TN
Teilnehmernummer (von Platzkarte)

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

03. Februar 2015

Algorithmik kontinuierlicher Systeme — 03. Februar 2015

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	9	13	8	11	6	8	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 bei einem Vertrauensarzt 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 (27 Seiten inklusive Deckblatt) und einwandfreies Druckbild.
- Vergessen Sie nicht, auf dem Deckblatt die Angaben zur Person inklusive der Teilnehmernummer (TN) von Ihrer Platzkarte 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, 03. Februar 2015 ....

(Unterschrift)

1 Theoriefragen (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?	$\mathcal{O}(\qquad)$
Welche Komplexität hat die Matrix-Matrix-Multiplikation $\mathbf{AB}$, wenn $\mathbf{A}$ und $\mathbf{B}$ tridiagonale $(n \times n)$-Matrizen sind?	$\mathcal{O}(\qquad)$
Welche Komplexität hat das Lösen eines Gleichungssystems $\mathbf{A}\vec{x} = \vec{b}$, wenn $\mathbf{A}$ eine $(n \times n)$-Matrix und $\vec{b}$ ein n-Vektor ist und die LR-Zerlegung $\mathbf{A} = \mathbf{LR}$ gegeben ist?	$\mathcal{O}(\qquad)$
Welche Komplexität hat die Berechnung des Betrags der Determinante $\lvert\det(\mathbf{A})\rvert$ einer $(n \times n)$-Matrix $\mathbf{A}$, wenn die QR-Zerlegung von $\mathbf{A} = \mathbf{QR}$ bekannt ist?	$\mathcal{O}(\qquad)$
Welche Komplexität hat die Durchführung eines Iterationsschrittes des JACOBI-Verfahrens für eine vollbesetzte $(n \times n)$-Matrix?	$\mathcal{O}(\qquad)$
Welche Komplexität hat die Berechnung der diskreten FOURIER-Transformation (DFT) bei direkter Bestimmung <u>ohne</u> FFT?	$\mathcal{O}(\qquad)$
Wie groß ist der Approximationsfehler des stückweise linearen Interpolanten im Falle äquidistanter Schrittweite h ?	$\mathcal{O}(\qquad)$
Wie groß ist der Approximationsfehler der iterierten SIMPSON-Regel für die Schrittweite h ?	$\mathcal{O}(\qquad)$

b) Gelten folgende Rechengesetze für Gleitpunktzahlen (mit **fester Mantissenlänge**)? Markieren Sie diese jeweils mit **r** („richtig“) bzw. **f** („falsch“)!

☐ Kommutativgesetz: $a + b = b + a$

☐ Assoziativgesetz: $a + (b + c) = (a + b) + c$

2 Dünnbesetzte Matrizen (8 Punkte)

Gegeben ist die Matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 8 & 0 \\ 0 & 2 & 3 & 0 & 0 \\ 0 & 3 & 2 & 1 & 0 \\ 0 & -4 & 0 & -3 & 0 \end{bmatrix}.$$

a) Sie sollen $\mathbf{A}$ im `CCS`-Format abspeichern (Compressed Column Storage). Geben Sie hierzu die interne Datenstruktur an. Die Indizierung beginnt stets bei 1.

b) Die Matrix $\mathbf{A}$ soll nun transponiert werden.Wie kann dies effizient durchgeführt werden?

Genauer: Welches Datenformat wird zur Speicherung von $\mathbf{A}^T$ verwendet, und wie muss die `CCS`-Datenstruktur von $\mathbf{A}$ für das neue Matrixformat modifiziert werden?

c) Gegeben ist nun eine Matrix $\mathbf{B}$ mit 6 Spalten und der Vektor $\vec{b} = [1, 1, 1, 1, 1, 1]^T$.
Das `CRS`-Format (Compressed Row Storage) von $\mathbf{B}$ ist bekannt:

Wertearray:	val =	[4 3 2 2 1 4]
Spaltenindexarray:	col_ind =	[2 1 4 2 3 5]
Zeilenpointerarray:	row_ptr =	[1 2 2 4 7 7]

Bestimmen Sie den Vektor $\mathbf{B}\,\vec{b}$.

