

1 title

1. Wie kann eine Wurfparabel approximiert werden?
LU-Faktorisierung, Cholesky-Faktorisierung,...
2. Wie werden A und b aufgestellt?
 b beinhaltet die Werte y_i der gegebenen Punkte. In A werden die Basisfunktionen gespeichert, also zB. Monome (wie Übung) werden in der ersten Zeile die $x_1^{0..n}$ und in den Spalten $x_{1..n}$ gespeichert. Somit steht in der i -ten Zeile und j -ten Spalte der Eintrag x_{i+1}^j .
3. Was ist eine Basisfunktion?
Eine Basisfunktion beschreibt eine Kurve indem die Kurve als eine endliche Anzahl von Parametern beschreibt. Die von uns verwendete Basisfunktion ist die Monom-Basis x^i
4. Wie funktioniert LU-Faktorisierung und wie löst man das Gleichungssystem?
Faktorisierung von A in untere (L) und obere Dreiecksmatrix (U) mittels Gauß.
$$\begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix} \rightarrow \begin{pmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \end{pmatrix} \rightarrow \dots$$

Zum Lösen dann (1) $Ly = b$ und (2) $Ux = y$ lösen.
Vorwärtssubstitution für (1):
Erstes Element: $y_1 = \frac{b_1}{l_{11}}$
 $y_i = \frac{1}{l_{ii}}(b_i - \sum_{j=1}^{i-1} l_{ij}y_j)$ für $i = 2, \dots, m$.
Rückwärts für (2):
 $x_1 = \frac{y_m}{u_{mm}}$
 $x_i = \frac{1}{u_{ii}}(y_i - \sum_{j=i+1}^m u_{ij}x_j)$ für $i = m-1, \dots, 1$.
5. Wie kann man LU-Faktorisierung stabiler machen?
Pivotisierung der Matrix, mittels Permutationsmatrix
6. Was ist der Aufwand?
 $PA = LU$: $\frac{2}{3}m^3$ FLOPs
7. Wann geht der LU-Faktorisierung kaputt? **Die LU-Faktorisierung scheitert sobald der Algorithmus versucht eine Division durch 0 durchzuführen. Bsp Matrix** $\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$

2 title

8. Was ist der Unterschied zwischen Approx und Interpolation? Polynomgrad bzw. Dimensionen der Matrix $A^{m \times n}$.
 - $m = n$: A quadratisch, voller Rang, existiert inverse, dann eindeutige Lösung.
 - $m > n$: System überbestimmt, nicht exakt lösbar. Approximation
9. Wie kann ein überbestimmtes Gleichungssystem approximiert werden?
Normalengleichung mit folgender LU-Faktorisierung, Cholesky-Faktorisierung, SVD,...
10. Warum sollte man manchmal ein Polynom kleineren Grades finden?
Durch ein Polynom mit einem kleineren Grad können verrauschte Daten besser approximiert werden.
11. Was ist der Grad eines Polynomes?
Der grad eines Polynoms entspricht dessen höchster Potenz
12. Was sind Normalengleichungen und wie sind sie definiert? Welche Dim hat $A^T A$?
Erster Teil siehe nächste Frage/Antwort.
 $A^T A \in (n \times n)$.
13. Wie leitet man die Normalengleichung her?

3 Herleitungen, geometrische:

$$\begin{aligned}
 Ax &= b \text{ nicht lösbar, da } b \text{ nicht in } \text{range}(A) \\
 &\Rightarrow y \text{ ist orthogonale Projektion von } b \text{ auf } \text{range}(A) \\
 &\Rightarrow r = b - Ax \\
 \|r\| \rightarrow \min &\Leftrightarrow r \perp \text{range}(A) = \langle a_1, \dots, a_n \rangle \\
 &\Leftrightarrow r \perp a_j, j = 1, \dots, n \\
 &\Leftrightarrow a_j^T r = \|a_j\| \cdot \|r\| \cdot \cos(a_j, r) = 0 \\
 &\Leftrightarrow A^T r = 0 \\
 &\Leftrightarrow A^T (b - Ax) = 0 \\
 &\Leftrightarrow A^T b - A^T Ax = 0 \\
 &\Leftrightarrow A^T Ax = A^T b
 \end{aligned}$$

14. Was ist die 2-Norm?

$$\|x\|_2 = \sqrt{\sum_i x_i^2}$$

15. Wann ist eine Matrix symmetrisch positiv definit?

Eine Matrix ist symmetrisch wenn : $A^T = A$ und sie ist positiv definit wenn gilt: $\forall x \neq 0; x^T \cdot A \cdot x > 0$

16. Welcher Löser muss verwendet werden, wenn A nicht quadratisch ist?

Cholesky, QR oder SVD. Je nach Eigenschaften von A.

