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

11. Februar 2014

Algorithmik kontinuierlicher Systeme — 11. Februar 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
Max. Punktzahl	6	13	7	19	11	11	9	8	6
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 (28 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, 11. Februar 2014 ....

(Unterschrift)

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

ja: ☐ nein: ☐

Erlangen, 11. Februar 2014 ....

(Unterschrift)

1 Theoriefragen und Gleitpunktzahlen (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 $\mathbf{A}\vec{b}$ wobei $\mathbf{A}$ eine **tridiagonale** $(n \times n)$-Matrix und $\vec{b}$ ein n-Vektor ist?	O( )
Welche Komplexität hat die Bestimmung der LR-Zerlegung einer $(n \times n)$-Matrix?	O( )
Welche Komplexität hat der Algorithmus von AITKEN-NEVILLE zur Bestimmung der Koeffizienten des Interpolationspolynoms?	O( )
Welche Komplexität hat die Bestimmung der diskreten FOURIER-Transformation eines n-Vektors bei Verwendung der FFT?	O( )
Welche Konvergenzordnung hat das GAUSS-SEIDEL-Verfahren zum Lösen eines linearen Gleichungssystems mit positiv definiter Koeffizientenmatrix?	
Was ist die Konvergenzordnung des NEWTON-Verfahrens zur Bestimmung einer einfachen Nullstelle?	
Wie groß ist der Approximationsfehler des stückweise linearen Interpolanten im Falle äquidistanter Schrittweite h ?	O( )
Wie groß ist der Approximationsfehler der iterierten SIMPSON-Regel für die Schrittweite h ?	O( )

b) Sind bei der Gleitpunktarithmetik die folgenden Operationen stabil?
Antworten Sie **jeweils** mit „ja“ oder „nein“!

- Addition
- Subtraktion
- Multiplikation
- Division

2 Lösen linearer Gleichungssysteme (13 Punkte)

a) Bestimmen Sie die LR-Zerlegung der folgenden Matrix:

$$\mathbf{B} = \begin{bmatrix} 2 & 1 & 3 \\ 4 & 3 & 4 \\ -4 & 1 & -10 \end{bmatrix}$$

b) Die LR-Zerlegung der Matrix $\mathbf{C} = \begin{bmatrix} 2 & 1 & 1 \\ 2 & 3 & -1 \\ -2 & -5 & 4 \end{bmatrix}$ ist bekannt:

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

Bestimmen Sie **unter Verwendung der LR-Zerlegung** von **C** die Lösung der linearen Gleichung

$$\mathbf{C}\vec{x} = \vec{b} \quad \text{für} \quad \vec{b} = [-1\,,\ 1\,,\ -1\,]^T \ .$$

Betrachten Sie nun das lineare Gleichungssystem $\mathbf{A}\vec{x} = \vec{b}$ mit $\mathbf{A} = \begin{bmatrix} 2 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}$, $\vec{b} = \begin{bmatrix} 16 \\ 0 \\ 0 \\ 16 \end{bmatrix}$

Dieses Gleichungssystem soll iterativ gelöst werden, beginnend mit dem Startvektor $\vec{x}^0 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$

c) Führen Sie **zwei Schritte** des Jacobi-Verfahrens (zum Startwert $\vec{x}^0$) durch.

Erinnerung:

Man betrachte das lineare Gleichungssystem $\mathbf{A}\vec{x} = \vec{b}$ mit $\mathbf{A} = \begin{bmatrix} 2 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}$, $\vec{b} = \begin{bmatrix} 16 \\ 0 \\ 0 \\ 16 \end{bmatrix}$

d) Führen Sie **einen Schritt** des GAUSS-SEIDEL-Verfahrens (zum Startwert $\vec{x}^0 = [0\,,0\,,0\,,0]^T$) durch.

e) Da die Matrix **A** positiv definit ist könnte man das lineare Gleichungssytem auch mit dem cg-Verfahren lösen.
Handelt es sich dabei um ein exaktes Verfahren oder um ein iteratives Verfahren?