3 Direkte Verfahren für lineare Gleichungssysteme (10 Punkte)

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

b) Bestimmen Sie die QR-Zerlegung der 2×2-Matrix $\mathbf{B} = \begin{bmatrix} 3 & 2 \\ 4 & 1 \end{bmatrix}$.

D.h. bestimmen Sie eine orthogonale 2×2-Matrix $\mathbf{Q}$ so dass $\mathbf{Q}^T\mathbf{B} = \begin{bmatrix} * & * \\ 0 & * \end{bmatrix}$. Geben Sie außerdem $\mathbf{R}$ an.

c) Von der Matrix **C** ist die QR-Zerlegung bekannt, es gilt

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

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

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

4 Singulärwertzerlegung (SVD) (9 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 Singulärwerte der Matrix $\mathbf{A}$.

b) Bestimmen Sie den Rang der Matrix $\mathbf{A}$ sowie deren FROBENIUS-Norm.

$\mathrm{rank}(\mathbf{A}) \quad =$

$|||\mathbf{A}|||_F \quad =$

c) Bestimmen Sie die beste Rang-1-Approximation von $\mathbf{A}$ (im Sinne der Frobenius-Norm).

Erinnerung:

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

c) 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!

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

5 Programmierung: Bilder, Filtern (13 Punkte)

In dieser Aufgabe sollen verschiedene Filteroperationen für Graustufenbilder in `C++` implementiert werden. Dazu werden die Klassen `Image` und `Filter` verwendet. Im Gegensatz zu den Übungsaufgaben ist **keine** Fehlerbehandlung erforderlich.

Ein `Image` repräsentiert ein Graustufenbild der Höhe `height` und Breite `width` und hat folgende Struktur:

```
class Image {
public:
    const unsigned int height;  //! Hoehe
    const unsigned int width;   //! Breite

    //! Konstruktor, initialisiert Attribute und legt Speicher an
    Image(unsigned int height, unsigned int width);

    //! Operator zum Lesezugriff an Stelle (i, j)
    const float &operator()(unsigned int i, unsigned int j) const;

    //! Operator zum Schreibzugriff an Stelle (i, j)
    float &operator()(unsigned int i, unsigned int j);

private:
    float *data;   //! Speicher
};
```

Die folgende Abbildung veranschaulicht das Speicherlayout des Bildes in der Variable `data`:

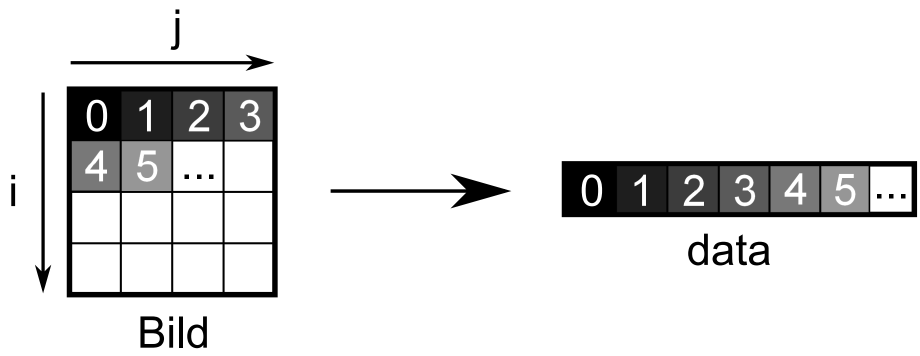


Abbildung 1: Layout eines 4×4-Bildes

Die Klasse `Filter`, welche einige Filteroperationen zur Verfügung stellt, hat folgende Struktur:

```
class Filter {
public:
    //! partielle Ableitung in x-Richtung (horizontal)
    static Image derivativeX(const Image &input);

    //! Laplace-Operator
    static Image laplacian(const Image &input);

    //! Mittelwertfilter der Groesse 3x3
    static Image mean3(const Image &input);
};
```

