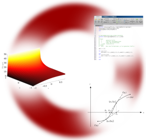
Praktikum Numerische Methoden für Ingenieure

Rechengenauigkeit, Interpolation und Kurvenanpassung

Christoph Ager

TUM – Lehrstuhl für Numerische Mechanik

October 27, 2016



Überprüfungstermine

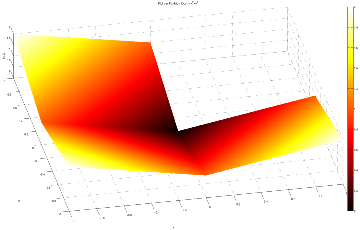
- Um das Praktikum positiv zu absolvieren, ist die Teilnahme an beiden Terminen zwingend erforderlich!
- Die Festlegung ob Sie am Montags- oder Mittwochs/Donnerstagstermin teilnehmen, wird ca. einen Monat vorher durch Sie (soweit möglich) festgelegt.
- Sie werden in ca. zwei Woche automatisch zu einem Prüfungstermin im TUMonline angemeldet. Das dabei angegebene Datum wird !nicht! dem Überprüfungsdatum entsprechen.

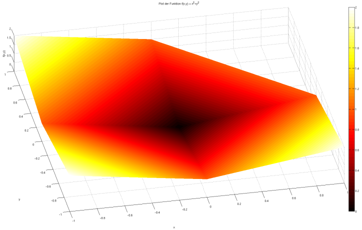
1. Aufgabenblatt

- Lösungszahlenwerte mit Matlab R2012b
- Funktion `disp` für Ausgabe verwenden
- Plots nicht symbolisch
- Generelle Unterschiede zwischen Matlab Versionen
- *.m-files erstellen und nicht direkt in Konsole!

- Aufgabe 3: Ein Viereck soll jeweils in zwei Dreiecke unterteilt werden (keine globale Delaunytriangulierung!)

Beispiel: 3 Vierecke zum Plotten von $f(x,y) = x^2 + y^2$

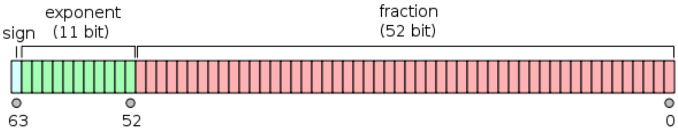




binäre Zahlendarstellung

- Darstellung durch bits (0 oder 1):
- Bsp.: 4 bit Zahl
  $[1001]_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = [9]_{10}$
- Achtung Subtraktion - Auslöschung
- Bsp.: simple Gleitkommazahl: ...

Double-precision floating-point format (64bit):



https://en.wikipedia.org/wiki/Double-precision_floating-point_format

$$(-1)^{sign}(1+\sum_{i=1}^{52} b_{52-i}2^{-i})2^{e-1023}$$

- Vorzeichen: 1 bit $(-1)^{sign} \longrightarrow -1, 1$
- Exponent: 11 bits $e \in [1, 2046]$ ($0, 2047$ used differently) $\longrightarrow \left[2^{-1022}, 2^{1023}\right] \sim \left[10^{-308}, 10^{308}\right]$
- Mantisse: 52 bits $m \in \left[1, 1+2^{-52}, 1+2^{-51}, ..., 2-2^{-52}\right] \sim \left[1, 1+2\cdot 10^{-16}, 1+4\cdot 10^{-16}, ..., 2-2\cdot 10^{-16}\right]$

Praktikum Numerische Methoden für Ingenieure

Aufgabenblatt 2

WS 2016/2017

Rechengenauigkeit, Interpolation und Kurvenanpassung

Aufgabe 1: Rechengenauigkeit

Erstelle eine Matlab Funktion, die den Schnittpunkt (x-Koordinate) einer durch zwei Punkte definierten Linie mit der x-Achse ($y = 0$) berechnet (`function x = lineintersection(P1,P2)`).

Nutze die Funktion, um für die gegebenen Punkte $\boldsymbol{P}_1 = \begin{pmatrix} 0.0 \\ 1.0 \end{pmatrix}$ und $\boldsymbol{P}_2 = \begin{pmatrix} 0.0 + \delta \\ 1.0 - \delta \end{pmatrix}$ den Schnittpunkt zu berechnen.

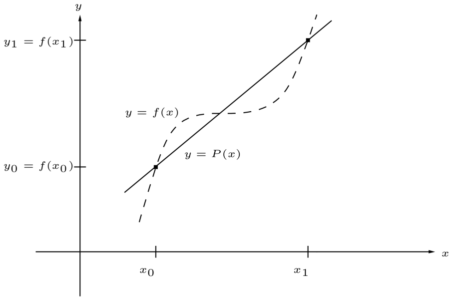
Berechne die Position für $10^{-20} \leq \delta \leq 10^5$ (HINWEIS: wähle eine geeignete Verteilung des Parameters δ) und plotte $(\delta, |x - x_{ex}|)$ den Betrag des absoluten Fehlers der Position in doppelt logaritmischem Maßstab im relevanten Bereich. Dabei ist x die ermittelte Position und x_{ex} die analytisch exakte Position.

