Fakultät für Technik
Bereich Informationstechnik

Labor Rechnergestützte Mathematik

Versuch 2

Approximation und Analysis

Laboranleitung

Prof. Dr.-Ing. Stefan Hillenbrand
Yvonne Beck, M.Sc.
Version 2021.1

Inhaltsverzeichnis

1 Versuchsdurchführung

- Arbeiten Sie vorbereitend diese Laboranleitung komplett gründlich durch. Bringen Sie die Vorbereitungsaufgaben, die per Hand zu bearbeiten sind, in die Veranstaltung mit.
- Ziehen Sie bei Bedarf Ihre Vorlesungsmitschriften oder weitere Literatur zu Rate.
- Bearbeiten Sie die Vorbereitungsaufgaben und berücksichtigen Sie Hinweise zur vorherigen Einreichung von Vorbereitungsaufgaben auf Moodle.
- Wiederholen Sie die Inhalte der vorherigen Versuche und mathematische Grundlagen, die bei der Vorbereitung der Aufgaben auftreten.
- Bereiten Sie Themen nach, die Sie im Labor nicht vollständig verstanden haben. Sie können für den weiteren Verlauf des Labors bzw. der zugehörigen Vorlesung relevant sein.
- Berücksichtigen Sie die Hinweise auf der Kursseite.

Die Vorbereitung wird zu Beginn des Versuches kontrolliert.
Sie ist Voraussetzung zum Bestehen des Versuchs.

2 Einleitung

Im ersten Versuch haben Sie erfolgreich den Einstieg in MATLAB bzw. Octave geschafft und sich anhand von Beispielen mit den Grundlagen der MATLAB-Sprache vertraut gemacht. Diese Grundlagen müssen Sie jetzt im zweiten Versuch beherrschen, um sich weitere mathematische Möglichkeiten von MATLAB/Octave zu erschließen:

- Approximation von Messdaten mit Polynomen und anderen Funktionen
- Darstellung und Auswertung von Polynomen
- Suche nach Nullstellen
- Numerische Integration

3 Approximation von Messdaten

MATLAB / Octave ist in der praktischen Ingenieurarbeit ein häufig genutztes Hilfsmittel zur Analyse von Messdaten. Daher haben Sie bereits bei der Einarbeitung in die MATLAB-Sprache Schaubilder der Geschwindigkeit $v(t)$ und des zurückgelegten Weges $s(t)$ eines Fahrzeuges erstellt. Mithilfe der Schaubilder konnten Sie die Messungen dann analysieren und haben festgestellt, dass die Schaubilder näherungsweise den bekannten Verläufen

$$v(t) = a \cdot t \qquad \text{und} \qquad s(t) = \frac{1}{2} \cdot a \cdot t^2$$

für konstante Beschleunigung entsprechen. Um das Fahrmanöver zu identifizieren haben Sie somit implizit den prinzipiellen funktionalen Zusammenhang zwischen den Messwerten ermittelt. Um die Größe a der Beschleunigung zu bestimmen, können Sie

- die gemessene Geschwindigkeit mit einem Polynom 1. Ordnung oder
- den gemessenen Weg mit einem Polynom 2. Ordnung

approximieren und daraus den Parameter a ablesen.

Auch wenn die zugrundeliegenden Gleichungen nicht genau bekannt sind, wird häufig eine Näherungsfunktion gesucht, die den Verlauf der Messdaten approximiert. Als Beispiel hierfür dienen in den folgenden Aufgaben die in Tabelle 1 zusammengestellten Einwohnerzahlen der USA im Zeitraum von 1900 bis 2020.

Tabelle 1: Einwohnerzahlen der USA[1]

Jahr	**Einwohner in Millionen**
1900	76,212
1910	92,228
1920	106,022
1930	122,775
1940	132,165
1950	150,697
1960	179,323
1970	203,302
1980	226,546
1990	248,710
2000	281,422
2010	308,746
2020	332,601

[1] Quelle: https://en.wikipedia.org/wiki/2020_United_States_census , aufgerufen am 26.01.2021, zuletzt geprüft am 05.02.2019.

3.1 Daten zeichnen

Aufgabe 1: Einwohnerzahlen plotten
Bearbeiten Sie diese Programmieraufgabe vor dem Labortermin!

Die Daten aus Tabelle 1 sind in der Datei **`Einwohner_USA_1900_2020.mat`** gespeichert. Erstellen Sie ein Skript **`V2_A1_Einw_USA.m`**, das die Daten einliest und graphisch darstellt:

- Löschen Sie zunächst alle Variablen.
- Laden Sie die Datei mit dem Befehl

  `load('Einwohner_USA_1900_2020.mat')`.

  Die Datei liegt im MATLAB-Datenformat **`.mat`** vor, in dem auch die Variablen gespeichert werden. Sie müssen daher keine Variable für den Dateiinhalt bereitstellen, sondern Sie können nach dem Laden direkt auf die Matrix **`Daten`** zugreifen.
- Erzeugen Sie zwei Vektoren **`Jahr`** und **`Einwohner`** mit den zugehörigen Daten.
- Erstellen Sie ein Schaubild, das die Zahl der Einwohner über den Jahren aufträgt. Gestalten Sie die Grafik so, dass sowohl ein durchgezogener Verlauf sowie die Datenpunkte dargestellt werden.
- Ergänzen Sie eine Beschriftung des Plots mit Titel, Achsenbeschriftung, Gitternetz und LEgende

3.2 Polynomapproximation der Einwohnerdaten

Die Einwohnerzahlen der USA werden alle 10 Jahre durch eine Volkszählung (United States Census[2]) ermittelt. Entsprechend liegen die Daten auch nur im Abstand von 10 Jahren vor. Um Zwischenwerte wie z. B. die Bevölkerungszahl im Jahr 2005 zu ermitteln, kann man zwischen den relativ wenigen bekannten Daten interpolieren. MATLAB erledigt dies beim Zeichnen der Daten mit dem `plot`-Befehl automatisch, indem es die Datenpunkte mit Geradenstücken (lineare Interpolation) verbindet.

Bei der Approximation der Messdaten werden nicht die Datenpunkte miteinander verbunden, sondern es wird eine mathematische Funktion gesucht, die die gegebenen Messdaten möglichst gut beschreibt. Betrachtet man den Verlauf der in Aufgabe 1 gezeichneten Einwohnerzahlen der USA, so vermutet man, dass diese Daten zum Beispiel durch ein Polynom oder durch eine Exponentialfunktion (exponentielles Wachstum) beschrieben werden können. Hat man eine solche Funktion gefunden, ist es sogar möglich eine Prognose zu wagen und die Einwohnerzahlen für das Jahr 2030 vorauszusagen.

