Richard Huber
richard.huber@uni-graz.at

8. Aufgabenblatt

Definition (LR-Zerlegung mit Permutationen)**.** *Wenn für die LR-Zerlegung andere Pivot-Strategien als die Diagonalstrategie genutzt werden, müssen ggf. die Zeilen der Matrix A (bzw. dem Zwischenergebnis $A^{(k)}$) vertauscht werden (sodass das Pivot-Element in der aktuellen Pivot-Zeile liegt) um Stufenform zu erhalten. Dies hat den Vorteil, dass man mit passenden Strategien für reguläre Matrizen stets eine Zerlegung finden kann, allerdings den Nachteil dass man die genutzten Permutationen berücksichtigen muss. So erhält man $PA = LR$, (mit Dreiecksmatrizen L, R und Permutationsmatrix P) wobei die Berechnung analog zur LR-Zerlegung mit Diagonalstrategie erfolgt. Genauer, ein Zwischenergebnis (vor dem $k+1$ ten Eliminations-Schritt) der Rechnung sieht wie folgt aus:*

$$P^{(k)} \cdots P^{(1)} A = \begin{pmatrix} 1 & & & & & \\ l_{21} & \ddots & & & & \\ \vdots & \ddots & \ddots & & & \\ l_{k1} & \dots & l_{kk} & 1 & & \\ \vdots & \vdots & \vdots & & \ddots & \\ l_{N1} & \dots & l_{Nk} & & & 1 \end{pmatrix} \begin{pmatrix} r_{11} & \dots & \dots & \dots & \dots & r_{1N} \\ & \ddots & \vdots & \vdots & \vdots & \vdots \\ & & r_{kk} & \dots & \dots & r_{kN} \\ & & & a^{(k)}_{k+1,k+1} & \dots & a^{(k)}_{k+1,N} \\ & & & \vdots & \ddots & \vdots \\ & & & a^{(k)}_{N,k+1} & \dots & a^{(k)}_{N,N} \end{pmatrix}.$$

und falls man Zeile in $A^{(k)}$ vertauscht, so müssen auch die l_{ij} Einträge in L zeilenmäßig vertauscht werden, aber nur für die ersten k Spalten (also unterhalb der Diagonale). Die Matrix $P = P^{(n-1)} \cdots P^{(1)}$ enthält in jeder Zeile und Spalte genau einen 1er und kann in effizienter Vektorschreibweise gespeichert werden, d.h.. man beginnt mit dem Vektor $\tilde{p} = [1, 2, \dots, n]$ und vertauscht die Einträge entsprechend wenn eine Zeile getauscht wird. Dann hat $\tilde{p}$ am Ende die Form sodass $P_{i,\tilde{p}_i} = 1$ und $P_{i,j} = 0$ sonst.

Aufgabe 1 (Diagonalstrategie für symetrische, positiv definite Matrizen)**.** *Eine Matrix $A = (a_{ij}) \in \mathbb{R}^{N\times N}$ heißt symmetrisch falls $a_{ij} = a_{ji}$ für $i, j \in \{1, \dots, N\}$ und positiv definit falls $x^T Ax > 0$ für alle $x \in \mathbb{R}^N \setminus \{0\}$. Insbesondere gilt für symmetrisch positiv definite (kurz SPD) Matrizen A, dass $a_{ii} > 0$ für $i = 1, \dots, N$ und A ist regulär. Im Folgenden zeigen wir dass für SPD Matrizen die Diagonalstrategie stets zulässig ist.*

a) *Zeigen Sie, dass*

$$x^T Ax = \sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}x_ix_j = \left(\sum_{i=1}^{N} C_i^{(1)} x_i\right)^2 + \sum_{i=2}^{N}\sum_{j=2}^{N} a_{ij}^{(1)} x_i x_j = (C^{(1)}\cdot x)^2 + \bar{x}^T A^{(1)}\bar{x}, \quad (1)$$

wobei $C_i^{(1)} = \frac{a_{1i}}{\sqrt{a_{11}}}$ für $i = 1, \dots, N$ und $a_{ij}^{(1)} = a_{ij} - \frac{a_{1j}a_{i1}}{a_{11}}$, $\bar{x} = (x_2, \dots, x_N)$ und $A^{(1)} = (a_{ij}^{(1)})_{i,j=2}^N \in \mathbb{R}^{N-1\times N-1}$ die entsprechende Teilmatrix ist, mit der nach dem Pivot-Schritt weiter gerechnet wird.

b) *Folgern Sie, dass A SPD ist genau dann wenn $A^{(1)}$ SPD ist und $a_{i1} = a_{1i}$ für $i = 1, \dots, N$.*

c) *Folgern Sie, dass für A SPD die Diagonalstrategie durchführbar ist (also keine Null-Einträge als Pivot-Element gewählt werden).*

Hinweis. *Für b) nutzen Sie (1) um die Positivität $x^T Ax$ und $\bar{x}^T A^{(1)}\bar{x}$ in Zusammenhang zu bringen. Außerdem können Sie x_1 stets so wählen dass $C^{(1)} \cdot x = 0$ (Warum?!) und* *unterscheiden Sie die Fälle $x_1 = 0$ und $x_1 \neq 0$. Für c) müssen Sie ausnutzen, dass die Diagonaleinträge einer SPD Matrix positiv sind und induktiv vorgehen.*

Bemerkung. *Viele in der Praxis relevante Matrizen sind positiv definit – was eine sehr starke Eigenschaft ist. SPD Matrizen sind stets regulär und darüber hinaus lässt sich stets die Diagonalstrategie verwenden, wodurch man ohne Zeilenvertauschungen und Permutationsmatrizen auskommt. Darüber hinaus haben solche Matrizen spezielle LR-Zerlegungen (Cholesky-Zerlegung) sodass $A = CC^T$ mit C unterer Dreiecksmatrix, deren Spalten mit $C^{(k)}$ übereinstimmt.*