17. Was ist der Unterschied zwischen Cholesky-Faktorisierung und QR-Faktorisierung?

QR-Faktorisierung	Cholesky-Faktorisierung
eher geometrisch (Basiswechsel)	eher analytisch

mehr schreiben

18. Was ist Cholesky-Faktorisierung? Wie funktioniert sie?

Robuster, effizienter Löser für spd Matrizen. Das Verfahren basiert auf der symmetrischen Gauß-Elimination. Dabei wird die Matrix

$$\begin{aligned}
 A &\text{ in die Faktoren } L \text{ und } L^T \text{ zerlegt. } A = \begin{pmatrix} a_{11} & w^T \\ w & K \end{pmatrix} \\
 A &= \begin{pmatrix} \sqrt{a_{11}} & 0 \\ \frac{w}{\sqrt{a_{11}}} & I \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \tilde{L} \tilde{L}^T \end{pmatrix} \begin{pmatrix} \sqrt{a_{11}} & \frac{w^T}{\sqrt{a_{11}}} \\ 0 & I \end{pmatrix} = \begin{pmatrix} \sqrt{a_{11}} & 0 \\ \frac{w}{\sqrt{a_{11}}} & \tilde{L} \end{pmatrix} \begin{pmatrix} \sqrt{a_{11}} & \frac{w^T}{\sqrt{a_{11}}} \\ 0 & \tilde{L}^T \end{pmatrix} = LL^T
 \end{aligned}$$

Im Algorithmus wird $L = A$ initialisiert und das obere Dreieck auf 0 gesetzt. Mit 3 for-Schleifen (aufgrund MatLab Schreibweise) werden dann die Elemente der Spalte wie folgt berechnet: $L(i, j) = L(i, j) - L(i, k) * (L(j, k) / L(k, k))$. Der Wert $x = 1 / \text{sqrt}(L(k, k))$ wird separat berechnet und zum Schluss werden die Elemente der Spalte durch $L(i, k) = L(i, k) * x$ endgültig berechnet.

19. Was ist fill-in und wie kommt es zustande?

Fill-in beschreibt das eine $a_{ij} = 0$ in A ist, aber nach der Faktorisierung mit Cholesky $a_{ij} \neq 0$. Fill-in tritt nur innerhalb des Bandes auf, alle Nullen außerhalb des Bandes bleiben erhalten. Es entsteht durch die gebildete Differenz $K - ww^T$.

20. Wie funktioniert QR-Faktorisierung?

- Erst orthonormale Basis $\{q_1, \dots, q_n\}$ von $\text{range}(A)$ konstruieren (Gram-Schmidt)
- Dann sind Vektoren aus A darstellbar in der gefundenen Basis $\{q_1, \dots, q_n\}$.
- $a_1 = r_{11}q_1, \dots, a_n = r_{1n}q_1 + \dots + r_{nn}q_n$. (Gram-Schmidt liefert auch Koeffizienten r)
- in Matrixschreibweise zusammengefasst ergibt sich: $A = QR$

Für die Lösung eines least-squares Problems:

- QR-Faktorisierung $A = QR$ (siehe oben).
- Berechne $b' = Q^T b$
- Löse $Rx = b'$ durch Rückwärtssubstitution.