3.2.1 Methode der kleinsten Quadrate

In der Vorlesung Rechnergestützte Mathematik haben Sie die Methode der kleinsten Quadrate (engl.: least squares method) [1, 2, 3] zur Polynomapproximation kennengelernt. Man nimmt an, dass die Messdaten näherungsweise durch ein Polynom der Ordnung m

$$P(x) = c_0 + c_1 \cdot x + c_2 \cdot x^2 + \cdots + c_m \cdot x^m$$

beschrieben werden können. Für jeden der n Datenpunkte $(x_i|y_i), i = 1, \dots, n$ wird daher das folgende Modell aufgestellt:

$$y_i = \underbrace{c_0 + c_1 \cdot x_i + c_2 \cdot x_i^2 + \cdots + c_m \cdot x_i^m}_{P(x_i)} + e_i, \qquad i = 1, \dots, n$$

Da das Polynom $P(x)$ die Datenpunkte nur näherungsweise beschreibt, stellt der Summand e_i dabei den Fehler zwischen $P(x_i)$ und dem tatsächlichen Wert y_i dar. Zur Berechnung der Polynomkoeffizienten $c_0, \dots, c_m$ wird die obige Gleichung für alle Datenpunkte $(x_i|y_i)$, $i = 1, \dots, n$ aufgeschrieben. In Matrix-Vektor-Notation ergibt sich damit:

[2] Erläuterungen dazu finden Sie in Wikipedia: http://de.wikipedia.org/wiki/United_States_Census oder ausführlicher in Englisch: https://en.wikipedia.org/wiki/United_States_Census. Die Webseite des United States Census Bureaus ist https://www.census.gov/

$$\underbrace{\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_n \end{pmatrix}}_{\vec{y}} = \underbrace{\begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^m \\ 1 & x_2 & x_2^2 & \cdots & x_2^m \\ 1 & x_3 & x_3^2 & \cdots & x_3^m \\ 1 & x_4 & x_4^2 & \cdots & x_4^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^m \end{pmatrix}}_{A} \cdot \underbrace{\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_m \end{pmatrix}}_{\vec{c}} + \underbrace{\begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ \vdots \\ e_n \end{pmatrix}}_{\vec{e}}$$

Hierbei sind

$\vec{y}$	Vektor der Länge n mit den y-Werten der Datenpunkte
A	Vandermonde-Matrix der Dimension $n \times (m+1)$ mit den x-Werten der Datenpunkte
$\vec{c}$	Vektor der Länge $m+1$ mit Koeffizienten des Approximationspolynoms $P(x)$
$\vec{e}$	Vektor der Länge n mit den bei der Approximation entstehenden Fehlern

Um das beste Modell für die Messdaten zu erhalten, müssen die Parameter des Polynoms $P(x)$ gefunden werden, bei denen die Fehler e_i zwischen dem Polynom und allen Messpunkten insgesamt möglichst klein werden. Hierfür wird bei der Methode der kleinsten Quadrate der Vektor $\vec{c}$ der Polynomkoeffizienten so bestimmt, dass das quadratische Gütemaß

$$Q = \sum_{i=1}^{n} e_i^2 = e_1^2 + e_2^2 + \cdots + e_n^2$$

minimiert wird. Im Anhang 6.1 wird gezeigt, dass der Parametervektor $\vec{c}$ durch Lösen des folgenden eindeutig lösbaren linearen Gleichungssystems berechnet werden kann:

$$\boxed{A^T A \cdot \vec{c} = A^T \cdot \vec{y}}$$

Durch Multiplikation beider Seiten mit $(A^T A)^{-1}$ kann nach $\vec{c}$ aufgelöst werden:

$$\boxed{\vec{c} = (A^T A)^{-1} A^T \vec{y}}$$

Diese Gleichung kann einfach mithilfe von MATLAB/Octave gelöst werden.

3.2.2 Polynomapproximation mit der Methode der kleinsten Quadrate

Der Verlauf der Einwohnerzahlen der USA soll mit einem Polynom 2. Ordnung approximiert werden.

Aufgabe 2: Approximation der Einwohnerzahlen der USA
Diese Aufgabe wird während des Labortermins bearbeitet!

Speichern Sie Ihr Skript **`V2_A1_Einw_USA.m`** aus Aufgabe 1 unter dem Namen **`V2_A2_Einw_USA_LS.m`** und modifizieren Sie es schrittweise:

- Erzeugen Sie aus den Vektoren **`Jahr`** und **`Einwohner`** die Vandermonde-Matrix A für ein Polynom 2. Ordnung sowie den Vektor $\vec{y}$.
- Berechnen Sie die Koeffizienten des Polynoms 2. Ordnung. Betrachten Sie die Ergebnisse im Workspace oder als Ausgabe im Command Window.
- Alternativ zur Berechnung von $\vec{c} = (A^T A)^{-1} A^T \vec{y}$ können Sie in MATLAB / Octave auch den Befehl **`c=A\y`** verwenden.
- Zeichnen Sie die Datenpunkte sowie das Polynom für $1900 \leq x \leq 2030$ (ausgewertet mit einer Schrittweite von einem Jahr) in ein gemeinsames Schaubild.
- Berechnen Sie eine Prognose der Einwohnerzahlen der USA für ein beliebiges Jahr zwischen 1900 und 2030 und zeichnen Sie den Prognosewert mit einem Marker in die Abbildung ein.
- Prüfen Sie, ob eine Warnung im Command Window ausgegeben wird. Ist dies nicht der Fall, ergänzen Sie bitte im Skript vor der Berechnung der Polynomapproximation den Befehl **`warning('on')`**

3.2.3 Polynome

Die Polynomapproximation mit der Methode der kleinsten Quadrate wird häufig eingesetzt. Daher gibt es in MATLAB / Octave hierfür die Funktion **`polyfit`**. Um diese Funktion zu verwenden, müssen Sie sich mit der Darstellung von Polynomen in MATLAB vertraut machen.

Aufgabe 3: Machen Sie sich wieder mit Polynomen vertraut
Bearbeiten Sie diese Aufgabe vor dem Labortermin: Handrechnung auf Papier!

