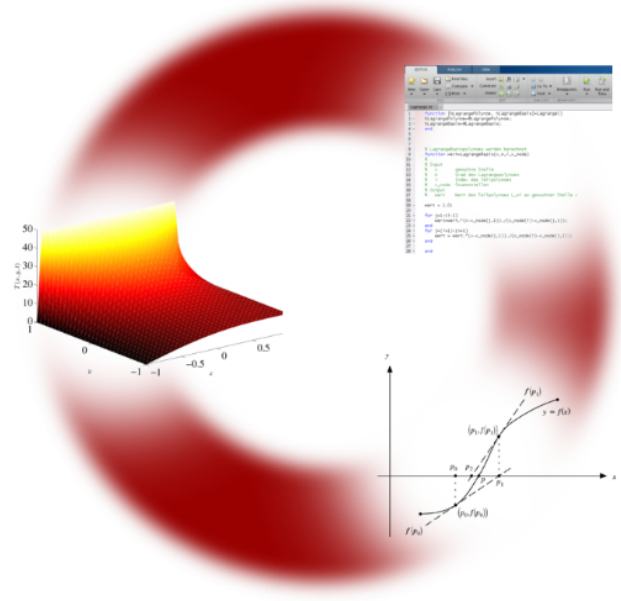
Lehrstuhl für Numerische Mechanik
TUM School of Engineering and Design
Technische Universität München

10. Lösung von nichtlinearen Gleichungssystemen

A Practical Course in Numerical Methods for Engineers

Barbara Wirthl, M.Sc.
Technische Universität München
Lehrstuhl für Numerische Mechanik



9. Aufgabenblatt: Vergleich der Methoden

- Was haben Sie beobachtet?
- Welche Vor- bzw. Nachteile der Methoden ergeben sich daraus?

$$\boldsymbol{A} = \begin{pmatrix} 1 & -2 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ -2 & \phi & -2 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & -2 & \phi & -2 & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & -2 & \phi & -2 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & -2 & \phi & -2 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & -2 & \phi \end{pmatrix} \qquad \boldsymbol{b} = \mathbf{1}$$

$$\phi = [10.0, 6.0, 5.1, 5.01, 5.001, 5.00001, 5.0000001, \dots]$$

Pingo

PINGO

Gauß'sches Eliminationsverfahren

Wie hoch ist der Rechenaufwand für die Durchführung des Gauß'schen Eliminationsverfahren für eine große Matrix $(n \times n)$?

- $\mathcal{O}(n)$
- $\mathcal{O}(n^2)$
- $\mathcal{O}(n^3)$ ✓

Reminder:

1. Berechne Multiplikatoren $m_{ji} = \frac{a_{ji}}{a_{ii}}$ für $j = i+1, i+2, \ldots, n$
2. Eliminiere die Matrixzeilen A_j mit

$$(A_j - m_{ji} A_i) \to (A_j)$$

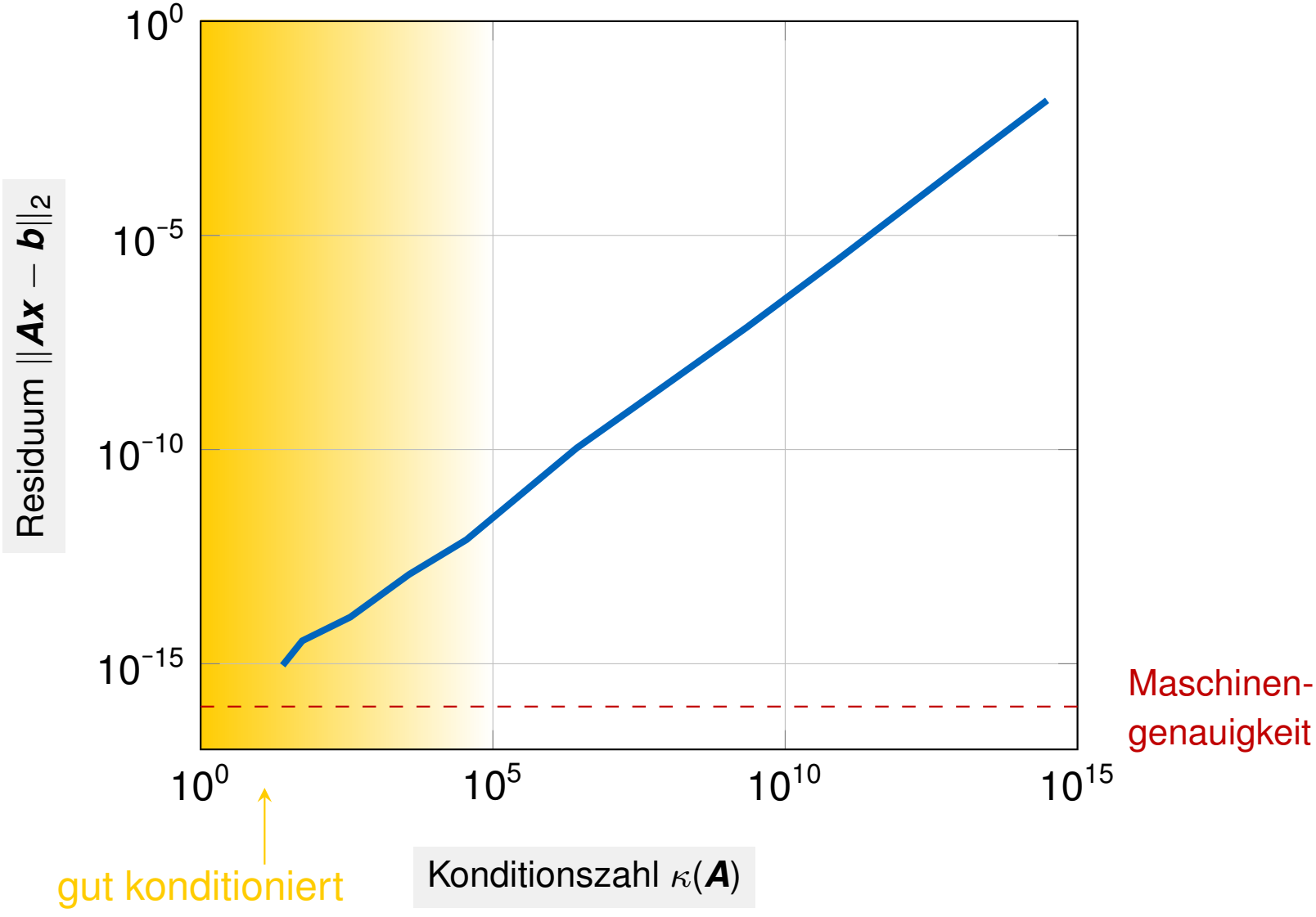
$\mathcal{O}(n^2)$

$\mathcal{O}(n)$

$\mathcal{O}(n^3)$

Genauigkeit des Gauß-Algorithmus

Konditionszahl: $\kappa(\boldsymbol{A}) = \|\boldsymbol{A}\| \, \|\boldsymbol{A}^{-1}\|$



Genauigkeit des Gauß-Algorithmus: Pivotisierung

$$(A_j - m_{ji} A_i) \rightarrow (A_j) \quad \text{mit} \quad m_{ij} = \frac{a_{ji}}{a_{ii}}$$

Pivotelement: Eintrag auf der Diagonalen a_{ii}

- muss ungleich Null sein
- sehr kleine Pivotelemente führen zu Rundungsfehlern

Lösung: Pivotisierung

- Zeilen tauschen, betragsmäßig größtes Pivotelement wählen

PINGO

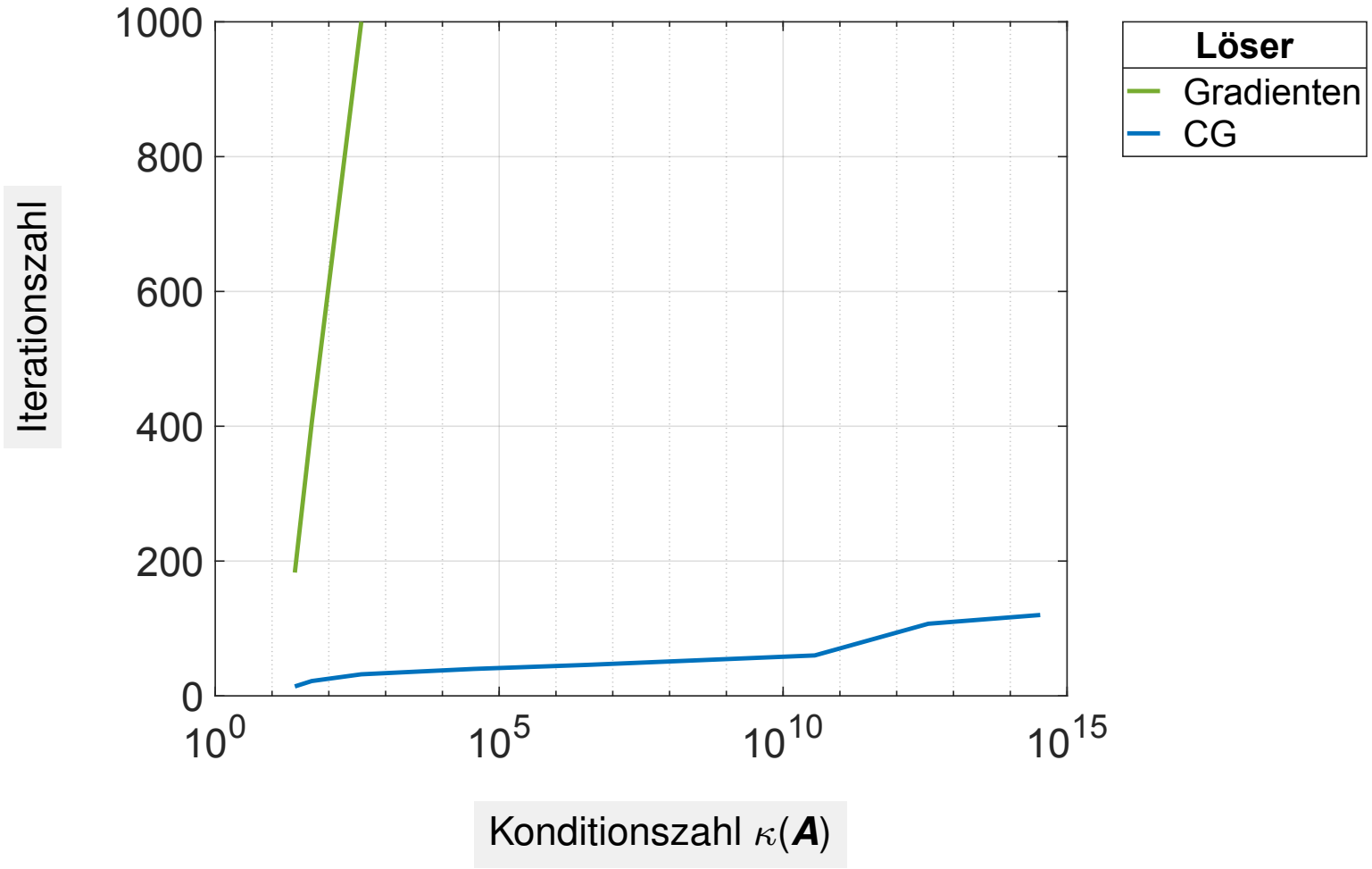
Vergleich Gauß, Gradienten & CG

Welche der folgenden Aussagen sind korrekt? Mehrere Antworten sind möglich.

- Das Gradientenverfahren benötigt mehr Iterationen als das CG-Verfahren.
- Die Lösungszeit des Gauß'schen Eliminationsverfahren ist abhängig von der Kondition der Matrix.
- Für kleine Systeme kann das Gauß'sche Eliminationsverfahren schneller sein als das CG-Verfahren.

