Übungen für Angewandte Numerik I
Sommersemester 2011

1. Gegeben seien die Koeffizienten $\{a_k\}_{k=0}^n$ eines Polynoms $n$ten Grades:

$$p(x) = \sum_{k=0}^{n} a_k x^k$$

Schreibe einen Matlab-Code zur Auswertung des Polynoms (a) Mittels des Horner Algorithmus und (b) durch eine gewichtete Summe der Potenzen. Vergleiche die Ergebnisse. Wie kann die Genauigkeit für den einen oder den anderen Code bestätigt werden? (*Gelöst von Herrn Hofstadler*.)

2. Finde eine IEEE Fließkommazahl Darstellung der Zahl $(100000.1)_{10}$ mit einfacher Genauigkeit. (*Gelöst von Herrn Hattenberger*.)

3. Die Nullstelle einer unbekannten Funktion f soll gefunden werden, und man hat Zugang zu den Werten von f nur durch ein Computerprogramm. Das Computerprogramm berechnet $f(x) = \arctan[4(x - \hat{x})]$, $\hat{x} = 1 + 2^{24}$, in doppelter Genauigkeit. Ein iteratives Verfahren wird zur Bestimmung der Nullstelle verwendet, und man speichert die Approximation $x_k \approx \hat{x}$ in der $k$ten Iteration mit einfacher Genauigkeit. Finde die kleinsten Toleranzen ε, δ und η, die für die Abbruchskriterien

$$|x_k - x_{k-1}| < \varepsilon |x_k|, \quad |x_k - x_{k-1}| < \delta, \quad |f(x_k)| < \eta$$

zuverlässig verwendet werden können. Welches Abbruchskriterium verlangt kein Vorwissen von der Funktion? (*Gelöst von Herrn Ramsauer*.)

4. Angenommen gilt $\lim_{x\to x_0} |F(x)/G(x)| < \infty$. Zeigen Sie, es gibt $K, \delta > 0$ sodass $|F(x)| \le K|G(x)|$, $\forall x \in (x_0 - \delta, x_0 + \delta)$. (*Gelöst von Herrn Ramsauer*.)

5. Das Gebiet $\Omega = (0,1) \subset \mathbb{R}^1$ sei mit einem Gitter $\{x_i\}_{i=0}^N \subset \bar{\Omega}$ diskretisiert, wobei die Gitterpunkte so definiert sind: $x_i = ih$, $0 \le i \le N$, $h = 1/N$. Für eine Funktion $u \in C^3(\bar{\Omega})$ (weitere Glattheit ist nicht garantiert) seien die Werte $\{u(x_i)\}_{i=0}^N$ gegeben. Zeige:

$$\max_{1\le i\le N-1} \left| u''(x_i) - \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2} \right| = \mathcal{O}(h).$$

(*Gelöst von Herrn Hofstadler*.)

6. Gegeben seien die Werte $u(x_i)$, $0 \le i \le N$, auf dem oben definierten Gitter $\{x_i\}_{i=0}^N$. Unter welchen Bedingungen der Funktion u gilt die Konvergenz:

$$h \sum_{i=0}^{N} u(x_i) \xrightarrow{h\to 0} \int_\Omega u(x)dx.$$

Erkläre. (*Gelöst von Herrn Hofstadler*.)

7. Gegeben seien die Werte $v(x_i)$, $0 \le i \le N$, auf dem oben definierten Gitter $\{x_i\}_{i=0}^N$. Schreibe eine Approximation $J_h(\boldsymbol{u})$ für

$$J(u) = \int_\Omega (u - v)^2 d\boldsymbol{x} + \mu \int_\Omega \sqrt{|u'|^2 + \varepsilon^2} d\boldsymbol{x}$$

und leite die Optimalitätsbedingung $\nabla J_h(\boldsymbol{u}) = 0$ her. (*Gelöst von Herrn Keeling*.)

8. Schreibe einen Pseudo-Code für Vorwärts Substitution und berechne und die Anzahl der gesamten Operationen.

9. Betrachte den Matlab-Code zur Implementierung der Methode der Konjugierten Gradienten für die Lösung des linearen Gleichungssystems $A\boldsymbol{x} = \boldsymbol{b}$ nach $\boldsymbol{x} \in \boldsymbol{R}^n$ wobei $\boldsymbol{b} \in \boldsymbol{R}^n$ und $A \in \boldsymbol{R}^{n \times n}$:

```
function x = cg(A,b,kmax)
x    = b;
r    = b - A*x;
rho1 = sum(r.^2);
for k=1:kmax
    if (k == 1)
        p = r;
    else
        beta = rho1/rho0;
        p    = r + beta*p;
    end
    w = A*p;
    alpha = rho1/(p'*w);
    x = x + alpha*p;
    r = r - alpha*w;
    rho0 = rho1;
    rho1 = sum(r.^2);
end
```

Welcher Wert soll man für `kmax` nehmen, damit der Code die Lösung des linearen Gleichungssystems genau (mit exakter Arithmetik) berechnet? Schreibe für jede Zeile *nur* die Großenordnung der Fließkomma-Operationen (`flops`) die durchgeführt werden, z.B. $\mathcal{O}(n)$, $\mathcal{O}(n^2)$, usw. Ist die Großenordnung der gesamten `flops` für diesen Code vergleichbar mit der für Gaußsche Elimination?

10. Die folgende Matrix sei gegeben:

$$A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,q+1} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ a_{p+1,1} & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & a_{n-q,n} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & a_{n,n-p} & \cdots & a_{n,n} \end{bmatrix} \in \mathbb{R}^{n \times n}$$

Bemerke:

$$a_{i,j} = 0 \text{ wenn } p < i - j \text{ oder } j - i > q.$$

Die *Bandbreite* von A is $p + q + 1$, die kleinste Anzahl der benachbarten Diagonalen, zu denen die nicht trivialen Elemente eingeschränkt sind.

(a) Schreibe einen Pseudo-Code zur Implementierung des Gauß Algorithmus für A mit möglichst wenig arithmetischen Operationen.

(b) Um Speicherplatz zu reduzieren, wird A als Liste der Diagonalen gespeichert:

$$B = \begin{bmatrix} a_{p+1,1} & \cdots & \cdots & a_{1,1} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ a_{n,n-p} & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & \ddots & \vdots & \vdots & \vdots & \vdots & a_{1,q+1} \\ \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & a_{n,n} & \cdots & \cdots & a_{n-q,n} \end{bmatrix} \in \mathbb{R}^{n \times (p+q+1)}$$

Bemerke: Die nicht trivialen Elemente von $A = \{a_{i,j}\}$ lassen sich bezüglich der Elemente von $B = \{b_{k,l}\}$ so darstellen:

$$b_{k,l} = a_{p+k-l+1,k} \quad a_{i,j} = b_{j,p+j-i+1}$$

Schreibe den Pseudo-Code vom Teil (a) bezüglich der Elemente von B um. (*Gelöst teilweise von Herrn Hofstadler und teilweise von Herrn Keeling*.)

11. Die Matrix $A \in \mathbb{R}^{n\times n}$ ist eine Bandmatrix mit Bandbreite $p + q + 1$, wobei p die linke und q die rechte Halbbandbreiten sind, d.h. die Anzahl der streng unteren Diagonalen ist p, und die Anzahl der streng oberen Diagonalen ist q. Für $\boldsymbol{b} \in \mathbb{R}^n$ soll eine Lösung $\boldsymbol{x} \in \mathbb{R}^n$ des linearen Gleichungssystems $A\boldsymbol{x} = \boldsymbol{b}$ durch Gaußshe Elimination berechnet werden, und zwar mit möglichst wenig arithmetischen Operationen. Leite eine Funktion $G(n, p, q)$ her, wobei die Größenordnung der Anzahl der Operationen für die Lösung durch $\mathcal{O}(G(n, p, q))$ gegeben ist.

    Hinweis: Für eine obere Schranke kann man im obigen Code die Schleifengrenzen `min(n,k+p)` und `min(n,k+q)`

    ```
    for k=1,...,n-1                        % Pivot-Index
        ...
        for i=k+1,...,min(n,k+p)           % Bis zur linken Grenze
            ...
            for j=k+1,...,min(n,k+q)       % Bis zur rechten Grenze
    ```

    mit `k+p` beziehungsweise `k+q` ersetzen. Für eine untere Schranke kann man die Schleifegrenze `n-1` für das Pivot-Index mit `min(n-p,n-q) = n - max(p,q)` ersetzen, und dann gelten `min(n,k+p) = k+p` und `min(n,k+q) = k+q`.

