Prof. Dr. Günther Greiner
Lehrstuhl für Graphische Datenverarbeitung
der Universität Erlangen–Nürnberg

24. Juli 2014

Algorithmik kontinuierlicher Systeme — 24. Juli 2014

Angaben zur Person (Bitte in DRUCKSCHRIFT ausfüllen!):

Name, Vorname: .....

Geburtsdatum: .....

Matrikelnummer: .....

Studienfach: .....

Nicht von der Kandidatin bzw. vom Kandidaten auszufüllen !

Bewertung:

Aufgabe	1	2	3	4	5	6	7	8	9	10
Max. Punktzahl	6	8	10	11	6	11	8	13	6	11
Erreichte Punkte										

Gesamtpunktzahl	
Note	

Organisatorische Hinweise

Die folgenden Hinweise bitte aufmerksam lesen und die Kenntnisnahme durch Unterschrift bestätigen!

- Bitte legen Sie Ihren Studentenausweis und einen Lichtbildausweis zur Personenkontrolle bereit.
- Hilfsmittel (außer Schreibmaterial **und** Taschenrechner) sind nicht zugelassen. Andere elektronische Geräte sind auszuschalten.
- Fragen zu den Prüfungsaufgaben werden grundsätzlich nicht beantwortet.
- Die Lösung einer Aufgabe muss auf das jeweilige Aufgabenblatt geschrieben werden. Sollte der Platz nicht reichen, so verwenden Sie die Zusatz-Seiten am Ende der Klausur. Fügen Sie einen Hinweis in Ihre Lösung ein, dass die Lösung auf den Zusatz-Seiten fortgesetzt wurde und beschriften Sie diese mit Namen und Aufgabennummer.
- Es können durch die Aufsicht zusätzlich Seiten eingeheftet werden, sollte mehr Platz benötigt werden. Bitte beschriften Sie den Kopf dieser Seiten mit Ihrem Namen und der Aufgabennummer. Streichen Sie alles, was nicht bewertet werden soll doppelt aus.
- Auf Ihrem Platz befinden sich einige lose Blätter Schmierpapier. Bei Bedarf können Sie zusätzliches Schmierpapier von der Aufsicht anfordern. Das Schmierpapier muss abgegeben werden, es wird aber nicht bewertet.
- Wenn Sie die Prüfung aus gesundheitlichen Gründen abbrechen müssen, so muss Ihre Prüfungsunfähigkeit durch eine Untersuchung in der Universitätsklinik nachgewiesen werden. Melden Sie sich bei der Aufsicht und lassen Sie sich das entsprechende Formular aushändigen.
- Die Bearbeitungszeit beträgt 90 Minuten.
- Überprüfen Sie die Prüfungsaufgaben auf Vollständigkeit (26 Seiten inklusive Deckblatt) und einwandfreies Druckbild.
- Vergessen Sie nicht, auf dem Deckblatt die Angaben zur Person einzutragen und die **Erklärungen auf dieser Seite zu unterschreiben**.
- Viel Erfolg!

Erklärungen

Durch meine Unterschrift bestätige ich den Empfang der vollständigen Klausurunterlagen und die Kenntnisnahme der obigen Informationen.

Erlangen, 24. Juli 2014 .....
(Unterschrift)

Ich bin damit einverstanden, dass mein Prüfungsergebnis unter Angabe der Matrikelnummer anonymisiert veröffentlicht wird:

ja: ☐ nein: ☐

Erlangen, 24. Juli 2014 .....
(Unterschrift)

1 Theoriefragen und Konvergenz (6 Punkte)

a) Beantworten Sie die folgenden Fragen! Schreiben Sie ihre Antwort in die rechte Spalte der Tabelle!

Welche Komplexität hat die Matrix-Vektor-Multiplikation $\mathbf{A}\vec{b}$, wenn $\mathbf{A}$ eine (beliebige) vollbesetzte $(n \times n)$-Matrix und $\vec{b}$ ein n-Vektor ist?	O( )
Welche Komplexität hat die Matrix-Vektor-Multiplikation $\mathbf{A}\vec{b}$ wenn $\mathbf{A}$ eine $(n \times n)$-Matrix vom Rang 1 ist, d.h. $\mathbf{A} = \vec{u}\,\vec{v}^T$?	O( )
Welche Komplexität hat die Bestimmung der LR-Zerlegung $\mathbf{A} = \mathbf{LR}$ einer 5-diagonalen $(n \times n)$-Matrix $\mathbf{A}$?	O( )
Welche Komplexität hat die Berechnung der Determinate einer $(n \times n)$-Matrix $\mathbf{A}$, wenn die LR-Zerlegung von $\mathbf{A} = \mathbf{LR}$ bekannt ist?	O( )
Welche Komplexität hat die Durchführung eines Iterationsschrittes des Gauss-Seidel-Verfahrens für eine vollbesetzte $(n \times n)$-Matrix?	O( )
Welche Komplexität hat das Auswerten eines einzelnen Punktes auf einer Bézier-Kurve mit n Kontrollpunkten mittels des Algorithmus von De Casteljau?	O( )
Wie groß ist der Approximationsfehler des Catmull-Rom-Interpolanten im Falle äquidistanter Schrittweite h ?	O( )
Wie groß ist der Approximationsfehler der iterierten Trapez-Regel für die Schrittweite h ?	O( )

b) Erklären Sie den Begriff **Konvergenzordnung** für eine konvergente Iterationsfolge (x_i) mit Grenzwert x^*.