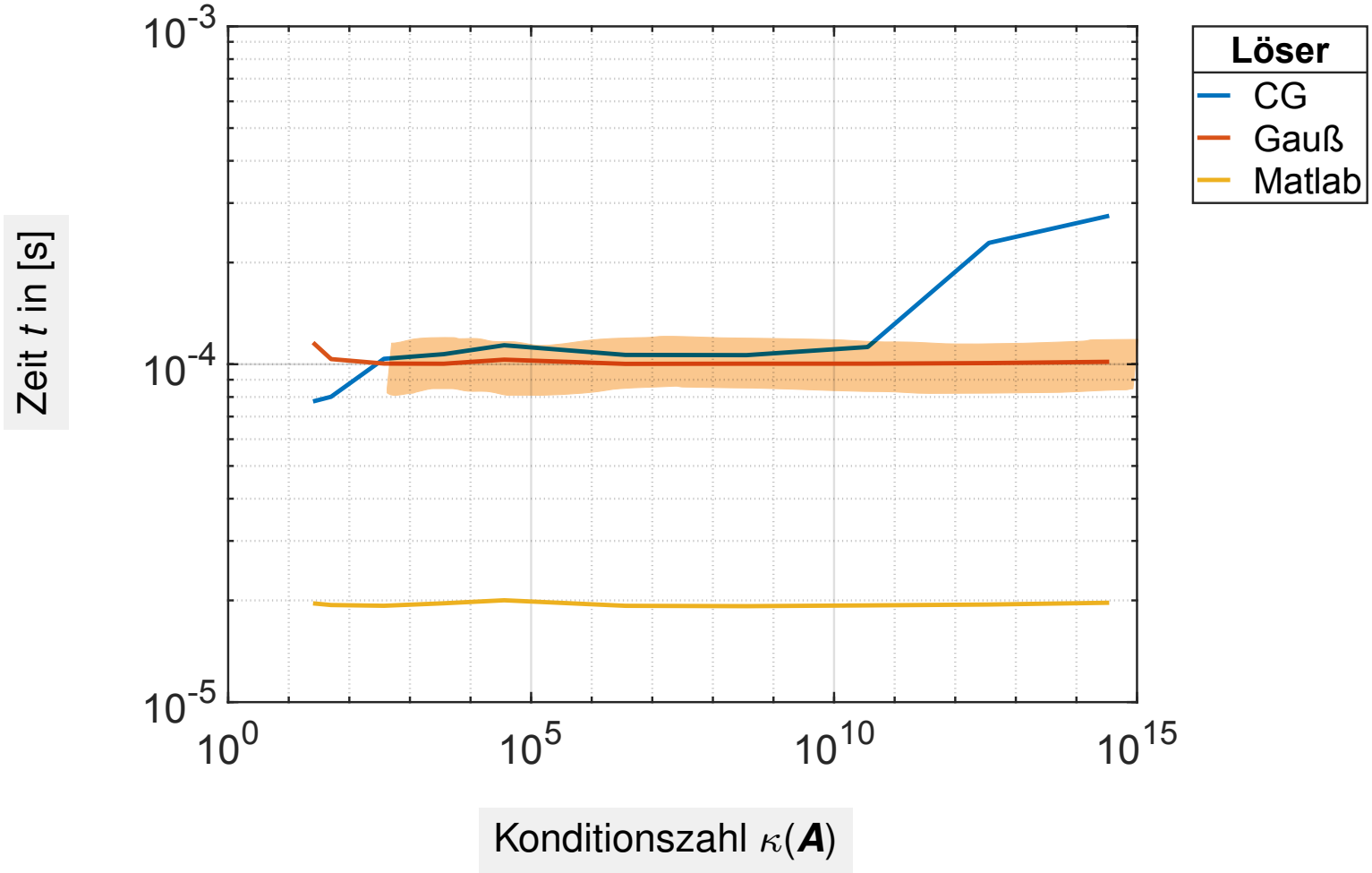
Iterationszahl – Gradienten vs. CG

- Das Gradientenverfahren benötigt mehr Iterationen als das CG-Verfahren.