3 Speicherung dünn besetzter Matrizen (7 Punkte)

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

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

a) Sie sollen $\mathbf{M}$ im `CCS`-Format abspeichern (Compressed Column Storage). Geben Sie hierzu die interne Datenstruktur an. Die Indizierung beginnt stets bei 0.

b) Die Matrix $\mathbf{M}$ soll nun transponiert werden. Wie können Sie $\mathbf{M}^T$ speichern, damit die Transformation kostengünstig durchgeführt werden kann? Inwiefern muss die interne Datenstruktur für das neue Matrixformat abgeändert werden?

c) Gegeben ist nun eine Matrix mit 4 Spalten im `CRS`-Format (Compressed Row Storage):

Wertearray: `4 3 8 2 -1 4`
Spaltenindexarray: `1 2 3 1 2 3`
Zeilenpointerarray: `0 1 1 3 6 6`

Geben Sie die Matrix in unkomprimierter Form an.

4 Falten und Filtern (19 Punkte)

a) Welche der folgenden 2D-Filter sind separierbar?

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

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

$$\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -3 & 0 & 3\end{bmatrix}$$

b) Ein (Grau-)bild mit $N \times N$ Pixeln soll mit einem <u>nicht</u> separierbaren Filter der Größe 5×5 gefiltert werden. Dabei werden nur die inneren $(N-4) \times (N-4)$ Pixel berechnet.

Bestimmen Sie den Rechaufwand (Anzahl der arithmetischen Operationen).

c) Nun wird das Bild von Teil b) mit einem separierbaren Filter der Größe 5×5 gefiltert. Bestimmen Sie wiederum den Rechenaufwand.

Gegeben ist eine Klasse `Image` in C++, welche die Werte eines quadratischen (Grau-)bildes als Werte vom Typ `double` speichert. Das Bild ist im Bereich `[0,dim-1]` × `[0,dim-1]` adressierbar.

d) Implementieren Sie die Methode `getValue`, die den Wert des Pixels an der Stelle `(x,y)` zurück gibt.

e) Implementieren Sie analog dazu die Methode `setValue`, die den Wert des Pixels `(x,y)` setzt.

```
class Image {

public:

        /* Konstruktoren, Destruktor und Methoden zum Laden
                müssen nicht berücksichtigt werden
        */

        unsigned int getDim() const {
                return dim;
        }

        //! gibt den Wert des Bildes an der Pixelposition (x,y) zurück
        double getValue(const unsigned int& x, const unsigned int& y) const {







        }


        //! setzt den Wert des Pixels (x,y) auf den Wert value
        void setValue(const unsigned int& x, const unsigned int& y, double value) {







        }

protected:

        //! Wertespeicher des Bildes
        std::vector<double> data;

        //! Breite/Höhe des quadratischen Bildes
        unsigned int dim;
};
```

f) Implementieren Sie die Methode `filter`, die ein Bild `src` bekommt, mit der Maske **F** filtert und das Ergebnis in das Bild `dest` speichert. Filtern Sie das Bild im Bereich `[1,dim-2]` $\times$ `[1,dim-2]`.

$$\mathbf{F} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Hinweis: Das Bild `dest` *hat die Dimension* `[0,dim-3]` $\times$ `[0,dim-3]`.

```
void filter(const Image& src, Image& dest) {

}
```

g) Nun sei ein separierbarer Filter $\mathbf{F}_2$ gegeben. Implementieren sie die Methode `filter` erneut unter Ausnutzung der Separierbarkeit.

$$\mathbf{F}_2 = \begin{bmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} * \begin{bmatrix} 3 & 10 & 3 \end{bmatrix}$$