2 Direkte Verfahren für lineare Gleichungssysteme (8 Punkte)

Von der Matrix $\mathbf{A}$ ist die QR-Zerlegung bekannt, es gilt

$$\mathbf{A} = \begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ -1 & -1 & 2 & 0 \\ 1 & -1 & 0 & 2 \end{bmatrix} = \begin{bmatrix} -1/2 & 1/2 & 1/2 & 1/2 \\ 1/2 & 1/2 & 1/2 & -1/2 \\ -1/2 & -1/2 & 1/2 & -1/2 \\ 1/2 & -1/2 & 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} 2 & 0 & -2 & 1 \\ 0 & 2 & -1 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

a) Lösen Sie das folgende lineare Gleichungssystem (unter Verwendung der QR-Zerlegung)

$$\mathbf{A}\vec{x} = \vec{b} \qquad \text{für} \qquad \vec{b} = [4, 2, 2, 4]^T.$$

b) Bestimmen Sie einen Vektor $\vec{w}$, so dass die zugehörige Householder-Spiegelung $\mathbf{H}_{\vec{w}}$ bei Anwendung auf $\mathbf{A}$ (von Teilaufgabe a)) Nullen in der ersten Spalte generiert, d.h.

$$\mathbf{H}_{\vec{w}} \cdot \mathbf{A} = \begin{bmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \end{bmatrix}.$$

c) Bestimmen Sie die LR-Zerlegung folgender Matrix $\mathbf{B} = \begin{bmatrix} 2 & 0 & -2 & 1 \\ 4 & 2 & -5 & 2 \\ -2 & 0 & 3 & 1 \\ 4 & -4 & -3 & 1 \end{bmatrix}$.

3 Singulärwertzerlegung (SVD) (10 Punkte)

Von der Matrix $\mathbf{A} = \frac{1}{25} \cdot \begin{bmatrix} 24 & -12 & -6 & -12 \\ 6 & -3 & 6 & 12 \\ -32 & 16 & 8 & 16 \\ 8 & -4 & 8 & 16 \end{bmatrix}$ ist die Singulärwertzerlegung bekannt und zwar:

$$\mathbf{A} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^T = \underbrace{\frac{1}{5} \cdot \begin{bmatrix} 3 & 0 & -4 & 0 \\ 0 & -3 & 0 & -4 \\ -4 & 0 & -3 & 0 \\ 0 & -4 & 0 & 3 \end{bmatrix}}_{\mathbf{U}} \underbrace{\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{\Sigma}} \underbrace{\frac{1}{5} \cdot \begin{bmatrix} 4 & -2 & -1 & -2 \\ -2 & 1 & -2 & -4 \\ -1 & -2 & 4 & -2 \\ -2 & -4 & -2 & 1 \end{bmatrix}}_{\mathbf{V}^T}$$

a) Bestimmen Sie die zu $||\cdot||_1$, $||\cdot||_2$ und $||\cdot||_\infty$ assoziierten Matrixnormen von $\mathbf{A}$.

$|||\mathbf{A}|||_1 \;=$

$|||\mathbf{A}|||_2 \;=$

$|||\mathbf{A}|||_\infty \;=$

b) Bestimmen Sie die Singulärwerte und den Rang der Matrix $\mathbf{A}$.

c) Bestimmen Sie den Kern und den Bildraum der Matrix $\mathbf{A}$.

$\mathbf{ker}(\mathbf{A}) \;=$

$\mathbf{im}(\mathbf{A}) \;=$

Erinnerung:

$$\mathbf{A} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^T = \underbrace{\frac{1}{5}\cdot\begin{bmatrix} 3 & 0 & -4 & 0 \\ 0 & -3 & 0 & -4 \\ -4 & 0 & -3 & 0 \\ 0 & -4 & 0 & 3 \end{bmatrix}}_{\mathbf{U}} \underbrace{\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{\Sigma}} \underbrace{\frac{1}{5}\cdot\begin{bmatrix} 4 & -2 & -1 & -2 \\ -2 & 1 & -2 & -4 \\ -1 & -2 & 4 & -2 \\ -2 & -4 & -2 & 1 \end{bmatrix}}_{\mathbf{V}^T}$$

d) Bestimmen Sie mit Hilfe der Pseudo-Inversen die Lösung des singulären Gleichungssystems

$$\mathbf{A}\vec{x} = \vec{b} \qquad \text{für} \qquad \vec{b} = [0, 0, 1, 1]^T \ .$$

Hinweis: Die **explizite** Bestimmung der Pseudo-Inversen ist hierfür nicht erforderlich!

e) In welchem Sinn löst die mit Hilfe der Pseudo-Inversen bestimmte Lösung das (singuläre) Gleichungssystem $\mathbf{A}\vec{x} = \vec{b}$?

4 Programmierung: Dreiecke, Baryzentrische Koordinaten (11 Punkte)

In dieser Aufgabe soll eine Klasse `Triangle` in `C++` implementiert werden, die ein Dreieck im zweidimensionalen Raum repräsentiert und verschiedene Methoden zur Verfügung stellt. Entgegen der Übungsaufgaben ist **keine** Fehlerbehandlung notwendig.

