AMSC/CMSC 666

NUMERICAL ANALYSIS I

FINAL TAKEHOME EXAM (due 12/14/19 noon)

1 (20 pts) *Basic 1-step methods.* Consider the following second order initial value problem modeling a spring-dashpot system:

$$y'' + 101y' + 100y = \sin t, \qquad y(0) = 2, y'(0) = 0. \tag{1}$$

(a) Solve this problem by hand. To this end find first the solution to the homogeneous equation (natural modes), and next a particular solution using the method of undetermined coefficients; note that the eigenvalues of the characteristic equation are integers. Explain whether the problem is *stiff* or not.

(b) Convert (1) into a first order system, and write the forward Euler (FE), backward Euler (BE) and Trapezoidal methods (TM).

(c) Write MATLAB programs that implement FE, BE, and TM with step-sizes $h = 10^{-k}$ for $k = 1, 2, 3$ on the interval $(0, 10)$.

(d) Find the error between computed solutions of (c) and exact solution of (a) at $t_n = nh$ and plot the results. Explain the results in terms of absolute stability, and draw conclusions.

2 (15 pts) *Runge-Kutta method.* Let $f(t, y)$ be a C^2 function with bounded second derivatives. Let $y(t)$ be the solution of the initial value problem

$$y' = f(t, y), \quad y(0) = y_0. \tag{2}$$

Consider the following implicit Runge-Kutta method with constant stepsize h:

$$y_{n+1} = y_n + hf\Big(t_n + \frac{h}{2}, \frac{1}{2}(y_{n+1} + y_n)\Big). \tag{3}$$

(a) Show that (3) has a unique solution for h sufficiently small.

(b) Define the truncation error τ_{n+1} and show that $\tau_{n+1} = O(h^2)$. Hint: obtain an expression for $y''(t)$ by differentiation of (2) and use Taylor expansion around $(t_n, y(t_n))$.

(c) Perform an error analysis and find the order of convergence.

(d) Consider the test equation $f(t, y) = \lambda y$ with $\lambda < 0$. Prove that (3) is absolutely stable for all $h > 0$. Is this method suitable for stiff problems? Justify your answer.

3 (20 pts) *Nystrom method.* Consider the following *multistep* method with constant stepsize h for (2):

$$y_{n+1} = y_{n-1} + \frac{h}{3}f(t_{n-2}, y_{n-2}) - \frac{2h}{3}f(t_{n-1}, y_{n-1}) + \frac{7h}{3}f(t_n, y_n). \tag{4}$$

(a) Derive (4) by writing (3) in the integral form

$$y(t_{n+1}) = y(t_{n-1}) + \int_{t_{n-1}}^{t_{n+1}} f(t, y(t))dt,$$

and approximating the integrand with a quadratic polynomial interpolating at $t = t_{n-2}, t_{n-1}, t_n$.

(b) Define truncation error for (4) and determine its order directly using Taylor expansion.

(c) Give a *direct* proof of convergence, thereby establishing the global rate of convergence of (4).

4 (15 pts) *Gear's Formula (BDF(2)).* Let the stepsize h be uniform. Interpolate (t_i, y_i) $(i = n-1, n, n+1)$ with a polynomial $p(t)$ of degree 2 and next set $p'(t_{n+1}) = f(t_{n+1}, y_{n+1})$.

(a) Deduce the following multistep backward differentiation formula (BDF)

$$y_{n+1} = \frac{4}{3}y_n - \frac{1}{3}y_{n-1} + \frac{2}{3}hf(t_{n+1}, y_{n+1}).$$

(b) Define truncation error τ_{n+1} and use Taylor's formula to prove that τ_{n+1} is $O(h^2)$. Even though this implies consistency, verify the consistency conditions directly.

(c) Study stability and convergence using the root condition. Conclude that this linear multistep method is of 2nd order, namely $\max_n \|y(t_n) - y_n\| = O(h^2)$.

(d) Verify absolute stability for all $\lambda \leq 0$. Explain whether or not BDF(2) is well-suited for *stiff* problems.

5 (15 pts). *Extra credit.* Solve Pb 1 with the following MATLAB built-in solvers:

- `ode15s`: this is similar to the Gear's multistep method or BDF;
- `ode113`: this is the multistep method which combines Adams-Bashforth and Adams-Moulton formulas;
- `ode23`: this is the single step method of Bogacki and Shampine of order 2-3.
- `ode45`: this is a single step method of Dormand and Prince similar to Runge-Kutta-Fehlberg (4-5).

The basic syntax is

```
[T,Y]=solver(@function,[t0 tf],y0,options),
```

where `solver` is any of the above solvers, `function` is an m file with the system of ODEs, `t0` is the initial condition, `tf` is the final condition, `y0` is the vector of initial values, `T` is a column vector of time points and `Y` is a solution array (see handout for details). The argument `options` sets, among other things, the relative and absolute accuracy, tells the solver if the Jacobian is constant in case of linear systems, and provides statistics of the computation useful to determine the computational cost.

Note that multistep methods are selfstarting, namely they do not need starting values; this is achieved by combining different methods of increasing order and adaptively selecting the mesh-size. Run the programs with relative accuracy `RelTol`$=10^{-2}$ and 10^{-4}, `JConstant=on` and `Stats=on`. The syntax could be

```
options = odeset('RelTol',1e-4,'JConstant',on,'Stats',on)
```

Compare the results and computational costs. Draw pictures of the computed and true solutions, and draw conclusions.