Interpretiere das Ergebnis qualitativ.

Lagrange-Interpolation: 2 Punkte

gegeben: (x_0, y_0), (x_1, y_1); $y_0 = f(x_0)$, $y_1 = f(x_1)$

$$P(x) = \underbrace{\frac{(x - x_1)}{(x_0 - x_1)}}_{=L_{10}(x)} f(x_0) + \underbrace{\frac{(x - x_0)}{(x_1 - x_0)}}_{=L_{11}(x)} f(x_1)\,.$$



Lagrange-Interpolation: n Punkte

$$L_{ni}\left(x_j\right) = 0, \quad \text{für } j \neq i \text{ und } i = 0, 1, \ldots, n,$$
$$L_{ni}\left(x_i\right) = 1, \quad \text{für } i = 0, 1, \ldots, n.$$

Basispolynome:

$$L_{ni}\left(x\right) = \frac{\left(x - x_0\right) \ldots \left(x - x_{i-1}\right)\left(x - x_{i+1}\right) \ldots \left(x - x_n\right)}{\left(x_i - x_0\right) \ldots \left(x_i - x_{i-1}\right)\left(x_i - x_{i+1}\right) \ldots \left(x_i - x_n\right)}$$

LagrangePolynom:

$$P_n\left(x\right) = f\left(x_0\right) L_{n0}\left(x\right) + \ldots + f\left(x_n\right) L_{nn}\left(x\right) = \sum_{i=0}^{n} f\left(x_i\right) L_{ni}\left(x\right)$$

Lagrange-Interpolation: Empfehlung für Umsetzung

- `function wert=LagrangeBasis(x,n,i,x_node)`
- `function wert=LagrangePolynom(x,n,x_node,f_node)`

Verwende immer die elementsweisen Operatoren in den Funktionen
`(.* , ./ , .^)`

Lagrange-Interpolation: n Punkte

Ableitung LagrangePolynom:

$$\frac{\mathrm{d}P_n(x)}{\mathrm{d}x} = f(x_0)\frac{\mathrm{d}L_{n0}(x)}{\mathrm{d}x} + \ldots + f(x_n)\frac{\mathrm{d}L_{nn}(x)}{\mathrm{d}x} = \sum_{i=0}^{n} f(x_i)\frac{\mathrm{d}L_{ni}(x)}{\mathrm{d}x}$$

Basispolynome:

$$L_{ni}(x) = \frac{(x-x_0)\ldots(x-x_{i-1})(x-x_{i+1})\ldots(x-x_n)}{(x_i-x_0)\ldots(x_i-x_{i-1})(x_i-x_{i+1})\ldots(x_i-x_n)}$$

$$= \prod_{k=0,k\neq i}^{n} \frac{(x-x_k)}{(x_i-x_k)}$$

Ableitung Basispolynome:

$$\frac{\mathrm{d}L_{ni}(x)}{\mathrm{d}x} = \sum_{m=0,m\neq i}^{n} \frac{1}{x_i-x_m} \prod_{k=0,k\neq(i,m)}^{n} \frac{(x-x_k)}{(x_i-x_k)}$$

Lagrange-Interpolation: Empfehlung für Umsetzung

- `function wert=LagrangeDerivBasis(x,n,i,x_node)`
- `function wert=LagrangeDerivPolynom(x,n,x_node,f_node)`

Verwende immer die elementsweisen Operatoren in den Funktionen
`(.* , ./ , .^)`

Aufgabe 2: Interpolation mit Lagrange-Polynomen

Erstelle ein MATLAB-Programm, das die Auswertung der Lagrange-Polynome und deren Ableitung für einen beliebigen Grad ermöglicht.

Es sind fünf Stützstellen x mit den zugehörigen Funktionswerten $f(x) = \left(\frac{x}{1+x}\right)^5$ gegeben:

x	0.0	1,0	2.0	3.0	4.0
$f(x)$	0.000000000000	0.031250000000	0.131687242798	0.237304687500	0.327680000000

Berechne mit Hilfe des erstellten Programmes den Funktionswert und die Ableitung an der Stelle $x = 0.6$ und plotte die jeweilige Funktion, sowie die Ableitung der untenstehenden Lagrange-Interpolation (Es ist exemplarisch der Plot für Polynome vom Grad 4 abgebildet).