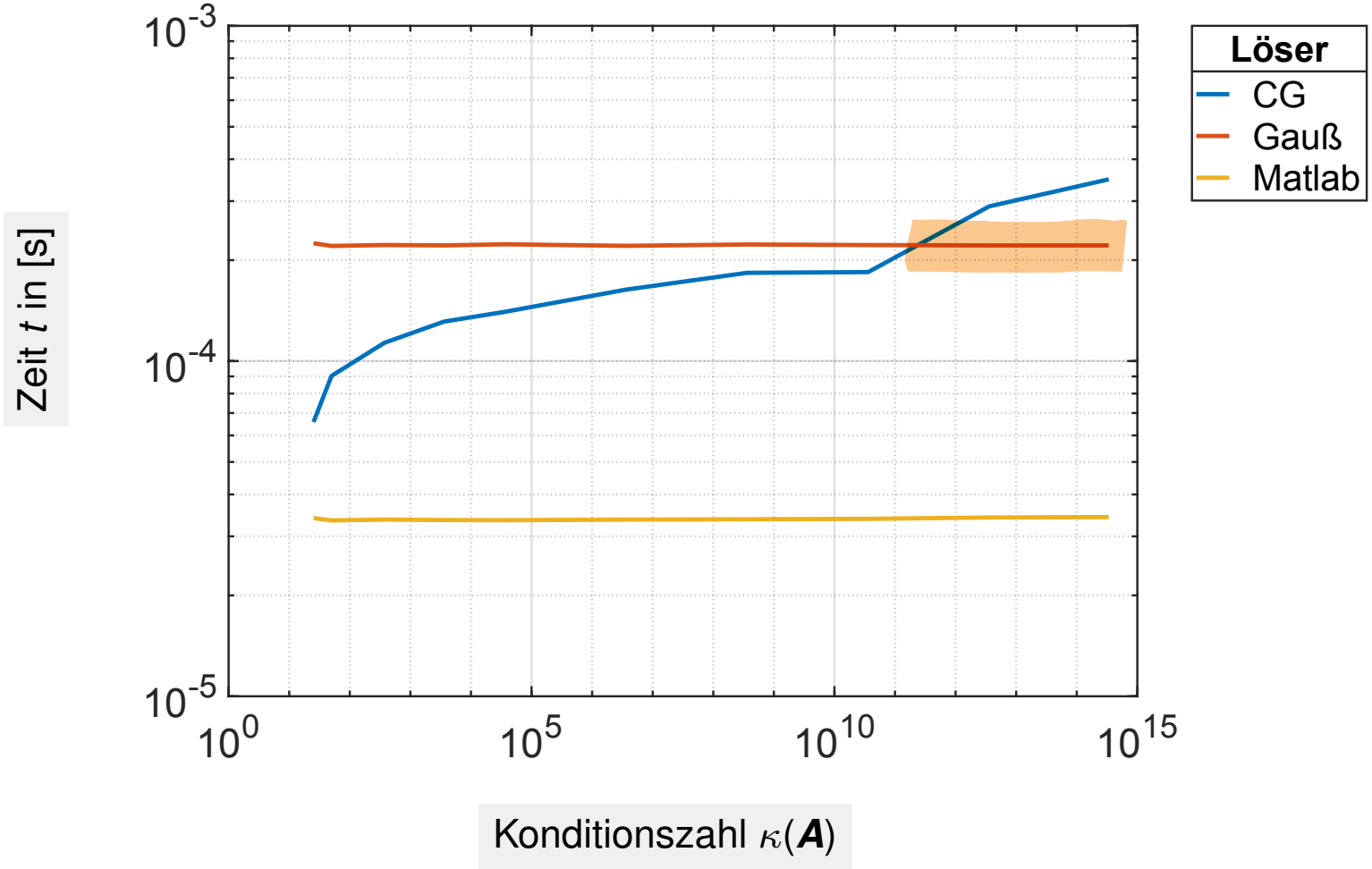
Lösungszeit für $n = 30$

- Für kleine Systeme kann das Gauß'sche Eliminationsverfahren schneller sein als das CG-Verfahren.



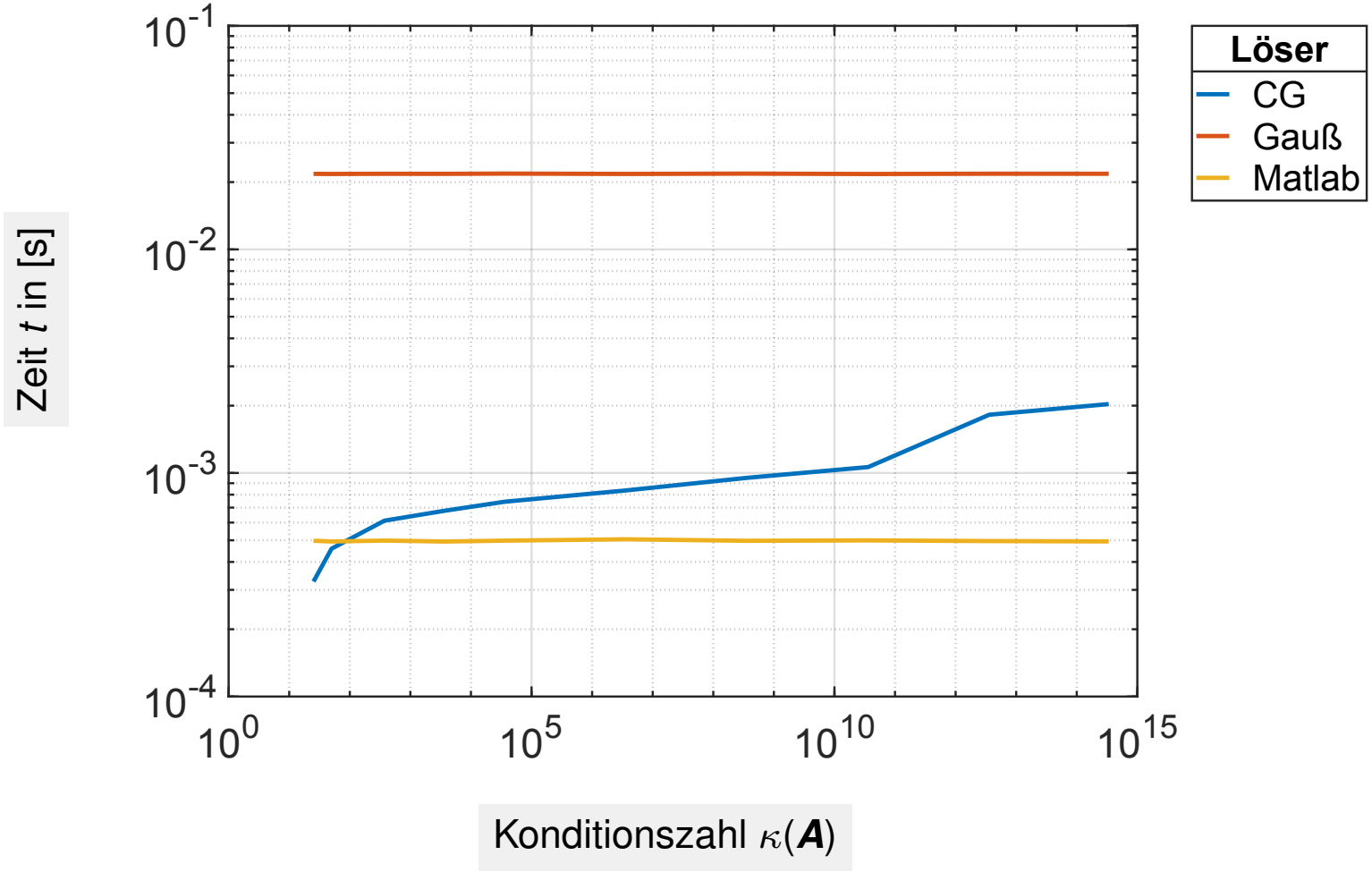
Lösungszeit für $n = 50$

- Für kleine Systeme kann das Gauß'sche Eliminationsverfahren schneller sein als das CG-Verfahren.

