien, Skripte „Numerik II für Ingenieure", „Numerische Lineare Algebra I", „Nichtlineare Optimierung"

Beurteilung der Prüfung und des/r Prüfers/in:

Herr Fuhrmann lässt einen viel reden, wobei ich irgendwann unsicher wurde, ob ich immer noch weiterreden soll. Aber ich denke, dass er es gut findet, umso mehr man von selbst erzählt. Außerdem hatte ich das Gefühl, dass er sich super spontan überlegt, was er gerade wissen will, ob man es weiß.

Fragen:

Was kann man machen, wenn man das Heat Problem mit Robin boundary conditions hat? $\rightarrow$ FEM, FVM

Und wie sieht das Problem aus? Wie macht man FVM?

$$\rightarrow \quad -\nabla \cdot \kappa \nabla u = f \quad \text{in } \Omega$$
$$\kappa \nabla u \cdot n + \alpha(u-g) = 0 \quad \text{on } \partial\Omega$$

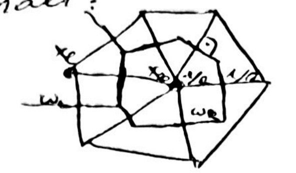
Die Idee ist nun, dass wir Ω in control volumes unterteilen können & für jedes c.v gilt der Satz von Gauß. Zusammen mit den boundary conditions kommen wir auf eine Matrix-Gleichung, die es zu lösen gilt. Hab die Formeln aufgeschrieben & immer ein bisschen gesagt, was ich grade umforme:

$$0 \underset{\text{Gauß}}{=} \int_{\omega_k} (-\nabla \cdot \kappa \nabla u - f)\,dx =$$
$$= -\int_{\partial\omega_k} \kappa \nabla u \cdot n_k \, d\gamma - \int_{\omega_k} f \, dx$$
$$= -\sum_{l \in N_k} \int_{\sigma_{kl}} \kappa \nabla u \cdot n_{kl} \, d\gamma - \int_{\gamma_k} \kappa \nabla u \cdot n \, d\gamma - \int_{\omega_k} f \, dx$$
$$\approx \underbrace{\sum_{l \in N_k} \frac{|\sigma_{kl}|}{h_{kl}} (u_k - u_l) + |\gamma_k| \alpha (u_k - g_k)}_{} - |\omega_k| f_k$$

$\rightarrow$ A: warum kriegen wir hier eine M-matrix?
- off-diagonal entries non-positive
- diagonal entries positive
- idd ~~wenn~~ wenn wir $|\gamma_k| \alpha (u_k)$ dazunehmen.

Wie kommt man an die control volumes?

→ Hier hab ich ein bisschen rumgestottert, dass wir die Voronoi-Zellen einer Delaunay-Triangulierung nehmen kann & man praktischerweise nur die Delaunay-Dreiecke zu speichern hat. Unsere collocation points sind die Ecken der Dreiecke, hab ein bisschen gemalt:



Was macht die Triangulierung Delaunay?

→ Hier hab ich etwas von Winkel $\leq 90°$ gesagt, was natürlich die weak acuteness ist. Daraufhin hab ich Kreise um Punkte gemalt, allerdings nur um 2: , worauf hin er meinte, das stimme noch nicht ganz, also: .

Wenn wir dann die Matrizen haben, wie kommen wir zur Lösung?

→ Direkte Löser, Iterative Löser oder zB CG.
Direkte Löser haben Probleme der Laufzeit & Speicherbedarf. Die klassischen iterativen Methoden konvergieren relativ langsam, also CG.

Wie erkennt man die Konvergenzrate der iterativen Methoden?

→ $u_{k+1} = u_k - M^{-1}(Au_k - b)$, also mit $\hat{u}$ exakte Lsg

$$u_{k+1} - \hat{u} = u_k - \hat{u} - M^{-1}(Au_k - A\hat{u}) = (I - M^{-1}A)(u_k - \hat{u}) =$$
$$= \ldots = (I - M^{-1}A)^k (u_0 - \hat{u})$$

⇒ Konvergenzrate: $\rho(I - M^{-1}A)$ muss < 1 sein!

Was ist die Idee von CG?

→ Gradientenverfahren: Richtung orthogonal, aber wenn man $Ax = b$ lösen möchte, entspricht das dem Minimierungsproblem $\min f(x) = \frac{1}{2} x^T A x - b^T x$ und hier sind in der A-Norm (A spd!) die Niveaulinien von f Kreise um die Lösung
⇒ A-orthogonale Richtungen
Da quadratische Minimierung, bekommen wir die exakte Schrittweite und sehen, dass wir die neue Suchrichtung als Linearkombination von d_{k-1} und r_k bekommen, brauchen auch nicht alle d_i speichern!

