Institut für Informationsverarbeitung
Leibniz Universität Hannover
http://www.tnt.uni-hannover.de

Scientific Computing I - Übung 11

1 Theorie

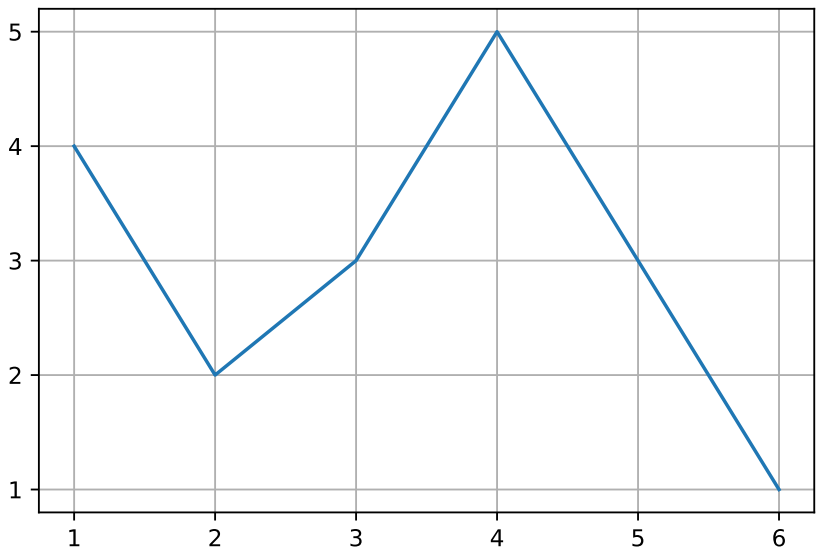
a) Nähern Sie das Integral

$$I(f) = \int_0^1 x \exp(-x^2) dx$$

mit Hilfe von Mittelpunktsregel, Trapezregel und Simpson-Regel an.

b) Schätzen Sie die in 1.b) mit der Mittelpunkts- bzw. der Trapezregel gemachten Fehler ab. Liefert eine der Regeln ein exaktes Ergebnis?

c) Nutzen Sie die Approximation mit zentralem Differenzenquotienten mit $h = 1$, um die erste und die zweite Ableitung an den Stellen $x \in \{2, 3, 4, 5\}$ der folgenden Funktion numerisch zu berechnen:



2 Numerische Integration

In dieser Aufgabe geht es darum, die numerische Integration, wie Sie sie im Theorieteil geübt haben, auch in MATLAB umzusetzen. Als Beispiel dient dazu die Funktion

$$f(x) = x \exp(-x^2).$$

a) Implementieren Sie eine MATLAB-Funktion, die zu einem gegeben x den Funktionswert $f(x)$ berechnet. Nutzen Sie ihre Funktion zusammen mit der MATLAB-Funktion `linspace` um einen Plot von f auf dem Intervall $x \in [0, 1]$ zu erstellen.

b) Implementieren Sie eine MATLAB-Funktion, welche zu gegebenen Intervallgrenzen $[a, b]$ das Integral

$$\int_a^b x \exp(-x^2) dx$$

mit Hilfe der Mittelpunktsregel schätzt und die Schätzung zurückgibt.

c) Implementieren Sie zwei Varianten ihrer Funktion aus b): eine Variante soll statt der Mittelpunktsregel die Trapezregel verwenden, die andere Variante soll die Simpson-Regel verwenden.

d) Nutzen Sie ihre Implementierungen aus b) und c) um eine adaptive Quadratur (Kapitel 8, Folie 36ff) als MATLAB-Funktion umzusetzen. Ihre Funktion bekommt dafür wieder Intervallgrenzen $[a, b]$ übergeben und soll zunächst den Fehler $E(f)$ (Kapitel 8, Folie 18) anhand dieser Grenzen abschätzen. Ist der Fehler größer als 10^{-6}, soll das Intervall $[a, b]$ in zwei gleich große Hälften aufgeteilt werden und ihre Funktion auf jede dieser Hälften angewendet werden. Das Ergebnis ist dann die Summe der beiden rekursiven Aufrufe. Ist der Fehler kleiner oder gleich dem Schwellwert, soll mit Hilfe der Simpson-Regel das Integral für das aktuelle Intervall geschätzt und zurückgegeben werden. Bei einer korrekten Implementierung ist der Rückgabewert ihrer Funktion etwa 0.3161.

e) Lassen Sie sich die Intervallgrenzen anzeigen, für die ihre Funktion aus d) die Simpson-Regel anwendet (eine Ausgabe in MATLABs Command Window reicht aus). Sind die Intervalle alle gleich groß oder passt sich der Algorithmus der Funktion an und verwendet für verschiedene Teilstücke der Funktion eine grobere bzw. feinere Aufteilung in Teilintervalle?

