

# Übung 12 - Numerisches Programmieren

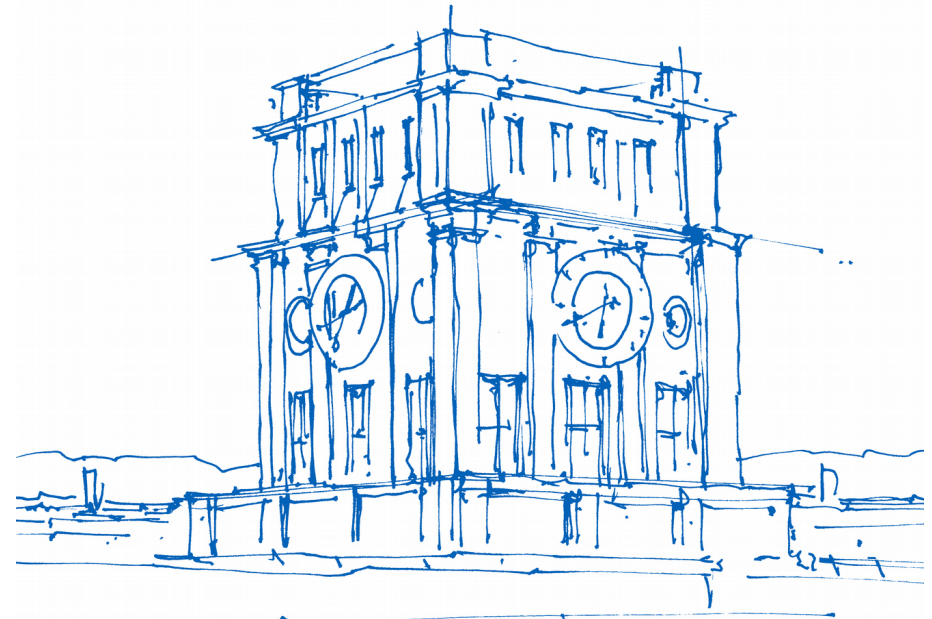
Michael Obersteiner

Technische Universität München

Fakultät für Informatik

Lehrstuhl für Wissenschaftliches Rechnen

Garching, 10. Februar 2021



*Uhrenturm der TUM*

# Recap – Iterative Verfahren

- Keine direkte Berechnung → **Iterative Annäherung** an Lösung
- Anfang mit Startwert  $x_0$  (Beliebig gewählt oder „educated guess“)
- Berechnungsvorschrift  $\Phi(x_k) = x_{k+1}$
- Stopp wenn ausreichend nah an Lösung:  $x_0 \xrightarrow{\Phi} x_1 \xrightarrow{\Phi} \dots \xrightarrow{\Phi} x_n$
- Fixpunkt bei unendlich viele Iterationen:  $\lim_{k \rightarrow \infty} x_k = x^* = \Phi(x^*)$
- Anwendung lineare Gleichungssysteme  $Ax = b$ :
  - Splitting Verfahren:
    - $\Phi(x) = x + M^{-1}(b - Ax)$   **$O(N^2)$**
    - Fehlerschätzung durch Residuum:  $r^{(k)} = b - Ax_k = Ax - Ax_k = A(x - x_k) = A\epsilon$
    - Je „ähnlicher“ M zu A desto schneller Konvergenz
    - Je teurer Invertierung von M desto teurer Iteration
    - Beispiele:
      - $M = \text{diag}(A)$  → Jacobi Verfahren
      - $M = L(A)$  (linke untere Dreiecksmatrix inklusive Diagonale) → Gauss-Seidel
  - Verfahren des steilsten Abstiegs:
    - Gradientenverfahren zur Approximation der Lösung x

# Recap – Verfahren des steilsten Abstiegs

- Indirektes Lösen des Gleichungssystems über Optimierungsproblem:

$$f(x) = \frac{1}{2}x^T Ax - b^T x$$

$$\nabla f(x) = Ax - b \stackrel{!}{=} 0 \Rightarrow Ax = b$$

- Wenn A **symmetrisch und positiv definit (spd)** dann ist Extremum ein Minimum!  
→ Abstiegsverfahren zur Suche des Minimum (Funktioniert **nur bei spd Matrix**)
- Negativer Gradient zeigt in Richtung des steilsten Abstiegs.
- Iterationsverfahren:

$$\Phi(x) = x + \alpha(b - Ax) \Rightarrow x^{(k+1)} = x^{(k)} + \alpha(b - Ax^{(k)})$$