12. Für die obige Bandmatrix A soll die LU-Zerlegung berechnet werden. Zeige, L hat die linke Halbbandbreite p und die rechte Halbbandbreite 0, und U hat die linke Halbbandbreite 0 und die rechte Halbbandbreite q.

13. Für die Matrix

    $$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 2 \\ 3 & 2 & 1 \end{bmatrix}$$

    führe Gaußsche Elimination mit einer geeigneten Pivotsuche durch, um eine Permutationsmatrix P, eine untere Dreiecksmatrix L und eine obere Dreiecksmatrix U zu bestimmen, sodass $PA = LU$ gilt. (*Gelöst von Herrn Ramsauer*.)

14. Die Vektoren $\boldsymbol{u}^k \in \mathbb{R}^N$ sollen durch die Lösung des folgenden Systems bestimmt werden:

    $$(I + \Delta t A)\boldsymbol{u}^{k+1} = \boldsymbol{u}^k, \quad k = 0, 1, 2, \ldots$$

    wobei für $\Delta x, \Delta t, d, f, c > 0$,

    $$\mathbb{R}^{N\times N} \ni A = \frac{d}{\Delta x^2} \begin{bmatrix} 2 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ 0 & & & -1 & 1 \end{bmatrix} + \frac{f}{\Delta x} \begin{bmatrix} 1 & & 0 \\ -1 & 1 & \\ & \ddots & \ddots \\ 0 & -1 & 1 \end{bmatrix} + cI$$

    Leite eine Bedingung für die Parameter $d, f, c, \Delta t, \Delta x$ her, sodass keine Pivot-Suche verwendet werden muss. Schätze die Eigenwerte der Matrix $(I + \Delta t A)$ und anschliessend der Matrix $(I + \Delta t A)^{-1}$ ab. (*Gelöst von Herrn Keeling*.)

15. Zeige, für eine reguläre schwach diagonal dominante Matrix $A \in \mathbb{R}^{n\times n}$ ist der Gauß Algorithmus *ohne* Pivotsuche immer anwendbar.

16. Die exakte Lösung des folgenden Systems,

$$\begin{bmatrix} 30.00 & 591400 \\ 5.291 & -6.130 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 591700 \\ 46.78 \end{bmatrix}$$

ist $x_1^* = 10.00$, $x_2^* = 1.000$. Das System soll aber mit 4-Dezimalzifferiger Arithmetik bei *jeder* Rechnung gelöst werden. Verwende eine geeignete Pivot-Strategie, um eine ziemlich genaue Approximation der Lösung zu berechnen.

17. Für die Lösung eines tridiagonalen Systems

$$\begin{bmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & a_n & b_n \end{bmatrix}$$

gibt es den Thomas Algorithmus,

$$c'_i = \begin{cases} c_1/b_1, & i = 1 \\ c_i/(b_i - c'_{i-1}a_i), & i = 2, \ldots, n-1 \end{cases} \qquad d'_i = \begin{cases} d_1/b_1, & i = 1 \\ (d_i - d'_{i-1}a_i)/(b_i - c'_{i-1}a_i) & i = 2, \ldots, n \end{cases}$$

$$\begin{aligned} x_n &= d'_n \\ x_i &= d'_i - c'_i x_{i+1}, i = n-1, \ldots, 1. \end{aligned}$$

Zeige, der Thomas Algorithmus kostet $\mathcal{O}(n)$ `flops`.

18. Zum Entrauschen der verrauschten Daten $\boldsymbol{v}$ wird das Randwertproblem

$$\begin{cases} -\mu u'' + u &= v, \quad \text{in } \Omega = (0,1) \\ u' &= 0, \quad \text{auf } \partial\Omega \end{cases}$$

wie folgt auf dem Gitter $x_i = ih$, $i = 0, \ldots, N$, $h = 1/N$, diskretisiert:

$$u(x_i) \approx u_i \quad v(x_i) \approx v_i \quad \left[\frac{\mu}{h^2}D + I\right]\boldsymbol{u} = \boldsymbol{v}, \quad D = \begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{bmatrix}$$

wobei $D \in \mathbb{R}^{(N+1)\times(N+1)}$ und $\mu > 0$ steuert, wie geglättet die Lösung $\boldsymbol{u}$ wird. Verwende den Thomas Algorithmus, um dieses System für eine verrauschte Treppenfunktion $\boldsymbol{v}$ zu lösen. Stelle $\boldsymbol{v}$ und $\boldsymbol{u}$ für verschiedene Werte von μ grafisch dar.

