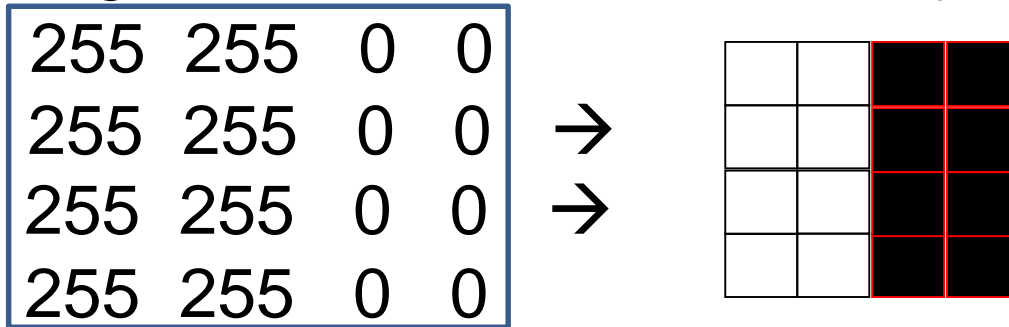
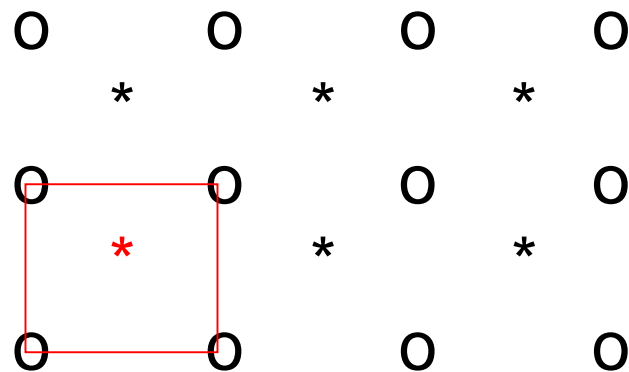


Bild als Matrix von Werten zwischen 0 und 255.

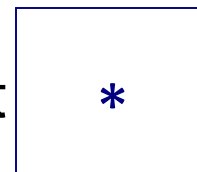
Jeder Wert gibt Grauwert an einer Stelle (einem Pixel) an.



(a) Umskalierung, Drehung, Verschiebung führt auf neues Gitter!
 Wie werden Pixelwerte des neuen Gitters bestimmt?



Bestimmung des neuen Wertes *
 z.B. durch bilineare Interpolation mit



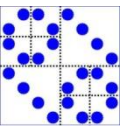
(b) Registrierung von Bilddaten:

Man verfügt über zwei verschiedene Aufnahmen desselben Objekts, z.B. zu verschiedenen Zeiten, von verschiedenen Aufnahmemedien, oder von verschiedenen Blickwinkeln aus aufgenommen. Z.B. Röntgen vs. Ultraschall.

Ziel ist es, diese Bilder (x, y) und (\tilde{x}, \tilde{y}) zur Deckung zu bringen. Z.B. eine neue Aufnahme mittels einer geeigneten Transformation (Verschiebung, Drehung) in das Referenzobjekt abzubilden.

Bestimme dazu a , b , und φ , so dass

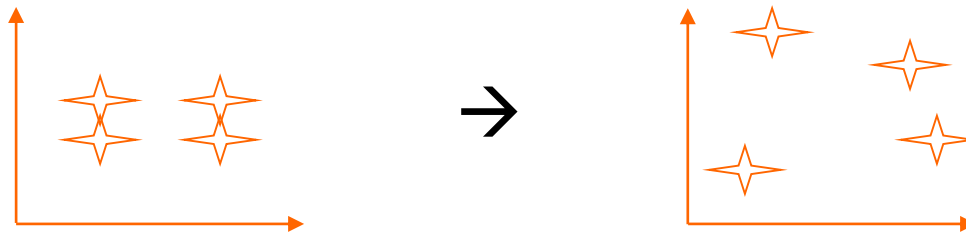
$$\min_{\varphi, a, b} \left\| \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} a \\ b \end{pmatrix} - \begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} \right\|$$



Wieder zwei verschiedene (aber deckungsgleiche) Aufnahmen desselben Objekts.

Nun ist aber eine der Aufnahmen verzerrt, z.B. im Falle von Röntgenaufnahmen durch Erdmagnetfeld, usw.

Wir nehmen an, dass wir Markierungspunkte in beiden Aufnahmen haben, die zur Übereinstimmung gebracht werden sollen, diesmal aber mit einer nichtlinearen Transformation.



Transformiere die Koordinaten der verzerrten Markierungspunkte möglichst gut auf die originalen Positionen,

z.B. durch zwei Polynome dritten Grades und Minimierung des Fehlers über alle Punkte:

(x_d, y_d) verzerrte Position von (x_p, y_p) :

Minimiere über alle Markierungspunkte den Abstand

$$\begin{pmatrix} x_p \\ y_p \end{pmatrix} - \begin{pmatrix} a_0 + a_1 x_d + a_2 y_d + a_3 x_d y_d + a_4 x_d^2 + a_5 y_d^2 + a_6 y_d x_d^2 + a_7 x_d y_d^2 + a_8 x_d^3 + a_9 y_d^3 \\ b_0 + b_1 x_d + b_2 y_d + b_3 x_d y_d + b_4 x_d^2 + b_5 y_d^2 + b_6 y_d x_d^2 + b_7 x_d y_d^2 + b_8 x_d^3 + b_9 y_d^3 \end{pmatrix}$$

Bikubischer Ansatz.