Gegeben sind die Polynome $\boxed{A(x) = 7x - 13}$ und $\boxed{B(x) = x^2 - 2x - 3}$

Lösen Sie die folgenden Aufgaben von Hand:

1. Ermitteln Sie die Nullstellen von $A(x)$ und $B(x)$
2. Berechnen Sie $C(x) = A(x) \cdot B(x)$
3. Berechnen Sie die Partialbruchzerlegung[3] von $\dfrac{A(x)}{B(x)} = \dfrac{7x - 13}{x^2 - 2x - 3}$
4. Berechnen Sie die Stammfunktion und die Ableitung von $B(x)$.

3.2.4 Polynome in MATLAB / Octave

MATLAB speichert Polynome in Form von Koeffizientenvektoren:

Polynom $\qquad P(x) = c_n x^n + c_{n-1} x^{n-1} + c_{n-2} x^{n-2} + \cdots + c_2 x^2 + c_1 x + c_0$

Vektor $\qquad \vec{p} = (c_n \;\; c_{n-1} \;\; c_{n-2} \;\; \cdots \;\; c_2 \;\; c_1 \;\; c_0)$

Für im Polynom nicht vorhandene Potenzen von x muss im Koeffizientenvektor entsprechend der Wert 0 eingegeben werden!

Wichtige Befehle für Polynome sind in Tabelle 2 zusammengestellt.

Tabelle 2: Wichtige Befehle für Polynome

Befehl	**Funktion**
`roots`	Berechnung aller (auch komplexer) Wurzeln (Nullstellen) eines Polynoms
`conv`	Polynommultiplikation
`residue`	Partialbruchzerlegung
`polyder`	Differentiation
`polyint`	Integration
`polyfit`	Kurvenanpassung (Least Squares - LS) eines Polynoms an gegebene Daten
`polyval`	Berechnung eines oder mehrerer Werte eines Polynoms

Aufgabe 4: Polynome in MATLAB / Octave
Bearbeiten Sie diese Programmieraufgabe vor dem Labortermin!

1. Lösen Sie die mathematischen Aufgaben aus Aufgabe 3 mithilfe von MATLAB / Octave:
   - Legen Sie eine Skriptdatei **`V2_A4_Polynome.m`** mit den dazu notwendigen Befehlen an.
   - Verwenden Sie die MATLAB-Befehle aus Tabelle 2.
   - Schauen Sie bei Bedarf in der MATLAB/Octave-Hilfe [5] nach Details zur Verwendung der Befehle.
2. Zeichnen Sie $C(x)$ im Bereich $-3 \leq x \leq 5$ mithilfe des Befehls **`polyval`**.

[3] In Mathematik 1 wurde die Partialbruchzerlegung im Zusammenhang mit der Integration behandelt.

3.2.5 Polynomapproximation mit polyfit

In MATLAB und Octave ist die Berechnung der Polynomapproximation mit der Funktion

```
p = polyfit(x, y, n)
```

realisiert. Hierbei sind

- **`x`** Vektor mit den x-Werten der zu approximierenden Daten
- **`y`** Vektor mit den y-Werten der zu approximierenden Daten
- **`n`** Ordnung des zu ermittelnden Polynoms
- **`p`** Polynom (Koeffizientenvektor)

Die Auswertung des Polynoms erfolgt dann mit dem bereits in Aufgabe 4 verwendeten Befehl **`polyval`**.

Aufgabe 5: Approximation der Einwohnerzahlen der USA mit polyfit
Diese Aufgabe wird während des Labortermins bearbeitet!

Speichern Sie Ihr Skript **`V2_A2_Einw_USA_LS.m`** aus Aufgabe 2 unter dem Namen **`V2_A5_Einw_USA_polyfit.m`** und modifizieren Sie es schrittweise:

- Berechnen Sie die Koeffizienten eines Polynoms 2. Ordnung aus den Vektoren **`Jahr`** und **`Einwohner`**. Beachten Sie die Ausgabe des Skripts auf dem Workspace.
- Geben Sie anhand des Polynoms 2. Ordnung eine Prognose für die Einwohnerzahl im Jahr 2030 an.
- Prüfen Sie, ob im Command Window eine Warnung ausgegeben wird. Ist dies nicht der Fall, so ergänzen Sie zu Beginn des Skripts die Codezeile **`warning('on')`** .

3.3 Approximation der Einwohnerdaten mit Exponentialfunktion

Anhand der gegebenen Einwohnerzahlen der USA muss man nicht zwangsläufig auf ein Polynom als geeignete Näherungsfunktion schließen. Es ist sicherlich genauso naheliegend, den Verlauf der Einwohnerzahlen mit einer Exponentialfunktion

$$y(x) = K \cdot e^{bx}$$

zu approximieren. Hierfür müssen die beiden Parameter K und b berechnet werden. Allerdings funktioniert das Konzept der Methode der kleinsten Quadrate jetzt nicht mehr direkt: es ist jetzt nicht mehr möglich, wie bei der Polynomapproximation die x-Werte in der Vandermonde-Matrix A getrennt von den Parametern im Vektor c anzuordnen.

Man kann den Ansatz jedoch so umformen, dass die Parameter K und b mit der bekannten Methode der kleinsten Quadrate berechnet werden können. Hierzu stellt man zunächst $K > 0$ als Exponentialfunktion dar:

$$K = e^a$$

Damit ergibt sich:

$$y(x) = e^a \cdot e^{bx} = e^{a+bx}$$

Durch Logarithmieren erhält man schließlich den folgenden Zusammenhang:

$$\ln\big(y(x)\big) = a + b \cdot x$$

Nach der Berechnung der Logarithmen der gegebenen y-Werte können die Parameter a und b der Geradengleichung mit der bekannten Polynomapproximation berechnet werden, , denn es handelt sich bei $a + b \cdot x$ um ein Polynom erste Ordnung (Gerade).

Aufgabe 6: Approximation der Einwohnerzahlen der USA mit Exponentialfunktion
Diese Aufgabe wird während des Labortermins bearbeitet!