Bei CG hab ich so viel wie möglich geredet, um möglichst wenig aufschreiben zu müssen.

Wie ist nun die Konvergenzrate von CG verglichen mit dem Gradientenverfahren?

→ Für CG bekommen wir

$$\|e_i\| \leq 2 \cdot \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i \|e_0\|$$

wobei die Wurzel eben den großen Unterschied macht.

Dann meinte Oliver, dass wir von der Zeit her theoretisch aufhören können. Da hat Herr Fuhrmann kurz überlegt & gesagt, er würde gerne noch was zu Parallelisierung fragen will.

→ Ich hab also auch hier einfach erzählt, was mir in den Kopf kam: Es gibt verschiedene Strukturen, wie wir das Memory verteilen: Shared, Non-Uniform Memory Access, Distributed, Distributed mit Software-Layer. Abhängig davon müssen wir uns Gedanken über Memory-Access machen, zB für Distributed sehr explizit. Es kann zu write-conflicts kommen → mutexes. Problem, dass es dann eventuell wieder sequentiell bearbeitet wird & Dead-Locks

→ Reduktion mit Reduktionsvariable, Zwischenergebnisse werden gespeichert & in der main kombiniert.

Prüfungsprotokoll Mathematik

Fach: Scientific Computing
Studiengang: Scientific Computing

☐ Bachelor
☒ Master

(Sonstiges bitte von Hand eintragen.)

Prüfer/in: Fuhrmann
Datum: 05.03.18
Prüfungsdauer: 30 min

Beisitzer/in: ~~[illegible]~~ Sele
Note: 1,0
Anzahl der Kandidaten: 1

Vorbereitungszeit: 2 Wochen intensiv
Literatur: Intro to High Performance Computing (Eijkhout) – für Rechnerarchitektur & Parallelisierung
Skripte Numerik 2 f. Ingenieure & Numerische LinA – für die Numerik
Online-Referenzen/Tutorials etc. – für C++

Beurteilung der Prüfung und des/r Prüfers/in:
Gute Atmosphäre, wenig "kritische" Nachfragen wenn man etwas nicht ganz so gut erklärt. Insgesamt ~~[illegible]~~ war es hilfreich, wenn man viel frei sprechen konnte. So begann die Prüfung mit "Wärmeleitungsgleichung mit homogenen Dirichlet Randbedingungen in 2D mit FEM, was geht das?", woraufhin ich die ersten 15 min fast allein geredet habe.

Fragen: 1. Thema wie oben geschrieben, konkreter:
- Formulierung d. Problems, Probleme der starken Formulierung
- Herleitung d. schwachen Formulierung mit Erklärung der Funktionenräume
- Lax-Milgram
- Galerkin-Methode: Diskretisierung, Herleitung des Gleichungssystems
- Small lemma of uniqueness
- Verwendung von Basisfunktionen mit kompaktem Träger (A wird sparse) auf finiten Elementen
- Baryzentrische Koordinaten: Definition der Basisfunktionen und Vorteile

2. Lösung des Gleichungssystems mit A s.p.d.
- Kurzer Vergleich iterative und direkte Methoden, für spd Matrix z.B. CG und Cholesky Zerlegung
- Erläuterung der Grundidee von CG, konjugierte Suchrichtungen, schrittweise Optimierung auf Unterraum
- Eigenschaften & Vorteile von CG (z.B. nur die letzten Suchrichtungen werden benötigt um die nächste zu berechnen)
- Konvergenzgeschwindigkeit (hinschreiben!) und Preconditioning
- Vergleich zum Spektralradius von iterativen Verfahren

Prüfungsprotokoll Mathematik

Fach: Scientific Computing
Studiengang: Mathematik

☐ Vordiplom
☐ Zwischenprüfung
☐ Diplom

☐ Bachelor
☒ Master

Prüfer/in: Fehrmann
Datum: 17.7.
Prüfungsdauer: 30'

Beisitzer/in: René Kehl
Note: 2,7
Anzahl der Kandidaten: 1

Vorbereitungszeit: 10 Tage à 3 Std.
Literatur: ~~SR~~ VL Folien + Wikipedia

Beurteilung der Prüfung und des/r Prüfers/in:

war sehr entspannt. Fehrmann hilft auch mal oder gibt Tipps, wenn man auf dem Schlauch steht.
René Kehl saß nur da, hat höflich gelächelt und nach 30 min. festgestellt, dass die Prüfung

