Engineering Analysis I

Midterm 1, Practice Exam

SOLUTIONS

Section number _____

Section number	Lecture time
20	9:00 a.m.
21	10:00 a.m.
22	11:00 a.m.
23	12:00 noon

This exam is closed-book and closed-notes. Calculators, computers, phones, or other computing/communication devices are not allowed.

Students should skip this page—it is only for graders.

Question	Points	Score
1	34	
2	25	
3	20	
4	21	
Total:	100	

Answer each question in the space provided. There are 4 questions for a total of 100 points.

1. Put a check mark ✓ in the box next to **EACH** correct answer. Note that there may be more than one correct answer for each question!

   (a) [6 points] Which of the following six MATLAB statements will create (or overwrite) a variable `A` and assign it the matrix $\begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 0 \end{bmatrix}$?

   - ✓ `A=[1 2 3; 0 0 0]`
   - ☐ `A=[1 2 3, 0 0 0]`
   - ✓ `A=[1:3; 0 0 0]`
   - ✓ `A=zeros(2,3); A(1,:)=1:3`
   - ☐ `A=zeros(2,3); A(:,1)=1:3`
   - ☐ `A=zeros(3,2); A(1,:)=1:3`

   (b) [4 points] Which of the following four MATLAB statements will replace the first row of an existing matrix variable `A` with the sum of its second and third rows? Assume that `A` has at least three rows and at least three columns.

   - ☐ `A(1)=A(2)+A(3)`
   - ✓ `A(1,:)=A(2,:)+A(3,:)`
   - ☐ `A(:,1)=A(:,2)+A(:,3)`
   - ✓ `A(1,:)=sum(A(2:3,:))`

   (c) [6 points] Which of the following six MATLAB statements will *not* generate an error message?

   - ✓ `x=sin([2 3;4 4])`
   - ✓ `x=[2 3]*[1;5]`
   - ☐ `x=[2 3]+[1;5]`
   - ✓ `x=[1 3].*[1 3]`
   - ☐ `x=[1 3]*[1 3]`
   - ✓ `x=2<[1 5]`

(d) [4 points] Which of the following four MATLAB logical expressions will return a value of logical `1` (meaning "true")?

- ✓ `~((1<3)&&(2<1))`
- ☐ `(1==3)||(2~=2)`
- ✓ `(1<3)||(~(2~=2))`
- ☐ `((1>3)||(1<3))&&(2>3)`

(e) [6 points] Which of the following six MATLAB statements is a valid first line of a loop? Assume that the variable `k` already exists in the workspace and contains the row vector `[0 0 0]`.

- ☐ `for k==1:3`
- ☐ `while k=1:3`
- ✓ `for k=1:3`
- ☐ `if k==[1 2 3]`
- ✓ `while k==1:3`
- ☐ `while k=k+1`

(f) [8 points] Which of the following eight blocks of code will add three to each element of an existing (and possibly non-square) matrix `A`?

- ✓
```
A = A+3;
```
- ✓
```
A = 3+A;
```
- ☐
```
A = A+3*size(A);
```
- ✓
```
A = A+3*ones(size(A));
```
- ☐
```
for k=1:size(A)
  A(k) = A(k)+3;
end
```
- ✓
```
for k=1:size(A,2)
  A(:,k) = A(:,k)+3;
end
```
- ☐
```
while ii=1:size(A,1)
  while jj=1:size(A,2)
    A(ii,jj) = A(ii,jj)+3;
  end
end
```
- ✓
```
for ii=1:size(A,1)
  for jj=1:size(A,2)
    A(ii,jj) = A(ii,jj)+3;
  end
end
```

2. Suppose each section of code below is run in MATLAB. If MATLAB generates an error message for the given code section, write "error" on the associated line, and otherwise write the value the variable `x` will have after the code section is run.

(a) [5 points]

```
clear
x = 0;
for k = [1 1 1]
    x = x + k;
end
```

(a) 3

(b) [5 points]

```
clear
x = 3;
if (x<0) || (x>5)
    x = 99;
end
```

(b) 3

(c) [5 points]

```
clear
x = 0;
for k = 1:10
    if k == 7
        x = 0;
    else
        x = x + 1;
    end
end
```

(c) 3

(d) [5 points]

```
clear
x = 0;
while y <= 10
    y = x + 2;
end
```

(d) error

(e) [5 points]

```
clear
x = 0;
while x == 0
    for k = 1:3
        x = x + 1;
    end
end
```

(e) 3

3. The MATLAB script below is supposed to use an iterative method to calculate the solution of a linear system of equations of the form

$$x = Mx + v\,,$$

where x is a vector of unknowns, M is a known square matrix, and v is a known vector. An iterative method for finding x takes the form

$$x_{\text{new}} = Mx_{\text{old}} + v\,,$$

where x_{old} is the old estimate for x and x_{new} is the new, updated estimate. The script should keep iterating until either the norm of the difference $x_{\text{new}} - x_{\text{old}}$ is less than some small tolerance `tol` or the maximum number of iterations `max_iter` is reached. The present state of the code is as follows:

```
1   % Get data from user
2   M = input('Enter M: ');
3   v = input('Enter v: ');
4
5   % Initialization
6   tol = 1e-10;
7   max_iter = 100;
8   x_old = zeros(size(v));
9
10
11
12  % Iteration
13  while (norm(x_new-x_old) >= tol) || (k < max_iter)
14
15
16      x_new = M*x_old + v;
17  end
18
19  % Show result
20  if k < max_iter
21      x_new
22  else
23      fprintf('The iteration did not converge.\n');
24  end
```

This program does not work! It is incomplete and it contains an error in logic. Your job is to complete and fix it by answering the questions on the next page.

Note: the expression `norm(x_new-x_old)` measures the size of the difference between the vectors `x_new` and `x_old`: it returns `0` when they are the same, and it returns a small positive number when they are nearly the same.

Answer the following questions to complete and fix the code on the previous page.

(a) [8 points] The initialization section is incomplete. Fill in the empty lines 9 and 10 to complete this section:

```
9   x_new = v;    % or x_new = ones(size(v)) or many others

10  k = 0;        % the order of lines 9 and 10 does not matter
```

(b) [4 points] The logical expression after the `while` keyword in line 13 is incorrect. Put a check mark ✓ in the box next to the correct expression (only one is correct):

- ☐ `(norm(x_new-x_old) >= tol) | (k < max_iter)`
- ☐ `(norm(x_new-x_old) >= tol) || (k >= max_iter)`
- ✓ `(norm(x_new-x_old) >= tol) && (k < max_iter)`
- ☐ `(norm(x_new-x_old) < tol) && (k < max_iter)`

(c) [8 points] The while loop body is incomplete. Fill in the empty lines 14 and 15 to complete the body of this while loop:

```
14  k = k + 1;

15  x_old = x_new;    % the order of lines 14 and 15 does not matter
```

4. [21 points] Write a MATLAB script which asks the user to enter two variables, a hyperlink matrix `H` and a scalar parameter `d`, and then calculates the vector `r` of page rank values which solves the equation

$$r = (1-d)\begin{bmatrix}1\\ \vdots \\ 1\end{bmatrix} + dHr\,.$$

Your script should use MATLAB's left-division operator `\` to calculate `r`, and display the result by simply leaving out the semicolon at the end of the calculation (no need to use `fprintf`). Your script does not need to check inputs for errors or include any comments. Useful MATLAB functions for use in your script may include `input`, `size`, `eye`, `ones`, and `zeros`.

```
1   H = input('Enter the hyperlink matrix H: ');
2   d = input('Enter the parameter d: ');
3
4   N = size(H,1);
5   r = (eye(N)-d*H) \ ((1-d)*ones(N,1))
6
7   % Other correct options for line 4 include
8   %   N = size(H,2);
9   %   [~,N] = size(H);
10  %   [N,~] = size(H);
11  %   [M,N] = size(H);
12  %   [N,M] = size(H);
13  %   [N,N] = size(H);
14  %   s = size(H);  N = s(1);
15  %   s = size(H);  N = s(2);
16
```