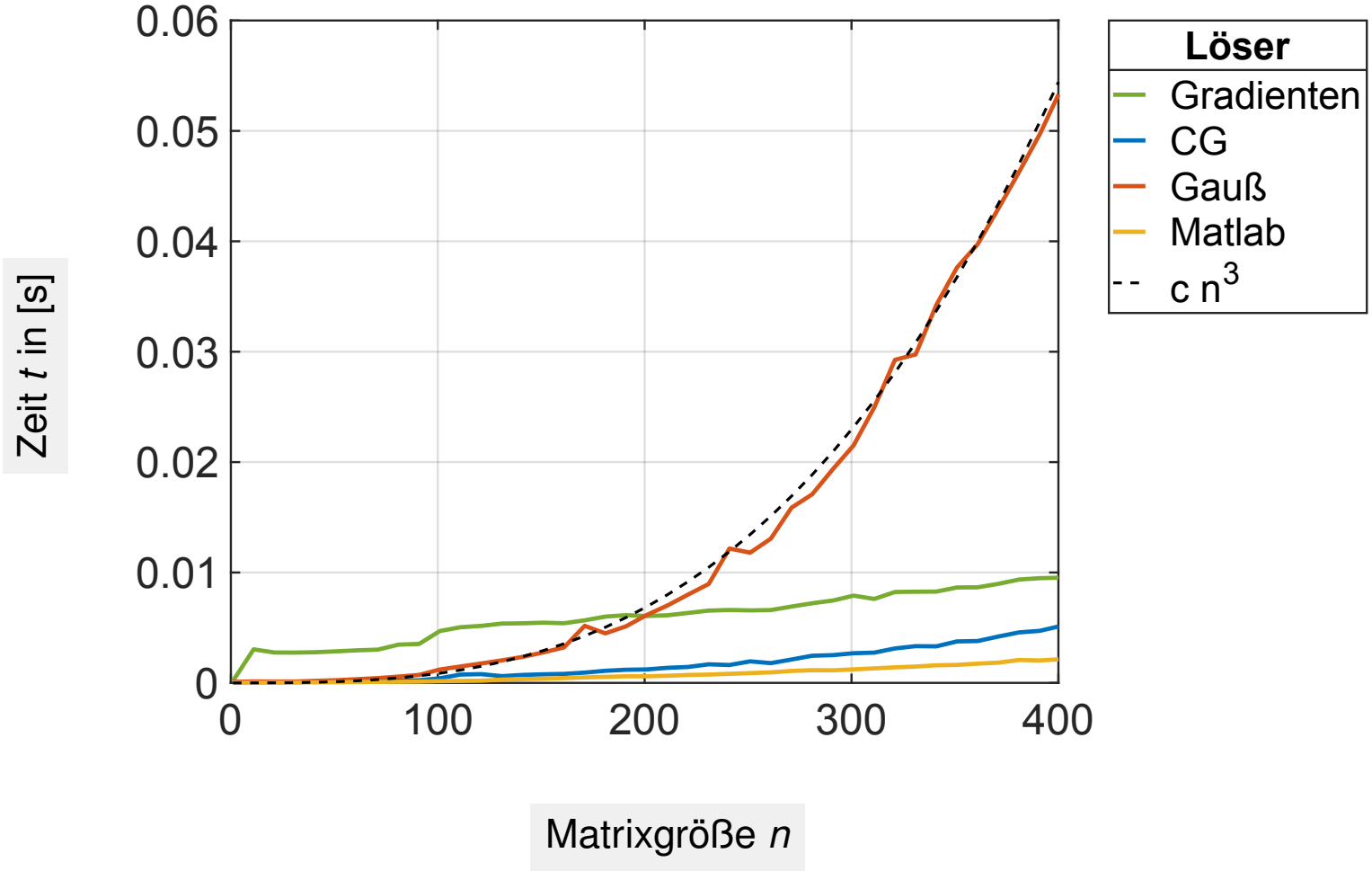


Lösungszeit für $n = 300$

- Die Lösungszeit des Gauß'schen Eliminationsverfahren ist abhängig von der Kondition der Matrix.



Lösungszeit für verschiedene Matrixgrößen



Aufgabenblatt 10

Aufgabe 1: Lösung von Gleichungen mit einer Variablen

Bestimmen Sie die Nullstelle der Funktion $f(x)$ mit Hilfe des Newton-Verfahrens mit Startwert x_0. Die Konvergenz des Verfahrens soll mit Hilfe des absoluten Fehlers des Funktionswertes mit der Toleranz 10^{-12} überprüft werden.

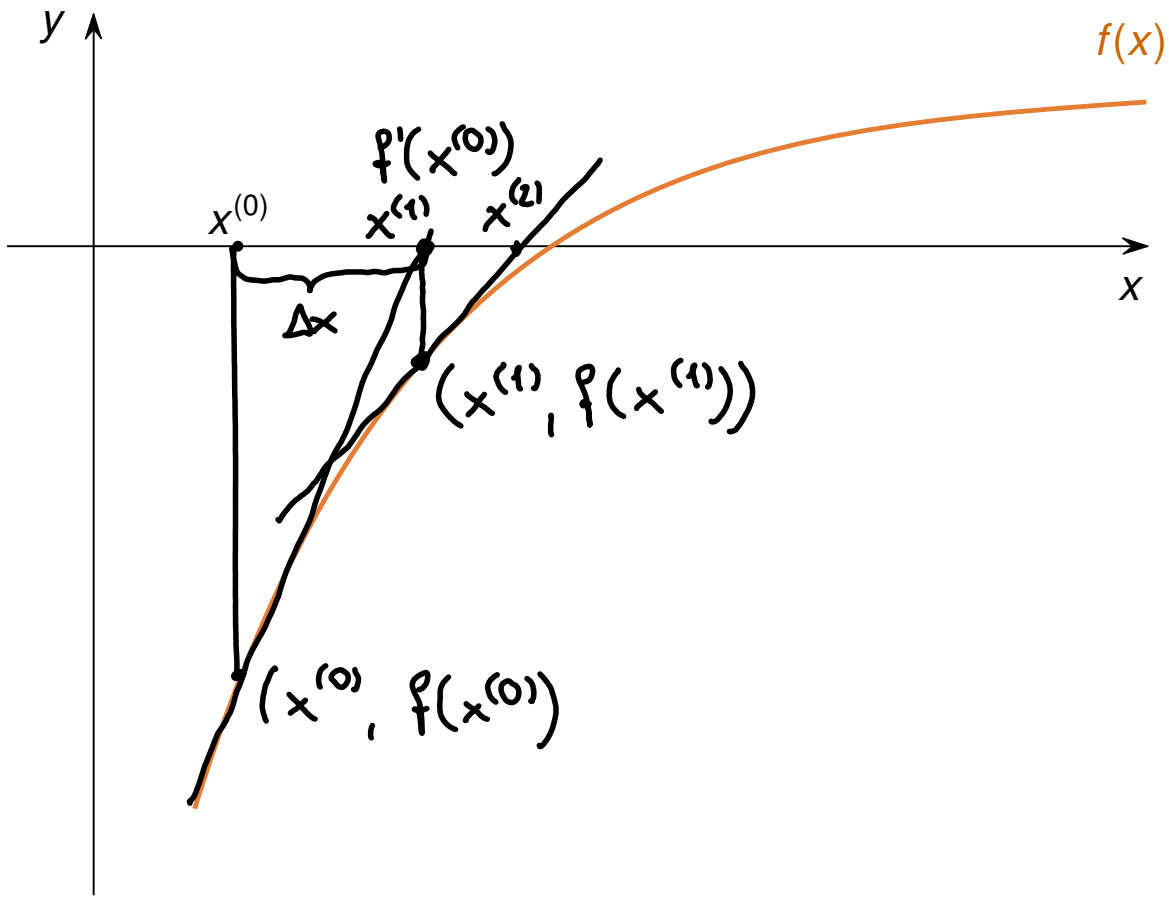
1. Gegeben ist die Funktion $f(x) = x^3 + 3^x$ mit $x_0 = 0.0$.