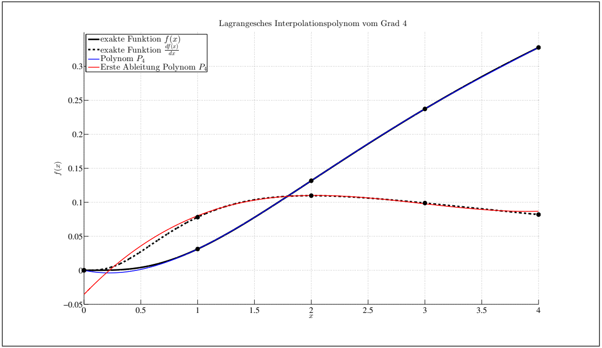
Wenden dies auf folgende Fälle an:

1. Lagrange-Interpolation: Polynome vom Grad 1 - verwenden Sie nur $f(x)$ bei $x = 0.0$; $x = 1.0$

$(Lsg.: f_{L1}(0.6) = 0.01875, f'_{L1}(0.6) = 0.03125)$

2. Lagrange-Interpolation: Polynome vom Grad 4

$(Lsg.: f_{L4}(0.6) = 0.0053987, f'_{L4}(0.6) = 0.046593)$



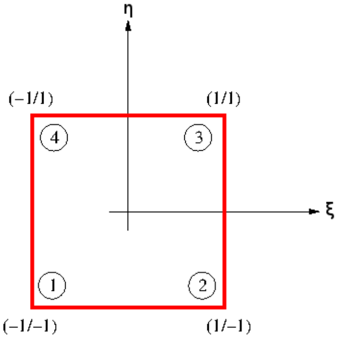
3. Lagrange-Interpolation: Polynome vom Grad 80 - Verwenden Sie dazu die gegebene Funktion $f(x)$ und werten Sie diese in gleichmäßigem Abstand im Intervall $[0.0, 4.0]$ aus.

$(Lsg. : f_{L80}(0.6) = 0.0074158, f'_{L80}(0.6) = 0.038624)$

(HINWEIS: Exakte Lösung der zugrunde liegenden Funktion: $f(0.6) = 0.0074158$, $f'(0.6) = 0.038624$)

gegeben:

- zweidimensionales vierknotiges Element Ω^e im $\boldsymbol{\xi} = (\xi, \eta)$-Koordinatensystem.
- $f(\xi, \eta)$



Lagrange-Interpolation: 2 dimensional

Die Lagrange'schen bilinearen Ansatzfunktionen in 2D sind wie folgt definiert:

$$N^1(\xi,\eta) = \frac{1}{4}(1-\xi)(1-\eta) \qquad N^3(\xi,\eta) = \frac{1}{4}(1+\xi)(1+\eta)$$

$$N^2(\xi,\eta) = \frac{1}{4}(1+\xi)(1-\eta) \qquad N^4(\xi,\eta) = \frac{1}{4}(1-\xi)(1+\eta)$$

Lagrange-Interpolation: Vorgabe für Umsetzung

- **Fkt. I**: `function val=linquadref(xi,eta)`
  Rückgabewert: Lagrange Polynome ausgewertet im Punkt (xi,eta).

- **Fkt. II**: `function deriv=linquadderivref(xi,eta)`
  Rückgabewert: Ableitungen der Lagrange Polynome ausgewertet im Punkt (xi,eta).

Praktikum Numerische Methoden für Ingenieure

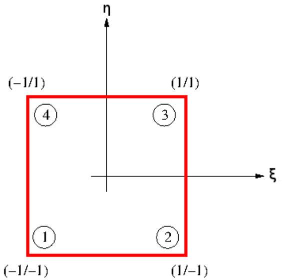
Aufgabenblatt 3

WS 2016/2017

Interpolation und Kurvenanpassung

Aufgabe 1: 2d - Interpolation

Gegeben ist ein zweidimensionales vierknotiges Element (siehe Abbildung) Ω^e im $\boldsymbol{\xi} = (\xi, \eta)$-Koordinatensystem.



An den vier Knoten sind folgende Funktionswerte $f(\xi_i, \eta_i)$ gegeben:

$(\xi \vert \eta)$	$(-1 \vert -1)$	$(+1 \vert -1)$	$(+1 \vert +1)$	$(-1 \vert +1)$
$f(\xi, \eta)$	0.0	1.0	3.0	1.0

Mithilfe von Lagrange'schen Ansatzfunktionen $N^i(\xi, \eta)$ sollen die Funktionswerte $f(\xi, \eta)$ sowie die Ableitungen $\frac{\partial f(\xi,\eta)}{\partial \xi}$, $\frac{\partial f(\xi,\eta)}{\partial \eta}$ an den Punkten $(\xi|\eta) = (0.0; 0.0)$ sowie $(\xi|\eta) = (0.577; -0.577)$ approximiert werden.