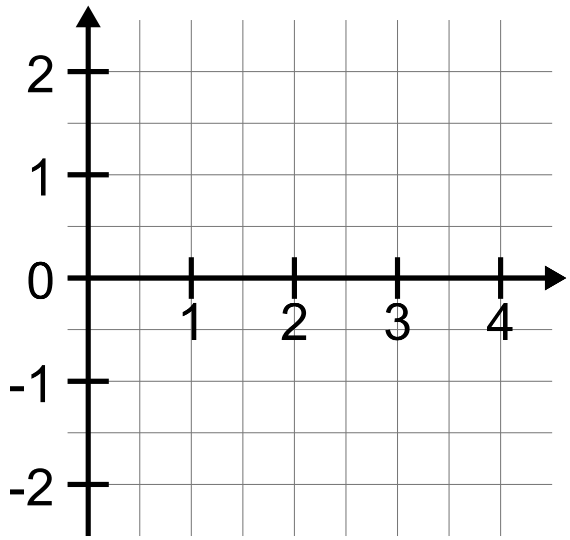
```
void filter(const Image& src, Image& dest) {








}
```

5 Interpolation (11 Punkte)

Gegeben seien folgende Punkte:

i	0	1	2	3
x_i	0	1	2	4
y_i	1	-1	1	2

a) **Zeichnen** Sie die Funktion $n(x) : [0, 4] \mapsto \mathbb{R}$, welche obige Werte, gemäß Nearest Neighbor Interpolation, stückweise konstant interpoliert.



(a) nearest neighbor

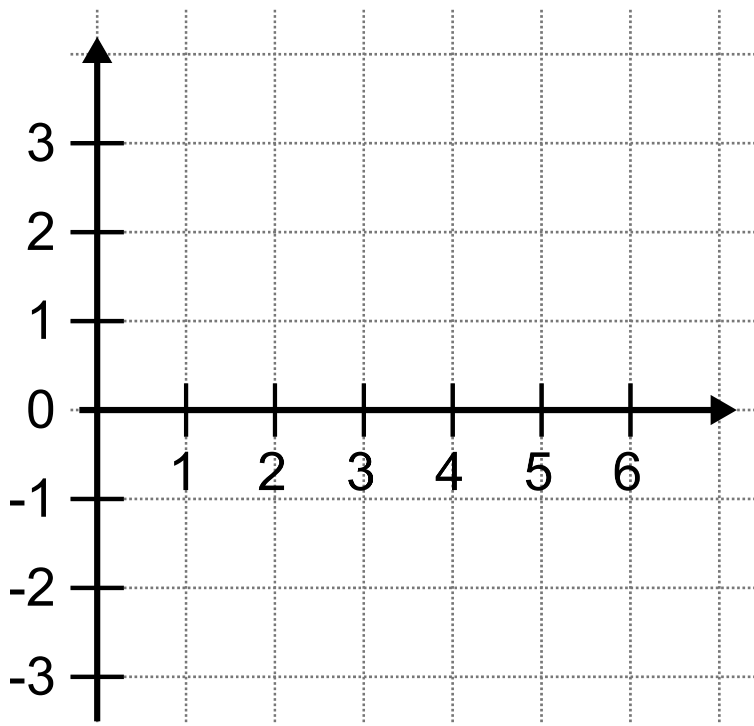
b) **Berechnen** Sie die Funktion $l(x) : [0, 4] \mapsto \mathbb{R}$, welche obige Werte stückweise linear interpoliert.

c) In dieser und der nachfolgenden Teilaufgabe sind andere Interpolationsdaten gegeben und zwar