Die aus den Übungen bekannte Klasse `Point2D`, die einen 2D-Punkt darstellt, ist gegeben:

```
class Point2D {
public:
    float x, y;                    //! Interne Darstellung
    Point2D(float x, float y);     //! Konstruktor
    ...                            //! weitere Konstruktoren und Operatoren
};
```

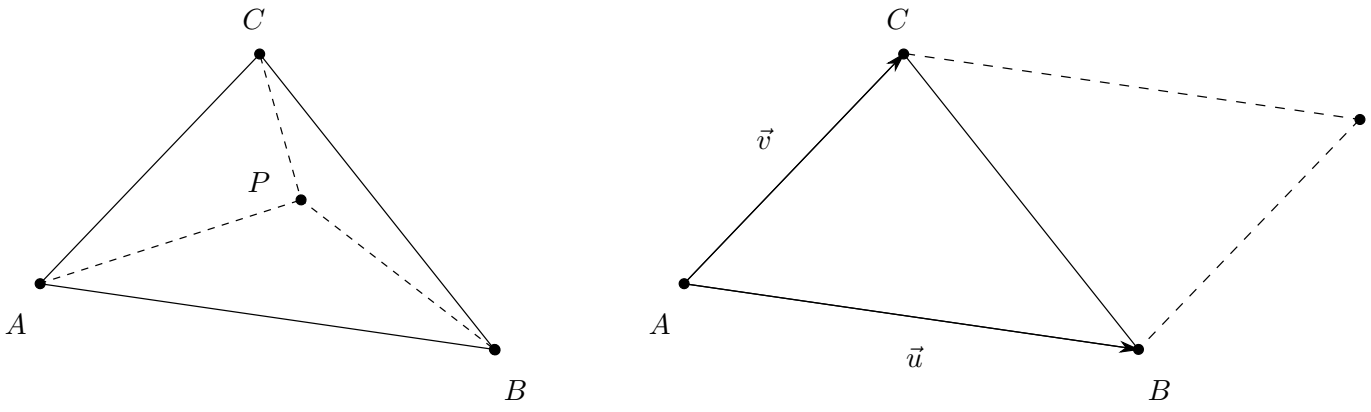
Hinweis: Sie können davon ausgehen, dass für Objekte der Klasse `Point2D` alle arithmetischen Standardoperatoren (`+, -, *, /, =, ==, !=, +=, -=, *=, /=`) zur Verfügung stehen.

Die Klasse `Triangle` hat folgende Struktur:

```
class Triangle {
public:
  //! Interne Darstellung: Drei Punkte
  Point2D A, B, C;

  //! Konstruktor
  Triangle(const Point2D &a, const Point2D &b, const Point2D &c);

  //! Berechnet die (vorzeichenbehaftete) Flaeche und gibt diese zurueck
  float area() const;

  //! Berechnet die baryzentrischen Koordinaten des Punktes P
  void barycentric(const Point2D &P, float &alpha, float &beta, float &gamma) const;

  //! Gibt true zurueck, wenn der Punkt P im Dreieck liegt
  bool inside(const Point2D &P) const;
};
```

Hinweis: Verändern Sie die Klassenstruktur nicht, d.h. führen Sie keine neuen Attribute oder Methoden ein.

Zur Veranschaulichung ist ein allgemeines Dreieck $\triangle(ABC)$, ein Punkt P und das Parallelogramm, das von den Vektoren $\vec{u} = B - A$ und $\vec{v} = C - A$ aufgespannt wird, gegeben:



a) Implementieren Sie zunächst den Konstruktor `Triangle(const Point2D &a, const Point2D &b, const Point2D &c)`, der die entsprechenden Membervariablen mit den übergebenen Punkten initialisiert.

```
Triangle::Triangle(const Point2D &a, const Point2D &b, const Point2D &c)









{









}
```

b) Implementieren Sie die Methode `area()`, die den (vorzeichenbehafteten) Flächeninhalt des Dreiecks $\triangle(ABC)$ berechnet und zurückgibt. Der vorzeichenbehaftete Flächeninhalt des oben gezeigten Parallelogramms ist gegeben durch: $\det(B - A, C - A)$.

```
float Triangle::area() const {









}
```

c) Die Methode `barycentric(const Point2D &P, float &alpha, float &beta, float &gamma)` soll zu einem übergebenen Punkt P die baryzentrischen Koordinaten α, β und γ berechnen.
Dabei soll gelten $P = \alpha A + \beta B + \gamma C$.

Implementieren Sie diese Methode und verwenden Sie dabei `area()` aus Teilaufgabe a) an entsprechender Stelle, auch wenn Sie diese Methode nicht implementiert haben.

```
void Triangle::barycentric(const Point2D &P, float &alpha, float &beta, float &gamma) const {

}
```

d) Implementieren Sie nun die Methode `inside(const Point2D &P)`, die prüft, ob ein gegebener Punkt P innerhalb des Dreiecks $\triangle(ABC)$ liegt. Der Rand soll dabei zum Dreieck gehören.