Die Lagrange'schen bilinearen Ansatzfunktionen in 2D sind wie folgt definiert:

$N^1(\xi, \eta) = \frac{1}{4}(1-\xi)(1-\eta) \qquad N^3(\xi, \eta) = \frac{1}{4}(1+\xi)(1+\eta)$

$N^2(\xi, \eta) = \frac{1}{4}(1+\xi)(1-\eta) \qquad N^4(\xi, \eta) = \frac{1}{4}(1-\xi)(1+\eta)$

Vorgehen:

1. Erstellen der Funktion **Fkt. I** (siehe unten), die alle Ansatzfunktionen $N^i(\xi, \eta)$ als Vektor zurückgibt.

2. Approximieren der Funktionswerte $f(0.0; 0.0)$ und $f(0.577; -0.577)$.

   $(Lsg.: f_L(0.0; 0.0) = 1.25$ und $f_L(0.577; -0.577) = 1.16676775)$

3. Erstellen der Funktion **Fkt. II**, die alle Ableitungen der Ansatzfunktionen $\frac{\partial N^i(\xi,\eta)}{\partial \xi}, \frac{\partial N^i(\xi,\eta)}{\partial \eta}$ als Matrix zurückgibt (Zeilen i, Spalten ξ, η).

4. Approximieren der Ableitungen $\frac{\partial f(\xi,\eta)}{\partial \eta}$, $\frac{\partial f(\xi,\eta)}{\partial \xi}$ in den Punkten $(\xi|\eta) = (0.0; 0.0)$ sowie $(\xi|\eta) = (0.577| - 0.577)$
$(Lsg.: \frac{\partial f(\xi,\eta)}{\partial \xi}_L|_{(0.0;0.0)} = 0.75; \qquad \frac{\partial f(\xi,\eta)}{\partial \eta}_L|_{(0.0;0.0)} = 0.75)$
$(Lsg.: \frac{\partial f(\xi,\eta)}{\partial \xi}_L|_{(0.577;\text{-}0.577)} = 0.60575; \qquad \frac{\partial f(\xi,\eta)}{\partial \eta}_L|_{(0.577;\text{-}0.577)} = 0.89425)$

Matlab Funktionen:

Folgende Funktionen sollen bei der Bearbeitung dieses Aufgabenblattes erstellt werden, da diese für spätere Aufgabenblätter wiederverwendet werden sollen. Erstellen Sie die Funktionen in Matlab und speichern Sie diese in eigenen *.m-files ab.

- **Fkt. I**: `function val=linquadref(xi,eta)`

  Rückgabewert: Lagrange Polynome ausgewertet im Punkt (xi,eta).

  Teste die Funktion mit:

$$(xi = 0.0, eta = 0.0) \rightarrow [0.25; 0.25; 0.25; 0.25]$$

$$(xi = 0.577, eta = -0.577) \rightarrow [0.16676775; 0.62173225; 0.16676775; 0.04473225]$$

- **Fkt. II**: `function deriv=linquadderivref(xi,eta)`

- **Fkt. I**: `function val=linquadref(xi,eta)`

  Rückgabewert: Lagrange Polynome ausgewertet im Punkt (xi,eta).

  Teste die Funktion mit:

$$(xi = 0.0, eta = 0.0) \rightarrow [0.25; 0.25; 0.25; 0.25]$$

$$(xi = 0.577, eta = -0.577) \rightarrow [0.16676775; 0.62173225; 0.16676775; 0.04473225]$$

- **Fkt. II**: `function deriv=linquadderivref(xi,eta)`

  Rückgabewert: Ableitungen der Lagrange Polynome ausgewertet im Punkt (xi,eta).

  Teste die Funktion mit:

$$(xi = 0.0, eta = 0.0) \rightarrow [-0.25, -0.25; 0.25, -0.25; 0.25, 0.25; -0.25, 0.25]$$

$$(xi = 0.577, eta = -0.577) \rightarrow [-b, -a; b, -b; a, b; -a, a] \qquad mit \quad a = 0.10575, b = 0.39425$$

Und Los ...

Nächste Tutorsprechstunden:

Montag 31.10. 16:00-18:15, 1264 - Computer-Red-Pool
Mittwoch 02.11. 15:30-17:45, 1264 - Computer-Red-Pool

Montag 07.11. 16:00-18:15, 1264 - Computer-Red-Pool
Mittwoch 09.11. 15:30-17:45, 1264 - Computer-Red-Pool

Nächste Aufgabenblätter:

Donnerstag 03.11. 17:00-17:45, 0350 - entfällt!
Donnerstag 10.11. 17:00-17:45, 0350