Fragen:

- homogenes Dirichlet Problem aufschreiben
- Idee von finiten Elementen erklären (+ Lax Milgram)
- für $\Omega \subset \mathbb{R}^2$ ein finites Element beschreiben
- Galerkin approx. erklären
- baryzentr. Koordinaten globalisieren auf Ω
- Übergang zum Lösen des Systems $Au = f$
- A PSD $\to$ Cholesky Zer. existiert
- Cholesky und Conjugate Gradients vergleichen

Ideen erklären nicht explizit aufschreiben

Prüfungsprotokoll Mathematik

Fach: Wissenschaftliches Rechnen (Scientific Computing) ☐ Bachelor
■ Master
Studiengang: Mathematik

(Sonstiges bitte von Hand eintragen.)

Prüfer/in: Jürgen Fuhrmann
Datum: 29.03.2017
Prüfungsdauer: 25 min

Beisitzer/in: René Kehl
Note: 1,0
Anzahl der Kandidaten: 1

Vorbereitungszeit: 12 Tage
Literatur: Folien der Vorlesung; Stroustrup: C++; Saad: Iterative methods for sparse linear systems; Varga: Matrix Iterative Analysis; Ern, Guermond: Theory and Practice of Finite Elements (alles Literatur aus der VL)

Beurteilung der Prüfung und des/r Prüfers/in: Sehr nette und entspannte Prüfung. Herr Fuhrmann lässt einen sehr frei reden, stellt manchmal Zwischenfragen und leitet zu anderen Themen über, René hat auch manchmal was gesagt. Relevante Gleichungen muss man aufschreiben. Es wurde überhaupt nicht gedrängelt. Manchmal wusste ich nicht, was ich alles erzählen soll und ob das gerade richtig war. Es war oft nicht sehr tiefgehend.

Fragen:

- Heat equation mit homogenen Dirichlet-Randbedingungen als Einstieg: Frage war, wie ich das mit P^1 Finiten Elementen in 2D jetzt umsetzen würde:
  - Habe angefangen, frei zu erzählen (Variationsgleichung aus partieller Integration, Räume dazu erklärt, Bilinearform eingeführt, vom unendlich-dim. zum [redacted]endlich-dim. Raum V_h, Galerkin-Methode ausführlich erklärt, Steifigkeitsmatrix A, wollen sie sparse und daher Funktionen mit kompaktem Träger, ...)
  - 2x Zwischenfrage: Existenz und Eindeutigkeit einer Lösung? (Lax-Milgram erklärt), was gilt für die Approximation $u_h \in V_h$? (Lemma von Céa mit Abschätzung erklärt)
- Wie berechnet man gesamte Steifigkeitsmatrix, bisher sind wir ja nur auf einem FE? (erst lineare FE erklärt, Basisfunktionen, baryzentrische Koordinaten erklärt, Einheitsdreieck und affine Trafo, lokale und globale Freiheitsgrade, dann Assemblierung der globalen Matrix aus lokalen Matrizen, ...)
- Wie löst man das Problem, wenn in der heat eqn. $\lambda(x)$ statt λ steht? (keine konstanten Koeffizienten, Quadraturformeln, habe Trapezregel erwähnt)
- Was gilt für die Steifigkeits-Matrix A und wieso? (SPD, wegen der selbstadjungierten Bilinearform)
- Wie löst man das diskrete Gleichungssystem dann?
  - Hier wieder frei geredet: Erklärt, wo es herkommt (Minimierung eines quadratischen Funktionals), wie CG-Verfahren abläuft, Methode des steilsten Abstiegs mit gut gewählten Suchrichtungen, A-Orthogonalität, energy norm, ...
  - Frage zur Konvergenz: Konvergiert in n Schritten, Konvergenzabschätzung über Konditionszahl $\kappa(A)$ (aufschreiben)

- Wie kann man die Konvergenz verbessern? (Präkonditionierer für A verwenden)

- Wie löst man das System von vorhin mit den $\lambda(x)$? Nichtlineares System, mit Newton-Verfahren, kurz Newton-Verfahren erklärt (aus Fixpunkt-Iteration, Linearisierung über lin. Operator, Jacobimatrix)

- Themensprung zur Parallelisierung: Rechnerarchitektur, Grundparadigmen (MIMD und SIMD erklärt, distributed/shared memory, Threads und MPI erwähnt, wie Threads funktionieren mit gleichem Adressraum, lock/unlock, Mutex-Variable erklärt, Deadlocks)