Betrachte z.B. $x_p = a_0 + a_1 x_{d,k} + \dots + a_9 y_{d,k}$ über alle Markierungspunkte:

ergibt Normalgleichungen zur Bestimmung der optimalen Koeffizienten a_0, \dots, a_9 .

Bestimme b_0, \dots, b_9 genauso.

10 Koeffizienten, also mindestens 10 Marker!

Integration \leftrightarrow *Flächenberechnung*

Problem:

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

In einigen Fällen kann das Integral mittels Stammfunktion berechnet werden:

Stammfunktion von x^k ist $x^{k+1}/(k+1)$;

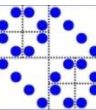
Daher
$$\int_a^b x^k \, dx = \left[\frac{x^{k+1}}{k+1} \right]_a^b = \frac{b^{k+1} - a^{k+1}}{k+1}$$

Stammfunktion von $\exp(x)$ ist $\exp(x)$, usw. (Formelsammlung)

In vielen Fällen existiert keine explizite Formel für die Stammfunktion!

Dann muss das Integral numerisch angenähert werden (vgl.

Riemann'sche Summen, Obersumme, Untersumme)

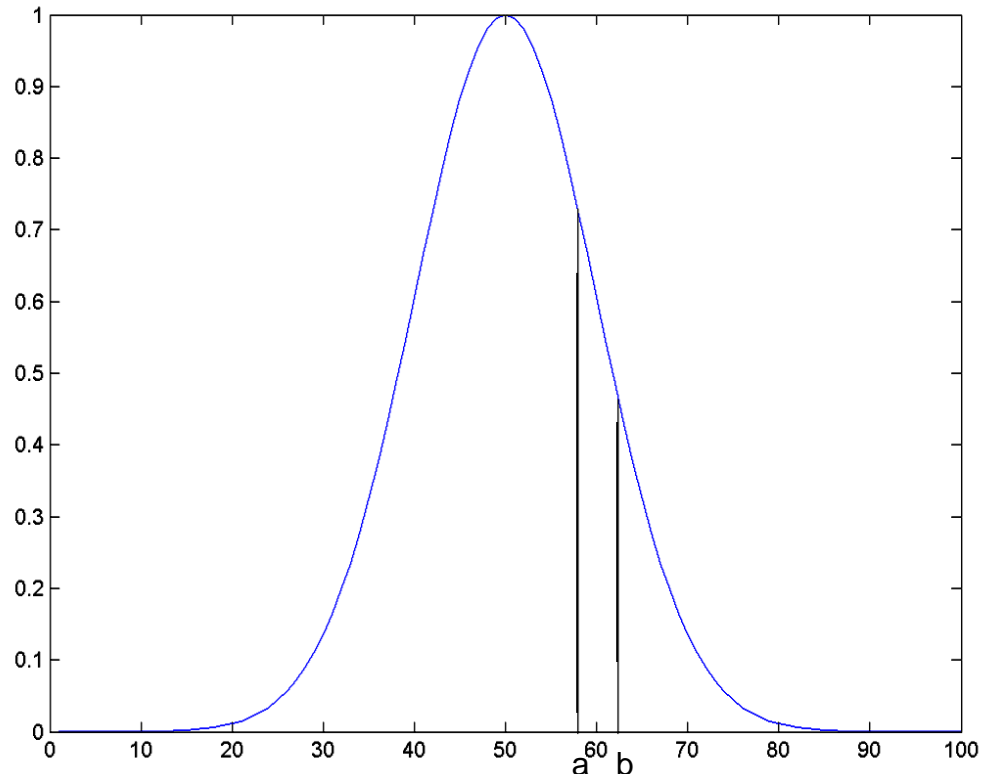


4.2.1. Beispiel aus der Wahrscheinlichkeitsrechnung:

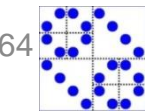
Gauss-Verteilung mit Mittelwert m und Standardabweichung σ ; gesucht ist die Wahrscheinlichkeit dafür, dass ein gemessener Wert zwischen a und b liegt:

$$\frac{1}{\sigma \sqrt{2\pi}} \cdot \int_a^b \exp\left(-\frac{(x-m)^2}{2\sigma^2}\right) dx$$

Entspricht der Fläche unter der Kurve:



$m=50$
 $\sigma=10$



4.2.2. Allgemeine Form numerischer Quadraturregeln zur näherungsweise Berechnung solcher Integrale:

$$I(f) = \int_a^b f(x) \, dx \approx \sum_{i=0}^n w_i f(x_i)$$

- Bestandteile:
- Stützstellen x_i mit Werten $f(x_i)$
 - Gewichte w_i

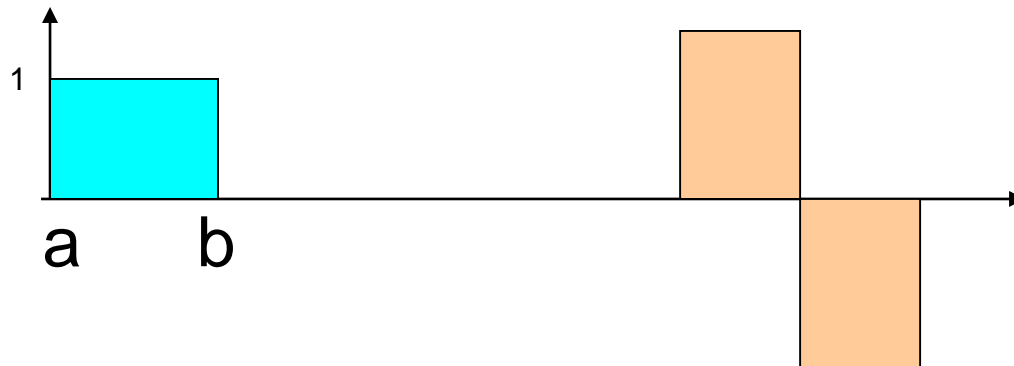
Frage: Wie sind die Stützstellen und Gewichte zu wählen, damit der Näherungswert möglichst gut ist.