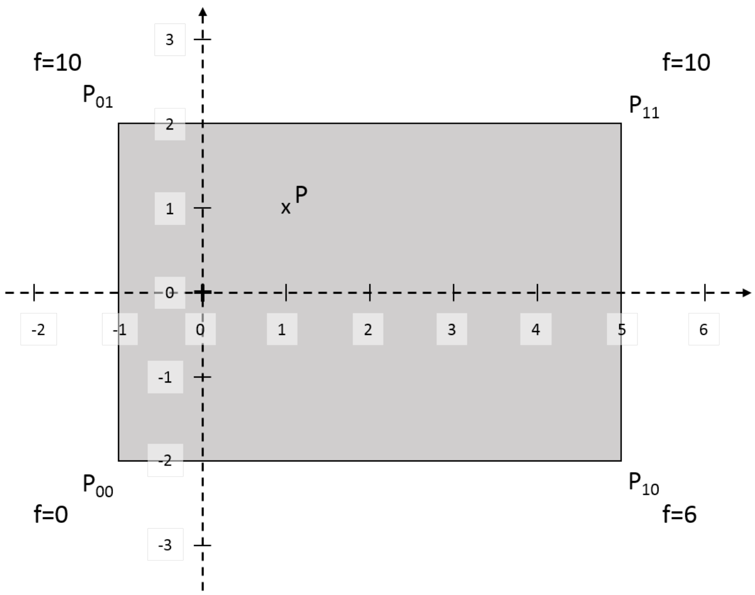
i	0	1	2	3
x_i	0	2	4	6
y_i	-1	-3	-1	3

Geben Sie die Ableitungen m_1 und m_2 des Catmull-Rom-Interpolanten an den Stellen x_1 und x_2 an.

d) Skizzieren Sie im Interval $[x_1, x_2]$ den Catmull-Rom-Interpolanten.

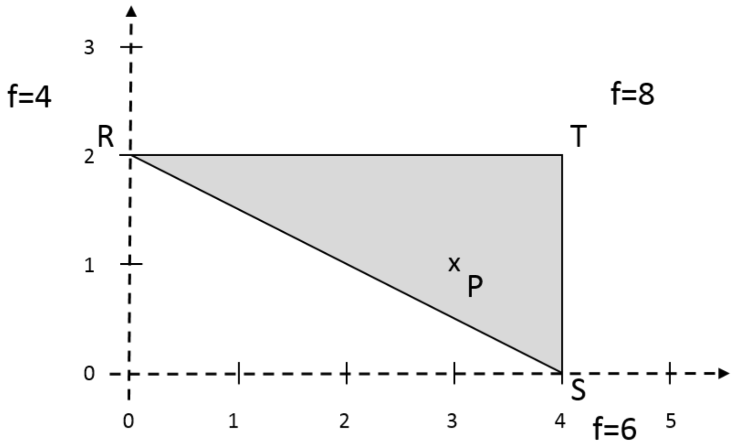


e) In den Ecken eines Rechtecks $P_{00} = [-1, -2]^T$, $P_{10} = [5, -2]^T$, $P_{11} = [5, 2]^T$, $P_{01} = [-1, 2]^T$ sind vier Werte $f_{00} = 0$, $f_{10} = 6$, $f_{11} = 10$ und $f_{01} = 10$ gegeben, diese werden ins Innere bilinear interpoliert.



Bestimmen Sie den Wert des Interpolanten im Punkt $P = [1, 1]^T$.

f) In den Ecken des Dreiecks $\Delta(R, S, T)$ mit $R = [0, 2]^T$, $S = [4, 0]^T$, $T = [4, 2]^T$ sind die drei Werte gegeben: $f_R = 4$ $f_S = 6$, $f_T = 8$. Diese werden ins Innere linear interpoliert.

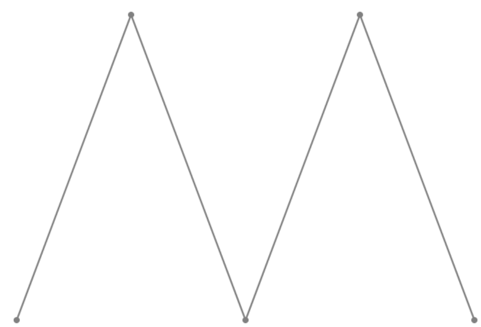


Bestimmen Sie den Wert des Interpolanten im Punkt $P = [3, 1]^T$.

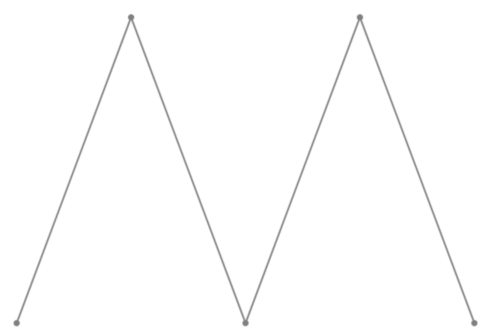
Hinweis: Bestimmen Sie zunächst (rechnerisch oder geometrisch) die baryzentrischen Koordinaten.