2. Gegeben ist die Funktion $f(x) = \arctan(x)$ mit $x_0 = 2.0$ sowie $x_0 = 1.0$.

Newton-Verfahren: 1D

Problem: $f(x) = 0 \quad \rightarrow$

$$x^{(k+1)} = x^{(k)} + \Delta x = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}$$



Newton-Verfahren: Mehrdimensional

1D: $x^{(k+1)} = x^{(k)} + \Delta x$ mit $\Delta x = -\frac{f\left(x^{(k)}\right)}{f'\left(x^{(k)}\right)}$

Mehrdimensional:
Nichtlineares Gleichungssystem der Form $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0}$

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \Delta\boldsymbol{x}^{(k)}$$

$$\text{mit } \Delta\boldsymbol{x}^{(k)} = -\boldsymbol{J}\left(\boldsymbol{x}^{(k)}\right)^{-1} \boldsymbol{F}\left(\boldsymbol{x}^{(k)}\right)$$

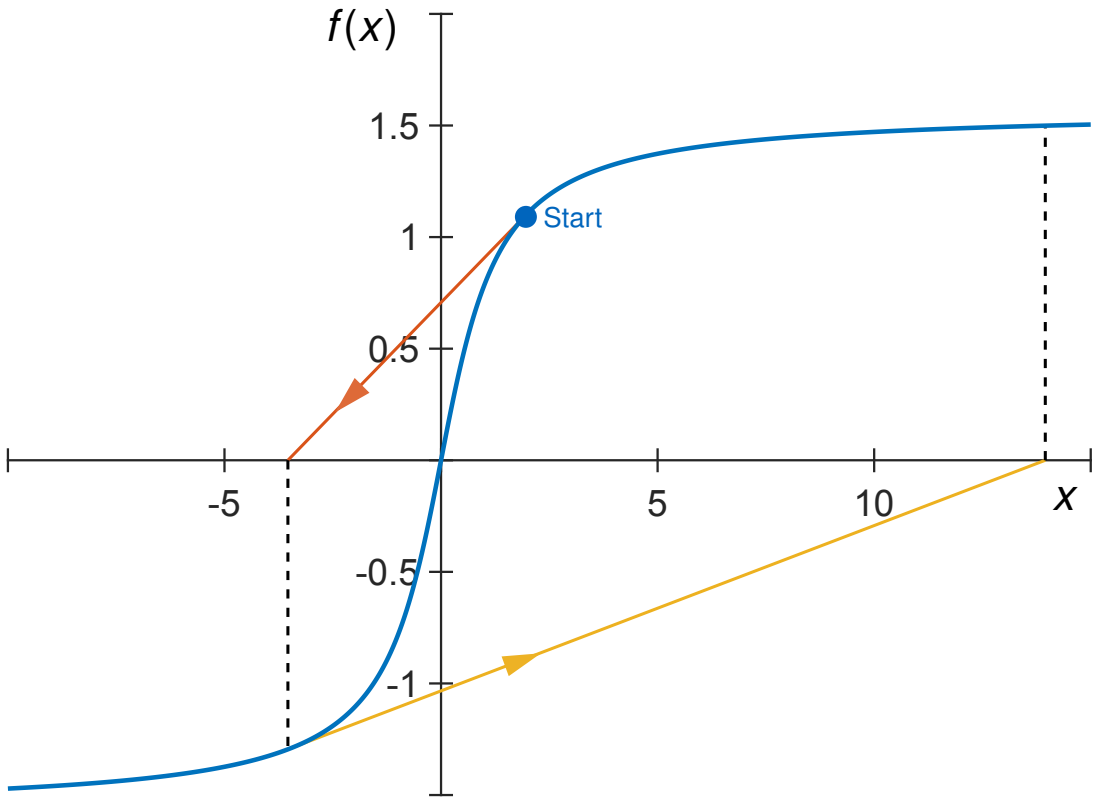
Mit der Jacobi-Matrix

$$\boldsymbol{J}(\boldsymbol{x}) = \begin{pmatrix} \frac{\partial f_1(\boldsymbol{x})}{\partial x_1} & \frac{\partial f_1(\boldsymbol{x})}{\partial x_2} & \cdots & \frac{\partial f_1(\boldsymbol{x})}{\partial x_{(n-1)}} & \frac{\partial f_1(\boldsymbol{x})}{\partial x_n} \\ \frac{\partial f_2(\boldsymbol{x})}{\partial x_1} & \frac{\partial f_2(\boldsymbol{x})}{\partial x_2} & \cdots & \frac{\partial f_2(\boldsymbol{x})}{\partial x_{(n-1)}} & \frac{\partial f_2(\boldsymbol{x})}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial f_{(n-1)}(\boldsymbol{x})}{\partial x_1} & \frac{\partial f_{(n-1)}(\boldsymbol{x})}{\partial x_2} & \cdots & \frac{\partial f_{(n-1)}(\boldsymbol{x})}{\partial x_{(n-1)}} & \frac{\partial f_{(n-1)}(\boldsymbol{x})}{\partial x_n} \\ \frac{\partial f_n(\boldsymbol{x})}{\partial x_1} & \frac{\partial f_n(\boldsymbol{x})}{\partial x_2} & \cdots & \frac{\partial f_n(\boldsymbol{x})}{\partial x_{(n-1)}} & \frac{\partial f_n(\boldsymbol{x})}{\partial x_n} \end{pmatrix}$$