Die Gewichte sollten dabei größer oder gleich 0 sein, um Rundungsfehler durch Auslöschung in der obigen Summe zu vermeiden!

Möglichst wenige Funktionsauswertungen für möglichst genaue Näherung!

4.2.3. Quadratur als Flächenberechnung

Das Integral zwischen a und b gibt den Flächeninhalt der Kurve an, die zwischen $f(x)$ und der x -Achse eingeschlossen ist.



Die blaue, linke Fläche entspricht $(b-a) \cdot 1$

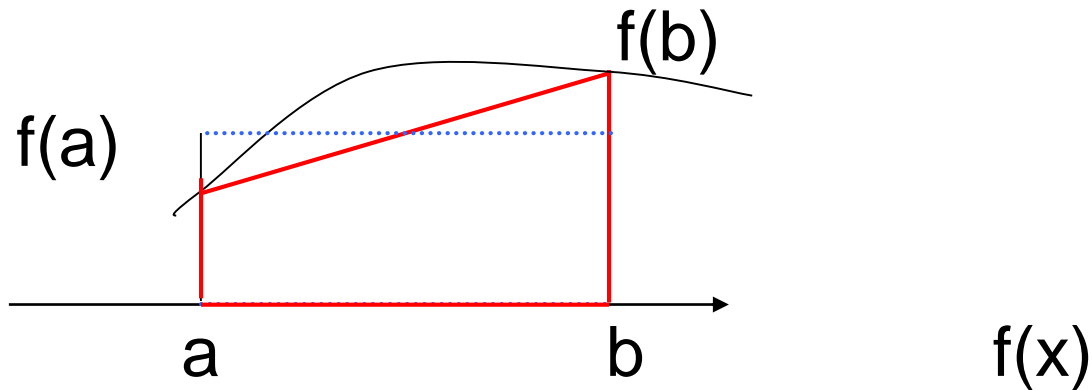
Und die rosa, rechte Fläche entspricht dem Wert 0, da obere und untere Fläche sich weg heben.

Trapezregel (n=1):

$$I(f) = \int_a^b f(x)dx \approx \frac{f(a) + f(b)}{2} \cdot (b - a) =$$

$$= \frac{b-a}{2} f(a) + \frac{b-a}{2} f(b)$$

Stützstellen a,b; Gewichte $w_0 = w_1 = (b-a)/2$

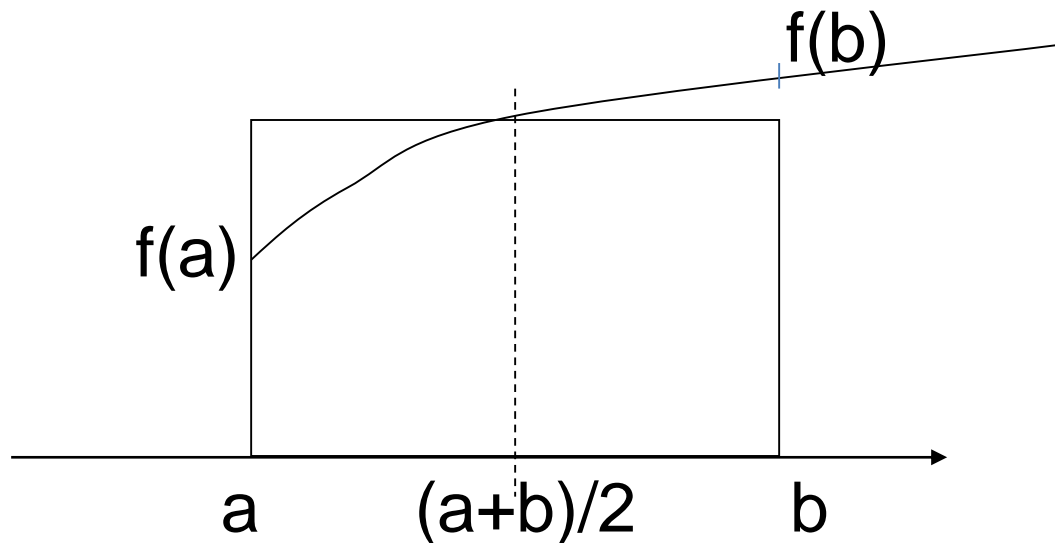


Mittelpunktregel ($n=0$):

Rechteck, begrenzt durch die Punkte $(a,0)$, $(b,0)$ und $((a+b)/2, f((a+b)/2))$

$$I(f) \approx (b-a) \cdot f\left(\frac{a+b}{2}\right)$$

Also mit $n=0$ nur eine Stützstelle $(a+b)/2$ und ein Gewicht $w_0 = (b-a)$.



4.2.4. Regeln aus der Interpolation

Nähere $f(x)$ durch einfach zu integrierende Funktion an, z.B. durch interpolierendes Polynom.

Der Einfachheit halber wählen wir äquidistante Stützstellen

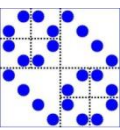
$$x_i = a + ih, \quad i = 0, 1, \dots, n, \quad h = (b - a) / n$$

Zu bestimmen sind noch passende Gewichte w_i .