6 Bézier Kurven (11 Punkte)

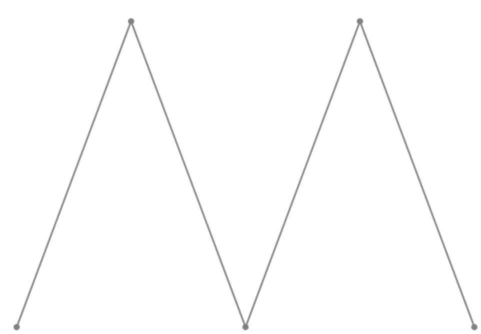
a) Zeichnen Sie die BÉZIER-Kurve zu folgendem Kontrollpolygon:



b) Zeichnen Sie nun eine Kurve, die die `Tangenten-Eigenschaft` nicht erfüllt. Alle übrigen Formeigenschaften müssen erhalten bleiben.



c) Verletzten Sie nun <u>nur</u> die Eigenschaft der `Variationsreduktion`.



d) Implementieren Sie die Methode `bezierMidpointSubdivision` in C++, die zu einem gegebenen Satz dreidimensionaler Kontrollpunkte eine Midpoint-Subdivision durchführt. Die Kontrollpunkte der entstandenen Teilkurven werden jeweils nach `left` und `right` geschrieben. Der Mittelpunkt soll in beiden Teilkurven vorhanden sein.

Hinweis: Sie können davon ausgehen, dass für Objekte der Klasse `vec3` alle Standardoperatoren (+, -, *, /, =, ==) zur Verfügung stehen.

```
void bezierMidpointSubdivision(const std::vector<vec3>& kp,
                               std::vector<vec3>& left,
                               std::vector<vec3>& right)
{



}
```

7 Singulärwertzerlegung (9 Punkte)

Gegeben sei die Singulärwertzerlegung (SVD) der Matrix $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$:

$$\underbrace{\frac{1}{6}\begin{bmatrix} 14 & -2 & 6 & 10 \\ 11 & 19 & -3 & -5 \\ 13 & 11 & 21 & -19 \end{bmatrix}}_{\mathbf{A}} = \underbrace{\frac{1}{9}\begin{bmatrix} 1 & 4 & 8 \\ 4 & 7 & -4 \\ 8 & -4 & 1 \end{bmatrix}}_{\mathbf{U}} \cdot \underbrace{\begin{bmatrix} 6 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \end{bmatrix}}_{\mathbf{S}} \cdot \underbrace{\frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{bmatrix}}_{\mathbf{V}^T}.$$

a) Bestimmen Sie für die Matrix $\mathbf{A}$:

- die Singulärwerte:

- den Rang:

- den Bildraum (welche Vektoren spannen den Bildraum auf?):

- den Kern (welche Vektoren spannen den Kern auf?):

- und die Konditionszahl bzgl. der Euklidischen Norm $|||\cdot|||_2$:

Zur Erinnerung:

$$\underbrace{\frac{1}{6}\begin{bmatrix} 14 & -2 & 6 & 10 \\ 11 & 19 & -3 & -5 \\ 13 & 11 & 21 & -19 \end{bmatrix}}_{\mathbf{A}} = \underbrace{\frac{1}{9}\begin{bmatrix} 1 & 4 & 8 \\ 4 & 7 & -4 \\ 8 & -4 & 1 \end{bmatrix}}_{\mathbf{U}} \cdot \underbrace{\begin{bmatrix} 6 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \end{bmatrix}}_{\mathbf{S}} \cdot \underbrace{\frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{bmatrix}}_{\mathbf{V}^T}.$$

b) Lösen Sie das folgende (unterbestimmte) Gleichungssystem mit Hilfe der Pseudo-Inversen.

$$\mathbf{A} \cdot \vec{x} = \begin{bmatrix} 36 \\ -36 \\ 0 \end{bmatrix}.$$

<u>Hinweis</u>: Die (explizite) Berechnung der Pseudo-Inversen ist dazu nicht erforderlich!