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

a) Implementieren Sie zunächst den Operator für den Lesezugriff `const float &operator()(unsigned int i, unsigned int j) const`, der eine (konstante) Referenz auf das Bild an der Stelle (i, j) zurückgibt. Beachten Sie das Speicherlayout aus Abbildung 1!

```
const float &Image::operator()(unsigned int i, unsigned int j) const {




}
```

b) Implementieren Sie den Operator für den Schreibzugriff `float &operator()(unsigned int i, unsigned int j)`, welcher eine Referenz auf das Bild an der Stelle (i, j) zurückliefert. Beachten Sie das Speicherlayout aus Abbildung 1!

```
float &Image::operator()(unsigned int i, unsigned int j) {




}
```

c) Die Methode `derivativeX(const Image &input)` soll die partielle Ableitung in x-Richtung $\frac{\partial I}{\partial x}(y, x)$ eines Eingabebildes I (`input`) berechnen. Verwenden Sie als Approximation zentrale Differenzen:

$$\frac{\partial I}{\partial x}(y, x) \approx \frac{1}{2}(I(y, x+1) - I(y, x-1)).$$

Der Rand wird hierbei vernachlässigt, sodass das Ausgabebild `output` um 2 Pixel in der Breite kleiner ist.

```
Image Filter::derivativeX(const Image &input) {
    Image output(input.height, input.width - 2);




    return output;
}
```

d) Der LAPLACE-Operator $\Delta I(y,x) = I_{xx} + I_{yy}$ wird im diskreten Fall durch folgende Filtermatrix K beschrieben:

$$K = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Implementieren Sie diesen Filter für das Eingabebild `input`, sodass im Bild `output` das Ergebnis des Operators zurückgegeben wird. Eine Behandlung des Rands ist nicht erforderlich, sodass das Ausgabebild um 2 Pixel in beide Dimensionen kleiner ist.

```
Image Filter::laplace(const Image &input) {
    Image output(input.height - 2, input.width - 2);



    return output;
}
```

e) Die Filtermatrix K des 3×3-Mittelwertfilters lautet

$$K = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \cdot \frac{1}{3}\left[1, 1, 1\right].$$

Implementieren Sie diesen Filter in der Methode `mean3(const Image &input)` für das übergebene Bild `input`. Nutzen Sie dabei für ein effizientes Vorgehen die Separierbarkeit des Filters aus. Beachten Sie, dass der Rand vernachlässigt wird, sodass das Ausgabebild in beiden Dimensionen um 2 Pixel kleiner ist.

```
Image Filter::mean3(const Image &input) {
    Image output(input.height - 2, input.width - 2);

    return output;
}
```

6 Iterative Lösungsverfahren (8 Punkte)

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

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix}, \qquad \vec{b} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 2 \\ 1 \end{bmatrix}.$$

a) Führen Sie **einen** Schritt des JACOBI-Verfahrens zur Lösung von $\mathbf{A}\vec{x} = \vec{b}$ durch. Verwenden Sie als Startvektor $[1, 1, 1, 1, 1]^T$.

Zur Erinnerung:

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix}, \qquad \vec{b} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 2 \\ 1 \end{bmatrix}.$$

b) Führen Sie **einen** Schritt des Gauss-Seidel-Verfahrens zur Lösung von $\mathbf{A}\vec{x} = \vec{b}$ durch. Verwenden Sie als Startvektor $[1, 1, 1, 1, 1]^T$.

c) Ist das Gauss-Seidel-Verfahren für diese Matrix konvergent? Begründen Sie Ihre Antwort!

7 Programmierung: Nichtlineare Optimierung (11 Punkte)

In dieser Aufgabe sollen verschiedene Methoden zur Optimierung nichtlinearer Funktionen in `C++` implementiert werden. Dabei soll das Minimum einer bivariaten Funktion $f\colon \mathbb{R}^2 \to \mathbb{R}$ gesucht werden:

$$\vec{x}^\star = \underset{\vec{x}\in\mathbb{R}^2}{\operatorname{argmin}} f(\vec{x}).$$

Entgegen der Übungen ist **keine** Fehlerbehandlung erforderlich. Zur Implementierung stehen Ihnen folgende Klassen zur Verfügung:

Ein `vec2` repräsentiert einen Vektor $\vec{v} \in \mathbb{R}^2$. Eine `mat2x2` repräsentiert eine 2×2-Matrix $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \in \mathbb{R}^{2\times 2}$:

```
class vec2 {
public:
  float x, y;                 //! Interne Darstellung: 2 Komponenten
  float length() const;       //! Gibt die euklidische Laenge des Vektors zurueck
};

class mat2x2 {
public:
  float a, b, c, d;           //! Interne Darstellung: 4 Komponenten
  mat2x2 inverse() const;     //! Gibt die inverse Matrix zurueck
};

class Function {
public:
  //! Gibt den Funktionswert an einer Stelle x zurueck
  float f(const vec2 &x) const;

  //! Gibt den Gradienten an einer Stelle x zurueck
  vec2 gradF(const vec2 &x) const;

  //! Gibt die Hesse-Matrix an einer Stelle x zurueck
  mat2x2 hessian(const vec2 &x) const;
};

class Optimizer {
public:
 //! Gradientenabstieg, Funktion f vom Startwert x0 aus, maximal maxIter Iterationen
 static vec2 gradientDescent(const Function &f, const vec2 &x0, int maxIter);

 //! Newtonverfahren, Funktion f vom Startwert x0 aus, maximal maxIter Iterationen
 static vec2 newton(const Function &f, const vec2 &x0, int maxIter);

 //! Berechnet fuer eine Funktion f an der Stelle x zur Richtung d
 //! eine geeignete Schrittweite (nur bei Gradientenabstieg noetig)
 static float stepLength(const Function &f, const vec2 &x, const vec2 &d);
};
```

Hinweise: Sie können davon ausgehen, dass für Objekte der Klassen `vec2` und `mat2x2` alle arithmetischen Standardoperatoren (`+, -, *, /, =, ==, !=, +=, -=, *=, /=`) zur Verfügung stehen. Sie können daher bspw. mit dem Operator `*` eine Matrix-Vektor-Mulitplikation durchführen.

Verändern Sie die Klassenstrukturen nicht, d.h. führen Sie keine neuen Attribute oder Methoden ein.

a) Implementieren Sie in der Methode `gradientDescent(const Function &f, const vec2 &x0, int maxIter)` das Gradientenabstiegsverfahren. Verwenden Sie zur Schrittweitenbestimmung die Methode `stepLength(...)`. Brechen Sie das Verfahren vorzeitig ab, falls die Länge des Gradienten $\|\vec{\nabla} f(\vec{x})\|_2 < 10^{-3}$. Geben Sie am Ende die gefundene Minimalstelle $\vec{x}^\star$ zurück.

```
vec2 Optimizer::gradientDescent(const Function &f, const vec2 &x0, int maxIter) {

}
```

b) Implementieren Sie in der Methode `newton(const Function &f, const vec2 &x0, int maxIter)` das Newton-Verfahren. Brechen Sie das Verfahren vorzeitig ab, falls die Länge des Gradienten $\|\vec{\nabla} f(\vec{x})\|_2 < 10^{-3}$. Geben Sie am Ende die gefundene Minimalstelle $\vec{x}^\star$ zurück.

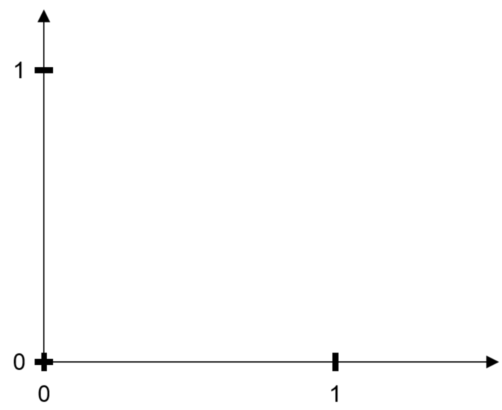
```
vec2 Optimizer::newton(const Function &f, const vec2 &x0, int maxIter) {

}
```

c) Zu welchem Verfahren degeneriert das Newton-Verfahren, falls die Hesse-Matrix gleich der Einheitsmatrix ist?

