University of Colorado
Department of Computer Science

Numerical Computation CSCI 3656

Spring 2015

Exam 3

1. *[6 pts]* Consider the following chunk of Matlab code, which implements one of the ODE solvers that we have discussed in the past month.

```
function [trajectory] = solver(current_state,h,n)
trajectory=[];
for steps=1:n
    x=current_state(1,1);
    v=current_state(2,1);
    xdot=x+v;  % these two lines
    vdot=sin(x);  % are the evaluation of the ODE
    slope_vector=[xdot,vdot]';
    new_state = current_state + h * slope_vector;
    trajectory=[trajectory,new_state];
    current_state=new_state;
end
end
```

   Which solver method is implemented here?

2. *[6 pts]* What is the local truncation error of the Forward Euler method? What about the *global* truncation error of the trapezoidal method?

3. *[5 pts]* Is the midpoint method for solving ODEs an *implicit* or *explicit* method? Circle one and explain your answer.

4. *[8 pts]* (a) What's the algorithmic difference between *single-step* and *multi-step* methods for solving ODEs — i.e., how do they *work* differently? (We're not asking about the error here, or how many points they use, but the inner workings of the algorithms that they use to find the next state).

   (b) Compare the computational complexity of single-step and multi-step methods for solving ODEs.

5. *[6 pts]* Circle the correct answers to these questions regarding QR factorization and least squares.

   - These two techniques solve completely different problems T/F
   - Both of these techniques are useful for factoring matrices T/F
   - Both of these techniques are useful for solving problems where there are more unknowns than equations T/F
   - Both of these techniques are useful for solving problems where there are more equations than unknowns T/F
   - QR factorization is computationally more expensive than least squares, for the same problem T/F
   - The least squares method is more accurate than QR factorization for ill-conditioned problems T/F

6. *[6 pts]* What is a Householder reflector? What kinds of problems is it used for?

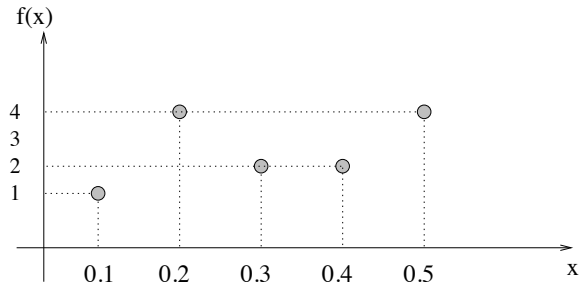
7. *[10 pts]* Using Forward Euler on this ODE system:

$$\begin{aligned} \dot{x}(t) &= x(t) - v(t) \\ \dot{v}(t) &= 2v(t) \end{aligned}$$

...compute $(x(t = 0.1), v(t = 0.1))$ starting from the initial condition $(x(t = 0), v(t = 0)) = (0, 1)$ with a time step of 0.1.

8. *[4 pts]* What is the *order* of the ODE system given in the previous problem? Is it *linear* or *nonlinear*? (circle one)

9. *[6 pts]* Draw arrows on the graph below representing the derivative $f'(0.3)$ computed using the simplest versions of the (a) forward (b) backward and (c) center difference methods. Please identify each arrow clearly.



10. *[6 pts]* Using the values in the graph above, compute *numerical* values for $f'(0.3)$ using those same three methods. Make sure that these values match up with the arrows that you drew.

11. *[4 pts]* Compute $f''(0.3)$ from those data using forward differences.

12. *[6 pts]* How does Romberg integration work? How is it similar to Richardson extrapolation?

13. *[5 pts]* How would you parallelize a 4th-order Runge Kutta solution of the ODE $\frac{dx}{dt} = \sin x$?

14. *[10 pts]* Compute $\int_1^{-1}(2x + 3)dx$ using two-term Gaussian quadrature. Show your work. (I'll put up Table 5.1 on the overhead projector.)

15. *[4 pts]* Why is it important to pay attention to how matrices are actually accessed in computer programs—e.g., how they are stored in the computer's memory?

16. *[4 pts]* What is the difference between motion capture and animation?

17. *[4 pts]* Tell us something interesting about chaos.