Speichern Sie Ihr Skript **`V2_A5_Einw_USA_polyfit.m`** aus Aufgabe 5 unter dem Namen **`V2_A6_Einw_USA_exponential.m`** und modifizieren Sie es:

- Berechnen Sie die Parameter der Exponentialfunktion
- Zeichnen Sie die Daten und die beiden Approximationen für $1800 \leq x \leq 2030$ in einem gemeinsamen Schaubild. Welcher Verlauf erscheint ihnen realistischer?
- Geben Sie anhand der beiden Approximationen jeweils eine Prognose für die Einwohnerzahl im Jahr 2030 an.

4 Funktionen

Nachdem Sie bei der Einarbeitung in die Grundlagen der MATLAB-Sprache bereits Funktionen programmiert und geplottet haben, werden Sie jetzt numerische Verfahren zur Nullstellensuche und zur Integration in MATLAB / Octave programmieren und einsetzen.

4.1 Beispielfunktion für weitere Untersuchungen

Für die numerische Nullstellensuche und die numerische Integration soll die folgende Funktion betrachtet werden:

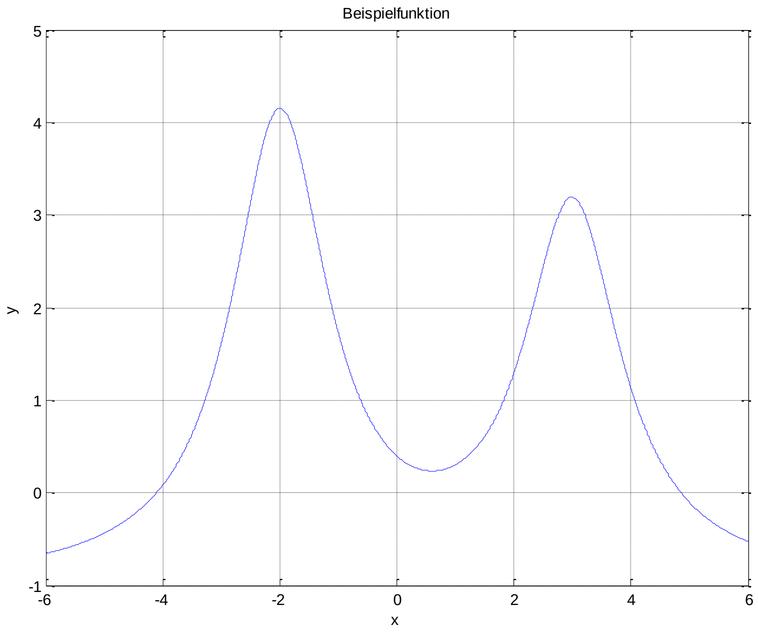
$$\boxed{f(x) = \frac{5}{x^2 + 4x + 5} + \frac{4}{x^2 - 6x + 10} - 1}$$

Aufgabe 7: Beispielfunktion programmieren

Bearbeiten Sie diese Programmieraufgabe vor dem Labortermin!

1. Programmieren Sie die oben angegebene Funktion als Matlab Function **`f()`** und speichern Sie sie unter dem Namen **`f.m`** in Ihrem Arbeitsverzeichnis. Programmieren Sie die Funktion so, dass ein Vektor aus x-Werten übergeben werden kann. Kommentieren Sie die Funktion vollständig und prüfen Sie die Aussagekraft der Kommentare durch Aufruf von **`doc f`**.[4]
2. Erstellen Sie ein Skript **`V2_A7_Plot_f.m`**, das die Funktion für $-6 \leq x \leq 6$ zeichnet. Wählen Sie dabei als Schrittweite des x-Vektors 0,001. Sie sollten das folgende Schaubild erhalten:

[4] Da weitere MATLAB-eigene Funktionen existieren, die den Namen f bzw. F tragen, sollten Sie sicherstellen, dass Ihr Arbeitsverzeichnis mit dem Verzeichnis, in dem die Funkton abgelegt ist, übereinstimmt. Alternativ müsste man der Beispielfunktion einen andere Namen geben, z.B. **`fbsp()`** und konsequent mit diesem Funktionsnamen arbeiten.


Beispielfunktion
5
4
3
2
1
0
-1
y
-6
-4
-2
0
2
4
6
x

4.2 Nullstellensuche

4.2.1 Sekantenverfahren

In der Vorlesung haben Sie das Sekantenverfahren [1, 2, 3] zur Berechnung der Nullstellen einer Funktion $f(x)$ kennengelernt. Ausgehend von zwei Startpunkten x_1 und x_2 wird – sofern das Verfahren konvergiert – iterativ eine immer bessere Näherung für die Nullstelle berechnet:

$$\boxed{x_{i+1} = x_i - \frac{f(x_i) \cdot (x_i - x_{i-1})}{f(x_i) - f(x_{i-1})}}$$

Aufgabe 8: Sekantenverfahren „von Hand“
Bearbeiten Sie diese Aufgabe vor dem Labortermin: Handrechnung auf Papier!

Berechnen Sie ausgehend von den beiden Punkten $x_1 = -5$ und $x_2 = -4$ einen Näherungswert für eine Nullstelle der Funktion aus 4.1. Geben Sie das Ergebnis auf vier Nachkommastellen genau an.

- Verwenden Sie zur Berechnung der Funktionswerte einen Taschenrechner.
- Notieren Sie alle Werte x_i und die zugehörigen Funktionswerte $f(x_i)$.
- Erläutern Sie, was die Anforderung „auf vier Nachkommastellen genau“ bedeutet.
- Brechen Sie die Iteration erst ab, wenn der Näherungswert auf vier Nachkommastellen genau berechnet ist.

Aufgabe 9: Programmierung des Sekantenverfahrens
Diese Aufgabe wird während des Labortermins bearbeitet!

Erstellen Sie ein Skript **`V2_A9_Sekantenverfahren.m`**, das die Nullstellen der Beispielfunktion aus Aufgabe 7 ausgehend von den Startwerten x_1 und x_2 berechnet. Bestimmen Sie damit die beiden Nullstellen der Beispielfunktion auf vier Nachkommastellen genau, indem Sie geeignete Startwerte zum Aufruf des Sekantenverfahrens wählen.

4.2.2 MATLAB / Octave – Funktion `fzero`

Mit der MATLAB-Funktion `x = fzero(fun, x0)` können die Nullstellen einer gegebenen Funktion gesucht werden:

- `fun` Handle auf die Funktion, z. B. `@f`
- `x0` Startwert bzw. Vektor mit den Startwerten

Aufgabe 10: Nullstellensuche mit `fzero`
Diese Aufgabe wird während des Labortermins bearbeitet!

