Lernziele

In diesem Praktikum sollen Sie üben und lernen:

- Verschiedene Speicherformate für Matrizen

Praktikumsaufgabe 15 – Matrix-Vektorprodukte

Lesen Sie die Aufgabenstellung, studieren Sie dann die vorgegebenen Programmzeilen. Ersetzen Sie dann die %% Kommentare im vorgegebenen Code durch Matlab-Anweisungen und führen Sie das Programm aus.

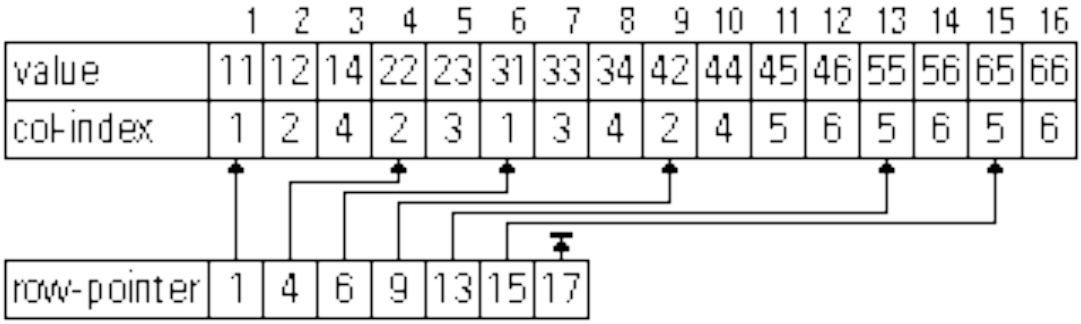
- **Problembeschreibung**

Sparse Matrizen müssen intern in speziellen Formaten kodiert werden. Benutzt man eine der zahlreichen Numerik-Bibliotheken mit eingebauten Routinen für die Rechnung mit sparsen Matrizen, so ist ein vertrauter Umgang mit diesen Formaten unerlässlich.Hier betrachten wir nun das sogenannte CRS-Format, welches die drei Arrays `val, col_ind, row_ptr` benutzt, um eine sparse Matrix zu speichern.

Das Format sei an einem Beispiel erklärt:

$$A := \begin{pmatrix} 11 & 12 & 0 & 14 & 0 & 0 \\ 0 & 22 & 23 & 0 & 0 & 0 \\ 31 & 0 & 33 & 34 & 0 & 0 \\ 0 & 42 & 0 & 44 & 45 & 46 \\ 0 & 0 & 0 & 0 & 55 & 56 \\ 0 & 0 & 0 & 0 & 65 & 66 \end{pmatrix}$$

... wird zu ...



Schreiben Sie eine Matlab-Funktion, die das Produkt der Matrix A mit einem Vektor x zurückgibt, wobei die Matrix A im sogenannten **compressed row-storage-Format (CRS)** gespeichert ist.

Der Funktionsaufruf lautet

```
% ex15x01.m
function ax = ex15x01(A,x)
```

wobei A eine Struktur mit den Feldern `m, n, value, colind` und `rowptr` ist.

In dem Beispiel sei somit:

```
A.m      = 6;
A.n      = 6;
A.value  = [11,12,14, ... ,56,65,66];
A.colind = [ 1, 2, 4, ... , 6, 5, 6];
A.rowptr = [1,4,6,9,13,15,17];
```

Praktikumsaufgabe 16 – Matrix-Vektorprodukte

Lesen Sie die Aufgabenstellung, studieren Sie dann die vorgegebenen Programmzeilen. Ersetzen Sie dann die %% Kommentare im vorgegebenen Code durch Matlab-Anweisungen und führen Sie das Programm aus.

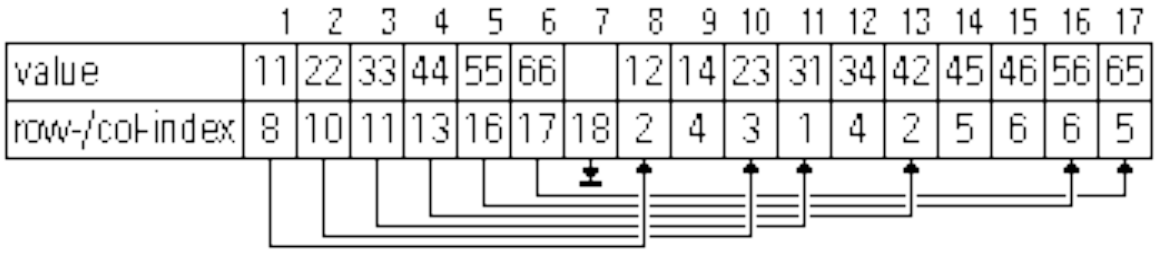
- **Problembeschreibung**

  Schreiben Sie eine Matlab-Funktion, die das Produkt der Matrix A mit einem Vektor x zurückgibt, wobei die Matrix A im sogenannten **modified compressed row-storage-Format** (MRS) gespeichert ist.

  Das Format sei an einem Beispiel erklärt:

$$A := \begin{pmatrix} 11 & 12 & 0 & 14 & 0 & 0 \\ 0 & 22 & 23 & 0 & 0 & 0 \\ 31 & 0 & 33 & 34 & 0 & 0 \\ 0 & 42 & 0 & 44 & 45 & 46 \\ 0 & 0 & 0 & 0 & 55 & 56 \\ 0 & 0 & 0 & 0 & 65 & 66 \end{pmatrix}$$

... wird zu ...



Der Funktionsaufruf lautet

```
% ex16x01.m
function ax = ex16x01(A,x)
```

wobei A ein Struktur mit den Feldern `m, n, value` und `index` ist.

In dem Beispiel ist somit

```
A.m = 6;
A.n = 6;
A.value = [11,22,33,44,55,66, 0,12,14, ... ,46,56,65];
A.ind =   [ 8,10,11,13,16,17,18, 2, 4, ... , 6, 6, 5];
```

Praktikumsaufgabe 17 –

Sparse Matrizen im CCS-Format

Lesen Sie die Aufgabenstellung, studieren Sie dann die vorgegebenen Programmzeilen. Ersetzen Sie dann die %% Kommentare im vorgegebenen Code durch Matlab-Anweisungen und führen Sie das Programm aus.

- **Problembeschreibung**

Matlab benutzt das sogenannte CCS-Format (Compressed Column Storage), welches gerade dem CRS-Format für die transponierte Matrix entspricht und die Arrays `val, row_ind, col_ptr` verwendet.

a) Lesen Sie nun zunächst auf den Matlab-Hilfeseiten den Matlab-Befehl `find` nach. Implementieren Sie dann die Matlab-Funktion

```
% CCS.m
function [val, row_ind, col_ptr] = CCS(A)
```

welche eine gegebene quadratische Matrix A im CCS-Format zurückgibt. Dabei kann davon ausgegangen werden, dass es in der Matrix A keine Spalten gibt, in denen alle Elemente gleich 0 sind. Testen Sie Ihr Programm zunächst für kleine Matrizen.

b) Vergleichen Sie die Laufzeiten Ihrer Implementierung von `CCS(A)` für Matrizen $A \in \mathbb{R}^{n \times n}$ im sparsen und vollbesetzten Format für $n = 2^{5,6,\ldots,12}$.

Benutzen Sie dazu die Matrix

```
A = [n+1, (n:-1:1); ones(n,1), eye(n,n)];
```

Verwenden Sie die Routinen `tic` und `toc`. Ihr Ergebnis müsste wie folgt aussehen: