MATH 567
Summer 2018
Practice Final Exam
Aug 2, 2018 Thu 1-3:30pm
Time Limit: 150 Minutes

Full Name (Print): ____________________

University ID ____________________

This exam contains 6 pages (including this cover page) and 5 problems. Check to see if any pages are missing. Enter your full name and university ID on the top of this page, and put also your initials on the top of every page, in case the pages become separated.

You may **not** use your any textbooks and notes on this exam.

You are **required** to show your work in detail on each problem.
The following rules apply:

- **Work on your own and no discussion is allowed.** Any form of cheating will not be tolerated and will result in a failing grade for the assignment or for the course. Only allow to use Calculator.
- **Organize your work**, in a reasonably neat and coherent way. Work scattered all over the page without a clear ordering will receive partial credit.
- **Mysterious or unsupported answers will not receive full credit**. A correct answer, unsupported by necessary calculations, explanation, or mathematical justification will receive no credit; an incorrect answer supported by substantially correct calculations and explanations might still receive partial credit.
- Only the last problem used MATLAB programming language. You are asked to fill in the blacked-out with MATLAB programming syntax. **Please provide full explanation of your understanding of the algorithm as well as the corresponding MATLAB codes.** You are not required to run the MATLAB codes on a computer.

Problem	Points	Score
1	20	
2	20	
3	20	
4	20	
5	20	
Total:	100	

1. (20 points) Let $h > 0$. Use Taylor series expansion to show the following one-sided finite difference scheme has a second-order accuracy, i.e.,

$$u'(\bar{x}) = \frac{1}{2h}\left(3u(\bar{x}) - 4u(\bar{x} - h) + u(\bar{x} - 2h)\right) + O(h^2).$$

2. (20 points) Consider the following ODE-BVP with periodic boundary condition

$$u''(x) - q(x)u(x) = f(x), \quad x \in (0,1)$$

$$u(0) = u(1), \quad \text{periodic BC.}$$

(a) (10 points) Descretize the above ODE-BVP with a **central difference scheme** using a uniform mesh size $h = 1/4$ and derive each equation. How many unknowns? Hint: $u_0 = u_{m+1}$.

(b) (10 points) Write down the system of linear equation in matrix form $A\vec{u} = \vec{f}$. Is A tridiagonal?

3. (20 points) Consider the following diffusion-advection equation

$$u_t(x,t) + u_x(x,t) = u_{xx}(x,t)$$

with initial and boundary conditions:

$$u(x,0) = \eta(x), \quad u(0,t) = a(t), \quad u(1,t) = b(t).$$

(a) (10 points) Discretize the above PDE by a Forward Euler scheme in time (with a Central Difference scheme in space) on a uniform mesh $(x_i, t_n) = (ih, nk), 0 \le i \le m+1; 0 \le n \le N$, h and k are step sizes.

(b) (10 points) Formulate your derived scheme into matrix-vector time-marching form.

4. (20 points) Consider the following heat equation (with a source term)

$$u_t = \kappa u_{xx} + f(x,t), \qquad 0 < x < 1,\ t > 0$$

with initial and boundary conditions

$$u(x,0) = \eta(x), \quad u(0,t) = a(t), \quad u(1,t) = b(t).$$

(a) (10 points) Derive a backward Euler scheme in time (with a Central Diference scheme in space) on a uniform mesh $(x_i, t_n) = (ih, nk), 0 \le i \le m+1; 0 \le n \le N$, h and k are step sizes.

(b) (10 points) Formulate your derived scheme into matrix-vector time-marching form.

5. (20 points) Newton's method for accurately approximating π:

   (a) (10 points) Let $f(x) = \tan(x/4) - 1$. Derive the Newton's iteration for solving $f(x) = 0$. Use Matlab to perform 3 iterations with an initial guess $x^{(0)} = 3$. What's the final error?

   (b) (10 points) Complete the blacked-out codes of a Newton's iteration for computing π:

```
format long
f=@(x) tan(x/4)-1;
df1=@(x) sec(x/4)^2/4;
Nmax=10;
tol=1e-10;

x=3; %initial guess
res=zeros(Nmax,1);

for i=1:Nmax
    xnew=[redacted];%Newton Iteration
    res(i)=abs(f(xnew)); %Residual
    if(abs(xnew-x)<tol) %Check Tolerance to Stop
        break;
    end
    x=[redacted];
end
err=abs(x-pi)  %compute error
semilogy(1:i,res(1:i),'o-'); %plot residuals
```

   Below is the output:

```
err =

     8.961720254774264e-12
```