Erweitern Sie Ihr Skript `V2_A7_Plot_f.m` um die Suche nach den Nullstellen der Funktion `f` mit `fzero` und speichern Sie es unter dem Namen `V2_A10_fzero.m`

- Stimmen die ermittelten Nullstellen mit den von Ihnen abgelesenen überein?
- Welche Erfahrungen können Sie für die numerische Nullstellensuche festhalten?

4.3 Numerische Integration

Als Beispiel für die numerische Integration soll das Integral

$$I = \int_{-4}^{4} f(x)\,dx = \int_{-4}^{4} \frac{5}{x^2+4x+5} + \frac{4}{x^2-6x+10} - 1\,dx$$

berechnet werden. Mit etwas Mühe wird es Ihnen – hoffentlich – gelingen, die Stammfunktion zu finden und damit das Integral zu berechnen. Mit dem Open-Source-Computeralgebrasystem Maxima[5] geht es einfacher (Abbildung 1).

```
(%i1) y:5/(x^2+4*x+5)+4/(x^2-6*x+10)-1;
            5            4
(%o1) ----------- + ------------ - 1
       x^2+4 x+5     x^2-6 x+10

(%i2) integrate(y, x, -4, 4);
(%o2) 4 atan(7)+5 atan(6)+5 atan(2)+π-8

(%i3) float(%);
(%o3) 13.42117157822453
```

Abbildung 1: Berechnung mit Maxima

Dasselbe Ergebnis erhält man auch in MATLAB unter Einsatz der Symbolic Math Toolbox

```
syms x      % symbolische Variable x
y = 5/(x^2+4*x+5)+4/(x^2-6*x+10)-1; % symbolischer Funktionsausdruck
```

[5] maxima.sourceforge.net

```
I = int(y,x,-4,4) % Berechnung des Integrals
vpa(I,14) % Ausgabe als Gleitkommazahl mit 14 Nachkommastellen

>> ans =
  13.421171578225
```

In vielen Anwendungsfällen wird die numerische Lösung des Integrals genügen oder der Integrand ist so kompliziert, dass eine symbolische Berechnung nicht möglich ist.

4.3.1 Zusammengesetzte Simpsonsche Regel

In der Vorlesung lernen Sie verschiedene Verfahren zur numerischen Integration kennen. Falls die jetzt verwendete zusammengesetzte Simpsonsche Regel [1, 2, 3] in Ihrer Vorlesung noch nicht hergeleitet wurde, können Sie dennoch damit arbeiten: alle für das Labor notwendigen Gleichungen finden Sie in dieser Anleitung, die Herleitung erfolgt dann in der Vorlesung.

Zur Berechnung des Integrals

$$I = \int_a^b f(x)\,dx$$

mit der zusammengesetzten Simpsonschen Regel wird der Integrationsbereich $[a\ b]$, wie in Abbildung 2, gezeigt in n gleich große Intervalle eingeteilt.

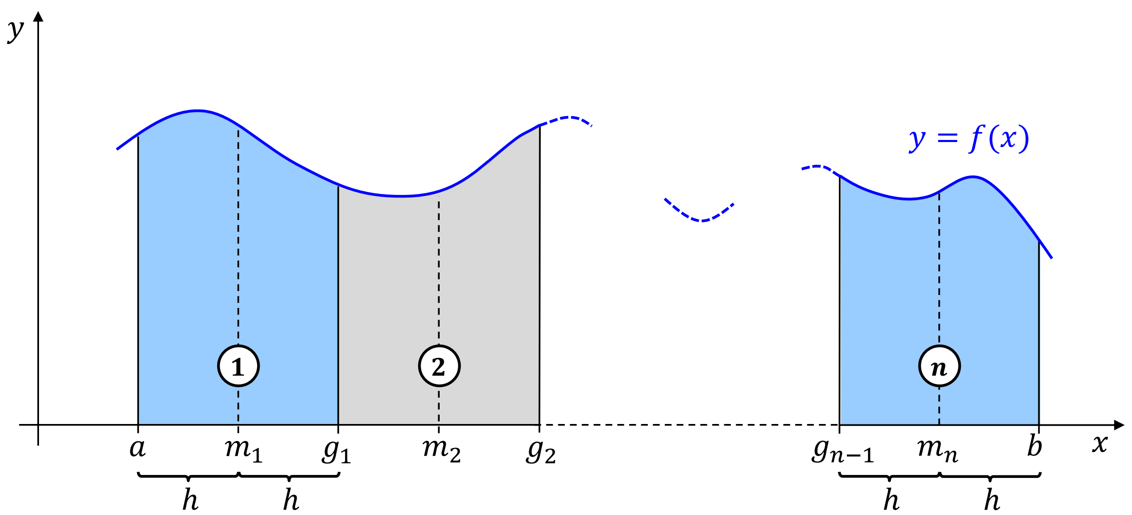


Abbildung 2: Intervalle und Stützstellen für die zusammengesetzte Simpsonsche Regel

Für jedes dieser Intervalle wird dann das Integral mit der Simpsonschen Regel berechnet. Damit ergibt sich insgesamt die folgende Gleichung zur Berechnung des Integrals:

$$\int_a^b f(x)\,dx = \frac{b-a}{6\,n} \cdot \left[f(a) + 2 \cdot \sum_{i=1}^{n-1} f(g_i) + 4 \cdot \sum_{i=1}^{n} f(m_i) + f(b) \right]$$

Aufgabe 11: Zusammengesetzte Simpsonsche Regel „von Hand"

Bearbeiten Sie diese Aufgabe vor dem Labortermin: Handrechnung auf Papier!

Berechnen Sie den Wert des Integrals

$$I = \int_{-4}^{4} \frac{5}{x^2 + 4x + 5} + \frac{4}{x^2 - 6x + 10} - 1\,dx$$

mit der zusammengesetzten Simpsonschen Regel.

- Teilen Sie den Integrationsbereich in $n = 4$ Intervalle ein.
- Ermitteln Sie mit einer Skizze die Lage der Stützstellen g_i und m_i.
- Berechnen Sie mithilfe Ihres Taschenrechners den Wert des Integrals mit der zusammengesetzten Simpsonschen Regel.
- Vergleichen Sie das Ergebnis mit dem Wert aus der symbolischen Berechnung.

Aufgabe 12: Berechnung der Stützstellen

Bearbeiten Sie diese Aufgabe vor dem Labortermin: Handrechnung auf Papier!