Aufgabe 2 (Gauß mit relativer Spaltenmaximumstrategie)**.** *Gegeben sei*

$$A = \begin{pmatrix} 2 & -4 & 6 & 2 \\ 3 & -6 & 10 & -4 \\ 1 & 3 & 13 & -6 \\ 0 & 5 & 11 & -6 \end{pmatrix} \qquad \text{und} \qquad b = \begin{pmatrix} 3 \\ -2 \\ 3 \\ -5 \end{pmatrix}.$$

Die relative Spaltenmaximumstrategie für eine Matrix $A \in \mathbb{R}^{N\times N}$ besteht darin, das k-te Pivot-Element $a_{i,k}^{(k-1)}$ so zu wählen, dass es maximalen Wert $\frac{|a_{i,k}^{(k-1)}|}{\sum_{j=k}^N |a_{i,j}^{(k-1)}|}$ unter allen verbleibenden nicht-Pivot Zeilen $k \leq i \leq N$ besitzt. Gegebenenfalls werden dann Zeilen so vertauscht, dass das Pivot-Element auf der Diagonale liegt, was der Permutationsmatrix $P^{(k)}$ entspricht.

a) *Berechnen Sie die LR-Zerlegung der Matrix A mittels relativer Spaltenmaximumstrategie. Geben Sie dabei L, R und P explizit an.*

b) *Nutzen Sie L, R und P um das Gleichungssystem $Ax = b$ mittels Vorwärtssubstitution und Rücksubstitution zu lösen.*

Bemerkung. *Pivot-Elemente die sehr klein sind können zu numerischen Problemen führen, da Division durch diese Elemente instabil sein kann, insbesondere wenn aufgrund von Rundungsfehlern nicht klar ist ob das Pivot-Element klein oder tatsächlich Null ist (bspw. durch Auslöschung). Da die Division sich auf alle Einträge der selben Zeile auswirkt ist es daher sinnvoll das Verhältnis des Pivot-Elements zu den restlichen Einträgen der Zeile zu betrachten, was mittels dem Quotienten $\frac{|a_{i,k}^{(k-1)}|}{\sum_{j=k}^N |a_{i,j}^{(k-1)}|}$ möglich ist.*

Matlab-Aufgabe 3 (Gauß-Elimination mit Permutationen)**.** *In dieser Aufgabe wollen wir das Programm aus Aufgabe 7.2 mit Pivot-Strategien erweitern. Schreiben Sie eine Matlab-Funktion $[A_k, L, P, flag]$=Matrixpermutation(A_k, L, P,strat) welche in Abhängigkeit von "strat" ein Pivot-Element aus der 1. Spalte von A_k auswählt, und die Zeilen der Matrizen $A_k \in \mathbb{R}^{n-k\times n-k}$ und $L \in \mathbb{R}^{n\times n}$ entsprechend vertauscht (sodass das Pivot-Element in der 1. Zeile steht), sowie P (die Permutationsgeschichte) entsprechend updatet. (hierbei steht A_k stellvertretend für $A^{(k)}$ und $P = P^{(k)} \cdots P^{(1)} \approx \tilde{p}$.)*

a) *Schreiben Sie die Matlab-Funktion Matrixpermutation sodass für strat='spalten' die Spaltenmaximunsstrategie anwendet wird um das nächste Pivot-Element der Matrix A_k zu wählen, und die Matrizen A_k, L und der Spaltenvektor P entsprechend zu permutieren, sowie die flag auf -1 zu setzen (und das Programm abzubrechen) falls das gewählte Pivot-Element betragsmäßig kleiner als 10^{-5}. (Spaltenmaximusstragie heißt der* *betragsmäßig größter Eintrag der Spalte wird das nächste Pivot-Element). Außerdem soll für strat='diagonal' die Diagonalstrategie angewendet werden (dann sind keine Vertauschungen notwendig).*

b) *Erweitern Sie die Matrixpermutation sodass mit strat='relative' die relative Spaltenmaximumsstrategie genutzt wird um das Pivot-Element zu wählen und entsprechende Permutationen anwendet. Insbesondere soll das Verfahren auch abgebrochen werden und flag*$= -1$ *gesetzt werden, falls* $\sum_{j=k}^{N} |a_{i,j}^{(k-1)}| < 10^{-5}$ *für alle* $i = k, \ldots, N$.

c) *Erweitern Sie Ihre Programme aus Aufgabe 7.2 sodass* [L, R, P,*flag*] = *my_LR(A,strat) die entsprechende Pivot-Strategie nutzt. Erweitern Sie auch [x, flag]=my_gauss(A, y,strat) sodass mit der entsprechenden LR (und P) Zerlegung das Gleichungssystem gelöst wird.*

Schreiben Sie ein Testskript um sich von der Richtigkeit Ihrer Programme zu überzeugen.

Hinweis. *Sie können die Funktionen my_LR.m my_gauss.m backsubstitution.m forwardsubstitution.m Test.m benutzen falls Sie 7.2 nicht gelöst haben bzw. unzufrieden mit Ihren Lösungen sind. Schreiben Sie die Funktion Matrixpermutation so, dass Sie für* A_k *mit kleineren Dimensionen als L bzw. P funktioniert, und beachten Sie wie Sie dann Zeilen L und P vertauschen müssen. (alternativ können Sie auch die kompakte Schreibweise in einer Matrix nutzen wenn Sie wollen.) Achten Sie auch darauf, wie sich P auf die Vorwärtssubstitution bzw. Rücksubstitution auswirkt. Dabei könnte die Matlab Funktionen max und strcmp nützlich sein.*