Bauer, Michael
Ehrenberg, Stephanie

Aufgabe 1)

b)

Testergebnisse:

```
>> Nadelimpuls
Fuer h = 2 ist die Summe: 13.9999 mit einem relativen Fehler von 0.99554
Fuer h = 1 ist die Summe: 250008.6874 mit einem relativen Fehler von 78.6309
Fuer h = 0.5 ist die Summe: 125019.3832 mit einem relativen Fehler von 38.8203
Fuer h = 0.25 ist die Summe: 62540.7641 mit einem relativen Fehler von 18.92
Fuer h = 0.125 ist die Summe: 31333.5031 mit einem relativen Fehler von 8.9801
Fuer h = 0.0625 ist die Summe: 15793.8055 mit einem relativen Fehler von 4.0305
Fuer h = 0.03125 ist die Summe: 8150.5191 mit einem relativen Fehler von 1.596
Fuer h = 0.015625 ist die Summe: 4571.9533 mit einem relativen Fehler von 0.45622
Fuer h = 0.0078125 ist die Summe: 3199.0658 mit einem relativen Fehler von 0.018943
Fuer h = 0.0039063 ist die Summe: 3000.179 mit einem relativen Fehler von 0.044405
Fuer h = 0.0019531 ist die Summe: 3107.2584 mit einem relativen Fehler von 0.010299
Fuer h = 0.00097656 ist die Summe: 3138.3293 mit einem relativen Fehler von 0.0004024
Fuer h = 0.00048828 ist die Summe: 3139.5906 mit einem relativen Fehler von 6.4526e-07
Fuer h = 0.00024414 ist die Summe: 3139.5927 mit einem relativen Fehler von 1.6634e-12
Fuer h = 0.00012207 ist die Summe: 3139.5927 mit einem relativen Fehler von 8.6906e-16
Fuer h = 6.1035e-05 ist die Summe: 3139.5927 mit einem relativen Fehler von 2.0278e-15
```

```
>> format LONG
>> [ergebniss,count]=  AdaptQuad(0.0001,-1,1,1)

ergebniss =

     3.139592654264642e+03


count =

      136574
```

Die Anzahl an benötigten Funktionsaufrufen bei dem mit AdaptQuad implementierten Verfahren ist deutlich geringer als bei Ausführung des Scripts Nadelimpuls.

Dies lässt sich dadurch erklären, dass in Intervallen mit geringer Steigung des Funktionsgraphen das adaptive Verfahren Funktionsaufrufe „einsparen“ kann – hier ist keine so feine Auflösung (also eine numerische Integration über eine sehr kleine Intervallbreite) notwendig.

Man erreicht also mit einer deutlich verringerten Anzahl an Funktionsaufrufen mit der AdaptQuad-Funktion eine vergleichbare Genauigkeit zum Nadelimpuls-Skript, welches stur über die gesamte Intervallbreite mit äquidistanten Stützstellen bei konstanter Breite der Teilintervalle rechnet, unabhängig davon, ob eine derart feine Auflösung über das gesamte Intervall notwendig ist.

c)

Bei den Werten sind wir von einem relativen Fehler von 0,1 ausgegangen.

Testergebnis:

Bauer, Michael
Ehrenberg, Stephanie

```
>> Aufgabe1c
Ergebnis mit AdaptQuad: 3139.6261 mit 262 Funktionsaurufe
Ergebnis mit Quad: 3139.8462 mit 89 Funktionsaurufe
Ergebnis mit Quadgk: 3133.5364
Ergebnis mit Quadl: 3139.593
```

Dabei fiel auf, dass das genaueste Ergebnis von der Matlab Funktion quadl ausgegeben wurde, gefolgt von unserer eigenen Funktion AdaptQuad, danach kommt quad, und am schlechtesten hat quadgk abgeschnitten.

Die höhere Genauigkeit bei AdaptQuad haben wir uns durch eine viel höhere Anzahl an Funktionsaufrufen erkauft; quad hat nur 89 Aufrufe benötigt, während AdaptQuad ganze 262 benötigt hat, die Differenz zwischen den Ergebniswerten liegt aber noch nicht mal bei 0,2 .

