Serie 8

1. Das “klassische” vierstufige Runge–Kutta–Verfahren (RK) der Ordnung $m = 4$ ist durch untenstehende Koeffiziententabelle gegeben:

$$\begin{array}{c|cccc}
0 & 0 & & & \\
\frac{1}{2} & \frac{1}{2} & 0 & & \\
\frac{1}{2} & 0 & \frac{1}{2} & 0 & \\
1 & 0 & 0 & 1 & 0 \\
\hline
 & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6}
\end{array}$$

a) Formulieren Sie dieses Verfahren mittels des Anfangswertproblemes (AWP)

$$\frac{d}{dt}y(t) = f(t, y) \quad , \quad y(0) = y_0. \tag{1}$$

b) Wenden Sie das Verfahren RK auf die genäherte Berechnung der speziellen Stammfunktion

$$Q(t) = \int_a^t f(\tau) d\tau$$

von f an.
Hinweis: Differentialgleichung und Anfangsbedingung für $Q(t)$ aufstellen.

2. (Extrapolation) Sei $y(t_f)$ die exakte Lösung des AWP (1) an der festen Stelle t_f, und sei $y_{RK}(t_f; h)$ die durch das Verfahren RK mit Schritt h erzeugte Näherungslösung von (1) an dieser Stelle. Allgemein gilt für den globalen Fehler $y_V(t; h) - y(t)$ eines Verfahrens V (asymptotisch für kleine h)

$$y_V(t; h) - y(t) = G(t)h^m + \mathcal{O}(h^{m+1}), \tag{2}$$

wobei m die Ordnung von V ist. Es seien nun die zwei Näherungen $Y_1 := y_{RK}(t_f; h_1)$ und $Y_2 := y_{RK}(t_f; h_2)$ aus Integrationen mit den verschiedenen Schrittlängen h_1 und h_2 bekannt. Berechnen Sie auf Grund von (2) einen neuen, besseren Näherungswert $y^\star(t_f)$ für $y(t_f)$ aus den Näherungen Y_1, Y_2: Kombinieren Sie dazu die beiden Gleichungen für die globalen Fehler von Y_1, Y_2 so, dass

$$y^\star(t_f) - y(t) = \mathcal{O}(h^p)$$

gilt mit $p > m$. Nehmen Sie an, zur Berechnung der Näherungen sei ein Verfahren RK der Ordung $m = 4$ verwendet worden, wobei die Schrittweiten $h_2 = h_1/2$ erfüllen.

Bitte wenden!

3. (**PA**: Konjugierte Gradienten mit Vorkonditionierung) Wir betrachten nochmals den cg–Algorithmus aus Serie 7, Aufgabe 2. Die Funktion `cg(A,b)` soll nun mit einer Matrix A im "sparse matrix"–Format von MATLAB aufgerufen werden.

a) (Vorkonditionierung) Nun soll der cg–Algorithmus mit der symmetrisch–positiv–definiten (spd) Matrix M vorkonditioniert werden. Dabei wird M nicht explizit definiert; es wird lediglich die Abbildung

$$r \mapsto z = M^{-1} r \tag{3}$$

gebraucht. Eine gut funktionierende Vorkonditionierung, mit welcher obige Abbildung schnell durchführbar ist, ist die *unvollständige Cholesky–Zerlegung*:

(i) Die (vollständige) Cholesky–Zerlegung $A = R^T R$ mit der Rechtsdreiecksmatrix R existiert für spd–Matrizen A (symmetrische Variante der LU–Zerlegung). In MATLAB: `R = chol(A)`.

(ii) Im Allgemeinen enthalten die Cholesky–Faktoren R^T, R einer schwach besetzten Matrix A viel mehr von Null verschiedene Einträge als A (*fill in*).
Bemerkung: `spy(A)` zeigt in MATLAB die Anordnung der Elemente $\neq 0$ von A.
Die unvollständige Cholesky–Zerlegung (`C = cholinc(A,'0')`, A, C im sparse–Format) besteht nun darin, die Elemente der Cholesky–Faktoren nur an den Stellen der von Null verschiedenen Elemente von A zu registrieren. Man erhält damit nicht A, sondern eine Näherung $M = C^T C$ von A nach Cholesky faktorisiert. Die Abbildung (3) ist dank der schwachen Besetzung von C durch Vor– und Rückwärtseinsetzen schnell durchführbar. Dies wird durch MATLAB–Befehle wie `y = C\r`, usf. automatisch geleistet.

Modifizieren Sie Ihre `cg(A,b)` Funktion so, dass A mit unvollständiger Cholesky–Zerlegung vorkonditioniert wird.

b) Betrachten Sie nun die Matrix A aus Serie 4, Aufgabe 3 mit $n = 10$ (also mit $N = 171$ Unbekannten). Das lineare Gleichungssystem $Au = b$ approximiert die Poisson–Gleichung $\triangle u = \varrho$ im Rechteck mit dem 5–Punkte–Differenzenstern, wobei Null–Randwerte vorgeschrieben sind. Der Vektor b enthält zeilenweise die Werte der Ladungsdichte ϱ. Bestimmen Sie die Spannung u zur Ladungsdichte $\varrho(x,y) = 1$ und plotten Sie die Spannungsverteilung (siehe dazu die Hinweise in Serie 4). Beobachten Sie auch das Konvergenzverhalten des Algorithmus.