Sie ergeben sich aus der Integration des $f(x)$ interpolierenden Polynoms in der Lagrangedarstellung:

$$p(x) = \sum_{i=0}^n f(x_i) L_i(x)$$

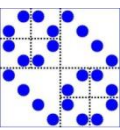
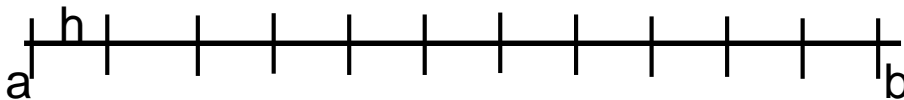
$$\text{also } \int_a^b f(x) dx \approx \int_a^b p(x) dx = \sum_{i=0}^n f(x_i) \int_a^b L_i(x) dx = \sum_{i=0}^n f(x_i) w_i$$



Dadurch lassen sich allgemein Integrationsregeln herleiten.

Für $n > 6$ ergeben sich auch negative Gewichte!
Numerisch problematisch (Auslöschung).

Daher ist dieses Vorgehen nur für kleine n sinnvoll.
Bedenke auch die beobachteten Oszillationen bei
Polynominterpolation hohen Grades!



1. Fall: $n=1$, nur $x_0=a$, $x_1=b$, $h=b-a$:

Das linear interpolierende Polynom ist

$$p(x) = f(a) + \frac{f(b) - f(a)}{b - a}(x - a)$$

Daher ergibt sich als Näherung:

$$\begin{aligned} \int_a^b f(x) dx &\approx \int_a^b p(x) dx = \int_a^b \left(f(a) + \frac{f(b) - f(a)}{b - a}(x - a) \right) dx = \\ &= f(a)(b - a) + \frac{f(b) - f(a)}{b - a} \left(\frac{b^2 - a^2}{2} - a(b - a) \right) = \\ &= h \left(\frac{1}{2} f(a) + \frac{1}{2} f(b) \right) = \frac{b - a}{2} (f(a) + f(b)) \end{aligned}$$

Also Gewichte $w_0 = w_1 = h/2 = (b-a)/2$.
Dies ist genau die Trapezregel.

Fehlerabschätzung nach Kap. 4.1.6:

$$\left| \int_a^b (f(x) - p(x)) dx \right| = \left| \int_a^b \left(\frac{f^{(2)}(\xi)}{2} (x-b)(x-a) \right) dx \right| \leq$$

$$\leq \frac{M_2}{2} \int_a^b (x-a)(b-x) dx = \frac{M_2}{2} \cdot \frac{(b-a)^3}{6} = \frac{M_2 (b-a)^3}{12} = \frac{M_2}{12} h^3$$

mit $M_k := \max_{x \in [a,b]} |f^{(k)}(x)|$

Daher ist die Trapezregel exakt für Polynome vom Grad 1

(da dann $M_2 = 0$),

d.h. Näherungswert und exakter Wert sind gleich falls $f(x)$ eine Polynom vom Grad ≤ 1 ist.

$n=2$, Simpson-Regel: $f \sim$ interpol. Parabel

$$x_0 = a, \quad x_1 = \frac{a+b}{2}, \quad x_2 = b, \quad h = \frac{b-a}{2}$$

$$w_0 = w_2 = \frac{h}{3} = \frac{b-a}{6}, \quad w_1 = \frac{4}{3}h = \frac{4(b-a)}{6}$$

$$I(f) \approx \frac{h}{3} \cdot \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right)$$

Fehlerabschätzung (ohne Beweis):

$$\left| \int_a^b (f(x) - p(x)) dx \right| \leq \frac{M_4 (b-a)^5}{2880} = \frac{M_4}{90} h^5$$

Daher ist die Simpsonregel sogar exakt für Polynome dritten Grades ($M_4 = 0$).

Allgemein:

$$\int_a^b f(x) dx \approx \int_a^b p_{0,1,\dots,n}(x) dx = h \sum_{i=0}^n \alpha_i f(a + ih)$$

Bis jetzt waren die Stützstellen vorgegeben (äquidistant), und nur die Gewichte wurden optimal gewählt.

- Vergeudete Freiheitsgrade!

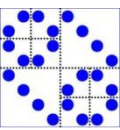
- Und aus Interpolation wissen wir aber, dass am Rand mehr Stützstellen sein sollten! Insbesondere bei der Integration!

Daher neue Problemstellung:

Wie sind Gewichte und Stützstellen zu wählen, dass sich eine möglichst ‚gute‘ Quadratur-Regel ergibt.

‚Gut‘ heisst: die Regel soll Polynome möglichst hohen Grades exakt integrieren! Polynome als Urfunktionen!

(Trapezregel: 2 Stützstellen, Grad 1 exakt,
Simpsonregel: 3 Stützstellen, Grad 3 exakt)



Finde $x_i \in [a, b]$, w_i , $i=0, \dots, n$, so dass für $j=0, \dots, m$:

$$\frac{b^{j+1} - a^{j+1}}{j+1} = \int_a^b x^j dx = I(x^j) \stackrel{!}{=} \sum_{i=0}^n w_i x_i^j$$

Dies sind $m+1$ nichtlineare Gleichungen für die $2n+2$ Unbekannten x_i und w_i .

Fallstudie n=1:

$$\frac{b^{j+1} - a^{j+1}}{j+1} = \int_a^b x^j dx = w_0 x_0^j + w_1 x_1^j$$

$j = 0 :$	$b - a$	$=$	$w_0 + w_1$
$j = 1 :$	$\frac{b^2 - a^2}{2}$	$=$	$w_0 x_0 + w_1 x_1$
$j = 2 :$	$\frac{b^3 - a^3}{3}$	$=$	$w_0 x_0^2 + w_1 x_1^2$
$j = 3 :$	$\frac{b^4 - a^4}{4}$	$=$	$w_0 x_0^3 + w_1 x_1^3$
$(j = 4 :)$	$\frac{b^5 - a^5}{5}$	$=$	$w_0 x_0^4 + w_1 x_1^4$

Aus den ersten vier Gleichungen ergibt sich
(Lösung per Hand)

$$x_{0,1} = \frac{a+b}{2} \pm \frac{b-a}{2\sqrt{3}}, \quad w_{0,1} = \frac{b-a}{2}$$


Gleichung $j=4$ ist mit diesen Werten nicht erfüllt!