Aufgabe 2.

a.

Matrix A:

// Zeilen 1 und 3 tauschen, Zeilen 2 und 4 tauschen

$$\begin{pmatrix} 4 & 8 & 16 & 8 \\ 2 & 12 & 48 & 28 \\ -2 & 4 & 24 & 0 \\ -1 & 2 & 16 & 12 \end{pmatrix}\begin{pmatrix} 4 \\ 10 \\ 2 \\ 5 \end{pmatrix} * (-0.25) \rightarrow \begin{pmatrix} 4 & 8 & 16 & 8 \\ 2 & 12 & 48 & 28 \\ -2 & 4 & 24 & 0 \\ 0 & 4 & 20 & 14 \end{pmatrix}\begin{pmatrix} 4 \\ 10 \\ 2 \\ 6 \end{pmatrix} * -(0.5)$$

$$\begin{pmatrix} 4 & 8 & 16 & 8 \\ 2 & 12 & 48 & 28 \\ 0 & 8 & 32 & 4 \\ 0 & 4 & 20 & 14 \end{pmatrix}\begin{pmatrix} 4 \\ 10 \\ 4 \\ 6 \end{pmatrix} * 0.5 \rightarrow \begin{pmatrix} 4 & 8 & 16 & 8 \\ 0 & 8 & 40 & 24 \\ 0 & 8 & 32 & 4 \\ 0 & 4 & 20 & 14 \end{pmatrix}\begin{pmatrix} 4 \\ 8 \\ 4 \\ 6 \end{pmatrix} * 0.5$$

$$\begin{pmatrix} 4 & 8 & 16 & 8 \\ 0 & 8 & 40 & 24 \\ 0 & 8 & 32 & 4 \\ 0 & 0 & 0 & 2 \end{pmatrix}\begin{pmatrix} 4 \\ 8 \\ 4 \\ 2 \end{pmatrix} * 1 \rightarrow \begin{pmatrix} 4 & 8 & 16 & 8 \\ 0 & 8 & 40 & 24 \\ 0 & 8 & -8 & -20 \\ 0 & 0 & 0 & 2 \end{pmatrix}\begin{pmatrix} 4 \\ 8 \\ -4 \\ 2 \end{pmatrix}$$

$$U = \begin{pmatrix} 4 & 8 & 16 & 8 \\ 0 & 8 & 40 & 24 \\ 0 & 8 & -8 & -20 \\ 0 & 0 & 0 & 2 \end{pmatrix} \qquad L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 1 & 0 & 0 \\ -0.5 & 1 & 1 & 0 \\ -0.25 & 0.5 & 0 & 1 \end{pmatrix}$$

$$P = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \qquad A * x = b \rightarrow x = \begin{pmatrix} -9 \\ 8 \\ -2 \\ 1 \end{pmatrix}$$

Matrix B

$$\begin{pmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{pmatrix}\begin{pmatrix}1&0&0&0&1\\-1&1&0&0&1\\-1&-1&1&0&1\\-1&-1&-1&1&1\\-1&-1&-1&-1&1\end{pmatrix}=\begin{pmatrix}1&0&0&0&0\\1&1&0&0&0\\1&0&1&0&0\\1&0&0&1&0\\1&0&0&0&1\end{pmatrix}\begin{pmatrix}1&0&0&0&1\\0&1&0&0&2\\0&-1&1&0&2\\0&-1&-1&1&2\\0&-1&-1&-1&2\end{pmatrix}$$

$$\begin{pmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{pmatrix}\begin{pmatrix}1&0&0&0&1\\-1&1&0&0&1\\-1&-1&1&0&1\\-1&-1&-1&1&1\\-1&-1&-1&-1&1\end{pmatrix}=\begin{pmatrix}1&0&0&0&0\\1&1&0&0&0\\1&1&1&0&0\\1&1&0&1&0\\1&1&0&0&1\end{pmatrix}\begin{pmatrix}1&0&0&0&1\\0&1&0&0&2\\0&0&1&0&4\\0&0&-1&1&4\\0&0&-1&-1&4\end{pmatrix}$$

