Familienname:

Vorname:

Matrikelnummer:

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

Gesamtpunktzahl:

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

03. Oktober 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([2 -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 = isIncreasing(x)
```

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

- `result =-1`, falls x unsortiert ist,
- `result = 0`, falls x aufsteigend sortiert ist, d.h. $x_\ell \leq x_{\ell+1}$ für alle ℓ,
- `result = 1`, falls x streng aufsteigend sortiert ist, d.h. $x_\ell < x_{\ell+1}$ für alle ℓ,
- `result = 2`, falls x exponentiell wachsend ist, d.h. $\sum_{j=1}^{\ell} x_j < x_{\ell+1}$ für alle ℓ.

```
function result = isIncreasing(x)
    result = 2;
    for i = ___________________________________
        if x(i-1) > x(i)
            ___________________________________
            return
        elseif result == 2
            if ___________________________________
                result = 1;
            elseif ___________________________________
                result = 0;
            end
        elseif ___________________________________
            if ___________________________________
                ___________________________________
            end
        end
    end
end
```

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

Aufgabe 5 (6 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

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

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

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

mit

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

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 (6 Punkte). Für gegebene differenzierbare Funktion $f : \mathbb{R} \to \mathbb{R}$ und $x \in \mathbb{R}$ approximiert der einseitige Differenzenquotient

$$\Phi(h) = \frac{f(x+h) - f(x)}{h}$$

die Ableitung $f'(x)$. Man kann das Neville-Verfahren dazu benutzen, durch geeignetes Postprocessing eine bessere Approximation von $f'(x)$ zu gewinnen: Für gegebenes $h > 0$ und $x_k := 2^{1-k}h$ sei $p_n(x)$ das eindeutige Interpolationspolynom vom Grad $n-1$ mit $p_j(x_k) = \Phi(x_k)$. Man kann zeigen, dass die Folge $p_n(0)$ dann „schneller“ gegen $f'(x)$ konvergiert als $\Phi(x_n)$. Schreiben Sie eine Funktion

```
p = richardson(f,x,h,N)
```

die dieses Vorgehen realisiert und den Vektor $p = \big(p_1(0), \ldots, p_N(0)\big) \in \mathbb{R}^N$ zurückgibt. Realisieren Sie die Funktion möglichst speicher- und rechenökonomisch, indem Sie das Schema

$$\begin{array}{lcccccccccccc}
\Phi(x_1) & = & p_{1,1} & \longrightarrow & p_{1,2} & \longrightarrow & p_{1,3} & \longrightarrow & \ldots & \longrightarrow & p_{1,N} & = & p_N(0) \\
 & & & \nearrow & & \nearrow & & & & \nearrow & & & \\
\Phi(x_2) & = & p_{2,1} & \longrightarrow & p_{2,2} & & & \nearrow & & & & & \\
 & & & \nearrow & & & & & & & & & \\
\Phi(x_3) & = & p_{3,1} & \longrightarrow & \vdots & & & & & & & & \\
\vdots & & \vdots & & \vdots & \nearrow & & & & & & & \\
\Phi(x_{n-1}) & = & p_{N-1,1} & \longrightarrow & p_{N-1,2} & & & & & & & & \\
 & & & \nearrow & & & & & & & & & \\
\Phi(x_N) & = & p_{N,1} & & & & & & & & & &
\end{array}$$

des Neville-Verfahrens mit

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

beachten. An welcher Stelle $p_{j,m}$ stehen die Werte $p_m(0)$? Bei geschickter Realisierung kommen Sie mit zwei Schleifen (d.h. quadratischem Aufwand) und zwei Vektoren $p, y \in \mathbb{R}^N$ aus (d.h. linearem Speicheraufwand) aus!