Greifen Sie bei der Implementierung auf die Methode `barycentric(const Point2D &P, float &alpha, float &beta, float &gamma)` zurück, auch wenn Sie diese Methode nicht implementiert haben.

```
bool Triangle::inside(const Point2D &P) const {

}
```

5 Interpolation in einem Prisma (6 Punkte)

a) Gegeben ist das Dreieck $\Delta(RST)$ mit den Eckpunkten

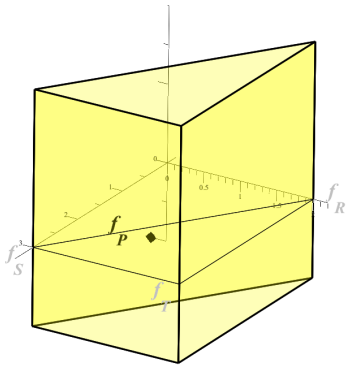
$$R = \begin{bmatrix}0\\2\end{bmatrix}\ , \quad S = \begin{bmatrix}3\\0\end{bmatrix}\ , \quad T = \begin{bmatrix}3\\2\end{bmatrix}\ , \qquad \text{sowie der Punkt} \quad P = \begin{bmatrix}2\\1\end{bmatrix}\ .$$

Bestimmen Sie die baryzentrischen Koordinaten (ρ, σ, τ) des Punktes P bezgl. $\Delta(RST)$.

b) Gegeben sei ein Prisma mit dreieckigen Deckflächen $\Delta(R_0\, S_0\, T_0)$ und $\Delta(R_1\, S_1\, T_1)$ (siehe Abbildung), wobei

$$R_0 = \begin{bmatrix}0\\2\\-1\end{bmatrix}\ , \quad S_0 = \begin{bmatrix}3\\0\\-1\end{bmatrix}\ , \quad T_0 = \begin{bmatrix}3\\2\\-1\end{bmatrix}\ ,$$

$$R_1 = \begin{bmatrix}0\\2\\2\end{bmatrix}\ , \quad S_1 = \begin{bmatrix}3\\0\\2\end{bmatrix}\ , \quad T_1 = \begin{bmatrix}3\\2\\2\end{bmatrix}\ .$$



In den Eckpunkten sind folgende Temperaturwerte bekannt:

$$f_{R_0} = 12\ , \quad f_{S_0} = 48\ , \quad f_{T_0} = 36\ , \quad f_{R_1} = 12\ , \quad f_{S_1} = 72\ , \quad f_{T_1} = 72.$$

Ermitteln Sie mittels Interpolation den Temperaturwert im Punkt $P = [2, 1, 0]^T$.

6 Iterative Lösungsverfahren (11 Punkte)

Gegeben seien die Matrix $\mathbf{A}$ sowie der Vektor $\vec{b}$ mit

$$\mathbf{A} = \begin{bmatrix} 6 & -1 & 1 \\ 2 & 4 & 1 \\ -2 & 2 & 6 \end{bmatrix}, \qquad \vec{b} = \begin{bmatrix} 12 \\ -4 \\ 6 \end{bmatrix}.$$

a) Warum konvergiert das JACOBI-Verfahren zur Lösung von $\mathbf{A}\vec{x} = \vec{b}$?

b) Führen Sie **zwei** Schritte des JACOBI-Verfahrens zur Lösung von $\mathbf{A}\vec{x} = \vec{b}$ durch. Verwenden Sie als Startvektor $[0, 0, 0]^T$.

Zur Erinnerung:

$$\mathbf{A} = \begin{bmatrix} 6 & -1 & 1 \\ 2 & 4 & 1 \\ -2 & 2 & 6 \end{bmatrix}, \qquad \vec{b} = \begin{bmatrix} 12 \\ -4 \\ 6 \end{bmatrix}.$$

c) Führen Sie **einen** Schritt des Gauss-Seidel-Verfahrens zur Lösung von $\mathbf{A}\vec{x} = \vec{b}$ durch. Verwenden Sie als Startvektor $[0, 0, 0]^T$.

Betrachten Sie nun die tridiagonale $(n \times n)$-Matrix $\mathbf{B}_n$ mit $b_{i,i} = \frac{1}{2}$ und $b_{i-1,i} = b_{i,i-1} = \frac{1}{4}$.

d) Ist das JACOBI-Verfahren für diese Matrix konvergent? Begründen Sie Ihre Antwort!

e) Ist es sinnvoll, das Gleichungssystem $\mathbf{B}_n\vec{x} = \vec{b}$ mithilfe des JACOBI-Verfahrens zu lösen? Begründen Sie Ihre Antwort!

7 Nichtlineare Optimierung (8 Punkte)

Die folgende, positiv definite Matrix $\mathbf{A}$ wird in Teilaufgabe a) und b) verwendet: $\mathbf{A} = \begin{bmatrix} 3 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 2 \end{bmatrix}$.

a) Beantworten Sie die folgenden Fragen und begründen Sie Ihre Antwort.

- Sind die beiden Vektoren $\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ und $\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$ $\mathbf{A}$-konjugiert?

- Sind die beiden Vektoren $\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$ und $\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$ $\mathbf{A}$-konjugiert?

b) Wieviele Iterationen (Abstiegsschritte) sind nötig, wenn man das Optimierungsproblem für das quadratische Funktional

$$Q(\vec{x}) = \vec{x}^T \mathbf{A} \vec{x} + 2\vec{b}^T \vec{x} + \gamma, \qquad (\vec{b} = [2, 0, -3]^T, \quad \gamma = -5)$$

mit dem cg-Verfahren löst (exakte Rechnung vorausgesetzt)?

In den Teilaufgaben c) und d) wird folgende zu minimierende nichtlineare Funktion zweier Variablen betrachtet:

$$F(x, y) = 2x^2 + y^2 - x^2y^2 + x - y + 2$$

c) Führen Sie einen Schritt des Gradientenabstiegs-Verfahrens durch (steepest descent).
Verwenden Sie die Schrittweite $t_0 = \frac{1}{2}$ und beginnen Sie mit dem Startwert
$[x_0, y_0] = [-1, 1]$.