21. Was ist eine orthogonale Projektion?

Entspricht der Projektion eines Punktes b auf das Bild einer Matrix A

beginne mit Projektion von b auf $\langle A \rangle$, mit $\|A\| = 1$

$r = b - b' = (I - AA^T)b$, mit $b' = (\sum_{i=1}^k A_i A_i^T) b = \hat{A} \hat{A}^T b$ mit $\hat{A} \hat{A}^T$ als orthogonale Projektion

22. Wann ist A von vollem Rang?

Die Matrix hat den vollen Rang, wenn: $\text{rank}(A) = n \Leftrightarrow \dim(\text{range}(A)) = n \Leftrightarrow \dim < a_1, \dots, a_n > = n \Leftrightarrow \text{Spalten } a_j \text{ linear unabhängig}$
 $\Leftrightarrow \text{null}(A) = 0$

SVD-Bild einfügen aus Skript S.47

23. Wie ist die Pseudoinverse definiert?

$$A^+ = (A^T A)^{-1} A^T$$

3 title

24. Was ist die Idee der Konditionierung?

Das Problem als Funktion modellieren, welche bei Eingabe x Ausgabe $f(x)$ liefert. Ein Problem ist gut konditioniert, wenn eine kleine Perturbation (Veränderung) in x auch nur eine kleine Perturbation der Ausgabe $f(x)$ bewirkt. Umgekehrt sollte klar sein.

25. Was bedeutete es, wenn ein Problem gut konditioniert ist?

Ein Problem ist gut konditioniert, wenn eine Störung (Perturbation) in \mathbf{x} auch nur eine kleine Störung in $\mathbf{f}(\mathbf{x})$ bewirkt. Dabei werden Perturbationen in relativen Änderungsraten gemessen mit: $\frac{\|\delta x\|}{\|x\|}, \frac{\|f(x+\delta x) - f(x)\|}{\|f(x)\|}$

26. Was sind finite Differenzen?

Diskretisierung auf einem regulärem Gitter, wobei die Ableitungen Approximiert werden.

(a) Vorwärtsdiff.: $f'[i] \approx \frac{f[i+1] - f[i]}{h}$

(b) Rückwärtsdiff.: $f'[i] \approx \frac{f[i] - f[i-1]}{h}$

(c) Zentrale Diff.: $f'[i] \approx \frac{f[i+1] - f[i-1]}{2h}$



Abbildung 8.2: Approximation der ersten Ableitung mittels Vorwärtsdifferenz (links), Rückwärtsdifferenz (Mitte) und zentrale Differenz (rechts).

$$\Delta u[i, j, t] = \frac{u[i-1, j, t] + u[i+1, j, t] + u[i, j-1, t] + u[i, j+1, t] - 4u[i, j, t]}{h^2}$$

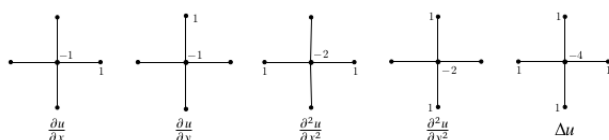


Abbildung 8.3: Schematische Darstellung der partiellen Ableitungen.

27. Was bedeuten parabolisch, hyperbolisch und elliptisch?

Parabolische PDEs beschreiben dynamische Gleichungen die gegen ein statisches Equilibrium konvergieren. Elliptische PDEs beschreiben statische Gleichgewichtszustände. Hyperbolische PDEs modellieren dynamische Systeme ohne Equilibrium.

28. Was besagt Diskretisierung?

Grundlegende Idee: Kurve (kontinuierlich) mit einer endlichen Zahl von Parametern zu beschreiben.

Dazu Kurve mittels Basisfunktionen φ_j und Koeffizienten f_j darstellen: $f(x) = \sum_{j=1}^n f_j \varphi_j(x)$.

Daraus ein LGS zusammenbasteln.