Für die Berechnung des Integralwertes mit der zusammengesetzten Simpsonschen Regel werden neben den Integrationsgrenzen a und b weitere Stützstellen benötigt:

- $g_i,\ i = 1, \dots, n-1$: Intervallgrenzen für die Simpson-Regel
- $m_i,\ i = 1, \dots, n$: Mittelpunkte der Intervalle

Leiten Sie mithilfe von Abbildung 2

- eine Gleichung zur Berechnung des Stützstellenabstands h abhängig von den Integrationsgrenzen a und b und der Zahl n der Intervalle und
- Gleichungen zur Berechnung der g_i und m_i abhängig vom Zählindex i, den Integralgrenzen und dem Stützstellenabstand h her.

Aufgabe 13: Zusammengesetzte Simpsonsche Regel mit For-Schleife

Diese Aufgabe wird während des Labortermins bearbeitet!

Erstellen Sie ein Skript **`V2_A13_Integral_Simpson_Schleife.m`**, das das Integral

$$I = \int_{-4}^{4} \frac{5}{x^2 + 4x + 5} + \frac{4}{x^2 - 6x + 10} - 1\,dx$$

berechnet:

- Die Anzahl $n \geq 2$ der Intervalle soll dabei variabel vorgegeben werden können.
- Die Funktionswerte werden durch Aufruf der Funktion **`f`** berechnet.
- h, g_i und m_i werden über die Gleichungen aus Aufgabe 12 berechnet.
- Die Summen werden mit **`for`**-Schleifen berechnet.

Berechnen Sie das Integral für $n = 2, 3, 4, 5, 6$ und vergleichen Sie das Ergebnis mit dem Wert aus der symbolischen Berechnung.

Aufgabe 14: Zusammengesetzte Simpsonsche Regel ohne Schleife
Diese Aufgabe wird während des Labortermins bearbeitet!

Erstellen Sie ein Skript **`V2_A14_Integral_Simpson.m`**, das das Integral

$$I = \int_{-4}^{4} \frac{5}{x^2 + 4x + 5} + \frac{4}{x^2 - 6x + 10} - 1\, dx$$

berechnet.

- Die Anzahl $n \geq 2$ der Intervalle soll dabei variabel vorgegeben werden.
- Verwenden Sie keine Schleife.
- Berechnen Sie die Vektoren **`g`** und **`m`** mit den Werten der Stützstellen g_i und m_i
- Nutzen Sie den **`sum`**-Befehl, um die Funktionswerte an den Stützstellen aufzusummieren.
- Berechnen Sie das Integral für $n = 2, 3, 4, 5, 6$ und vergleichen Sie das Ergebnis mit dem Wert aus der symbolischen Berechnung.
- Vergleichen Sie die beiden Programme aus Aufgaben 13 und 14 bezüglich Rechenzeit und Speicherbedarf.

Sobald das Skript zufriedenstellend funktioniert, machen Sie daraus eine Funktion

```
I = simpson(fun, a, b, n)
```

und testen Sie diese.

4.3.2 MATLAB / Octave – Funktion zur numerischen Integration

Zur numerischen Integration sind in MATLAB und Octave in den Funktionen **`quad`**, **`quadgk`**, und **`quadl`** verschiedene Verfahren implementiert, die in MATLAB in Zukunft von der zur Zeit noch nicht in Octave implementierten Funktion **`integral`** abgelöst werden sollen.

Im Labor wird die Funktion **`I = quad(fun, a, b)`** verwendet, mit der das Integral

$$I = \int_{a}^{b} \text{fun}\ dx$$

mit einem adaptiven Simpson-Algorithmus berechnet wird:

- **`fun`** Handle auf die Funktion, z. B. **`@f`**
- **`a, b`** Integralgrenzen

Aufgabe 15: Flächenberechnung

Diese Aufgabe wird während des Labortermins bearbeitet!

Berechnen Sie die durch die Funktion **`f`** und die x-Achse umschlossene Fläche. Erweitern Sie hierzu Ihr Skript aus Aufgabe 7 entsprechend, nutzen Sie die Nullstellensuche aus Aufgabe 10 und speichern Sie das Ergebnis unter **`V2_A15_Flaeche_f.m`**.

5 Literatur

[1] Frank Thuselt, Felix Paul Gennrich: Praktische Mathematik mit MATLAB, Scilab und Octave für Ingenieure und Naturwissenschaftler, Springer Verlag, 2013.
Als E-Book verfügbar!

[2] Frank Thuselt: Das Arbeiten mit Numerik Programmen MATLAB, Scilab und Octave in der Anwendung. Beiträge der Hochschule Pforzheim, Nr. 129, 2009.
Download von der Laborseite.

[3] Rainer Dietz: Skript zur Vorlesung Rechnergestützte Mathematik, 2016.
Download von der Vorlesungsseite.

[4] Stefan Hillenbrand: MATLAB / Octave Kurzreferenz.
Download von der Laborseite.

[5] MATLAB Online-Hilfe.
Verfügbar auf de.mathworks.com – Support – Documentation

[6] Lothar Papula: Mathematik für Ingenieure und Naturwissenschaftler, Band 1, Springer Verlag, 2014.
Als E-Book verfügbar!

[7] Lothar Papula: Mathematik für Ingenieure und Naturwissenschaftler, Band 2, Springer Verlag, 2015.
Als E-Book verfügbar!

Hinweis zu den E-Books

Die Hochschule hat zahlreiche Lehrbücher, insbesondere vom Springer-Verlag, abonniert, die Sie kostenlos herunterladen können. Weitere Informationen zu den E-Books finden Sie auf den Webseiten der Bibliothek:
https://www.hs-pforzheim.de/hochschule/organisation/bibliothek/

Die hier genannten E-Books können Sie auch direkt bei Springer herunterladen:

1. Verbinden Sie sich mit dem Hochschulnetz:
    - Rechner der Hochschule oder
    - Eigener Rechner über VPN
2. Rufen Sie http://link.springer.com auf.
3. Suchen Sie nach dem Buch.
4. Abonnierte Bücher können Sie kapitelweise oder vollständig als PDF mit Wasserzeichen herunterladen.