19. Zum Entrauchen der verrauschten Daten $\boldsymbol{v}$ wird das Randwertproblem

$$\begin{cases} -\mu(u_{xx} + u_{yy}) + u &= v, \quad \text{in } \Omega = (0,1)^2 \\ u_n &= 0, \quad \text{auf } \partial\Omega \end{cases}$$

wie folgt auf dem Gitter $(x_i, y_j) = h(i,j)$, $i,j = 0,\ldots,N$, $h = 1/N$, diskretisiert:

$$\left[\frac{\mu}{h^2}D + I\right]\boldsymbol{u} = \boldsymbol{v} \qquad \begin{array}{rcll} \boldsymbol{u} & = & \langle u_{11},\ldots,u_{N1},\ldots,u_{1N},\ldots,u_{NN}\rangle^{\mathrm{T}} & u(x_i,y_j) \approx u_{ij} \\ \boldsymbol{v} & = & \langle v_{11},\ldots,v_{N1},\ldots,v_{1N},\ldots,v_{NN}\rangle^{\mathrm{T}} & v(x_i,y_j) \approx v_{ij} \end{array}$$

$$\left[\frac{\mu}{h^2}D + I\right] = \frac{\mu}{h^2}\begin{bmatrix} D_1 & D_0 & & & & & \\ D_0 & D_2 & D_0 & & & & \\ & D_0 & D_2 & D_0 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & D_0 & D_2 & D_0 & \\ & & & & D_0 & D_2 & D_0 \\ & & & & & D_0 & D_1 \end{bmatrix} + I$$

$$D_0 = -I, \quad D_1 = \begin{bmatrix} 2 & -1 & & & \\ -1 & 3 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 3 & -1 \\ & & & -1 & 2 \end{bmatrix}, \quad D_2 = \begin{bmatrix} 3 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 3 \end{bmatrix}$$

wobei $D_0, D_1, D_2 \in \mathbb{R}^{(N+1)\times(N+1)}$ und $\mu > 0$ steuert, wie geglättet die Lösung $\boldsymbol{u}$ wird. Schreibe die Koeffizeitenmatrix mit *sparse* format und löse das System für eine verrauschte Treppenfunktion $\boldsymbol{v}$ mit

(a) der Jacobi Methode
(b) der (Symmetrisch) Gauß-Seidel Methode
(c) der Methode der Konjugierten Gradienten.

Stelle $\boldsymbol{v}$ und $\boldsymbol{u}$ für verschiedene Werte von μ grafisch dar. (Hinweise: Ein nützliches Code-Stück:

```
NN = N*N;
Dxdiag      = -ones(N,N);
Dxdiag(N,:) = 0;
Dxsup       = ones(N,N);
Dxsup(1,:)  = 0;
Dx  = spdiags(Dxdiag(:),0,NN,NN) + spdiags(Dxsup(:),1,NN,NN);
Dydiag      = -ones(N,N);
Dydiag(:,N) = 0;
Dysup       = ones(N,N);
Dysup(:,1)  = 0;
Dy  = spdiags(Dydiag(:),0,NN,NN) + spdiags(Dysup(:),N,NN,NN);
D   = Dx'*Dx + Dy'*Dy;
```

Der `imagesc`-Befehl in `MATLAB` ist geeignet für die grafische Darstellung von Bildern.)

20. Zeige Äquivalenz der Normen $\|\cdot\|_1$ und $\|\cdot\|_2$.

21. Die gemessenen pH-Werte einer chemischen Lösung sind $\boldsymbol{f} = \langle a, b, \ldots, b\rangle \in \mathbb{R}^{n+1}$. Für $\boldsymbol{e} = \langle 1, \ldots, 1\rangle$ zeige
$$\frac{a+nb}{1+n} = \arg\min_{c\in\mathbb{R}} \|\boldsymbol{f} - c\boldsymbol{e}\|_2^2 \quad \text{während} \quad b = \arg\min_{c\in\mathbb{R}} \|\boldsymbol{f} - c\boldsymbol{e}\|_1$$

22. Zeige für eine gegeben Vektornorm $\|\cdot\|_{\mathrm{V}}$ über $\mathbb{R}^n$, die folgende Funktion
$$\|A\|_{\mathrm{M}} = \sup_{\boldsymbol{x}\in\mathbb{R}^n} \frac{\|A\boldsymbol{x}\|_{\mathrm{V}}}{\|\boldsymbol{x}\|_{\mathrm{V}}}$$
ist eine Matrixnorm über $\mathbb{R}^{n\times n}$.

