Thema:
Berechnung des Eigenvektors zum größten Eigenwert einer stochastischen blocktridiagonalen Matrix

Betreuer: M. Pester

Beschreibung:

Bei einer aus der Teilchenphysik stammenden Simulationsaufgabe werden bis zu 110 Teilchen mit jeweils bis zu 1000 möglichen Energiezuständen (bzw. Zustandsintervallen) betrachtet.

Mit Zufallszahlen wird der Übergang von einem Zeitpunkt zum nächsten simuliert, wobei jeweils (höchstens) ein Teilchen mehr oder weniger existiert. Die Matrix „zählt“ zunächst die Übergänge von Energiezuständen.

Bei n Teilchen im Zeitschritt t und m Teilchen im Zeitschritt $t+1$, wobei $\max(0, n-1) < m < \min(N, n+1)$, wird im Block A_{mn} ein Element um 1 erhöht, das dem Übergang von einem Energiezustandsintervall zum anderen entspricht (N ist die maximale Teilchenzahl, NB die Zahl der möglichen Zustände).

Wegen $|n-m| \le 1$ ist diese Matrix blocktridiagonal. Die Simulation bewirkt außerdem, dass jeder Block schwachbesetzt und diagonaldominant ist. Da es bei dieser statistischen Zählung um die Bestimmung von Wahrscheinlichkeiten geht, werden am Ende die Spalten der Matrix auf Spaltensumme 1 normiert und die Matrixelemente sind nichtnegativ (d.h. $a_{ij} \ge 0,\ \sum_i a_{ij} = 1$). Folglich hat diese Matrix als betragsgrößten Eigenwert $\lambda_{max} = 1$. ($\rightarrow$ Perron-Frobenius-Theorem).

Da alle anderen Eigenwerte im Bereich $|\lambda| < 1$ liegen (und viele Eigenwerte sehr nahe bei 1), konvergiert bspw. die Potenzmethode sehr schlecht.

Aufgabe:

Die Blocktridiagonalmatrix muss platzsparend gespeichert werden, d. h. bei N Blöcken mit je NB×NB Elementen benötigt man nur ein (3·NB)×(N·NB)-Array, in dem noch 2 unbenutzte „Reserveblöcke“ enthalten sind (zwei Blockspalten enthalten nur je 2 Blöcke).

$$A = \begin{bmatrix} \square & \square & & & & \\ \square & \square & \square & & & \\ & \square & \square & \square & & \\ & & \ddots & \ddots & \ddots & \\ & & & \square & \square & \square \\ & & & & \square & \square \end{bmatrix}, \quad \text{bzw.} \quad R = \begin{bmatrix} \square & \square & \square & & & \\ & \square & \square & \square & & \\ & & \ddots & \ddots & \ddots & \\ & & & \square & \square & \square \\ & & & & \square & \square \\ & & & & & \square \end{bmatrix} \longrightarrow \begin{bmatrix} \square\square & \cdots & \square\square \\ \square\square & \cdots & \square\square \\ \square\square & \cdots & \square\square \end{bmatrix}$$

Da der Eigenwert $\lambda = 1$ bereits bekannt ist, kann der Eigenvektor als Lösung des homogenen Gleichungssystems $(A - I)x = 0$ bestimmt werden durch eine QR-Zerlegung (oder QL): $A - I = QR$.

Die Lösung vereinfacht sich weiter (da Q orthogonal, also invertierbar):

$$(A - I)x = 0 \quad \Rightarrow \quad QR\,x = 0 \quad \Rightarrow \quad R\,x = 0 \quad (= Q^\top 0)\,.$$

Die orthogonale Matrix Q muss also nicht gespeichert werden. Bei Betrachtung der QR-Zerlegung erkennt man, dass die Dreiecksmatrix R ebenfalls eine Blockstruktur besitzt (Diagonalblöcke und zwei darüber liegende Blöcke). Sie kann also im Speicher die ursprüngliche Blocktridiagonalmatrix überschreiben, ggf. mit bis zu 3 „Hilfsblöcken“ als Zwischenspeicher.

