Familienname:

Vorname:

Matrikelnummer:

Aufgabe 1 (6 Punkte):
Aufgabe 2 (2 Punkte):
Aufgabe 3 (4 Punkte):
Aufgabe 4 (2 Punkte):
Aufgabe 5 (8 Punkte):
Aufgabe 6 (4 Punkte):
Aufgabe 7 (3 Punkte):
Aufgabe 8 (1 Punkte):

Gesamtpunktzahl:

Schriftlicher Test zu Matlab (90 Minuten)
VU Einführung ins Programmieren für TM

17. Juni 2011

Aufgabe 1 (6 Punkte). Was macht die folgende MATLAB-Funktion mit einem Vektor $x \in \mathbb{R}^n$? Was ist die Bedeutung des Parameters $i \in \mathbb{R}$, d.h. was tut die Funktion in Abhängigkeit von i genau? Unterscheiden Sie die Fälle $i < 0$, $i = 0$ und $i > 0$!

```
function y = foobar(x,i)
    n = length(x);
    y = abs(x);
    for j = 1:n
        for k = 1:n-j
            % Zeitpunkt der Auswertung
            if i*y(k) < i*y(k+1)
                y([k k+1]) = y([k+1 k]);
            end
        end
    end
end
```

Tabellieren Sie zunächst die Werte von j, k, n und y zum markierten Zeitpunkt bei Aufruf der Funktion in der Form `y = foobar([1 -1 -2 3 5],-1)`? Was ist die abschließende Rückgabe der Funktion?

j	k	n	y

Aufgabe 2 (2 Punkte). Die MATLAB-Funktion `y = sort(x)` übernimmt einen Vektor $x \in \mathbb{R}^n$ und gibt ihn aufsteigend sortiert als $y \in \mathbb{R}^n$ zurück. Schreiben Sie eine Version der Funktion `foobar` aus der vorausgegangenen Aufgabe, die ohne `for`-Schleifen auskommt.

Aufgabe 3 (4 Punkte). Füllen Sie den folgenden Lückentext aus, sodass der Aufruf

```
result = isSorted(x)
```

für einen Vektor $x \in \mathbb{R}^n$ folgende Rückgabe liefert:

- `result =-1`, falls x unsortiert ist,
- `result = 0`, falls alle Einträge von x gleich sind,
- `result = 1`, falls x aufsteigend sortiert ist,
- `result = 2`, falls x absteigend sortiert ist.

```
function result = isSorted(x)
    result = 0;
    for i = 1:_______________-1
        if x(i) < x(i+1)
            if result == 2
                result = _______________;
                return
            end
            _______________ = _______________;
        elseif _______________
            if _______________
                _______________ = _______________;
                return
            end
            _______________ = _______________;
        end
    end
end
```

Aufgabe 4 (2 Punkte). Schreiben Sie eine Version von `isSorted`, die ohne `for`-Schleife auskommt. Benutzen Sie dazu die Vektorarithmetik in geeigneter Weise.

Aufgabe 5 (8 Punkte). Zu gegebenen reellen Stützstellen $x_1 < \cdots < x_n$ und Funktionswerten $y_j \in \mathbb{R}$ garantiert die Lineare Algebra ein eindeutiges Polynom $p(t) = \sum_{j=1}^{n} a_j t^{j-1}$ vom Grad $n-1$ mit $p(x_j) = y_j$ für alle $j = 1, \ldots, n$. Nun sei $t \in \mathbb{R}$ fixiert und $p(t)$ gesucht. Man kann $p(t)$ mit dem *Neville-Verfahren* berechnen, ohne zunächst den Koeffizientenvektor $a \in \mathbb{R}^n$ berechnen zu müssen: Dazu definiere man für $j, m \in \mathbb{N}$ mit $m \geq 2$ und $j + m \leq n + 1$ die Werte

$$
\begin{aligned}
p_{j,1} &:= y_j, \\
p_{j,m} &:= \frac{(t - x_j)p_{j+1,m-1} - (t - x_{j+m-1})p_{j,m-1}}{x_{j+m-1} - x_j}.
\end{aligned}
$$