23. Zeige für eine Matrix $\{A_{ij}\} = A \in \mathbb{R}^{n\times n}$, die Frobenius Matrixnorm,

$$\|A\|_{\mathrm{F}} = \left[\sum_{i,j=1}^{n} |A_{ij}|^2\right]^{\frac{1}{2}}$$

ist mit der Vektornorm $\|\boldsymbol{x}\|_2$ kompatibel.

24. Finde die Iterationsmatrix T_{SGS} für die Symmetrische Gauß-Seidel Methode.

25. Leite die Approximierte Inverse M_{SGS} für die Symmetrische Gauß-Seidel Methode her.

26. Zeige, wenn A symmetrisch ist, ist M_{SGS} symmetrisch, während M_{GS} im allgemeinen nicht symmetrisch ist.

27. Schreibe einen Matlab-Code zur Implementierung der Vektoriteration zur Bestimmung des dominanten Eigensystems einer gegebenen Matrix $A \in \mathbb{R}^{n\times n}$, die die Voraussetzungen der Konvergenz erfüllt. Wie könnte man den Code anschliessend verwenden, um die nächsten Eigensysteme zu bestimmen?

28. Zeige für den folgenden Algorithmus,

$$\begin{array}{rclcrcl} \boldsymbol{v} &=& \boldsymbol{x}/\|\boldsymbol{x}\|_\infty & & \theta &=& \tau u_1 \\ \tau &=& \mathrm{sign}(v_1)\|\boldsymbol{v}\|_2 & & \sigma &=& \tau\|x\|_\infty \\ \boldsymbol{u} &=& \boldsymbol{v} + \tau\hat{\boldsymbol{e}}^{(1)} & & P &=& I - \boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}, \quad P\boldsymbol{x} = -\sigma\hat{\boldsymbol{e}}^{(1)} \end{array}$$

es gilt $\theta = \frac{1}{2}\|\boldsymbol{u}\|_2^2$.

29. Schreibe einen Matlab-Code, der für eine gegebene Matrix $A \in \mathbb{R}^{n\times n}$ eine orthogonale Matrix P bestimmt, die $PA = U$ erfüllt, wobei U eine obere Dreiecksmatrix ist.

30. Für die orthogonale Transformation

$$G = \begin{bmatrix} 1 & 0 & \cdots & & & & & \cdots & 0 \\ 0 & \ddots & & & & & & & \vdots \\ \vdots & & 1 & & & & & & \\ & & & c & \cdots & s & & & \\ & & & \vdots & \ddots & \vdots & & & \\ & & & -s & \cdots & c & & & \\ & & & & & & 1 & & \vdots \\ \vdots & & & & & & & \ddots & 0 \\ 0 & \cdots & & & & & \cdots & 0 & 1 \end{bmatrix} \quad \text{und} \quad x = \begin{bmatrix} \vdots \\ \\ \vdots \\ a \\ \vdots \\ b \\ \vdots \\ \\ \vdots \end{bmatrix}$$

sollen durch das Produkt Gx nur die Elemente a und $b \neq 0$ in x geändert werden, wobei b vernüllt wird. Bestimme die Parameter c, s und r, die erfüllen:

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}$$

Zeige, unter der Bedingung $r > 0$ ist die Lösung (c, s, r) so eindeutig bestimmt:

$$r = \sqrt{a^2 + b^2}, \quad c = a/r, \quad s = b/r.$$

31. Implementiere die in der Vorlesung gegebenen Pseudo-Codes, um den QR-Algorithmus zur Bestimmung der Eigensysteme einer gegebenen SPD Matrix $A \in \mathbb{R}^{n\times n}$ durchzuführen. D.h. $A = P\tilde{A}P^{\mathrm{T}}$ soll zuerst mittels Householder Transformationen auf tridiagonale Form $\tilde{A} \to A^{(1)}$ transformiert werden. Anschliessend sollen QR-Zerlegungen $R^{(k-1)}Q^{(k-1)} =: A^{(k)} = Q^{(k)}R^{(k)}$ für tridiagonale Matrizen $A^{(k)}$ mittels Givens Transformationen durchgeführt werden, bis $A^{(k)}$ ausreichend diagonal ist.