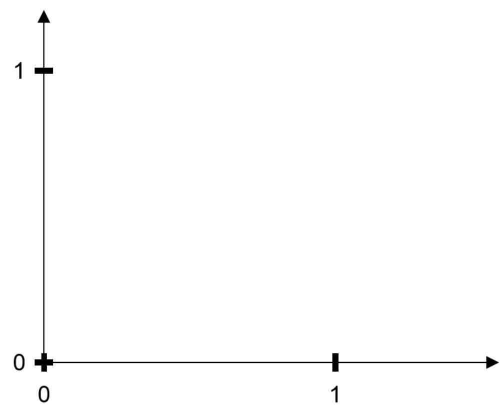
8 Bézier-Kurven (6 Punkte)

$\{B_i^n(u)\},\ i = 0, 1, \ldots, n$ seien die BERNSTEIN-Polynome vom Grad $n \in \mathbb{N}$.

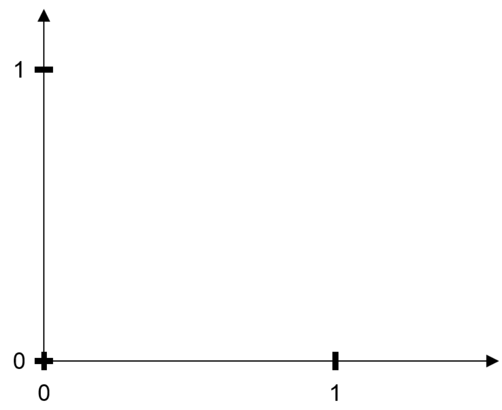
a1) Skizzieren Sie die BERNSTEIN-Polynome vom Grad $n = 1$ in folgende Abbildung.



a2) Skizzieren Sie die BERNSTEIN-Polynome vom Grad $n = 2$ in folgende Abbildung.



a3) Skizzieren Sie die BERNSTEIN-Polynome vom Grad $n = 3$ in folgende Abbildung.



b) Betrachten Sie die folgende kubische BÉZIER-Kurve

$$C : [0,1] \to \mathbb{R}^2 \ , \quad C(u) = \begin{bmatrix} -2 \\ 0 \end{bmatrix} \cdot B_0^3(u) + \begin{bmatrix} 2 \\ 4 \end{bmatrix} \cdot B_1^3(u) + \begin{bmatrix} 6 \\ 4 \end{bmatrix} \cdot B_2^3(u) + \begin{bmatrix} 6 \\ 0 \end{bmatrix} \cdot B_3^3(u)$$

Bestimmen Sie **rechnerisch** mittels **midpoint subdivision** die Kontrollpunkte der beiden Teilkurven $C_{|[0,\frac{1}{2}]}$ (Kontrollpunkte $c_0, \ldots, c_3$) und $C_{|[\frac{1}{2},1]}$ (Kontrollpunkte $d_0, \ldots, d_3$).

9 Faltung (8 Punkte)

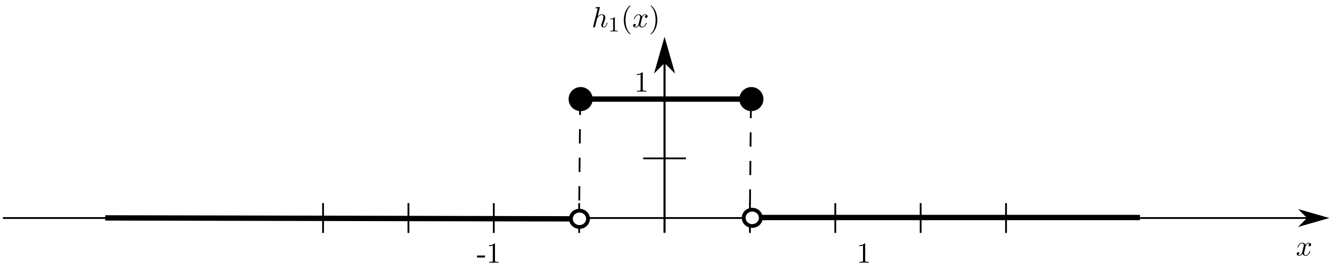
a) Prüfen Sie die folgenden Behauptungen und markieren Sie diese jeweils mit **r** („richtig“) bzw. **f** („falsch“)! Hierbei seien die Funktionen so gewählt, dass die Faltungen existieren.