Also werden durch diese Parameter Polynome bis zum Grad 3 exakt integriert.

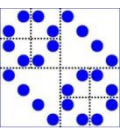
Im Spezialfall: $\int_{-1}^1 f(x) \frac{dx}{\sqrt{1-x^2}}$ ergeben sich als optimale Stützstellen, die Polynome möglichst hohen Grades exakt integrieren, wieder die Nullstellen der Tchebycheff-Polynome

$$\cos\left(\frac{(2j+1)\pi}{2n+2}\right), j=0,1,\dots,n$$

Allgemein lässt sich zeigen, dass die optimalen Stützstellen gerade die Nullstellen von Orthogonalpolynomen sind, z.B. Legendre-, Hermite-, Laguerre-Polynome. (siehe Formelsammlung)



$$\langle p_i, p_j \rangle = \int_a^b p_i(x) p_j(x) g(x) dx = 0 \quad \text{für } i \neq j$$



Vorteil der Gauss-Quadratur mit optimalen Stützstellen :
Für festes n ergibt sich optimales Ergebnis!

Nachteil: Falls Resultat nicht genau genug, will man mehr Stützstellen verwenden; die neuen Stützstellen sind dann aber Nullstellen eines anderen Polynoms, also muss man an allen Stellen $f(x_i)$ neu berechnen.

Ausweg: Behalte die n alten Stützstellen. Suche nun n neue Stützstellen und $2n$ Gewichte zu den $2n$ Stützstellen, so dass Polynome möglichst hohen Grades exakt integriert werden.

4.2.6. Monte-Carlo-Methoden:

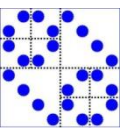
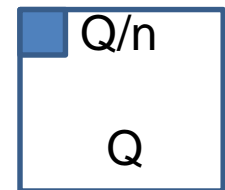
Wähle n Zufallspunkte aus dem Gebiet $Q \in \mathbb{R}^n$,
 berechne Funktionswerte an diesen Stellen und setze

$$I = \int_Q f(\vec{x}) d\vec{x} \approx \sum_{i=1}^n f(x_i) \frac{|Q|}{n}$$

Besonders nützlich bei hochdimensionalen Integralen!

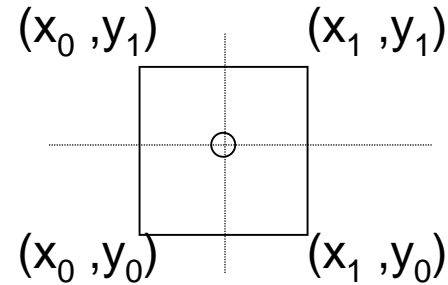
Stützstellen: Zufallsstellen $f(x_i)$

Gewichte: immer $|Q| / n$ an jeder Stützstelle,
 da $|Q|$ auf n Terme verteilt wird.



4.2.7. Zweidimensionale Integration

Mittelpunktsregel:



$$\int_Q f(x, y) dx dy \approx (y_1 - y_0) \int_{x_0}^{x_1} f\left(x, \frac{y_0 + y_1}{2}\right) dx \approx$$

$$\approx (x_1 - x_0)(y_1 - y_0) f\left(\frac{x_1 + x_0}{2}, \frac{y_1 + y_0}{2}\right)$$

Mittelpunktsregel erst für jedes x in y , dann für fixiertes y in x .

Allgemein Quadratur-Regeln aus der 2D-Interpolation:

Bestimme zu $f(x,y)$ ein interpolierendes Polynom $p(x,y)$, so dass Stützstellen und Grad zusammenpassen

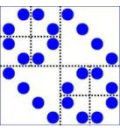
(vgl. Kap. 4.1.10).

Gesucht: Integral über Quadrat Q mit Ecken (x_0, y_0) , (x_0, y_1) ,
 (x_1, y_0) , (x_1, y_1)

$$\begin{aligned} \int_Q f(x, y) dx dy &\approx \int_Q p(x, y) dx dy = \int_Q \sum_{r,s} a_{r,s} x^r y^s dx dy = \\ &= \sum_{r,s} a_{r,s} \int_{x_0}^{x_1} x^r dx \int_{y_0}^{y_1} y^s dy = \sum_{r,s} \frac{a_{r,s} (x_1^{r+1} - x_0^{r+1}) (y_1^{s+1} - y_0^{s+1})}{(r+1)(s+1)} \end{aligned}$$



Andere Möglichkeit: Lagrange-Ansatz wie 4.1.10.