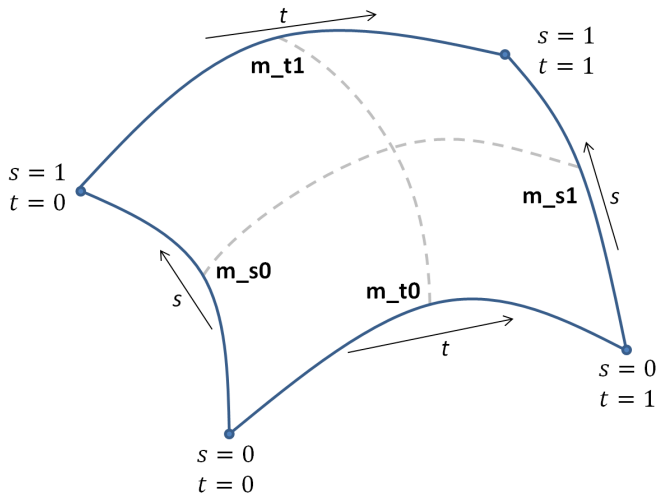
d) Führen Sie einen Schritt des NEWTON-Verfahrens durch und beginnen Sie mit dem Startwert
$[x_0, y_0] = [0, 0]$.

8 Programmierung: Bilineare Interpolation, Coons Patches (13 Punkte)

In dieser Aufgabe soll ein Coons Patch (`CoonsPatch`) in `C++` implementiert werden. Entgegen der Übungen ist **keine** Fehlerbehandlung erforderlich.

```
class CoonsPatch {
  public:
  ...
  //! Wertet den den bilinearen Anteil an (s, t) aus
  float evaluateBilinear(float s, float t) const;

  //! Wertet den Patch an (s, t) aus
  float evaluateAt(float s, float t) const;

  //! Berechnet die Ableitung des Patches in t-Richtung an der Stelle (s, t)
  float evaluateDerivativeT(float s, float t) const;

 protected:
  //! Funktionen, die den Patch definieren (siehe Grafik!)
  const ParametricFunction &m_s0;    // linke Funktion; hat den Parameter s
  const ParametricFunction &m_s1;    // rechte Funktion; hat den Parameter s
  const ParametricFunction &m_t0;    // untere Funktion; hat den Parameter t
  const ParametricFunction &m_t1;    // obere Funktion; hat den Parameter t
};

class ParametricFunction {
  public:
  //! Wertet die Funktion an der Stelle t (in [0,1]) aus
  virtual float f(float t) const = 0;

  //! Auswertung der Ableitung der Funktion an der Stelle t (in [0,1])
  virtual float df(float t) const = 0;
};
```

Visualisierung des Coons Patches mit entsprechender Zuordnung und Parametrisierung der Randkurven:



a) Implementieren Sie die Methode `evaluateBilinear(float s, float t)`, die zunächst die Funktionswerte an den vier Eckpunkten durch Auswertung der Randkurven bestimmt und diese **bilinear** für die Stelle (s, t) interpoliert. Der interpolierte Wert soll dann zurückgegeben werden. Sie können davon ausgehen, dass sowohl s als auch t im Intervall $[0, 1]$ liegen. Achten Sie dabei auf die korrekte Zuordnung der Randkurven (siehe Grafik!).

```
float CoonsPatch::evaluateBilinear(float s, float t) const {









}
```

b) Implementieren Sie die Methode `evaluateAt(float s, float t)`, welche das **Coons Patch** an der Stelle (s, t) auswertet. Greifen Sie dabei auf die Methode `evaluateBilinear(float s, float t)` an entsprechender Stelle zurück, auch wenn Sie diese Methode nicht implementiert haben. Sie können davon ausgehen, dass sowohl s als auch t im Intervall $[0, 1]$ liegen. Geben Sie anschließend den berechneten Wert zurück. Achten Sie dabei auf die korrekte Zuordnung der Randkurven (siehe Grafik!).

```
float CoonsPatch::evaluateAt(float s, float t) const {









}
```

c) Implementieren Sie die Methode `evaluateDerivativeT(float s, float t)`, welche die Ableitung des Coons Patches in t-Richtung an der Stelle (s, t) auswertet. Sie können davon ausgehen, dass sowohl s als auch t im Intervall $[0, 1]$ liegen. Geben Sie anschließend den berechneten Wert zurück. Achten Sie dabei auf die korrekte Zuordnung der Randkurven (siehe Grafik!).

Hinweis: Überlegen Sie sich zunächst, wie die Ableitungen (in t-Richtung) der einzelnen Bestandteile des Coons Patches zu berechnen sind.

