Step 1 of 3

When using numerical approximation techniques, it is useful to have programs written to aid in numerical calculations to avoid calculations by hand. Using Matlab, create an

M-file named “fixp.m” which will contain Matlab code that implements the fixed point method.

Inside this file, write the code to implement the method. One example of a correctly executable routine based on the algorithm in the text is as follows:

```
function rtn = fixp (f,x0,Tol)

%fixed point reaches a certain tolerance to approximate root of fx

%inputs are function f, initial point x0, and tolerance

i=1; %initialize iteration count

n=0; %loop control

while (n == 0)%checks if Tolerance is reached

x1 = x0; x0 = f(x0);

X = [i, x1]; % vector containing iteration count apx.root

i=i+1; %increment iteration count by one

if (abs(x0 - x1)< Tol) %tolerance based on norm convergence

n = 1;

disp (X) %display vector X

end

end % of while loop
```

Step 2 0

a) To approximate a root of $f(x)=e^x-2x^2$ using the fixed point method, find an arrangement of $f(x)=0$ such that $x=g(x)$ (it is a trial and error approach to find an appropriate function that converges to a desired root).

The text suggests to first try $x=\sqrt{(e^x/2)}$ using the point $x_0=1.5$ to initialize the method to approximate the root.
For this problem, let the tolerance be .00001. The root approximation will be in the second column of the row output.
INPUT:
>> format longG
>> f=@(x)sqrt(exp(x)/2)
OUTPUT:
f =

Step 3 0

@(x)sqrt(exp(x)/2)

Step 4 0

INPUT:
>> fixp(f,1.5,.00001)
OUTPUT:

Step 4

0

INPUT:
```
>> fixp(f,1.5,.00001)
```
OUTPUT:
```
21 1.48799494367049
```

Step 5

0

Looking at the Matlab outputs, it appears that $\boxed{x = 1.48799494367049}$ is the root near 1.5.
Note: Answers may vary due to choice of tolerance, convergence criteria, and root finding algorithm.

Step 6

0

Next, the text suggests trying $x = -\sqrt{(e^x/2)}$ using the point $x_0 = -0.5$ to initialize the method to approximate the root.
For this problem, let the tolerance be .00001. The root approximation will be in the second column of the row output.
INPUT:
```
>> f=@(x)sqrt(exp(x)/2)
```
OUTPUT:
```
f =
```

Step 6 0

Next, the text suggests trying $x = -\sqrt{(e^x/2)}$ using the point $x_0 = -0.5$ to initialize the method to approximate the root.
For this problem, let the tolerance be .00001. The root approximation will be in the second column of the row output.
INPUT:
>> f=@(x)sqrt(exp(x)/2)
OUTPUT:
f =

Step 7 0

@(x)-sqrt(exp(x)/2)

Step 8 0

INPUT:
>> fixp(f,-0.5,.00001)
OUTPUT:
8 -0.539839467642105

Step 9 0

Step 9

0

Looking at the Matlab outputs, it appears that $\boxed{x = -0.539839467642105}$ is the root near -0.5.
Note: Answers may vary due to choice of tolerance, convergence criteria, and root finding algorithm.

Step 10

0

b) The text suggests trying $x = \sqrt{(e^x/2)}$ using the points $x_0 = 2.5, 2.7$ to initialize the method to approximate the root near 2.6.
For this problem, let the tolerance be .00001. The root approximation will be in the second column of the row output.
INPUT:

```
>> f=@(x)sqrt(exp(x)/2)
```

OUTPUT:

```
f =
```

Step 11

0

```
@(x)sqrt(exp(x)/2)
```

Step 12

0

Step 12 0

INPUT:
>> fixp(f,2.5,.00001)
OUTPUT:
44 1.48799858547352

Step 13 0

INPUT:
>> fixp(f,2.7,.00001)
OUTPUT:
There is no output unless an error message is programmed. Using this point triggers an infinite loop.

Step 14 0

Looking at the Matlab outputs, it appears that the fixed point method converges to $x = 1.48799858547352$ when started with 2.5, and does not converge for 2.7, which does not produce a result near 2.6.
Note: Answers may vary due to choice of tolerance, convergence criteria, and root finding algorithm.

Step 15 0

Step 15

0

c) To find an arrangement that converges to the other positive root, try

$$e^x - 2x^2 = 0$$
$$e^x = 2x^2$$
$$x = \ln\left(2x^2\right)$$

Step 16

0

Next, give the following sequence of commands in the Matlab command window to apply the fixed point method to $x = \ln\left(2x^2\right)$, using the point $x_0 = 2.6$ to initialize the method to approximate the root.
For this problem, let the tolerance be .00001. The root approximation will be in the second column of the row output.
INPUT:
```
>> f=@(x)log(2*x^2)
```
OUTPUT:
```
f =
```

Step 17

0

```
@(x)log(2*x^2)
```

```
>> f=@(x)log(2*x^2)
OUTPUT:
f =
```

Step 17 0

```
@(x)log(2*x^2)
```

Step 18 0

```
INPUT:
>> fixp(f,2.6,.00001)
OUTPUT:
24 2.61782952318925
```

Step 19 0

Looking at the Matlab outputs, it appears that $\boxed{x = 2.61782952318925}$ is the root near 2.6.
Note: Answers may vary due to choice of tolerance, convergence criteria, and root finding algorithm.