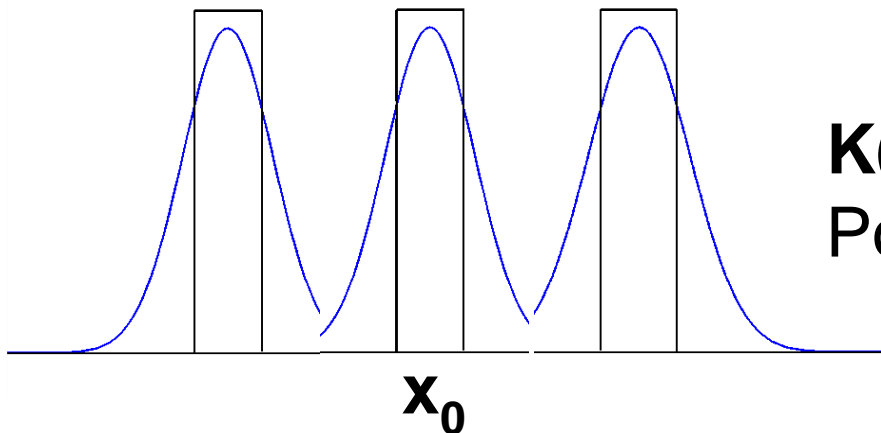


4.2.8. Deblurring bei verschmierten Bildern, Gauss'scher Weichzeichner:

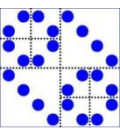
Modell: Jeder Pixelpunkt wird entsprechend einer Gauss-Verteilung, verschmiert, z.B. Aufnahme durch Atmosphäre:

(Zur Vereinfachung hier nur eindimensional)

An der Stelle x_0 wird der ursprüngliche Wert $f(x_0)$ ersetzt durch gestörten Wert $g(x_0)$, der sich aus der Überlagerung von Nachbarpunkte ergibt.



$K(x)$ Kernfunktion
Point Spread Function



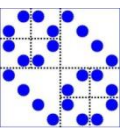
Das beobachtete Bild g entsteht durch eine ‚Faltung‘ des ursprünglichen Bildes f mit dem Gauss-Kern

$K(x)=\exp(-x^2/(2\pi\sigma^2))$, der genau die obige Funktion beschreibt

Um ein diskretes Modell zu erhalten, ersetze man das Faltungs-Integral an jeder diskreten Stelle x_j z.B. durch die Mittelpunktsumme (oder Trapezsumme) \rightarrow

$$g(x_j) = \int \exp\left(-\frac{(x_j - t)^2}{2\pi\sigma^2}\right) \cdot f(t) dt \approx$$
$$\approx h \sum_k \exp\left(-\frac{(x_j - x_k)^2}{2\pi\sigma^2}\right) \cdot f(x_k)$$

Dies führt auf ein lineares Gleichungssystem, das die Beziehung zwischen ursprünglichen und gestörten Werten beschreibt:



$$\left(\exp\left(-\frac{(j-k)^2}{2\pi\sigma^2 n^2}\right) \right)_{j,k=1}^n \cdot \vec{f} = \vec{g}$$

Durch Lösung des Gleichungssystems kann man aus den gestörten Daten $g(x_j)$ das Originalbild $f(x_k)$ wieder gewinnen.

Aber das Gleichungssystem $K \cdot \vec{f} = \vec{g}$

ist sehr schlecht konditioniert.

Daher verwendet man zusätzlich Regularisierung, z.B.:

$$\left(K^T K + \rho^2 I \right) \cdot \vec{f} = K^T \vec{g}$$

Die Beziehung

$$\left(\exp\left(-\frac{(j-k)^2}{2\pi\sigma^2 n^2}\right) \right)_{j,k=1}^n \cdot \vec{f} = \vec{g}$$

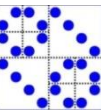
modelliert einen Weichzeichner oder Blur.

„Verwasche“ die Pixel durch Mittelwertbildung mit Nachbarn,

z.B. $x_k \rightarrow (x_{k-1} + 2x_k + x_{k+1}) / 4$

oder Maske $\begin{bmatrix} & 1 & \\ 1 & 4 & 1 \\ & 1 & \end{bmatrix} / 8$

Matlab `gauss_filter.m`



Unbekannter Blur:

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} b_1^{(i)} \\ b_2^{(i)} \end{pmatrix} = \begin{pmatrix} c_1^{(i)} \\ c_2^{(i)} \end{pmatrix} \Leftrightarrow$$

$$\Leftrightarrow \begin{cases} a_{11}b_1^{(i)} + a_{12}b_2^{(i)} = c_1^{(i)} \\ a_{21}b_1^{(i)} + a_{22}b_2^{(i)} = c_2^{(i)} \end{cases}$$

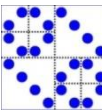
Für unbekanntem Blur A benutze „Testbilder“ b und betrachte Daten c nach Anwendung von A: $Ab=c$

$$\begin{pmatrix} b_1^{(1)} & b_2^{(1)} & 0 & 0 \\ 0 & 0 & b_1^{(1)} & b_2^{(1)} \\ b_1^{(2)} & b_2^{(2)} & 0 & 0 \\ 0 & 0 & b_1^{(2)} & b_2^{(2)} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} \cdot \begin{pmatrix} a_{11} \\ a_{12} \\ a_{21} \\ a_{22} \end{pmatrix} = \begin{pmatrix} c_1^{(1)} \\ c_2^{(1)} \\ c_1^{(2)} \\ c_2^{(2)} \\ \vdots \\ \vdots \end{pmatrix}$$

Ergibt überbestimmtes
Least Squares Problem
 $Ba=c$

Zu Bestimmung des unbekanntem
Blur-Operators A.

Damit kann dann ein unbekanntes Bild b aus gemessenem c rekonstruiert werden durch Lösen von $Ab=c$.



Ermittle unbekanntem Bluroperator durch Testbilder und Lösen des Least Squares Problems.

Ergibt Modellierung des Blur.

Danach liefert Bildaufnahme das gestörte Bild.