29. Wie lässt sich die erste Ableitung mittels finiter Differenzen approximieren?

Taylor-Entwicklung erster Ordnung $f(x) + hf'(x) + O(h^2)$ nach $f'(x)$ auflösen

$$\Rightarrow f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

Wird $f(x)$ diskretisiert mit $f[i] = f(ih)$

$$\xrightarrow{\text{Vorwärtsdifferenz}} f'[i] = \frac{f[i+1] - f[i]}{h}$$

$$\xrightarrow{\text{Rückwärtsdifferenz}} f'[i] = \frac{f[i] - f[i-1]}{h}$$

$$\xrightarrow{\text{Zentraldifferenz}} f'[i] = \frac{f[i+1] - f[i-1]}{2h}$$

30. Was ist der Unterschied zwischen $-\Delta u$ und Δu ?
 $\Delta u[i, j, t] = \frac{u[i-1, j, t] + u[i+1, j, t] + u[i, j-1, t] + u[i, j+1, t] - 4u[i, j, t]}{h^2}$: Ausgehend von den Punkten um die Mitte wird davon der mittlere Punkt 4mal abgezogen.
 $\Delta u[i, j, t] = \frac{4u[i, j, t] - u[i-1, j, t] - u[i+1, j, t] - u[i, j-1, t] - u[i, j+1, t]}{h^2}$: Ausgehen von dem mittleren Punkt, werden die umliegenden Nachbarn abgezogen.
 vgl. Abb. Frage 26

Skript 59 Bild

4 title

31. Wie sieht die Heat-Equation aus? $\dot{u} = \Delta u$
 32. Was besagt $\dot{U} = \text{Laplace}U$? Siehe vorige Frage.
 33. Was ist der explizite/implizite Euler? (Definition und Vorteile/Nachteile sowie Unterschiede)
 Definition vgl. 59

explizit	implizit
Vorteile: einfach zu verstehen einfach zu implementieren Nachteile: Instabilität bei großen Zeitsprüngen, nicht mehr δt konvergiert Fehler schaukeln stark \rightarrow keine Lösung mehr möglich	verwendet Zeitableitung \dot{u} am nächsten Zeitpunkt $u[t+1]$ stabil für jeden Zeitschritt höherer Rechenaufwand

34. Wie kann Bildrauschen eliminiert werden? (Am Bsp.: Heat-Equation?)
 35. Was passiert bei zu hohen Schritten im explizite Euler?
 36. Was ist die CFL-Bedingung?(Vllt?)
 $\delta t \leq \frac{h}{||v||}$ mit $||v||$ = Informationsgeschwindigkeit.
 37. Wann ist der Gleichgewichtszustand erreicht?
 Der implizite Euler konvergiert immer gegen den statischen Gleichgewichtszustand der Heat-Equation, der durch die Laplace-Equation $\Delta u = 0$ beschrieben wird. Während der explizite nur bei ausreichend kleinen Zeitschritten konvergiert.
 38. Was beschreiben die Matrizen U und V in der Heat-Equation?
 In U werden die Höhen der Funktion an jedem Gridpunkt gespeichert. In V wird die Ableitung der Zeit in Bezug auf eine bestimmte Höhe U , also die Geschwindigkeit, gespeichert.
 39. Was passiert mit den Randpunkten?
Die Randpunkte in der Heat-Equation werden nicht verändert.
 40. Wie stellt man das lineare Gleichungssystem für den Laplace auf?
Jeder Punkt im Gitter ist eine Unbekannte die berechnet werden muss, wir haben als bei einem $N \times N$ mit Gitter $N-2$ Unbekannten (Randpunkte verändern sich nicht). Dies führt zu einem $(N-2)^2 \times (N-2)^2$ Gleichungssystem $-Lu = b$. U ist ein Spaltenvektor in dem die neuen Werte U für das Gitter stehen. Die Reihenfolge in der die Gitterpunkte in U stehen führt zu einer Index Abbildung: $u_k = u(i, j)$ mit $k = \text{idx}(i, j) = (i-1) + (j-2) \cdot (N-2)$. Für die Matrix L gilt: $l_{kk} = 4, l_{kl} = -1/h^2$, wenn (i, j) Nachbarn von i, j und $l = \text{idx}(i, j)$ sonst $l_{kl} = 0$.
 41. Warum stellt man das lineare Gleichungssystem für den Laplace auf?