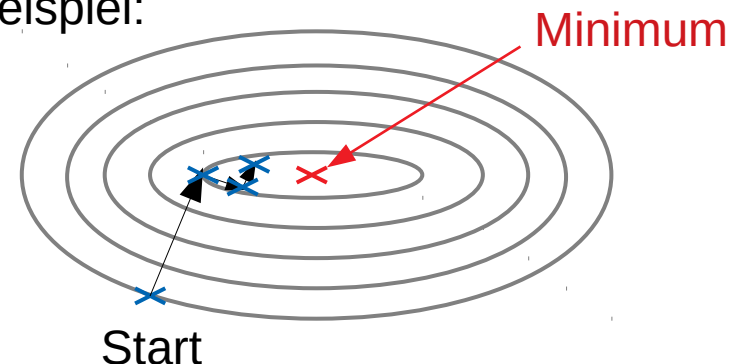
- Fehlerschätzer:

$$r^{(k)} = b - Ax^{(k)}$$

- Optimale Schrittweite:

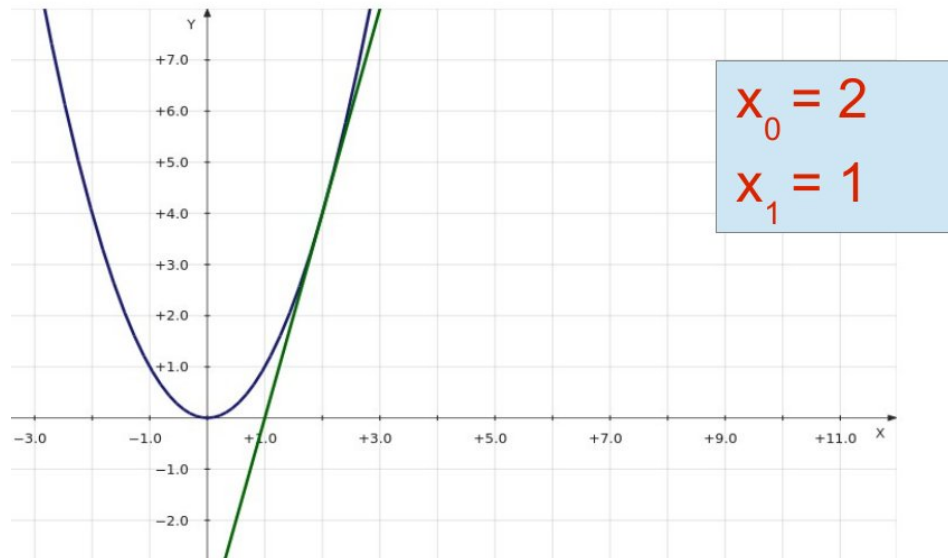
$$\alpha^{(i)} = \frac{r^{(i)T} r^{(i)}}{r^{(i)T} A r^{(i)}}$$

Beispiel:



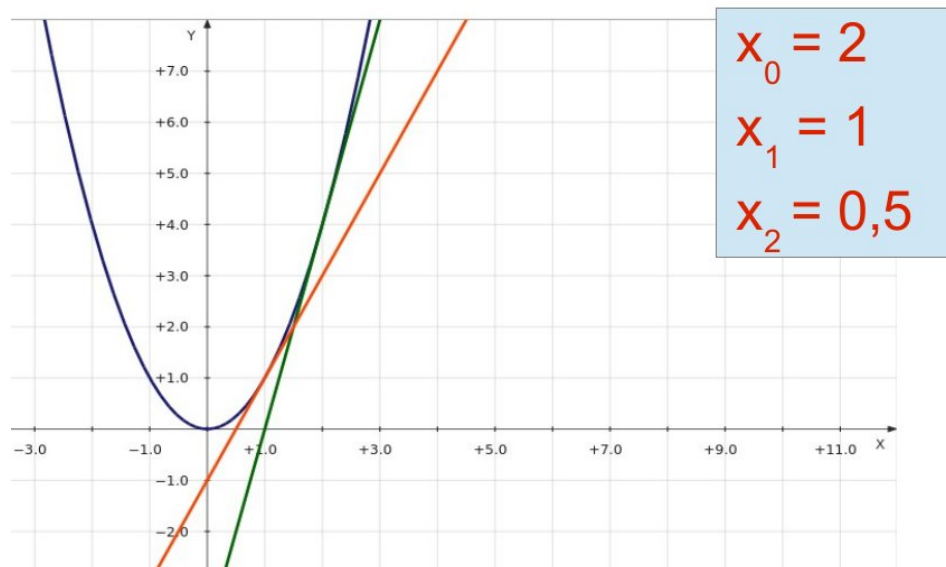
# Recap – Newton Verfahren

- Nullstellenberechnung einer (nicht-linearen) Funktion:  $f(x) \stackrel{!}{=} 0$
- Idee: Approximation der Funktion mittels Tangente  
→ Nullstelle der Tangente als nächste Annäherung
- Vorschrift: 
$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$



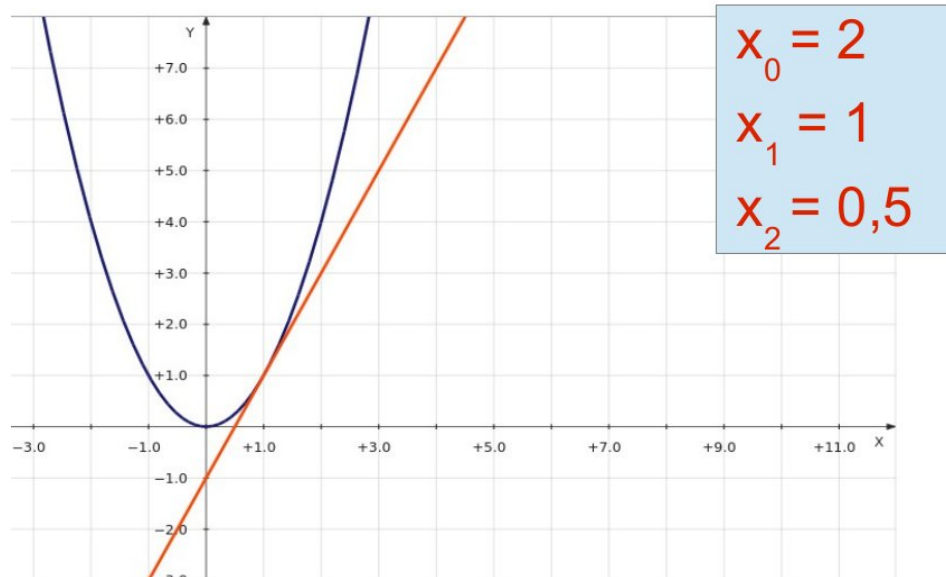
# Recap – Newton Verfahren

- Nullstellenberechnung einer (nicht-linearen) Funktion:  $f(x) \stackrel{!}{=} 0$
- Idee: Approximation der Funktion mittels Tangente  
→ Nullstelle der Tangente als nächste Annäherung
- Vorschrift: 
$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$



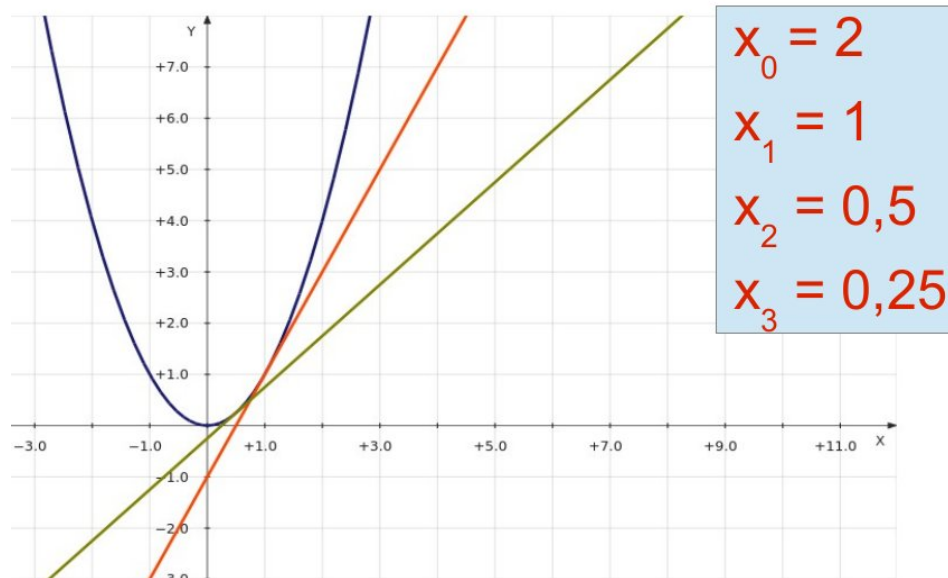
# Recap – Newton Verfahren

- Nullstellenberechnung einer (nicht-linearen) Funktion:  $f(x) \stackrel{!}{=} 0$
- Idee: Approximation der Funktion mittels Tangente  
→ Nullstelle der Tangente als nächste Annäherung
- Vorschrift: 
$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$