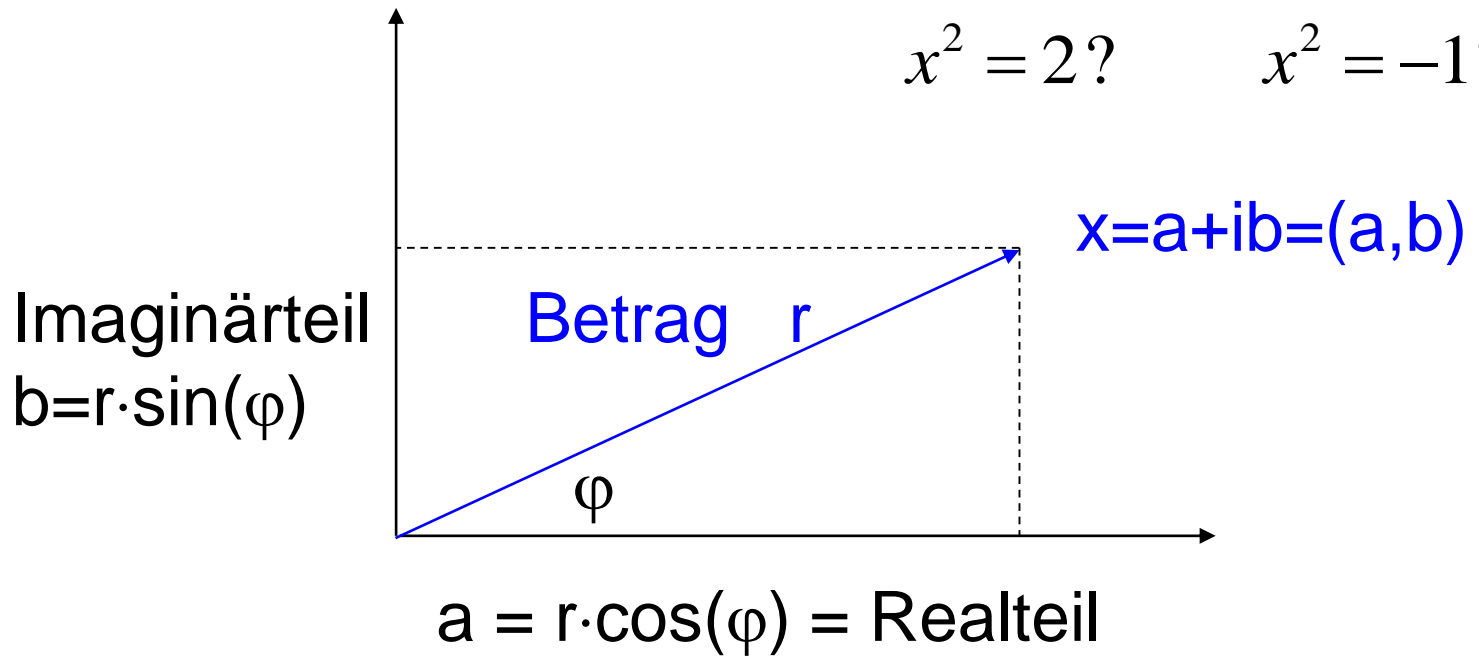
Formuliere Least Squares Problem zur Rekonstruktion des Originalbildes.

Regularisiere das Problem.

V. Diskrete Fourier-Transformation

5.1. Komplexe Zahlen und trigonometrische Interpolation

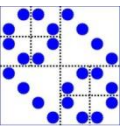
$i = \sqrt{-1}$, nicht schlimmer als $\sqrt{2}$
 $x^2 = 2?$ $x^2 = -1?$



IN
 Z
 Q
 IR
 C

Komplexe Zahl: *Anschaulich:*

Vektor (a, b) in der Ebene mit spezieller Multiplikation



Mögliche Darstellungen:

$$\mathbf{x} = \mathbf{a} + i \mathbf{b} = r \cdot (\cos(\varphi) + i \cdot \sin(\varphi)) = r \cdot \exp(i \varphi)$$
$$(\mathbf{a}, \mathbf{b}) = r \cdot (\cos(\varphi), \sin(\varphi)).$$

mit $r = |\mathbf{x}| = \sqrt{a^2 + b^2}, \quad \tan(\varphi) = \frac{b}{a}$

$$e^{i\varphi} = \sum_{n=0}^{\infty} \frac{(i\varphi)^n}{n!} = \sum_{k=0}^{\infty} \frac{(i\varphi)^{2k}}{(2k)!} + \sum_{k=0}^{\infty} \frac{(i\varphi)^{2k+1}}{(2k+1)!} =$$
$$= \sum_{k=0}^{\infty} \frac{(-1)^k \varphi^{2k}}{(2k)!} + i \cdot \sum_{k=0}^{\infty} \frac{(-1)^k \varphi^{2k+1}}{(2k+1)!} = \cos(\varphi) + i \cdot \sin(\varphi)$$

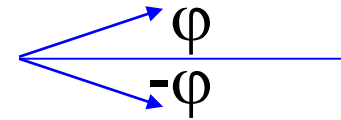
Multiplikation:

$$\mathbf{x} * \mathbf{y} = (\mathbf{a}+i\mathbf{b}) * (\mathbf{c}+i\mathbf{d}) = (\mathbf{ac}-\mathbf{bd})+i(\mathbf{ad}+\mathbf{bc}) =$$
$$= r \exp(i \varphi) * s \exp(i \theta) = rs \exp(i (\varphi+\theta))$$

(Beträge multipliziert, Winkel addiert)