Es gilt dann $p(t) = p_{1,n}$. Schreiben Sie eine Funktion `neville`, die den Auswertungspunkt $t \in \mathbb{R}$ sowie die Vektoren $x, y \in \mathbb{R}^n$ übernimmt und $p(t)$ mittels Neville-Verfahren berechnet. Dazu berücksichtige man das folgende schematische Vorgehen

$$
\begin{array}{cccccccccccccc}
y_1 & = & p_{1,1} & \longrightarrow & p_{1,2} & \longrightarrow & p_{1,3} & \longrightarrow & \cdots & \longrightarrow & p_{1,n} & = & p(t) \\
 & & & \nearrow & & \nearrow & & & & \nearrow & & & \\
y_2 & = & p_{2,1} & \longrightarrow & p_{2,2} & & & & & & & & \\
 & & & \nearrow & & & & \nearrow & & & & & \\
y_3 & = & p_{3,1} & \longrightarrow & \vdots & & & & & & & & \\
\vdots & & \vdots & & \vdots & \nearrow & & & & & & & \\
y_{n-1} & = & p_{n-1,1} & \longrightarrow & p_{n-1,2} & & & & & & & & \\
 & & & \nearrow & & & & & & & & & \\
y_n & = & p_{n,1} & & & & & & & & & &
\end{array}
$$

Schreiben Sie eine Funktion `pt = neville(x,y,t)`, die das Neville-Verfahren mittels geeigneter Schleifen realisiert und dazu die Matrix $(p_{j,m})_{j,m=1}^{n}$ vollständig aufbaut.

Aufgabe 6 (4 Punkte). Man kann das Neville-Verfahren

$$
\begin{aligned}
p_{j,1} &:= y_j,\\
p_{j,m} &:= \frac{(t-x_j)p_{j+1,m-1} - (t-x_{j+m-1})p_{j,m-1}}{x_{j+m-1}-x_j}
\end{aligned}
$$

mit

$$
\begin{array}{ccccccccccccc}
y_1 & = & p_{1,1} & \longrightarrow & p_{1,2} & \longrightarrow & p_{1,3} & \longrightarrow & \cdots & \longrightarrow & p_{1,n} & = & p(t)\\
 & & & \nearrow & & \nearrow & & & & \nearrow & & & \\
y_2 & = & p_{2,1} & \longrightarrow & p_{2,2} & & & & & & & & \\
 & & & \nearrow & & & & \nearrow & & & & & \\
y_3 & = & p_{3,1} & \longrightarrow & \vdots & & & & & & & & \\
\vdots & & \vdots & & \vdots & \nearrow & & & & & & & \\
y_{n-1} & = & p_{n-1,1} & \longrightarrow & p_{n-1,2} & & & & & & & & \\
 & & & \nearrow & & & & & & & & & \\
y_n & = & p_{n,1} & & & & & & & & & &
\end{array}
$$

so programmieren, dass zur Speicherung der Werte *keine* Matrix $(p_{j,m})_{j,m=1}^{n}$ aufgebaut wird, sondern der gegebene y-Vektor geeignet überschrieben wird. Realisieren Sie dieses Vorgehen als Funktion `pt = neville(x,y,t)`

Aufgabe 7 (3 Punkte). Modifizieren Sie den Code aus der vorausgegangenen Aufgabe so, dass er für Vektoren $x, y \in \mathbb{R}^n$ und $t \in \mathbb{R}^N$ den Vektor $p = \big(p(t_j)\big)_{j=1}^N \in \mathbb{R}^N$ der Auswertungen zurückgibt. Dabei soll keine weitere Schleife verwendet werden, sondern nur geeignete MATLAB-Arithmetik. Ferner soll Ihr Code sowohl für Spaltenvektoren x, y, t als auch für Zeilenvektoren aufrufbar sein und einen Zeilenvektor p zurückliefern.

Aufgabe 8 (1 Punkt). Man kann das Neville-Verfahren auch rekursiv programmieren? Ist das sinnvoll? Begründen Sie Ihre Antwort!