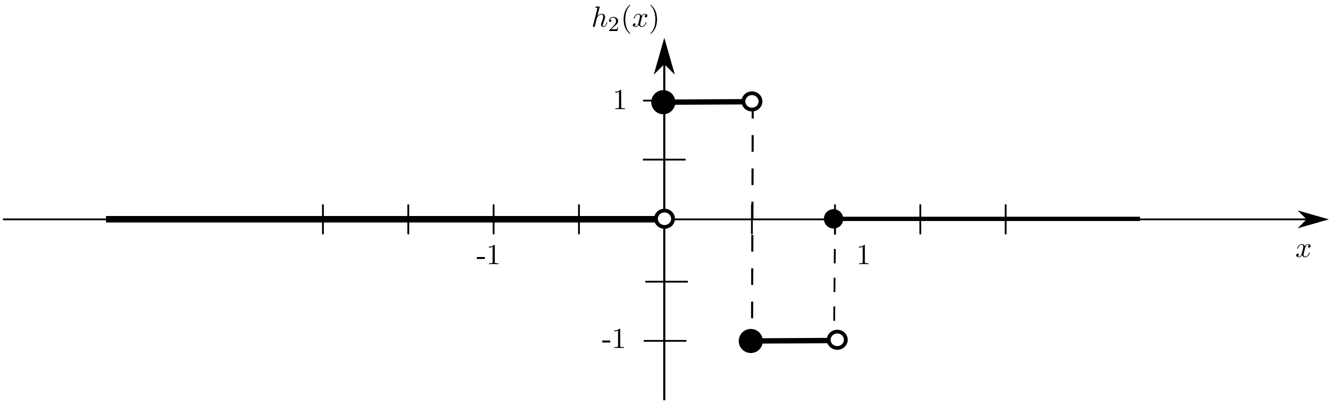
☐	für alle f, g, h gilt:	$f * (g + h) = (f * g) + (f * h)$;
☐	für alle f, g gilt:	$f * g = g * f$;
☐	für alle f gilt:	$f * f = f$;
☐	für alle f, g, h gilt:	$f * (g * h) = (h * g) * f$.

b) Gegeben sind die Funktionen

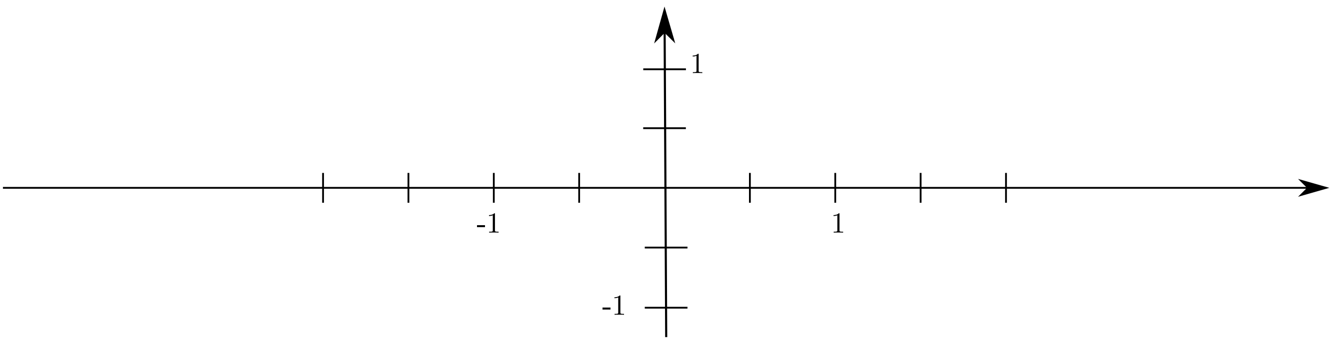
$$h_1(x) := \begin{cases} 1 & \text{falls} \quad |x| \leq 0.5 \\ 0 & \text{sonst} \end{cases} , \qquad h_2(x) := \begin{cases} 1 & \text{falls} \quad x \in [0, \frac{1}{2}) \\ -1 & \text{falls} \quad x \in [\frac{1}{2}, 1) \\ 0 & \text{sonst} \end{cases}$$

sowie die Funktionsgraphen der Funktionen h_1 und h_2:





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



c) Gegeben sind die Funktionen

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

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).