3 Gradientenbilder

In vielen Anwendungen der Bildverarbeitung, z.B. bei der Kantendetektion, kommen Gradientenbilder zum Einsatz. Dafür ist numerische Differentiation notwendig.

a) Erstellen Sie ein MATLAB-Skript, welches das zur Verfügung gestellte Bild `cameraman.png` lädt und anzeigt.

b) Erweitern Sie ihr Skript aus a) um die Berechnung und Anzeige eines Bilds, welches die horizontale Ableitung jedes Bildpunkts zeigt. Hierfür wechseln Sie wieder wie zuvor bei der Hauptkomponentenanalyse vom Datentyp `uint8` auf den Datentyp `double`. Zur Berechnung der Ableitung betrachten Sie jede Zeile i des Bilds als eine eigene Funktion f_i. Ist die Matrix A das ursprüngliche Bild, dann gilt $f_i(j) = A_{ij}$. Nun können Sie mit der Approximation der ersten Ableitung mit zentralem Differenzenquotienten (Kapitel 8, Folie 51) und der Schrittweite $h = 1$ für jede Zeile i die Ableitung bestimmen. Für die erste bzw. letzte Spalte setzen Sie die Ableitung einfach auf den Wert 0. Die Ableitung ist also wieder eine Zeile einer Matrix, die genauso lang ist wie die ursprüngliche Bildzeile. Zusammen ergeben alle so berechneten Ableitungszeilen wieder ein Bild.

   Für das Anzeigen des Bildes sollten Sie nicht auf den Datentyp `uint8` zurückwechseln, aber den Wertebereich auf $[0, 1]$ skalieren. Ziehen Sie dafür von allen Einträgen in ihrem Ableitungsbild den kleinsten Wert, der im gesamten Ableitungsbild vorkommt, ab. Anschließend dividieren Sie ihr Ableitungsbild durch den größten Wert im gesamten Ableitungsbild. Das so skalierte Bild können Sie wie gewohnt mittels `imshow` anzeigen. Es sollte ein größtenteils graues Bild mit hellen und dunklen Flächen dort sein, wo sich die Helligkeit in horizontaler Richtung ändert.

c) Erweitern Sie ihr Skript aus b) um die Berechnung und Anzeige der vertikalen Ableitung. Gehen Sie dazu analog zu b) vor, betrachten nun aber jede Spalte j statt jeder Zeile i als eine Funktion f_j. Es gilt $f_j(i) = A_{ij}$.

d) Die beiden berechneten Bilder aus b) und c) ergeben zusammen den Gradienten für jeden Bildpunkt. Erweitern Sie ihr Skript um die Berechnung und die Anzeige des Betrags des Gradienten. Sei H das in b) berechnete Bild und V das in c) berechnete, dann ist der Gradient ∇A_{ij} des Bildpunkts A_{ij} gleich dem Vektor (H_{ij}, V_{ij}). Berechnen Sie für ihr Gradientenbild G den Betrag des Gradienten für jeden Bildpunkt, d.h. $G_{ij} = \|\nabla A_{ij}\|_2$.

   Auch für G gilt, dass Sie es auf den Wertebereich $[0, 1]$ skalieren sollten, bevor Sie es anzeigen. G ist allerdings ein größtenteils schwarzes Bild mit hellen Flächen dort, wo in eine beliebige Richtung eine große Helligkeitsänderung in A zu sehen ist.

e) Erstellen Sie eine Kopie ihres Skripts. In der Kopie sollen Sie statt einer Approximation der ersten Ableitung die Approximation der zweiten Ableitung mit zentralem Differenzenquotienten (Kapitel 8, Folie 52) berechnen.