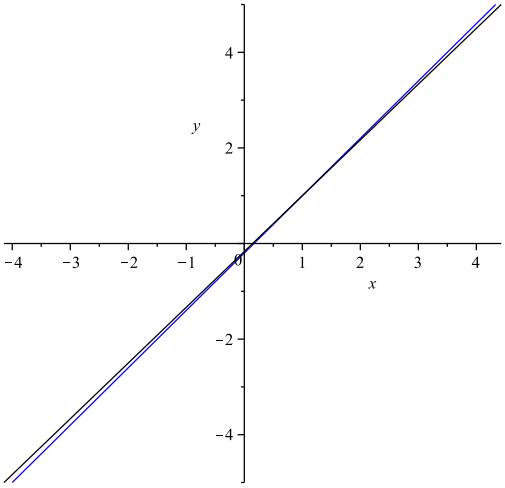
```
float CoonsPatch::evaluateDerivativeT(float s, float t) const {

}
```

9 Gemischte Aufgaben (6 Punkte)

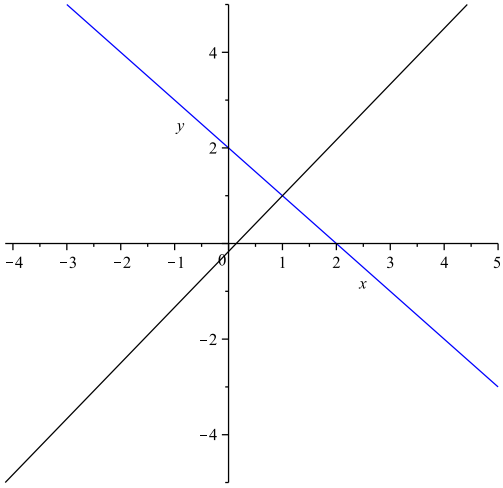
9.1 Kondition

a) Der Schnittpunkt zweier Geraden soll bestimmt werden.

Geben Sie an, ob in den folgenden Situationen die Bestimmung des Schnittpunktes gut oder schlecht konditioniert ist (keine Begründung erforderlich!).



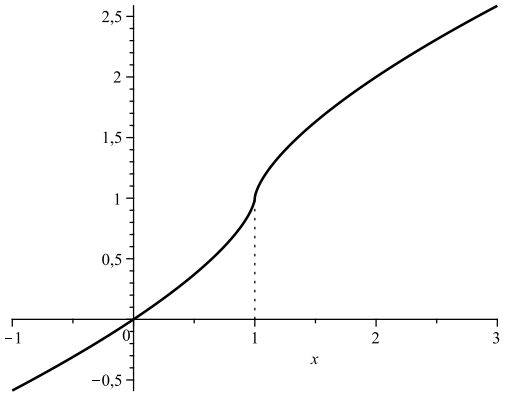
Die Geraden G und H_1 .



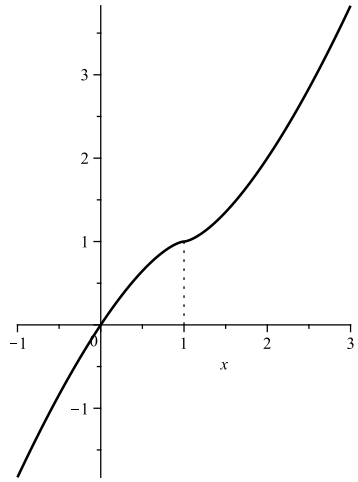
Die Geraden G und H_2 .

b) Die Funktionen $f_1(x)$ und $f_2(x)$ sollen in der Nähe der Stelle 1, also für $x \approx 1$, mit einem (nicht näher spezifiziertem) Algorithmus ausgewertet werden. Die folgenden Abbildungen zeigen jeweils den Graph der Funktion f_1 (links) und f_2 (rechts).

Geben Sie jeweils an, ob das Problem gut oder schlecht konditioniert ist (keine Begründung erforderlich!).



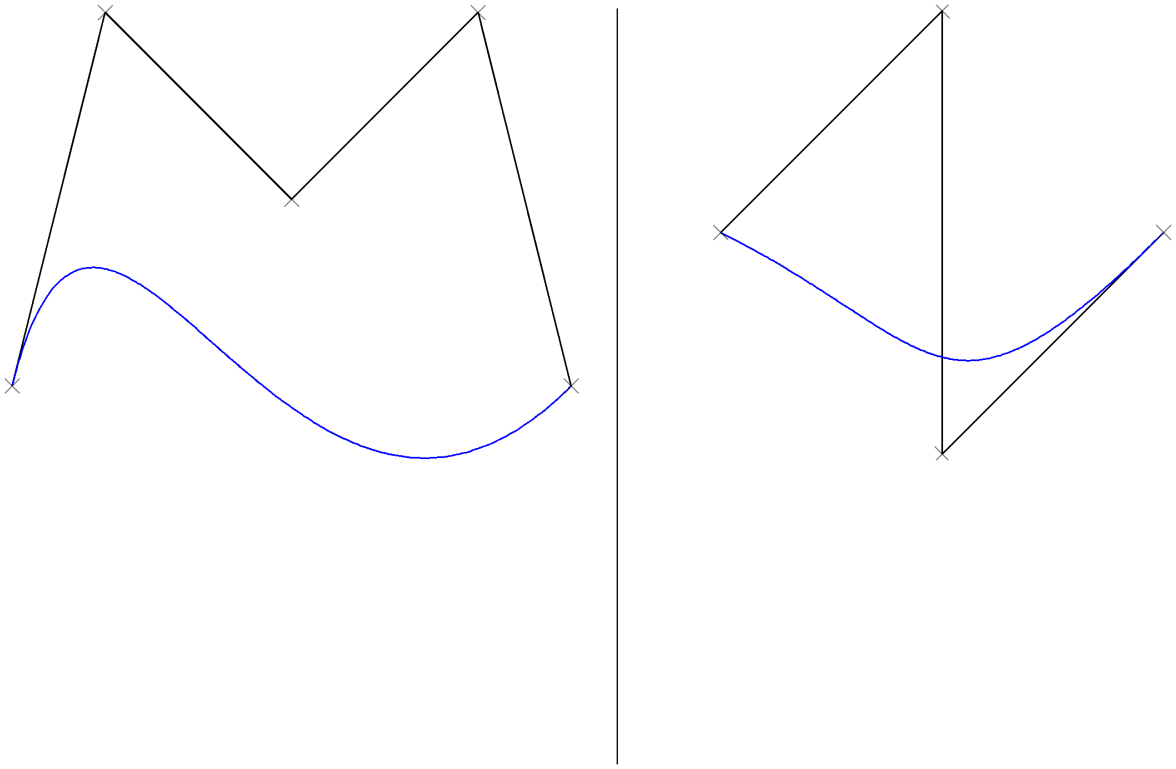
Graph von f_1(x) = 1 + signum $(x-1)\,|x-1|^{2/3}$.