32. Zeige, wenn die Werte $\{x_i\}_{i=0}^{m-1}$ verschieden sind, ist die Vandermonde Matrix $V^{\mathrm{T}}V$ SPD, wobei $V = [\boldsymbol{e}, \boldsymbol{x}, \ldots, \boldsymbol{x}^{n-1}]$ die Vandermonde Matrix ist.

33. Zeige, für lineare Regression

$$E(a,b) = \sum_{i=0}^{m-1} [y_i - (a + bx_i)]^2$$

ist die minimierende Lösung explizit und eindeutig so gegeben:

$$a^\star = \bar{\boldsymbol{y}} - b^\star \bar{\mathbf{x}}, \quad b^\star = \frac{\bar{\boldsymbol{x}\boldsymbol{y}} - \bar{\boldsymbol{x}}\bar{\boldsymbol{y}}}{\bar{\boldsymbol{x}^2} - \bar{\boldsymbol{x}}^2} \quad \text{wobei z.B. } \bar{\boldsymbol{x}\boldsymbol{y}} = \frac{1}{m}\sum_{i=0}^{m-1} x_i y_i$$

wenn die Werte $\{x_i\}_{i=0}^{m-1}$ verschieden sind.

34. Konstruiere die Hermite Basis Funktionen $\{\phi_i\}_{i=0}^N$, $\{\psi_i\}_{i=0}^N$ für ein regelmäßiges Gitter, wobei ϕ_i und ψ_i den Träger in $[x_{i-1}, x_{i+1}]$ haben.

35. Sei ein nicht notwendigerweise regelmäßiges Gitter $\{x_i\}_{i=0}^N$ gegeben. Die spline Basis-Funktion $s_{i,k}(x) \in \mathcal{S}^k(\boldsymbol{x})$ hat den Träger in $[x_i, x_{i+k+1}]$. Basierend auf Glattheitsbedingungen schreibe das Gleichungssystem für die Koeffizienten von $s_{i,k}(x)$ für $k = 1$, 2 und 3.

36. Gegeben seien zu interpolierende Daten $\{(x_i, f_i)\}_{i=0}^N$. Schreibe einen Matlab-Code für die Interpolation,

   (a) mit linearen Splines,

   (b) mit kubischen Splines durch die Randbedingung $s'(x) = 0$ eingeschränkt,

   (c) mit kubischen Splines durch die Randbedingung $s''(x) = 0$ eingeschränkt, und

   (d) mit der Hermite Basis. (Hinweise: $f'(x_i)$ kann mit den Werten $\{f_i\}$ approximiert werden.)

37. Sei $g \in \mathcal{C}^1([a,b])$ mit $g([a,b]) \subset [a,b]$. Angenommen $\exists\gamma \ni |g'(x)| \le \gamma < 1, \forall x \in (a,b)$. Zeige

$$|x^\star - x_k| \le \frac{\gamma^k}{1-\gamma}|x_1 - x_0|, \quad k \ge 1$$

wobei $x^\star$ der eindeutige Fixpunkt in $[a,b]$ ist.

38. Definiere $g(x) = xe^{r(1-x)}$, $r \ge 1$, $f(x) = x - g(x)$. Für $r = 1$, finde $\epsilon_0 \ni \forall\epsilon \in (0, \epsilon_0)$

   (a) $g(B(1,\epsilon)) \subset B(1,\epsilon)$ und

   (b) $\exists\gamma \in (0,1) \ni |g'(x)| \le \gamma, \forall x \in B(1,\epsilon)$.

   Erkläre die Konsequenzen für die Fixpunkt-Iteration $x_{k+1} = g(x_k)$.

39. Sei $f(x)$ eine Funktion mit Nullstelle der Multiplizität $m \ge 1$. Zeige asymptotisch quadratische Konvergenz für die Fixpunkt-Funktion $g_m(x) = x - mf(x)/f'(x)$.