Zur Erinnerung:

$$\underbrace{\frac{1}{6}\begin{bmatrix} 14 & -2 & 6 & 10 \\ 11 & 19 & -3 & -5 \\ 13 & 11 & 21 & -19 \end{bmatrix}}_{\mathbf{A}} = \underbrace{\frac{1}{9}\begin{bmatrix} 1 & 4 & 8 \\ 4 & 7 & -4 \\ 8 & -4 & 1 \end{bmatrix}}_{\mathbf{U}} \cdot \underbrace{\begin{bmatrix} 6 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \end{bmatrix}}_{\mathbf{S}} \cdot \underbrace{\frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{bmatrix}}_{\mathbf{V}^T}.$$

c) Bestimmen Sie die Matrix vom Rang 1, die $\mathbf{A}$ im Sinne der FROBENIUS-Norm am besten approximiert.

8 Nichtlineare Optimierung (8 Punkte)

Die folgende Kostenfunktion (Zielfunktional) soll minimiert werden.

$$F(x, y) = \frac{x^4}{2} + y^2 - 2\,xy + 3\,x - 4\,y + 2$$

a) Führen Sie einen Schritt des **Gradienten-Verfahrens** durch. Wählen Sie als Startwert $(x_0, y_0) = (1, 3)$ und als Schrittweite $t = \frac{1}{8}$.

b) Nennen Sie drei mögliche **Abbruchkriterien** für Iterationsverfahren.

Erinnerung: Die folgende Kostenfunktion (Zielfunktional) soll minimiert werden.

$$F(x,y) = \frac{x^4}{2} + y^2 - 2\,xy + 3\,x - 4\,y + 2$$

c) Führen Sie einen Schritt des NEWTON-**Verfahrens** durch. Wählen Sie als Startwert $(x_0, y_0) = (1, 3)$.

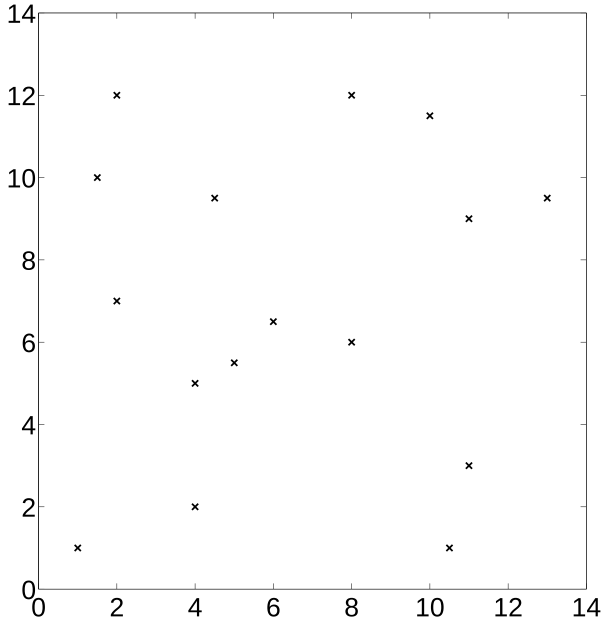
Hinweis: Die Inverse einer 2-Matrix $\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ *ist gegeben durch* $\mathbf{A}^{-1} = \frac{1}{ad-bc} \cdot \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$

d) Gegeben ist die Matrix $\mathbf{A} = \begin{bmatrix} -1 & 1 \\ 1 & 3 \end{bmatrix}$.