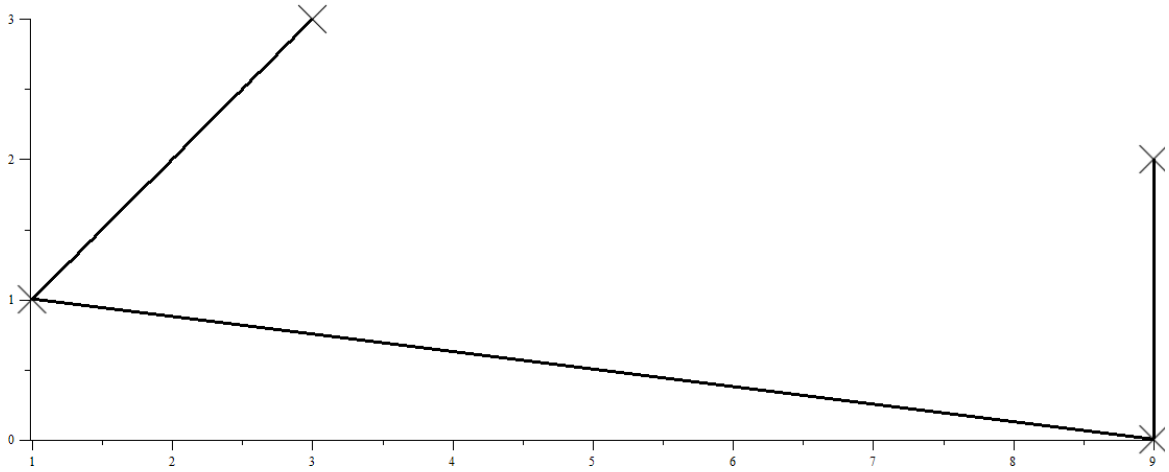
Graph von f_2(x) = 1 + signum $(x-1)\,|x-1|^{3/2}$.

9.2 Bézier-Kurven

a) Die nachfolgend gezeigten Kurven sind keine Bézier-Kurven, denn nicht alle Formeigenschaften sind erfüllt. Geben Sie für jedes Beispiel die Formeigenschaften an, die **nicht** erfüllt sind.
(pro richtiger Angabe $+\frac{1}{2}$ Punkt, pro falscher Angabe $-\frac{1}{2}$ Pkt!)



b) Die folgende Abbildung zeigt das Kontrollpolygon einer kubischen Bézier-Kurve $C : [0, 1] \to \mathbb{R}^2$.
Bestimmen Sie graphisch (mit Hilfe des Algorithmus von de Casteljau) den Kurvenpunkt $C(0.5)$.



10 Falten und Filtern (11 Punkte)

10.1 Diskrete Faltung – Filter

a) Welche der folgenden 2D-Filter sind separierbar? Geben Sie im Falle der Separierbarkeit auch die zugehörigen 1D-Filter an.

$$\frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}$$

$$\frac{1}{4}\begin{bmatrix}0 & 1 & 0\\ 1 & 0 & 1\\ 0 & 1 & 0\end{bmatrix}$$

$$\begin{bmatrix}1 & 2 & 1\\ 0 & 0 & 0\\ -1 & -2 & -2\end{bmatrix}$$

$$\frac{1}{8}\begin{bmatrix}1 & 1 & 0 & 0\\ 0 & 0 & 1 & 1\\ 1 & 1 & 0 & 0\\ 0 & 0 & 1 & 1\end{bmatrix}$$

b) Ein $N \times N$ großer (Grau-)Bildausschnitt wird mit einem nicht separierbaren Filter der Größe $M \times M$ gefiltert. Bestimmen Sie den Rechenaufwand (Anzahl **aller arithmetischen Operationen**) unter der Voraussetzung, dass um den Bildausschnitt ein ausreichend großer Rand vorhanden ist.