Keine Ahnung, die Frage ist doof

42. Wie unterscheidet sich das Lösen des Gleichungssystems beim Laplace zu den konjugierten?
 Die konjugierten Verfahren lösen dieses iterativ, während bei der Laplace Gleichung genau eine Lösung gefunden wird. Da er unabhängig vom der Zeit ist, kann die explizite Euler Integration nicht angewendet werden.
 43. Wie löst man dieses Gl?
Dieses Gl ist mit der Cholesky-Faktorisierung lösbar. Die Zeitkomplexität ist allerdings mit $O(m^3)$ hoch in diesem Fall wodurch die lösungsweise durch iterative lösungsverfahren besser ist.
 44. Welche Eigenschaft hat die Laplace-Matrix?
 $-L$ hat folgende Eigenschaften:
- Die Matrix ist sparse: Maximal 5 Einträge pro Zeile, die ungleich null sind.
 - $-L$ ist symmetrisch.
 - $-L$ ist positiv definit (L negativ definit).

5 title

45. Wie funktionieren konjugierte Gradienten?

Beliebige Suchrichtung p_n für die Minimierung von $f(x)$ zulassen:

$$\Rightarrow x_{n+1} = x_n + \alpha_n p_n,$$

wobei α_n ähnlich zum Gradientenabstieg bestimmt werden kann:

$$\frac{\delta}{\delta \alpha_n} f(x_n + \alpha_n p_n) = 0$$

$$\Rightarrow \alpha_n = \frac{p_n^T r_n}{p_n^T A p_n} \text{ wenn } p_n = -\nabla f(x_n) = r_n \text{ äquiv. zu Gradientenabstieg}$$

46. Wann werden konjugierte Gradienten angewendet?

Für große Matrizen ($m > 10000$), da sie die Struktur ausnutzen.

47. Was ist α ?

α ist die Schrittweite in der wir uns Richtung Lösung bewegen bei einem iterativem Löser. Im Gradientenabstieg ist $\alpha = \frac{r_n^T r_n}{r_n^T A r_n}$; in konjugierten Gradienten ist $\alpha = \frac{r_n^T r_n}{p_n^T A p_n}$

48. Was ist die Konvergenz des Gradientenabstiegs?

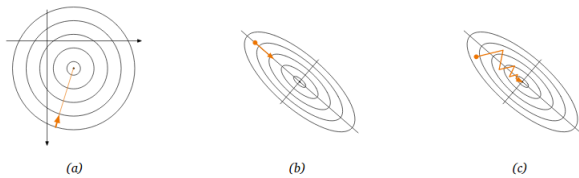


Abbildung 11.3: Konvergenz des Gradientenabstiegs. (a) Sphärische Konturlinien: Konvergenz in einem Schritt. (b) Elliptische Konturlinien, Startpunkt auf Hauptachse: Konvergenz in einem Schritt. (c) Elliptische Konturlinien, beliebiger Startpunkt: Langsame Konvergenz!

49. Wie ist der Gradientenabstieg definiert?

Löse $Ax = b$, mit A s.p.d. formuliert als Minimierung $f(x) = \frac{1}{2}x^T Ax - b^T x \rightarrow \min$

$f(x)$ hat ein Minimum, wenn gilt: $0 = \nabla f(x) = Ax - b$

minimiere $f(x)$ iterativ, indem bei jedem Schritt von x_n ein Stückchen im Paraboloid heruntergelaufen wird, dies entspricht dem negativen Gradienten $-\nabla f(x)$.

Da $r_n = b - Ax_n = -\nabla f(x)$ folgt r_n als Abstiegsrichtung der n -ten Iteration und $x_{n+1} = x_n + \alpha_n r_n$, mit $\alpha_n = \arg \min_{\alpha} f(x_{n+1})$

50. Was ist der Unterschied zwischen dem Gradientenabstieg und konjugierten Gradienten?

Bei dem Gradientenabstieg ist die Schrittweite abhängig von dem Resduum r_n . Die konjugierten Gradienten ersetzen r_n durch die Suchrichtung p_n , wobei jedes p_n einmalig vorkommt.

51. Was ist eine Datenstruktur für eine dünnbesetzte Matrix?

Eine einfache Datenstruktur für eine dünnbesetzte Matrix währe das Triple-Format wo alle Einträge $a_{ij} \neq 0$ in einem Tripel (i, j, v) gespeichert werden wobei i der Zeilenindex, j der Spaltenindex und v der Wert von a_{ij} ist. Überlicherweise gespeichert als 3 Arrays für jeweils i, j, v .