Da $(A - I)$ eine singuläre Matrix ist, muss das unterste Diagonalelement von R gleich Null sein (d.h. praktisch $r_{nn} \approx$ Maschinenepsilon). Beim Rückwärtseinsetzen sollte man also z.B. mit $x_n = 1$ beginnen, um ein $x \ne 0$ zu erhalten (aber: Überlaufrisiko !?)

Die Berechnung des Eigenvektors soll als Unterprogramm geschrieben werden, wobei die Matrix A, deren Blockanzahl N und Blockgröße NB als Input-Parameter übergeben werden und der Eigenvektor x als Output-Parameter zurückgegeben wird.

Zum Testen muss in einem separaten Unterprogramm vorher eine Matrix mit den o.g. Eigenschaften generiert werden (Zufallszahlen, Diagonaldominanz sichern, Spalten normieren).

Zur Beurteilung des Ergebnisses wäre eine grafische Anzeige der Eigenvektorkomponenten geeignet (Diagramm: $i \rightarrow x_i$), bzw. dasselbe für den „Defekt“ $z = Ax - x$. Die grafische Anzeige kann auch außerhalb des Programms mit anderen Werkzeugen wie Gnuplot oder Matlab erfolgen, wenn das Testprogramm den Vektor in eine Datei schreibt.

Die QR-Zerlegung:

Die Elimination einer Spalte (unterhalb des Diagonalelementes) erfolgt durch eine Householdertransformation $(I - 2vv^\top)$, wobei sich der Spiegelungsvektor v aus der aktuellen (k-ten) Spalte von A folgendermaßen berechnet:

$$s := -\mathrm{sign}(a_{kk})\sqrt{\sum_{i=k}^{n} a_{ik}^2} \qquad \text{(Spaltensumme unterhalb der Diagonale)}$$

$$\begin{aligned}
v_j &:= 0 \qquad (j = 1, \ldots, k-1) \\
v_k &:= \sqrt{\frac{1}{2}\left(1 - \frac{a_{kk}}{s}\right)}, \qquad w := -\frac{1}{2sv_k} \\
v_j &:= wa_{jk}, \qquad j = k+1, \ldots, n
\end{aligned}$$

Die Spiegelungsmatrix $Q_k = Q_k^\top = I - 2vv^\top$ wird nicht berechnet, sondern lediglich auf die restlichen Spalten der Matrix A angewendet, d.h. $\forall \ell \geq k$:

$$a_{*\ell} := (I - 2vv^\top)a_{*\ell} = a_{*\ell} - 2(v^\top a_{*\ell})v,$$

d. h. pro Spalte nur ein Skalarprodukt und eine Vektoraddition, wobei jeweils Rechenoperationen eingespart werden können, weil der Vektor v erst ab Index k von Null verschieden ist.

„Normalerweise" werden bei der QR-Zerlegung die Spiegelungsvektoren v_k im unteren Dreieck der Matrix A und der Faktor R im oberen Dreieck gespeichert. In diesem speziellen Fall kann man auf das Aufheben der v_k verzichten.

Dieser Algorithmus soll an die spezielle Blocktridiagonalstruktur angepasst werden, wobei für die Blöcke von R auch der ursprüngliche Speicher von A wiederverwendbar sein sollte, sobald die Elemente von A nicht mehr benötigt werden.

Bei der Festlegung der Datenstrukturen ist auf einen möglichst effektiven Zugriff zu achten, denn bei einer Blockgröße NB= 1000 ergeben sich in den Spalten der Matrix Teilvektoren der Länge 3000, so dass sich die Verwendung von BLAS-Routinen für die Vektoroperationen durchaus lohnen kann.

Zusätzlich kann untersucht werden, ob mit OpenMP eine teilweise Parallelisierung der Rechnung möglich ist.