6 Anhang: Methode der kleinsten Quadrate

6.1 Herleitung der Gleichung zu Berechnung der Polynomparameter

Bei der Polynomapproximation mit der Methode der kleinsten Quadrate (engl.: least squares method) [1, 2, 3] werden die Messdaten näherungsweise durch ein Polynom der Ordnung m

$$P(x) = c_0 + c_1 \cdot x + c_2 \cdot x^2 + \cdots + c_m \cdot x^m$$

beschrieben. Für jeden der gegebenen n Datenpunkte $(x_i|y_i), i = 1, \dots, n$ wird somit das folgende Modell aufgestellt:

$$y_i = \underbrace{c_0 + c_1 \cdot x_i + c_2 \cdot x_i^2 + \cdots + c_m \cdot x_i^m}_{P(x_i)} + e_i, \qquad i = 1, \dots, n$$

Da das Polynom $P(x)$ die Datenpunkte nur näherungsweise beschreibt, stellt der Summand e_i dabei den Fehler zwischen $P(x_i)$ und dem tatsächlichen Wert y_i dar. Zur Berechnung der Polynomkoeffizienten $c_0, \dots, c_m$ muss man die obige Gleichung für alle Datenpunkte $(x_i|y_i), i = 1, \dots, n$ aufschreiben. In Matrix-Vektor-Notation ergibt sich damit:

$$\underbrace{\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_n \end{pmatrix}}_{\vec{y}} = \underbrace{\begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^m \\ 1 & x_2 & x_2^2 & \cdots & x_2^m \\ 1 & x_3 & x_3^2 & \cdots & x_3^m \\ 1 & x_4 & x_4^2 & \cdots & x_4^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^m \end{pmatrix}}_{A} \cdot \underbrace{\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_m \end{pmatrix}}_{\vec{c}} + \underbrace{\begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ \vdots \\ e_n \end{pmatrix}}_{\vec{e}}$$

Hierbei sind

- $\vec{y}$ Vektor der Länge n mit den y-Werten der Datenpunkte
- A Vandermonde-Matrix der Dimension $n \times (m+1)$ mit den x-Werten der Datenpunkte
- $\vec{c}$ Vektor der Länge $m+1$ mit Koeffizienten des Approximationspolynoms $P(x)$
- $\vec{e}$ Vektor der Länge n mit den bei der Approximation entstehenden Fehlern

Der Koeffizientenvektor $\vec{c}$ wird nun so berechnet das die Fehler $\vec{e}$ möglichst klein werden. Durch einfaches Umformen kann der Fehlervektor $\vec{e}$ abhängig von den Messdaten und den Koeffizienten berechnet werden:

$$\vec{e} = \vec{y} - A \cdot \vec{c}$$

Für die Berechnung des Koeffizientenvektors $\vec{c}$ müssen alle Elemente $e_1, \dots, e_n$ des Fehlervektors $\vec{e}$ berücksichtigt werden. Bei der Methode der kleinsten Quadrate summiert man hierfür die Quadrate der Fehler auf und berechnet daraus das Gütemaß

$$Q = \sum_{i=1}^{n} e_i^2 = e_1^2 + e_2^2 + \cdots + e_n^2 = \underbrace{\begin{pmatrix} e_1 & e_2 & \cdots & e_n \end{pmatrix}}_{\vec{e}^T} \cdot \underbrace{\begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}}_{\vec{e}} = \vec{e}^T \cdot \vec{e}$$

Mit $\vec{e} = \vec{y} - A \cdot \vec{c}$ folgt daraus:

$$Q = (\vec{y} - A \cdot \vec{c})^T \cdot (\vec{y} - A \cdot \vec{c}) = (\vec{y}^T - \vec{c}^T \cdot A^T) \cdot (\vec{y} - A \cdot \vec{c})$$
$$= \vec{y}^T \vec{y} - \vec{y}^T A\, \vec{c} - \vec{c}^T A^T \vec{y} + \vec{c}^T A^T A\, \vec{c}$$

Da Q ein Skalar ist, müssen auch alle Summanden Skalare sein. Daher gilt:

$$\vec{y}^T A \vec{c} = (\vec{y}^T A \vec{c})^T = \vec{c}^T A^T \vec{y}$$

Damit folgt schließlich:

$$Q = \vec{y}^T \vec{y} - 2\, \vec{c}^T A^T \vec{y} + \vec{c}^T A^T A \vec{c}$$

Die Vandermonde-Matrix A und der Vektor $\vec{y}$ enthalten die bekannten x- und y- Werte der gegebenen Datenpunkte. Damit ist das Gütemaß Q eine skalare Funktion, die von den $m + 1$ Variablen $c_0, c_1, c_2, \dots, c_m$ aus dem Vektor $\vec{c}$ abhängt. Dies wird nun in der Notation verdeutlicht:

$$Q(\vec{c}) = \vec{y}^T \vec{y} - 2\, \vec{c}^T A^T \vec{y} + \vec{c}^T A^T A \vec{c}$$

Aus der Vorlesung Mathematik 1 ist die notwendige Bedingung für den gesuchten Extremwert bekannt:[6] Die partiellen Ableitungen nach allen Variablen müssen gleich Null sein.

$$\begin{aligned} \frac{\partial Q(\vec{c})}{\partial c_0} &= 0 \\ \frac{\partial Q(\vec{c})}{\partial c_1} &= 0 \\ &\vdots \\ \frac{\partial Q(\vec{c})}{\partial c_m} &= 0 \end{aligned} \qquad \Leftrightarrow \qquad \operatorname{grad} Q(\vec{c}) = \frac{\partial Q(\vec{c})}{\partial \vec{c}} = \begin{pmatrix} \frac{\partial Q}{\partial c_0} \\ \frac{\partial Q}{\partial c_1} \\ \vdots \\ \frac{\partial Q}{\partial c_m} \end{pmatrix} = \vec{0}$$

Damit folgt:

$$\frac{\partial Q(\vec{c})}{\partial \vec{c}} = \underbrace{\frac{\partial}{\partial \vec{c}} \vec{y}^T \vec{y}}_{\vec{0}} - 2 \frac{\partial}{\partial \vec{c}}\, \vec{c}^T A^T \vec{y} + \frac{\partial}{\partial \vec{c}}\, \vec{c}^T A^T A \vec{c} = \vec{0}$$

[6] In Mathematik 1 haben Sie der Einfachheit halber nur mit Funktionen mit 2 Veränderlichen gerechnet, die Methoden sind jedoch auf beliebig viele Variablen übertragbar.