52. Was sind die Vorteile von konjugierten Gradienten?

- Fehler $\|e_n\|_A$ nimmt monoton ab.
- Der Algorithmus terminiert nach maximal m Iterationen.
- Aufwand $O(m^2)$, typischerweise $O(m^{1.5})$.
- Speicheraufwand $O(m)$.
- Algorithmus muss nicht direkt auf Matrix-Einträge zugreifen. Daher Nutzung einer effizienten Datenstruktur für das Matrix-Vektorprodukt Ap möglich.

53. Was ist ein Stopkriterium von konjugierten Gradienten?

Wenn $A^{n \times n}$ dann kann man nach maximal n -Schritte aufhören, ebenso wenn $\beta < \epsilon$

6 title

54. Wie ist die Wave-Equation definiert? Wie kann man sie als PDE erster Ordnung schreiben?

$\ddot{u} = c^2 \Delta u$. Durch eine Hilfsfunktion $v := \dot{u}$ kann die Gleichung wie folgt modelliert werden: $\dot{u} = v$ $\dot{v} = c^2 \Delta u$

55. Was erfordert eine doppelt so große Wellengeschwindigkeit?
Eine doppelt so hohe Wellengeschwindigkeit erfordert einen halb so großen Zeitschritt, damit die explizite Integration nicht instabil wird und weiterhin die CFL bedingung erfüllt.
56. Wie diskretisiert man die PDE's mit finiten Differenzen im Falle der Wave-Equation?
 $u[i, j, t] = u(ih, jh, t\delta t)$, $v[i, j, t] = v(ih, jh, t\delta t)$. Die PDEs hängen nun im 2-dim Gitter von Raum und Zeit ab.
57. Wieso müssen Zeitschritte manchmal gesplittet werden?
 Um im expliziten Euler eine große Wellengeschwindigkeit zu berechnen, müssen Zeitschritte gesplittet werden.
58. Welche Randbedingungen sind möglich?
- (a) periodisch
 - (b) gespiegelt
 - (c) Null
59. Was ist der Unterschied zwischen implizit und explizit?(vgl Heat.)
Explizite Integration berechnet die Werte des nächsten Zeitpunktes $u[t+1]$ aus dem momentanen Werten am Punkt sowie dessen Nachbarn: $u[t+1] = u[t] + \delta t \Delta u[t]$. Implizite Integration berechnet die Werte des nächsten Zeitpunktes $u[t+1]$ mithilfe des nächsten Zeitpunktes: $u[t+1] = u[t] + \delta t \Delta u[t+1]$. Zum berechnen von $u[t+1]$ müssen wir das Gleichungssystem: $(I - \delta t \Delta)u[t+1] = u[t]$ lösen.
60. Wie sieht das lineare Gleichungssystem im impliziten Fall im Vergleich zum expliziten (Wave Equation) aus?
 Im expliziten Fall kann das Gleichungssystem in einem Schritt gelöst werden, da die Gleichungen nicht voneinander abhängen. Im impliziten Fall hängt v_{t+1} aber von u_{t+1} ab und umgekehrt, und muss daher als Gleichungssystem gelöst werden:
- $(I - \delta t^2 c^2 L) v_{t+1} = v_t \delta t c^2 L u_t$.
 - Dabei ist $A = (I - \delta t^2 c^2 L)$, $x = v_{t+1}$ (gesucht) und $b = v_t \delta t c^2 L u_t$

7 sparse Cholesky-Faktorisierung

61. Was ist der Unterschied von sparse Cholesky-Faktorisierung und normaler Cholesky-Faktorisierung?
 Für die sparse Matrix A wird die Spalte j nicht mehr durch all Spalten $k < j$ aus L modifiziert. Die j -Spalte wird nur noch durch die Spalten k modifiziert, für die $l_{jk} \neq 0$ ist.
62. Was sind fill-ins?
 s. Frage 19
63. Was ist eine Bandmatrix?
Eine Bandmatrix ist eine dünnbesetzte Matrix deren Nicht-Null-Einträge sich auf ein 'Band' um die Diagonale der Matrix beschränken. wobei die Bandbreite $\beta(A)$ definiert ist als die maximale Distanz einer nicht-Null von der Diagonalen der Matrix