$$\begin{pmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{pmatrix}\begin{pmatrix}1&0&0&0&1\\-1&1&0&0&1\\-1&-1&1&0&1\\-1&-1&-1&1&1\\-1&-1&-1&-1&1\end{pmatrix}=\begin{pmatrix}1&0&0&0&0\\1&1&0&0&0\\1&1&1&0&0\\1&1&1&1&0\\1&1&1&0&1\end{pmatrix}\begin{pmatrix}1&0&0&0&1\\0&1&0&0&2\\0&0&1&0&4\\0&0&0&1&8\\0&0&0&-1&8\end{pmatrix}$$

$$\begin{pmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{pmatrix}\begin{pmatrix}1&0&0&0&1\\-1&1&0&0&1\\-1&-1&1&0&1\\-1&-1&-1&1&1\\-1&-1&-1&-1&1\end{pmatrix}=\begin{pmatrix}1&0&0&0&0\\1&1&0&0&0\\1&1&1&0&0\\1&1&1&1&0\\1&1&1&1&1\end{pmatrix}\begin{pmatrix}1&0&0&0&1\\0&1&0&0&2\\0&0&1&0&4\\0&0&0&1&8\\0&0&0&0&16\end{pmatrix}$$

b.

Wir würden in diesen Fall eine QR-Zerlegung machen, da deren Komplexität geringer ist als die Komplexität einer LU-Zerlegung und ihre numerische Stabilität zudem höher ist.

Die LU Zerlegung hat eine Komplexität von $O(n^3)$, während QR „nur“ eine Komplexität von $O(n^2)$ hat.

Bei großen n dauert die Berechnung mit LU Zerlegung deutlich länger als mit QR Zerlegung.

Aufgabe 3.

a.

$$A=\begin{pmatrix}3&-1&1\\4&-8&3\\0&3&1\end{pmatrix}\qquad s_{21}=\frac{4}{\sqrt{4^2+3^2}}=0{,}8\qquad c_{21}=\frac{3}{\sqrt{4^2+3^2}}=0{,}6\qquad Q_{21}=\begin{pmatrix}0{,}6&0{,}8&0\\-0{,}8&0{,}6&0\\0&0&1\end{pmatrix}$$

$$Q_{21}A=\begin{pmatrix}0{,}6&0{,}8&0\\-0{,}8&0{,}6&0\\0&0&1\end{pmatrix}*\begin{pmatrix}3&-1&1\\4&-8&3\\0&3&1\end{pmatrix}=\begin{pmatrix}5&-7&3\\0&-4&1\\0&3&1\end{pmatrix}$$

$$s_{32}=\frac{3}{\sqrt{3^2+(-4)^2}}=0{,}6\qquad c_{32}=\frac{-4}{\sqrt{3^2+(-4)^2}}=-0{,}8\qquad Q_{32}=\begin{pmatrix}1&0&0\\0&-0{,}8&0{,}6\\0&-0{,}6&-0{,}8\end{pmatrix}$$

$$R=Q_{32}Q_{21}A=\begin{pmatrix}1&0&0\\0&-0{,}8&0{,}6\\0&-0{,}6&-0{,}8\end{pmatrix}*\begin{pmatrix}0{,}6&0{,}8&0\\-0{,}8&0{,}6&0\\0&0&1\end{pmatrix}=\begin{pmatrix}-5&7&1\\0&4{,}2&0\\0&0&-1{,}4\end{pmatrix}$$

Bauer, Michael
Ehrenberg, Stephanie

c.

Es wird die gleiche Anzahl an Rechenoperationen benötigt wie bei einer Hessenberg Matrix, da alle Stellen, die bereits 0 sind nicht mehr berechnet werden müssen. Auch entspricht die Anzahl an Stellen, die mit 1 belegt sind, exakt der Anzahl an Stellen der Nebendiagonalen bei der Hessenberg Matrix (bei Verwendung der Givens-Rotation).

Als Lösungsverfahren würde sich die Householder Methode anbieten, da diese nur einen Funktionsaufruf benötigt, um das Gleichungssystem zu berechnen.