Mit den Rechenregeln[7]

$$\frac{\partial}{\partial \vec{c}}\, \vec{c}^T \cdot A^T \cdot \vec{y} = A^T \cdot \vec{y} \qquad \text{und} \qquad \frac{\partial}{\partial \vec{c}}\, \vec{c}^T \cdot A^T A \cdot \vec{c} = 2\, A^T A \cdot \vec{c}$$

folgt damit

$$\frac{\partial Q(\vec{c})}{\partial \vec{c}} = -2\, A^T \cdot \vec{y} + 2\, A^T A \cdot \vec{c} = \vec{0}$$

Man erhält somit ein eindeutig lösbares lineares Gleichungssystem:

$$\boxed{A^T A \cdot \vec{c} = A^T \cdot \vec{y}}$$

Durch Multiplikation beider Seiten mit $(A^T A)^{-1}$ kann nach $\vec{c}$ aufgelöst werden:

$$\boxed{\vec{c} = (A^T A)^{-1} A^T \vec{y}}$$

Diese Gleichung kann einfach mithilfe von Matlab/Octave gelöst werden. Auf die Prüfung der hinreichenden Bedingung kann verzichtet werden, da die Funktion $Q(\vec{c})$ als Summe der Fehlerquadrate nur ein Minimum haben kann.

6.2 Rechenregeln: Ableitung von Vektorfunktionen

6.2.1 Lineare Form

Beim Term

$$f(\vec{c}) = \vec{c}^T \cdot A^T \cdot \vec{y}$$

handelt es sich um eine sogenannte lineare Form. Zur Berechnung der Ableitung wird zunächst die Matrix A durch Ihre $m + 1$ Spaltenvektoren dargestellt:

$$A = \begin{pmatrix} \vec{a}_0 & \vec{a}_1 & \cdots & \vec{a}_m \end{pmatrix} \quad \Leftrightarrow \quad A^T = \begin{pmatrix} \vec{a}_0^T \\ \vec{a}_1^T \\ \vdots \\ \vec{a}_m^T \end{pmatrix}$$

Damit folgt

$$f(\vec{c}) = \begin{pmatrix} c_0 & c_1 & \cdots c_m \end{pmatrix} \cdot \underbrace{\begin{pmatrix} \vec{a}_0^T \\ \vec{a}_1^T \\ \vdots \\ \vec{a}_m^T \end{pmatrix}}_{A^T} \cdot \vec{y} = \begin{pmatrix} c_0 & c_1 & \cdots c_m \end{pmatrix} \cdot \underbrace{\begin{pmatrix} \vec{a}_0^T \cdot \vec{y} \\ \vec{a}_1^T \cdot \vec{y} \\ \vdots \\ \vec{a}_m^T \cdot \vec{y} \end{pmatrix}}_{A^T \cdot \vec{y}}$$

$$= c_0 \cdot \vec{a}_0^T \cdot \vec{y} + c_1 \cdot \vec{a}_1^T \cdot \vec{y} + \cdots + c_m \cdot \vec{a}_m^T \cdot \vec{y}$$

[7] Die Herleitung der Regeln finden Sie im folgenden Abschnitt 6.2.

Jetzt können die partiellen Ableitungen berechnet werden:

$$\begin{aligned} \frac{\partial f(\vec{c})}{\partial c_0} &= \vec{a}_0^T \cdot \vec{y} \\ \frac{\partial f(\vec{c})}{\partial c_1} &= \vec{a}_1^T \cdot \vec{y} \\ &\vdots \\ \frac{\partial f(\vec{c})}{\partial c_m} &= \vec{a}_m^T \cdot \vec{y} \end{aligned} \qquad \Rightarrow \qquad \mathrm{grad} f(\vec{c}) = \frac{\partial f(\vec{c})}{\partial \vec{c}} = \begin{pmatrix} \vec{a}_0^T \cdot \vec{y} \\ \vec{a}_1^T \cdot \vec{y} \\ \vdots \\ \vec{a}_m^T \cdot \vec{y} \end{pmatrix} = A^T \cdot \vec{y}$$

Damit ist die Ableitungsregel gefunden: $\boxed{\frac{\partial}{\partial \vec{c}}\, \vec{c}^T \cdot A^T \cdot \vec{y} = A^T \cdot \vec{y}}$

Da $f(\vec{c})$ eine skalare Funktion ist, gilt $\left(f(\vec{c})\right)^T = f(\vec{c})$ und somit:

$$\vec{c}^T \cdot A^T \cdot \vec{y} = \left(\frac{a}{b}\vec{c}^T \cdot A^T \cdot \vec{y}\right)^T = \vec{y}^T \cdot A \cdot \vec{c}$$

Daraus folgt die weitere Ableitungsregel $\boxed{\frac{\partial}{\partial \vec{c}}\, \vec{y}^T \cdot A \cdot \vec{c} = A^T \cdot \vec{y}}$

6.2.2 Quadratische Form

Beim Term

$$g(\vec{c}) = \vec{c}^T \cdot A^T \cdot A \cdot \vec{c}$$

handelt es sich um eine sogenannte quadratische Form. Der Vektor $\vec{c}$ geht zweimal in $g(\vec{c})$ ein. Damit für die Berechnung der Ableitung die Produktregel angewendet werden kann, wird $g(\vec{c})$ in zwei Faktoren aufgeteilt:

$$g(\vec{c}) = \underbrace{\vec{c}^T \cdot A^T}_{\vec{y}^T} \cdot \underbrace{A \cdot \vec{c}}_{\vec{y}}$$

Nun wird die Produktregel angewandt dabei die in Abschnitt 6.1 hergeleiteten Ableitungsregeln verwendet:

$$\frac{\partial g(\vec{c})}{\partial \vec{c}} = \underbrace{\frac{\partial}{\partial \vec{c}}(\vec{c}^T \cdot A^T) \cdot \vec{y}}_{A^T \cdot \vec{y}} + \underbrace{\frac{\partial}{\partial \vec{c}}\, \vec{y}^T \cdot (A \cdot \vec{c})}_{A^T \cdot \vec{y}} = 2 \cdot A^T \cdot \vec{y}$$

Mit $\vec{y} = A \cdot \vec{c}$ folgt die Ableitungsregel

$$\boxed{\frac{\partial}{\partial \vec{c}}\, \vec{c}^T \cdot A^T \cdot A^T \cdot \vec{c} = 2 \cdot A^T \cdot A^T \cdot \vec{c}}$$