Konvergenzverhalten des Newton-Verfahrens

Beispiel Arkustangens:

$f(x) = \arctan(x)$ mit $x_0 = 2.0$

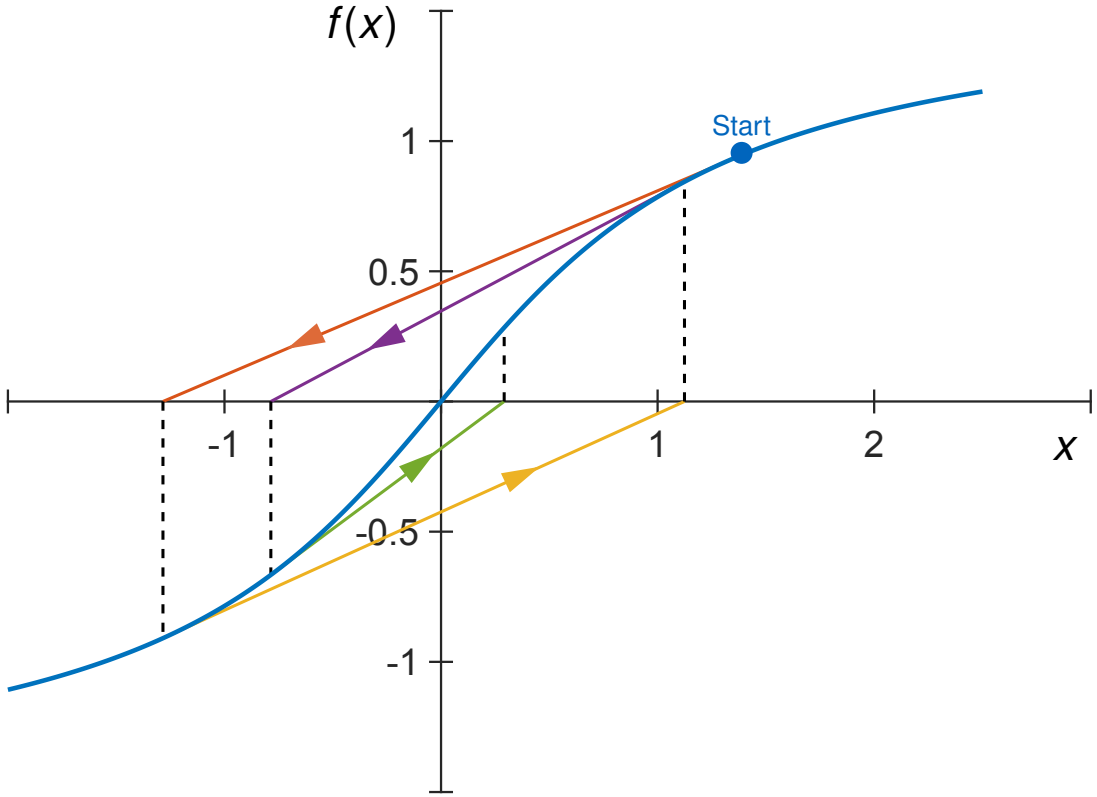


$x_0 = 2.0$
$x_1 = -3.54$
$x_2 = 13.95$
$x_3 = -279.34$
$\vdots$
$x_{10} = \text{Inf}$
$x_{11} = \text{NaN}$

Konvergenzverhalten des Newton-Verfahrens

Beispiel Arkustangens:

$f(x) = \arctan(x)$ mit $x_0 = 2.0$ 135



$$x_0 = 1.35$$
$$x_1 = -1.28$$
$$x_2 = 1.12$$
$$x_3 = -0.79$$
$$\vdots$$
$$x_6 = 2.86 \cdot 10^{-6}$$
$$x_7 = 1.56 \cdot 10^{-17}$$

Lokale Konvergenz:
Startwert muss *ausreichend nahe* an der Lösung sein

Aufgabenblatt 10

Aufgabe 2: Newton-Verfahren für nichtlineare Gleichungssysteme

Gegeben ist die stationäre Wärmeleitungsgleichung in 2D:

$$-\nabla \cdot (\lambda \nabla T) = \dot{q}(T) \quad \text{in } \Omega$$

$$\text{mit} \quad T = T_D \quad \text{auf} \quad \Gamma_D, \qquad \lambda \nabla T \cdot \boldsymbol{n} = 0 \quad \text{auf} \quad \Gamma_N$$

Die Wärmeleitfähigkeit λ ist eine vom Material abhängige bekannte Größe. Es soll angenommen werden, dass an den Rändern (Γ_D-Dirichlet Rand) entweder die Temperatur T vorgegeben wird oder die Ränder (Γ_N-Neumann Rand) adiabat sind ($\lambda \nabla T \cdot \boldsymbol{n} = 0$).

Durch eine temperaturabhängige, exotherme, chemische Reaktion wird Wärme in das System eingebracht. Folgende Gleichung für $\dot{q}(T)$ soll diesen Effekt modellieren:

$$\dot{q}(T) = c_1 e^{-\frac{c_2}{T}}$$

Diskretisieren Sie die Gleichung mithilfe der Finiten-Elemente-Methode mit bilinearen Viereckselementen.

Schwache Form

Starke Form der stationären Wärmeleitungsgleichung:

$$-\nabla \cdot (\lambda \nabla T) = c_1 e^{-\frac{c_2}{T}} \quad \text{in } \Omega$$

1. Schritt: Multiplikation mit der Testfunktion v und Integration über das Gebiet Ω

$$-\int_\Omega v \nabla \cdot (\lambda \nabla T)\, \mathrm{d}\boldsymbol{x} = \int_\Omega v c_1 e^{-\frac{c_2}{T}}\, \mathrm{d}\boldsymbol{x}$$

2. Schritt: Partielle Integration des Terms auf der linken Seite

$$\int_\Omega \lambda \nabla v \cdot \nabla T\, \mathrm{d}\boldsymbol{x} - \int_\Gamma \lambda v \nabla T \cdot \boldsymbol{n}\, \mathrm{d}\gamma = \int_\Omega v c_1 e^{-\frac{c_2}{T}}\, \mathrm{d}\boldsymbol{x}$$