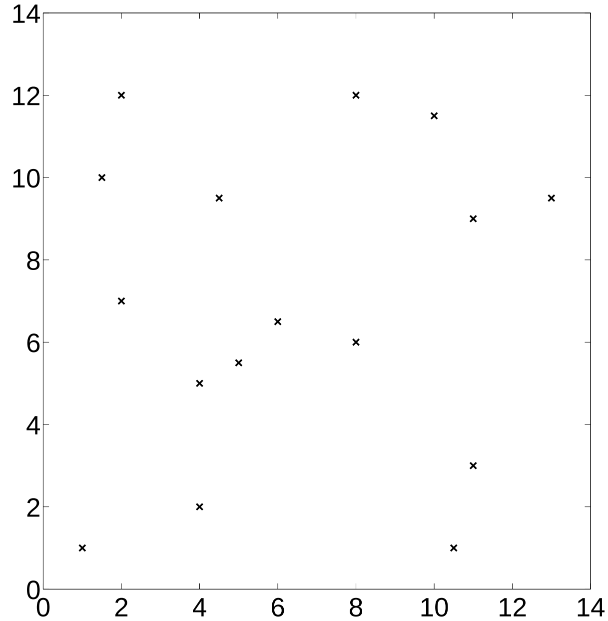
Bestimmen Sie zum Vektor $\mathbf{b} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ eine **A**-konjugierte Richtung.

9 Median Cut (6 Punkte)

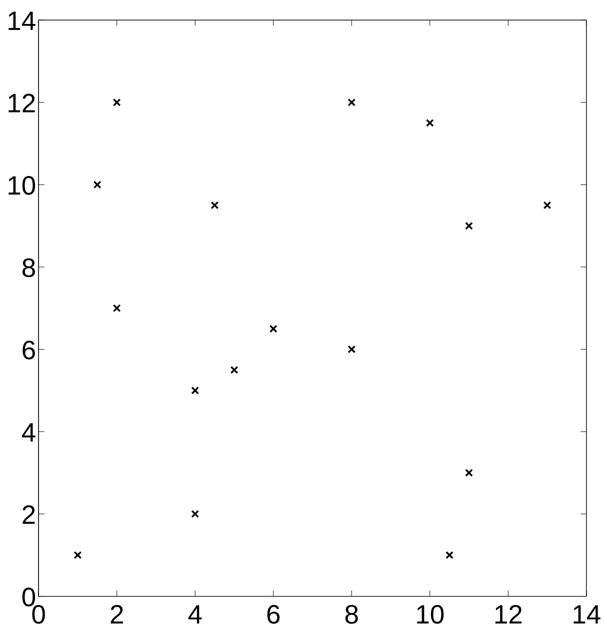
Gegeben ist folgende Punktewolke mit 16 Punkten:



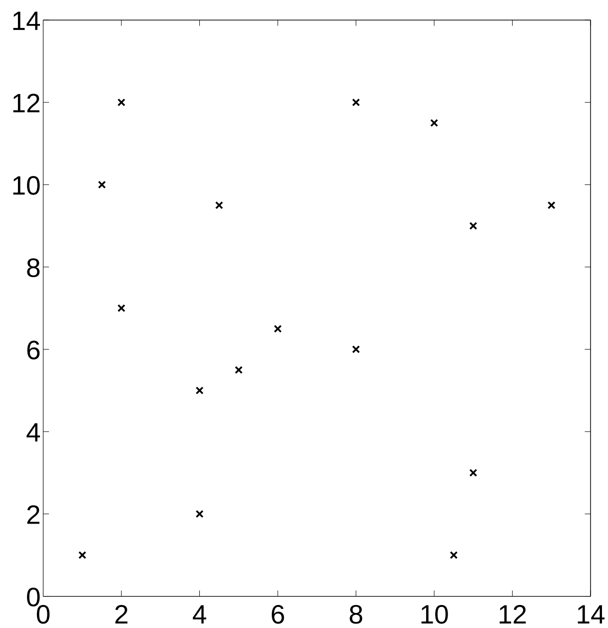
Gegebene Punktewolke



Schritt 1



Schritt 2



Schritt 3

Führen Sie 3 Schritte des **Median-Cut** Verfahrens durch!
<u>Hinweis</u>: Benutzen Sie dazu die obigen Vorlagen!