$$\begin{pmatrix}
 4 & -1 & & & & & & & & \\
 -1 & 4 & -1 & & & & & & & \\
 & -1 & 4 & & & & & & & \\
 -1 & & & \ddots & & & & & & \\
 & \ddots & & \ddots & \ddots & & & & & \\
 & & & \ddots & \ddots & \ddots & & & & \\
 & & & & \ddots & \ddots & \ddots & & & \\
 & & & & & \ddots & \ddots & \ddots & & \\
 & & & & & & \ddots & \ddots & \ddots & \\
 0 & & & & & & & & & -1 \\
 & & & & & & & 4 & -1 & \\
 & & & & & & & -1 & 4 & -1 \\
 & & & & & & & -1 & -1 & 4
 \end{pmatrix}$$

$\underbrace{\hspace{10em}}_N$

A.

64. Kommt in der sparse Cholesky-Faktorisierung fill-in vor?
65. Welche Schritte muss man im sparse Cholesky-Faktorisierung machen?
 Ändere die Reihenfolge der x_i mittels Permutationsmatrix P , so dass $\tilde{x} = Px$
 Sortiere analog A um; also $\hat{A} = AP^T$. Da A nun nicht mehr symmetrisch, permutiere nun auch die Zeilen von A und b respektive $\tilde{A} = P\hat{A} = PAP^T$, $\tilde{b} = Pb$.
 Wende nun Cholesky auf $\tilde{A}\tilde{x} = \tilde{b}$ an.
66. Welche Vorteile hat der Sparse Cholesky-Faktorisierung?
 Effizientes, robustes Verfahren für dünn-besetzte Matrizen. Durch eine Umsortierung der Matrix kann es keine fill-ins geben. Durch die Konstruktion von \tilde{L} enthält sie weniger 0en.
67. Was ist die Band-Cholesky-Faktorisierung?
Zunächst lege eine Band-matrix-Datenstruktur für L an. der Speicherverbrauch ist nun $O(mb)$ anstatt $O(m^2)$, wobei b die bandbreite von der Bandmatrix ist. Bei der Berechnung der l_{ij} von L müssen nur Einträge $l_{ij} \in \text{Band}(L)$ berücksichtigt werden, da alle anderen einträge null sind. Dies reduziert den Rechenaufwand von $O(m^3)$ auf $O(mb^2)$ Löse nun $Ly = b$ und $L^T x = y$ unter ausnutzung der Bandbegrenzung von L , dies reduziert den aufwand von $O(m^2)$ auf $O(mb)$

Bilder 102

68. Wieso macht man den Band-Cholesky-Faktorisierung und was ist der Vorteil davon?
69. Was macht der Minimum degree-Algorithmus?
 Ziel: Permutation finden, sodass $LL^T = PAP^T$ möglichst wenig fill-in besitzt.
 Dazu suche den Knoten v mit den wenigsten Kanten und nummeriere diesen. Des Weiteren verbinde die Nachbarn von v mit einander.

8 title

70. Wie parallelisiert man?
 Auf einem Prozess mittels SIMD und SSE. Auf mehreren Prozessoren mittel OpenMP (1 Masterthread). Der Code von OpenMP ist:

```
# pragma omp parallel
id = omp_get_thread_num (); Welcher Thread ?
int nthreads = omp_get_num_threads () Wieviele?
```
71. Welche Kommandozeilen Argumente?

- **-O:** schaltet einfache optimierungen ein
- **-O3:** mehr Optimierungen, insbesondere function inlining
- **-Ofast:** noch mehr Optimierungen kann aber Ergebnisse verändern!
- **-march=native:** Aktiviert die passenden AVX/SSE/SSE3 Instruktionen
- **-fexpensive-optimizations:** Noch mehr Optimierungen
- **-funroll-loops** Schreibt Schleifen aus.
- **-ftree-vectorize:** Automatische SSE-Vektorisierung

72. Worauf muss man bei Parallelisierung achten?

Zu beachten ist, dass manche Variablen private gesetzt werden müssen und dass der Speicherzugriff eindeutig sein muss, also Ergebnisse separat gespeichert werden müssen.

73. Wie kann man Code optimieren?

Bsp.: Call by Reference, Return by Reference, Inlining von Funktionen