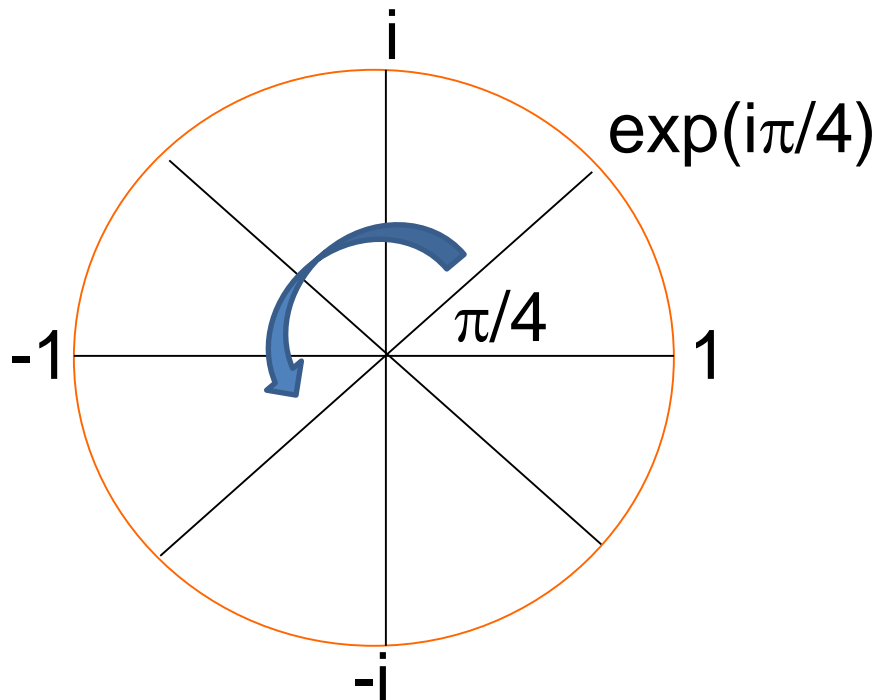


konjugiert komplexe Zahl: $\bar{x} = a - ib = r \cdot e^{-i\varphi}$



n-te Einheitswurzeln: $x^n = 1 = e^{2\pi i}$ hat n Lösungen

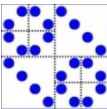
$$\exp\left(\frac{2\pi i \cdot j}{n}\right) = \cos\left(\frac{2\pi j}{n}\right) + i \sin\left(\frac{2\pi j}{n}\right), \quad j = 0, \dots, n-1$$



$$e^{2\pi i} - 1 = 0$$

für $n=8$

$$\left(e^{(i\pi/4)}\right)^8 = e^{(2\pi i)} = 1$$



Wir betrachten nun ein komplexes Interpolationsproblem mit den n -ten Einheitswurzeln als Stützstellen:

$$\omega^j, \quad j = 0, 1, \dots, n-1 \quad \text{mit} \quad \omega := \exp\left(\frac{2\pi i}{n}\right)$$

Zu den n Stützstellen ω^j und vorgegebenen Werten $v_j, j=0, 1, \dots, n-1$, finde man das Polynom mit den Koeffizienten $c_j, j=0, 1, \dots, n-1$, so dass gilt: $p(\omega^j) = v_j, \text{ für } j=0, 1, \dots, n-1$, also

$$\begin{aligned} v_j &= p(\omega^j) = p\left(\exp\left(\frac{2ij\pi}{n}\right)\right) = \\ &= c_0 + c_1 \cdot \exp\left(\frac{2ij\pi}{n}\right) + \dots + c_{n-1} \cdot \exp\left(\frac{2ij\pi(n-1)}{n}\right) = \\ &= \sum_{k=0}^{n-1} c_k \cdot \exp\left(\frac{2ijk\pi}{n}\right) = \sum_{k=0}^{n-1} c_k \cdot \omega^{jk} = \sum_{k=0}^{n-1} c_k (\omega^j)^k \\ &\quad \text{für } j=0, 1, \dots, n-1 \end{aligned}$$



Wie üblich erhalten wir lineares $n \times n$ – Gleichungssystem zur Bestimmung der c_j :

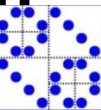
$$\begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \dots & \omega^{(n-1)(n-1)} \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-1} \end{pmatrix} = \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ \vdots \\ v_{n-1} \end{pmatrix}, \quad \mathbf{F}_n^* \mathbf{c} = \mathbf{v}$$

Diese Matrix F_n hat spezielle Eigenschaften: $\sum_{j=0}^{n-1} (\omega^k)^j = \frac{1 - \omega^{kn}}{1 - \omega^k}$

$$F_n^T = F_n, \quad F_n^H = \bar{F}_n = n \cdot F_n^{-1} = n \cdot \text{inv}(F_n)$$

(dabei ist $A^H := \text{conj}(A^T)$ das Hermite'sche von A)

Damit erhält man die Lösung durch $\mathbf{c} = \text{inv}(F_n)^* \mathbf{v} = F_n^H \mathbf{v} / n$



$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-1} \end{pmatrix} = \frac{1}{n} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \bar{\omega} & \bar{\omega}^2 & \dots & \bar{\omega}^{n-1} \\ 1 & \bar{\omega}^2 & \bar{\omega}^4 & \dots & \bar{\omega}^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \bar{\omega}^{n-1} & \bar{\omega}^{2(n-1)} & \dots & \bar{\omega}^{(n-1)(n-1)} \end{pmatrix} \cdot \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ \vdots \\ v_{n-1} \end{pmatrix}$$

mit $\bar{\omega} = \text{conj}(\omega) = \exp(-2\pi i/n)$, also

$$c_k = \frac{1}{n} \sum_{j=0}^{n-1} v_j \bar{\omega}^{jk}, \quad k = 0, 1, \dots, n-1$$

Dies beschreibt gerade die
Diskrete Fourier- Transformation (DFT), also