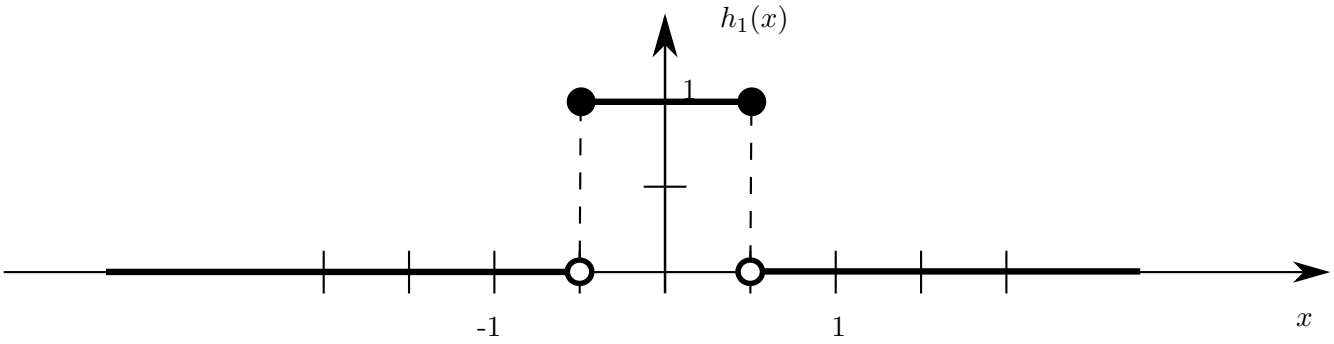
c) Nun soll das Bild aus Teil b) mit einem separierbaren Filter der Größe $M \times M$ gefiltert werden. Wie viele **Multiplikationen** sind hierbei nötig?

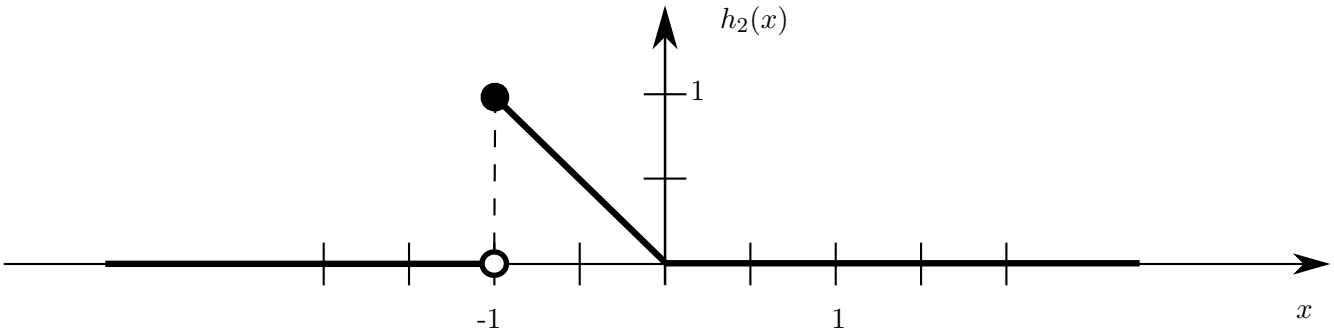
10.2 Kontinuierliche Faltung – Zeichnerisch

Gegeben sind die Funktionen

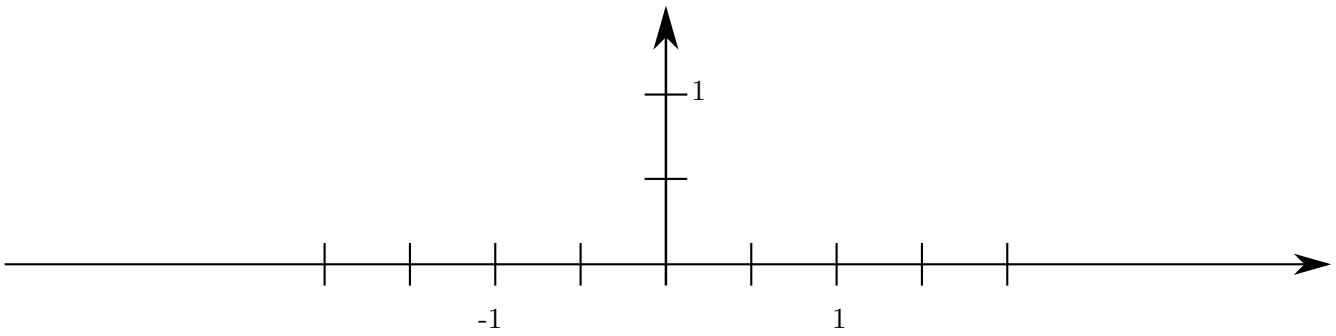
$$h_1(x) := \begin{cases} 1 & \text{falls} \quad |x| \leq 0.5 \\ 0 & \text{sonst} \end{cases}, \qquad h_2(x) := \begin{cases} -x & \text{falls} \quad x \in [-1, 0] \\ 0 & \text{sonst} \end{cases}$$

sowie die Funktionsgraphen der Funktionen h_1 und h_2:





Zeichnen Sie das Ergebnis der Faltung $h_2 * h_1$:



10.3 Kontinuierliche Faltung – Rechnerisch

Gegeben sind die Funktionen

$$h_3(x) := \begin{cases} 1 & \text{falls} \quad |x| \leq 2 \\ 0 & \text{sonst} \end{cases}, \quad h_4(x) := \begin{cases} \frac{3}{4} & \text{falls} \quad 1 \leq x \leq 3 \\ 0 & \text{sonst} \end{cases}.$$

Geben Sie das Ergebnis der Faltung $h_3 * h_4$ als mathematischen Ausdruck an, indem Sie das Faltungsintegral lösen (Hinweis: Überlegen Sie sich zunächst eine sinnvolle Fallunterscheidung).