40. Für das Abstiegsverfahren mit der Fixpunkt-Funktion,

$$\boldsymbol{G}_{\mathrm{A}}(\boldsymbol{u}) = (1-\alpha)\boldsymbol{u} + \alpha\boldsymbol{G}(\boldsymbol{u}), \quad \boldsymbol{G}(\boldsymbol{u}) = \boldsymbol{v} - \frac{\mu}{h^2}D(\boldsymbol{u})\boldsymbol{u},$$

$$D(\boldsymbol{u}) = \begin{bmatrix} D_1 & -D_1 & & & \\ -D_1 & D_1 + D_2 & -D_2 & & \\ & \ddots & \ddots & \ddots & \\ & & -D_{N-1} & D_{N-1} + D_N & -D_N \\ & & & -D_N & D_N \end{bmatrix}, \quad D_i = \left[\left(\frac{u_i - u_{i-1}}{h} \right)^2 + \epsilon^2 \right]^{-\frac{1}{2}}$$

zeige für $\alpha \in (0,1)$, es gilt $\boldsymbol{G}_{\mathrm{A}}(D) \subset D$ für $D = B_\infty(\boldsymbol{v}, \frac{2\mu}{h})$ und $\|\boldsymbol{G}'_{\mathrm{A}}(\boldsymbol{u})\|_\infty \leq 1 - \alpha + \frac{4\alpha\mu}{\epsilon h^2}$, $\forall \boldsymbol{u} \in D$.

41. Für die implizite Fixpunkt Iteration mit der Fixpunkt-Funktion,

$$\boldsymbol{G}_{\mathrm{I}}(\boldsymbol{u}) = \left[I + \frac{\mu}{h^2} D(\boldsymbol{u}) \right]^{-1} \boldsymbol{v}$$

zeige es gilt $\boldsymbol{G}_{\mathrm{I}}(D) \subset D$ für $D = B_2(0, \|\boldsymbol{v}\|_2)$ und $\|\boldsymbol{G}'_{\mathrm{I}}(\boldsymbol{u})\|_2 \leq \frac{16\mu}{(\epsilon^3 h^4)} \|\boldsymbol{v}\|_2^2$, $\forall \boldsymbol{u} \in D$.

42. Leite ein 5×5 System für die Koeffizienten $\{a_{-2}, \ldots, a_2\}$ her,

$$f''(x) = \sum_{k=-2}^{2} a_k f(x + kh) + \mathcal{O}(h^r)$$

um die Genauigkeit $r = 4$ zu erreichen.

43. Leite die Zusammengesetzte Trapez-Regel her:

$$\int_a^b f(x) dx = \frac{h}{2} \left[f(a) + 2 \sum_{j=1}^{m-1} f(x_j) + f(b) \right] + \mathcal{O}(h^2)$$

wobei $x_j = a + jh$, $j = 0, \ldots, m$, $h = (b - a)/m$.

44. Zeige, die Genauigkeitsordnung vom Crank-Nicholson Verfahren ist $n = 2$.

45. Das Sturm-Liouville Randwertproblem

$$\begin{cases} -(pu')' + qu &= f, \quad x \in \Omega \\ \alpha u + \omega u' &= g, \quad x \in \partial\Omega \end{cases} \qquad \Omega = (0, 1)$$

wird mit linearen Splines $\mathcal{S}^1(\boldsymbol{x})$ auf dem Gitter $\boldsymbol{x} = \{x_i\}$, $x_i = a + hi$, $i = 0, \ldots, N$ und $h = 1/N$ approximiert.

(a) Für die Dirichlet Randbedingung, $u = g$, leite ein lineares Gleichungssystem, $A\boldsymbol{a} = \boldsymbol{b}$, $u(x) = \sum_{i=0}^{n} a_i s_i(x)$, mit den Daten

$$p = q = f = 1, g = g_i \text{ in } [x_{i-1}, x_i), i = 1, \ldots, N.$$

her.

(b) Für die Neumann Randbedingung, $u' = 0$, leite ein lineares Gleichungssystem, $A\boldsymbol{a} = \boldsymbol{b}$, $u(x) = \sum_{i=0}^{n} a_i s_i(x)$, mit den Daten

$$p = p_i, q = q_i, f = f_i \text{ in } [x_{i-1}, x_i), i = 1, \ldots, N.$$

her.