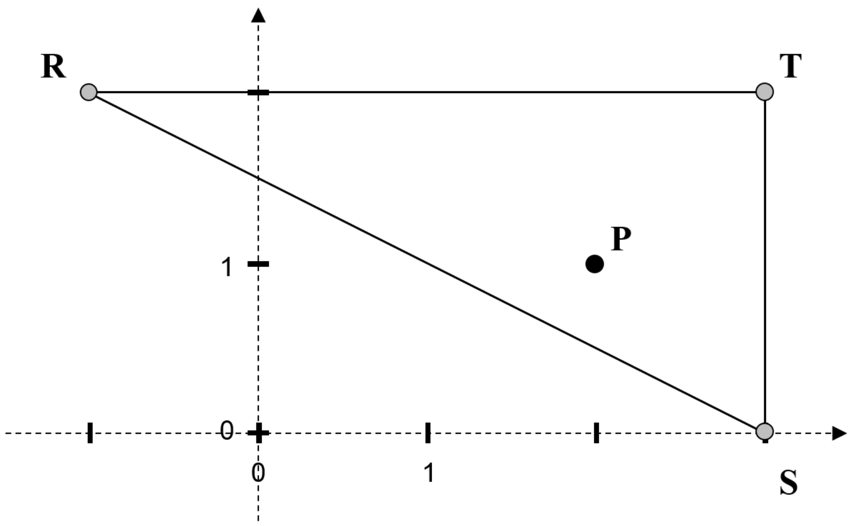
10 Multivariate Interpolation (11 Punkte)

10.1 Baryzentrische Koordinaten

Gegeben sind die Eckpunkte eines Dreiecks $\Delta(\mathbf{R}, \mathbf{S}, \mathbf{T})$:

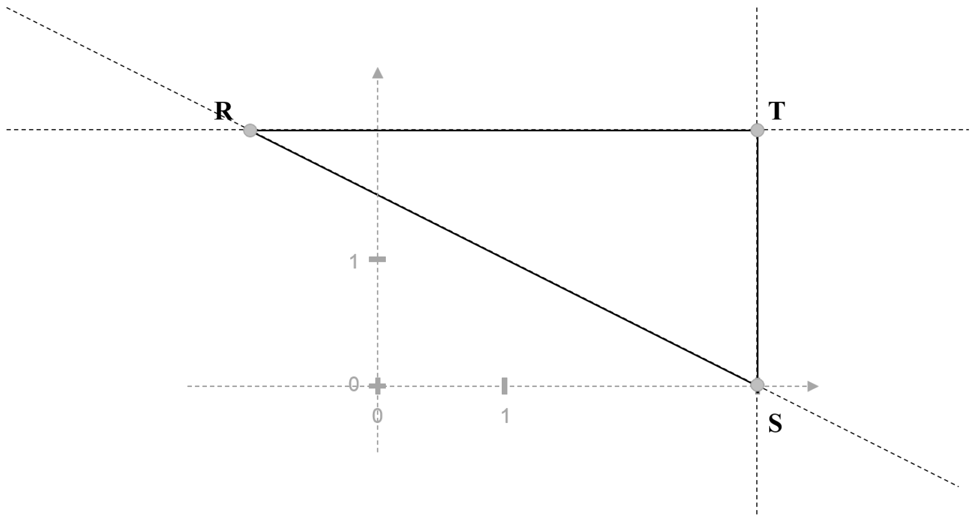
$$\mathbf{R} = [-1, 2]^T \ , \quad \mathbf{S} = [3, 0]^T \ , \quad \mathbf{T} = [3, 2]^T$$

sowie der Punkt $\mathbf{P} = [2, 1]^T$:

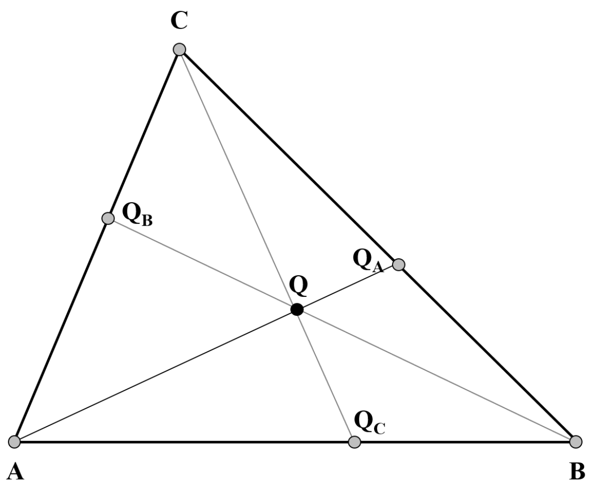


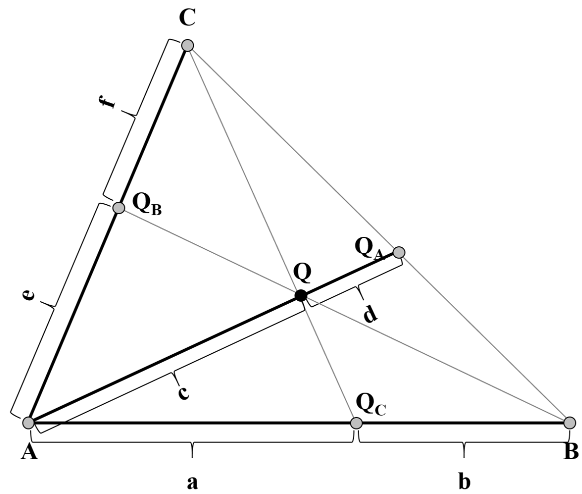
a) Bestimmen Sie die baryzentrischen Koordinaten $[\rho, \sigma, \tau]^T$ des Punktes $\mathbf{P} = [2, 1]$ bzgl. des Dreiecks $\Delta(\mathbf{R}, \mathbf{S}, \mathbf{T})$.

b) Markieren Sie in der nachfolgenden Zeichnung die Gerade $\tau = 1$ sowie die Menge der Punkte, für deren baryzentrische Koordinaten gilt: $\sigma \leq 0$ und $\rho \leq 0$!



c) Der Punkt $\mathbf{Q}$ liegt im Dreieck $\Delta(\mathbf{A}, \mathbf{B}, \mathbf{C})$ und seine baryzentrischen Koordinaten α, β und γ bzgl. $\Delta(\mathbf{A}, \mathbf{B}, \mathbf{C})$ seien bekannt (siehe linke Abb.).





Mit Hilfe der baryzentrischen Koordinaten von $\mathbf{Q}$ sollen die Teilungsverhältnisse der Strecken $\overline{AB}$, $\overline{AQ_A}$ und $\overline{AC}$ (siehe rechte Abb.) bestimmt werden.

Drücken Sie diese Teilungsverhältnisse mit Hilfe der baryzentrischen Koordinaten α, β, γ aus:

$a : b \quad = \quad \boxed{\quad : \quad}$

$c : d \quad = \quad \boxed{\quad : \quad}$

$e : f \quad = \quad \boxed{\quad : \quad}$

10.2 Bilineare Interpolation

Im Rechteck mit den Eckpunkten $A = [-1, -1]$, $B = [5, -1]$, $C = [5, 3]$ und $D = [-1, 3]$ sind die Werte $f_A = 0$, $f_B = 6$, $f_C = 2$ bzw. $f_D = 6$ gegeben. Diese Werte werden ins Innere bilinear interpoliert.

Bestimmen Sie die Werte des Interpolanten im Punkt $P = [1, 2]$ sowie im Mittelpunkt $M = \frac{1}{4}(A+B+C+D)$ des Rechtecks.

Hinweis: Eine Skizze ist unter Umständen hilfreich.