3. Schritt: Ausnutzung der Randbedingungen $\Gamma = \Gamma_N + \Gamma_D$

$$\int_\Omega \lambda \nabla v \cdot \nabla T\, \mathrm{d}\boldsymbol{x} = \int_\Omega v c_1 e^{-\frac{c_2}{T}}\, \mathrm{d}\boldsymbol{x}$$

Endlich-dimensionaler Ansatzraum

Ansatzfunktionen für Testfunktion und Temperaturfeld:

$$v^h(\boldsymbol{x}) = \sum_i N^i(\boldsymbol{x})\hat{v}_i \qquad T^h(\boldsymbol{x}) = \sum_j N^j(\boldsymbol{x})\hat{T}_j$$

Einsetzen:

$$\int_\Omega \lambda \nabla v^h \cdot \nabla T^h \,\mathrm{d}\boldsymbol{x} = \int_\Omega v^h c_1 e^{-\frac{c_2}{T^h}} \,\mathrm{d}\boldsymbol{x}$$

$$\sum_i \hat{v}_i \left[\sum_j \int_\Omega \lambda \nabla N^i(\boldsymbol{x}) \cdot \nabla N^j(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} \hat{T}_j - \int_\Omega N^i(\boldsymbol{x}) c_1 e^{-\frac{c_2}{\sum_j N^j(\boldsymbol{x})\hat{T}_j}} \mathrm{d}\boldsymbol{x} \right] = 0$$

Muss für beliebige Werte $\hat{v}_i$ gelten:

$$\sum_j \int_\Omega \lambda \nabla N^i(\boldsymbol{x}) \cdot \nabla N^j(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} \hat{T}_j - \int_\Omega N^i(\boldsymbol{x}) c_1 e^{-\frac{c_2}{\sum_j N^j(\boldsymbol{x})\hat{T}_j}} \mathrm{d}\boldsymbol{x} = 0 \qquad \forall\, i$$

Newton-Verfahren

Newton Verfahren für Gleichungssystem $\boldsymbol{F}(\boldsymbol{T}) = \boldsymbol{0}$:

$$\boldsymbol{J}\left(\boldsymbol{T}^{(k)}\right)\Delta\boldsymbol{T}^{(k)} = -\boldsymbol{F}\left(\boldsymbol{T}^{(k)}\right)$$

$$\boldsymbol{T}^{(k+1)} = \boldsymbol{T}^{(k)} + \Delta\boldsymbol{T}^{(k)} \qquad k = 0, 1, 2, \ldots$$

In unserem Fall:

$$\boldsymbol{F}(\boldsymbol{T}) = \sum_j \int_\Omega \lambda \nabla N^i(\boldsymbol{x}) \cdot \nabla N^j(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}\, \hat{T}_j - \int_\Omega N^i(\boldsymbol{x}) c_1 e^{-\frac{c_2}{\sum_j N^j(\boldsymbol{x})\hat{T}_j}} \,\mathrm{d}\boldsymbol{x} = 0 \qquad \forall\, i$$

Newton-Verfahren

Komponenten $F_i\left(\boldsymbol{T}^{(k)}\right)$ von $\boldsymbol{F}\left(\boldsymbol{T}^{(k)}\right)$:

$$F_i\left(\boldsymbol{T}^{(k)}\right) = \sum_j \int_\Omega \lambda \nabla N^i(\boldsymbol{x}) \cdot \nabla N^j(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}\, \hat{T}_j^{(k)} - \int_\Omega N^i(\boldsymbol{x}) c_1 e^{-\frac{c_2}{\sum_j N^j(\boldsymbol{x}) \hat{T}_j^{(k)}}} \,\mathrm{d}\boldsymbol{x}$$

Komponenten $J_{ij}\left(\boldsymbol{T}^{(k)}\right) = \frac{\partial F_i}{\partial \hat{T}_j}$ der Jacobi-Matrix $\boldsymbol{J}\left(\boldsymbol{T}^{(k)}\right)$:

$$J_{ij}\left(\boldsymbol{T}^{(k)}\right) = \int_\Omega \lambda \nabla N^i(\boldsymbol{x}) \cdot \nabla N^j(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} - \int_\Omega N^i(\boldsymbol{x}) N^j(\boldsymbol{x}) \frac{c_1 c_2}{\left(\sum_j N^j(\boldsymbol{x}) \hat{T}_j^{(k)}\right)^2} e^{-\frac{c_2}{\sum_j N^j(\boldsymbol{x}) \hat{T}_j^{(k)}}} \,\mathrm{d}\boldsymbol{x}$$

Umsetzung im Programm

Bestimmung von $J_{ij}\left(\boldsymbol{T}^{(k)}\right)$ und $-F_i\left(\boldsymbol{T}^{(k)}\right)$ analog zu den Arbeitsblättern 7 & 8:

- Aufteilen des Integrals in Elemente $\rightarrow \Omega^{(e)}$
- Transformation auf das Referenzelement $\rightarrow \Omega^{(e)}_{ref}$
- Integration mit der Gauß-Quadratur

$$\sum_j J_{ij}\left(\boldsymbol{T}^{(k)}\right)\,\Delta\hat{T}_j^{(k)} = -F_i\left(\boldsymbol{T}^{(k)}\right)$$

$$T_j^{(k+1)} = T_j^{(k)} + \Delta T_j^{(k)} \qquad k = 0, 1, 2, \ldots .$$

Umsetzung im Programm: Dirichlet-Randbedingungen

```
[dbcsysmat,dbcrhs] = assignDBC_nlin(sysmat,rhs, dbc)
```

- Berücksichtigt Dirichlet-Ränder in der Systemmatrix bzw. im Systemvektor
- Wenden Sie folgendes Vorgehen auf alle Zeilen mit einer Dirichlet-Randbedingung an:
  1. Ersetzen Sie die Zeile der Systemmatrix durch 0-Einträge und setzen Sie den Wert auf der Hauptdiagonalen zu 1.
  2. Ersetzen Sie den entsprechenden Eintrag im Systemvektor durch einen 0-Eintrag.

Und los...

Nächste Tutorsprechstunden:

Montag	23.01. 10:00 – 12:15 Uhr	MW1264
Mittwoch	25.01. 15:30 – 17:45 Uhr	MW1264
Montag	30.01. 10:00 – 12:15 Uhr	MW1264

2. Überprüfung (Achtung Terminänderung!)

Dienstag **31.01.2023** MW1264