# Recap – Newton Verfahren

- Nullstellenberechnung einer (nicht-linearen) Funktion:  $f(x) \stackrel{!}{=} 0$
- Idee: Approximation der Funktion mittels Tangente  
→ Nullstelle der Tangente als nächste Annäherung
- Vorschrift: 
$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$



# Recap – Newton Verfahren

- Nullstellenberechnung einer (nicht-linearen) Funktion:  $f(x) \stackrel{!}{=} 0$
- Idee: Approximation der Funktion mittels Tangente  
→ Nullstelle der Tangente als nächste Annäherung
- Iterationsvorschrift:  $\Phi(x) = x - \frac{f(x)}{f'(x)} \Rightarrow x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$
- Vorteile:
  - Quadratische Konvergenz **bei einfacher Nullstelle** (da  $\Phi'(x^*) = 0$ )
  - Bei mehrfachen Nullstellen existieren Modifikationen für quadratische Konvergenz
- Nachteil:
  - Nur lokale Konvergenz (Startpunkt „**muss bereits in der Nähe liegen**“)  
→ guter Anfangspunkt bekannt oder Annäherung mit anderem Verfahren



# Übung 12 – Eigenwerte

- Eigenwerte definieren den Streckungsfaktor des zugehörigen Eigenvektors nach Anwendung der Matrix:  $Av = \lambda v$

- Bei Symmetrischen Matrizen gibt es Basis aus Eigenvektoren so dass:

$$x = \sum_{i=1}^n \tilde{x}_i v_i \Rightarrow Ax = \sum_{i=1}^n \lambda_i \tilde{x}_i v_i$$

- Berechnung der Eigenwerte über Nullstellen des charakteristischen Polynoms:

$$Av = \lambda v$$

$$Av - \lambda v = \mathbf{0}$$

$$(A - \lambda I_n)v = \mathbf{0}; \quad v \neq \mathbf{0}$$

$$\Rightarrow \det(A - \lambda I_n) = 0$$

- Zugehöriger Eigenvektor wird dann über Gleichungssystem bestimmt (vorletzte Zeile)
- Beobachtung: Eigenwerte liegen in Gerschgorin Kreisscheiben (Siehe Übung 2)

# Übung 12 – iterative Berechnung der Eigenwerte

- Beobachtung:
  - Mithilfe von Eigenvektor lässt sich Eigenwert berechnen
    - Rayleigh Quotient:  $\lambda = \frac{v^T A v}{v^T v}$

# Übung 12 – iterative Berechnung der Eigenwerte

- Beobachtung:

- Mithilfe von Eigenvektor lässt sich Eigenwert berechnen

→ Rayleigh Quotient:  $\lambda = \frac{v^T A v}{v^T v}$

- Maximaler Eigenwert dominiert bei wiederholter Anwendung von A (symmetrisch):

$$x = \sum_{i=1}^n \tilde{x}_i v_i \Rightarrow Ax = \sum_{i=1}^n \lambda_i \tilde{x}_i v_i \Rightarrow A^k x = \sum_{i=1}^n \lambda_i^k \tilde{x}_i v_i \xrightarrow{\text{large } k} \approx \lambda_{max}^k v_{max} \tilde{x}_{max}$$

- Power Iteration (für maximalen Eigenwert):

$$v^{(k+1)} = A v^{(k)}; \quad \lambda^{(k)} = \frac{v^{(k)T} A v^{(k)}}{v^{(k)T} v^{(k)}}$$

Konvergenzrate:  $\left| \frac{\lambda_{max-1}}{\lambda_{max}} \right|$

- Shiften der Matrix um alternative Eigenwerte zum größten zu machen:

$$\lambda(A - \mu I_n) = \lambda(A) - \mu$$

→ Iteration mit:  $\tilde{A} = A - \mu I_n$

- Iteration mit Inversen um gegen kleinsten Eigenwert zu iterieren