Inleiding programmeren en numerieke methoden

Exam

8/11-2011

For this exam no books or graphical electronic calculators are allowed.
Location: Tentamenhal 03 Blauwborgje 4, Time: 14:00-17:00.

1 MATLAB commands

Briefly explain what the following build-in MATLAB commands do. (Max. 5 lines for each command.) Write if the function is a simple MATLAB command, a numerical method command, or a symbolic toolbox command.

1. function
2. input
3. feval
4. while
5. fzero
6. rref
7. randn
8. subs
9. diff
10. eps

2 Debugging and Runge-Kutta

1. The following program should use Runge-Kutta method to solve the differential equation

$$\frac{dy}{dt} = (1-t)y \qquad (1)$$

defined in the function *funcrk*. At the end the program plots the solution compared to the exact solution and after the user presses a button it plots the relative error. However, the program does not work. Find the 6 bugs and correct them! (Line numbers are indicated on the left. You can use them to report the location of the bugs.)

```
 1:%
 2:% runge_kutta.m
 3:%
 4:t=0.0:0.1:5.0:                   % Define time interval
 5:delta_t=t(2)-t(1);               % Determine time step
 6:a=0.5;                           % Select free R-K parameter
 7:b=1-a; alpha=0.5/b; beta=0.5/b;  % Set values of the other R-K parameters
 8:n=length(t);                     % Find number of time points
 9:y=zeros(n);                      % Clear y vector
10:y(0)=1;                          % Set initial value
11:for ii=1:n-1                     % Solve differential equation
12:    k_1=delta_t*funcrk(t(ii),y(ii));
13:    k_2=delta_t*funcrk(t(ii)+alpha*delta_t,y(i)+beta*k_1);
14:    y(ii+1)=y(ii]+a*k_1+b*k_2;
15:end
16:exact=exp(t-t.^2/2);             % Calculate exact result
17:plot(t,y,’r’,t,exact,’b’)        % Plot Runge-Kutta solution and exact one
18:pause                            % Wait for user
19:diff=(exact-y)/exact;            % Find relative error
20:plot(t,diff)                     % Plot the relative error
```

2. Briefly explain the difference between the Runge-Kutta method and the Euler method.
3. Discuss the value of the parameter a used above and what happens if the parameter is set equal to 1.

3 Matrices

1. Write a code to construct the following matrix using one for loop and **not** using the *diag* command.

$$\begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ i & 0 & \sqrt{2} & 0 & \cdots & 0 & 0 \\ 0 & \sqrt{-2} & 0 & \sqrt{3} & \cdots & 0 & 0 \\ 0 & 0 & \sqrt{-3} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & \sqrt{10} \\ 0 & 0 & 0 & 0 & \cdots & \sqrt{-10} & 0 \end{bmatrix} \tag{2}$$

2. Construct the same matix using the *diag* command.
3. What are the commands for diagonalizing (finding the eigen values and eigen vectors) and inverting matrices numerically?

4 Build algorithm

1. Describe how the Newton-Raphson method works. You are encouraged to use a figure.
2. Make a step by step outline of an algorithm that use the Newton-Raphson method.
3. Write a code following your algorithm described in the point above, where you use one while loop. Assume that the name of the function that you want to find the root for is called *func* and the derivative of the function is provided by *der_func*.