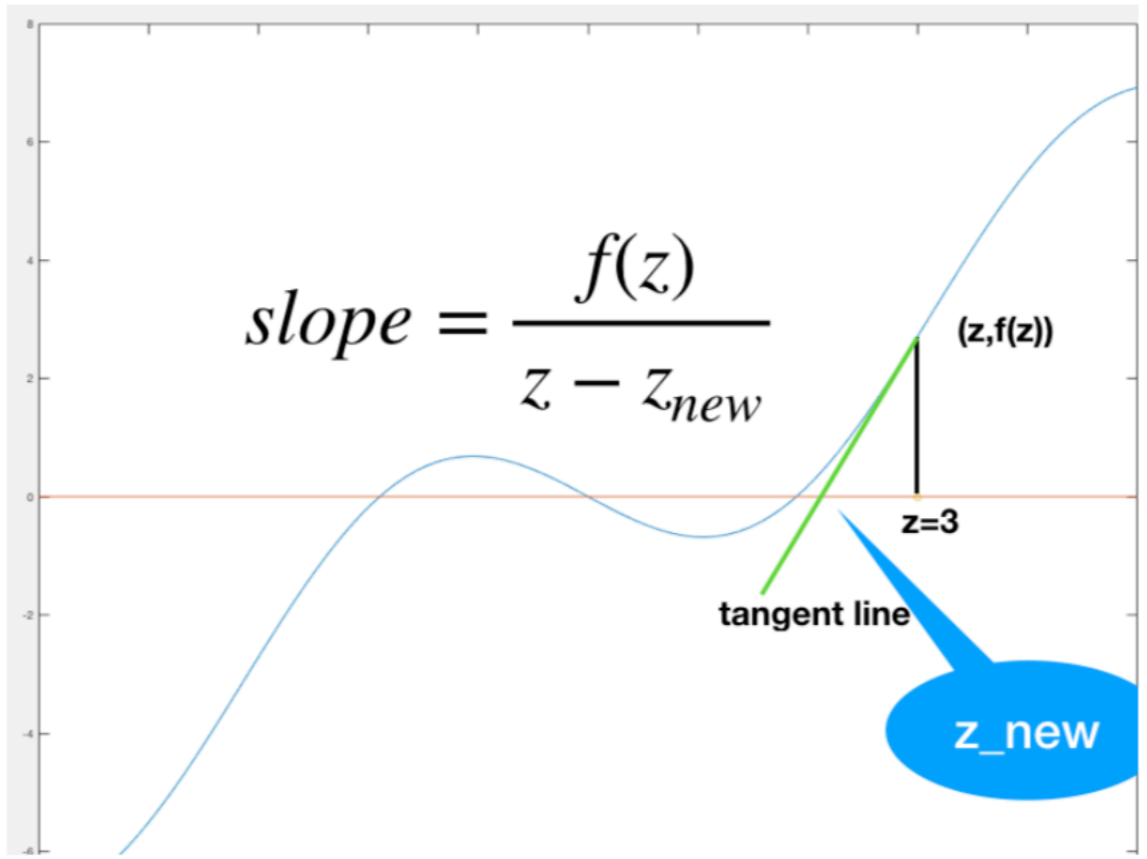
Final & nonlinear system solving

Numerical methods 2019

Topics

- The Newton Method of solving a nonlinear function
- Numerical Integration
- Data Fitting by the least mean square method
- Checking four conditions of the Reduced Echelon Form
- Forming a reduced echelon form

- The Newton Method of solving a nonlinear function
-



J. (9 points) Implement the Newton's method

____

____

____

```
s = 'x- 2*sin(x)';
f = inline(s);
df = inline(diff(str2sym(s)));
z = rand;
while  ~( [redacted]  < 10^-6)
    slope = [redacted];
    if ~( abs(slope)) < 10^-6
        z = [redacted];
       else
              break;
    end
end
f(z)
```

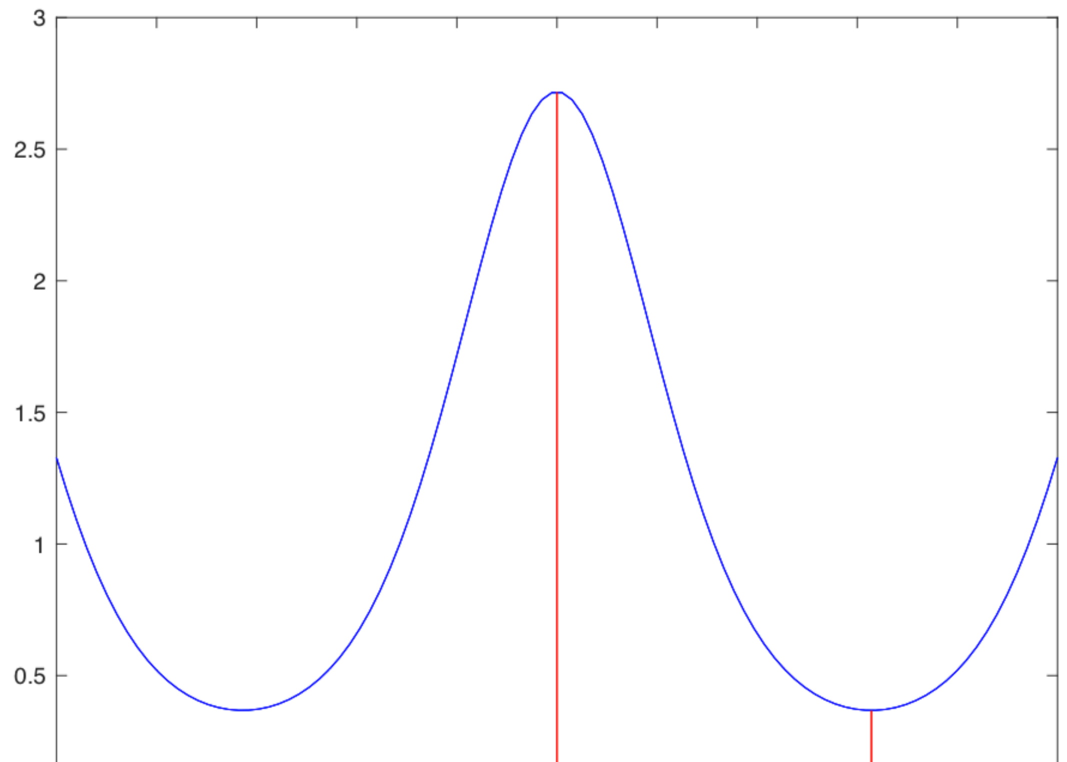
Numerical Integration

% Draw function exp(cos(x)) and
% apply integral to find its definite integration from 0 to pi

```
f = inline('exp(cos(x))');
Z = linspace(-5,5);
plot([redacted],'b'); hold on
plot([0 0],[0 f(0)],'r');
plot([pi pi],[0 f(pi)],'r');

ans = [redacted](@(x) exp(cos(x)),0,pi);
fprintf('%18.17f\n',ans);
```

3.97746326050642285



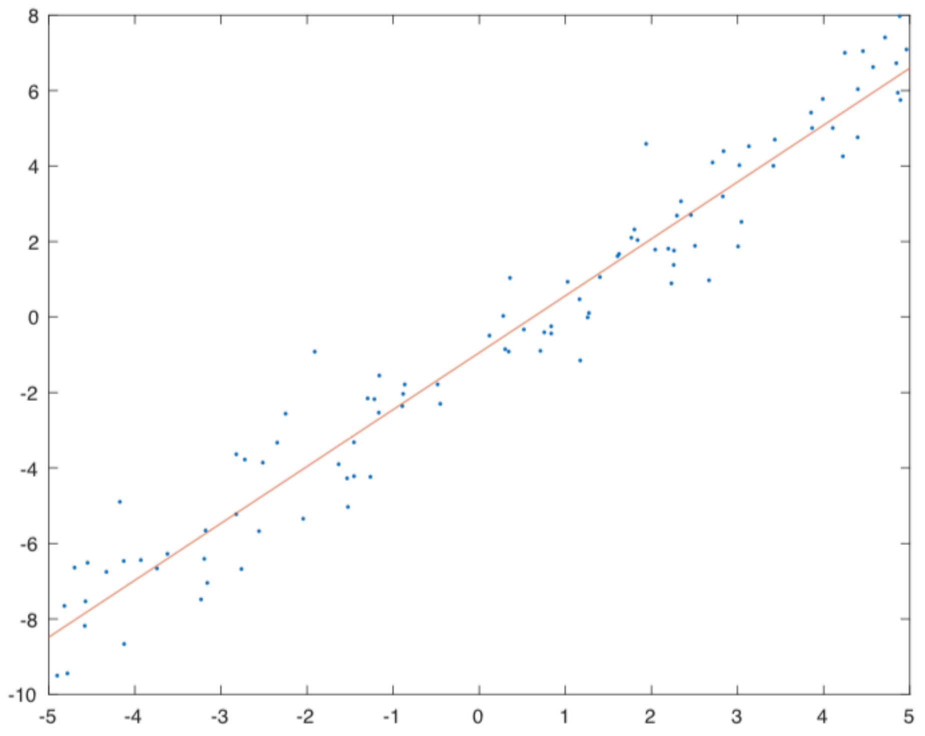
Data Fitting

Apply polyfit to estimate polynomial coefficients subject to given data for line fitting

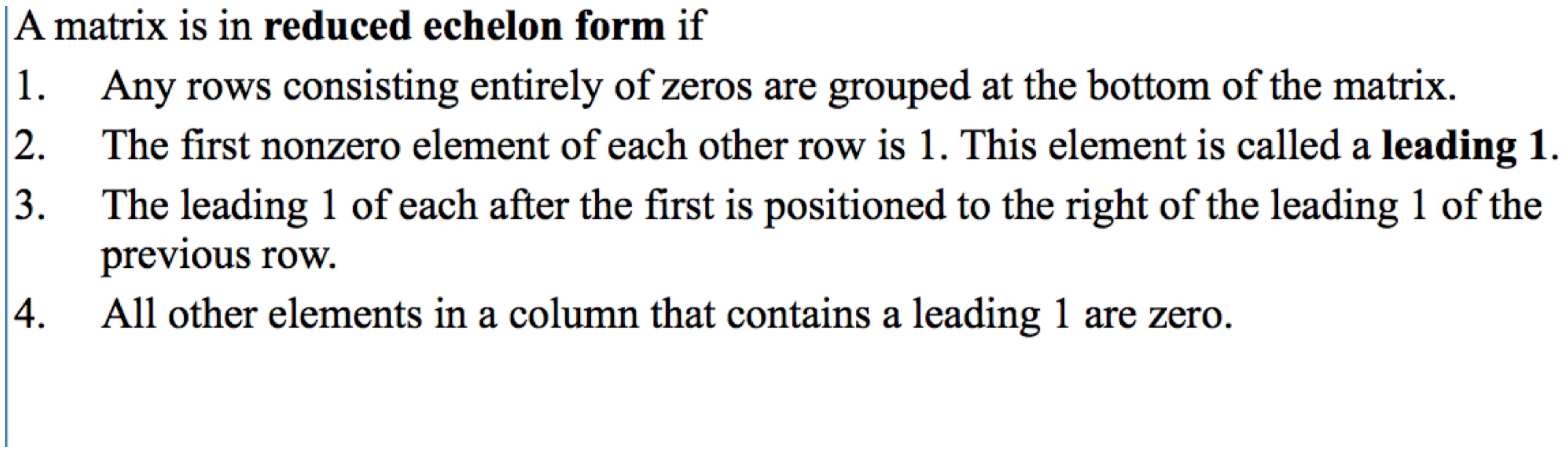
```
n = 100;
x = rand(1, n) * 10 - 5;
noise = randn(1,n) ;
y = polyval([1.5 1], x) + noise;
plot(x, y, '.'); hold on;

p = polyfit([redacted]);
x_new = linspace(-5,5);
y_new = polyval([redacted]);

plot(x_new, y_new)
```



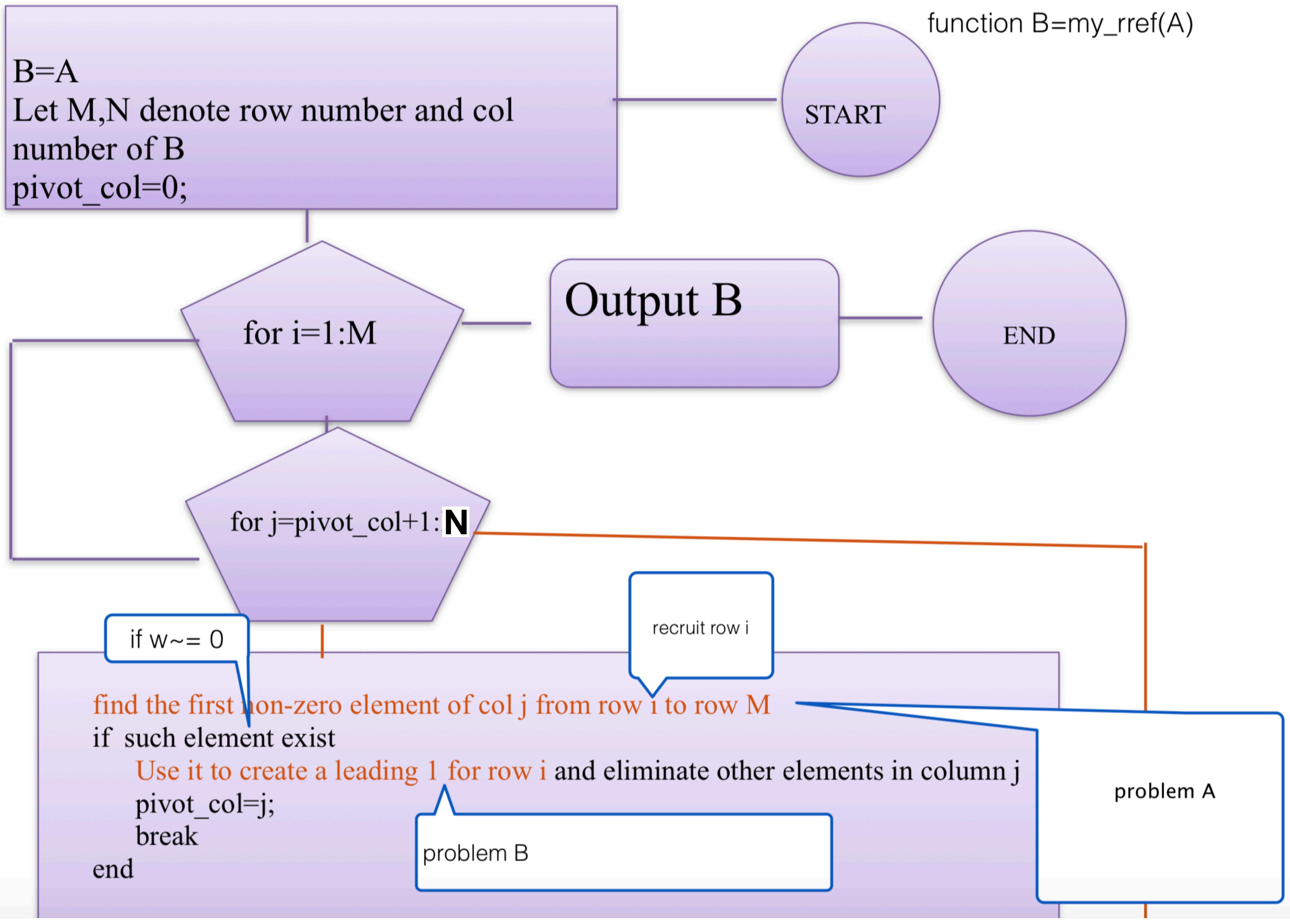
- Checking four conditions of the Reduced Echelon Form



A matrix is in **reduced echelon form** if

1. Any rows consisting entirely of zeros are grouped at the bottom of the matrix.
2. The first nonzero element of each other row is 1. This element is called a **leading 1**.
3. The leading 1 of each after the first is positioned to the right of the leading 1 of the previous row.
4. All other elements in a column that contains a leading 1 are zero.

Gauss Jordon Elimination



function B=my_rref(A)
START
B=A
Let M,N denote row number and col number of B
pivot_col=0;
for i=1:M
Output B
END
for j=pivot_col+1:N
if w~= 0
recruit row i
find the first non-zero element of col j from row i to row M
if such element exist
Use it to create a leading 1 for row i and eliminate other elements in column j
pivot_col=j;
break
end
problem A
problem B

Body statements

```
for i = 1: M
for j = pivot_col + 1 : N

    Find the first non-zero element of col j from
    row I ro row M

    Such element exists

    Use it to create a leading one from row i

    Use it to eliminate other elements
    in col j

end
end
```

```
function B=my_rref(A)
B=A;
[M N]=size(B);
pivot_col = 0;
for i = 1:M
    for j=pivot_col+1:N

            w=0;
            for k=i:M
                if B(k,j) ~= 0
                    w=B(k,j);
                    break;
                end
            end
            if w~=0
                temp=B(i,:);
                B(i,:)=B(k,:);
                B(k,:)=temp;
                r=B(i,j);
               B(i,:)=B(i,:)/r;
               for p=1:M
                   if (p~= i) & B(p,j)~=0
                   B(p,:)=B(p,:)-B(i,:)*B(p,j);
                   end
               end
               pivot_col = j;
               break
        end
    end
end
```

Find the first non-zero element of col j from row I ro row M

Such element exists

Use it to create a leading one from row i

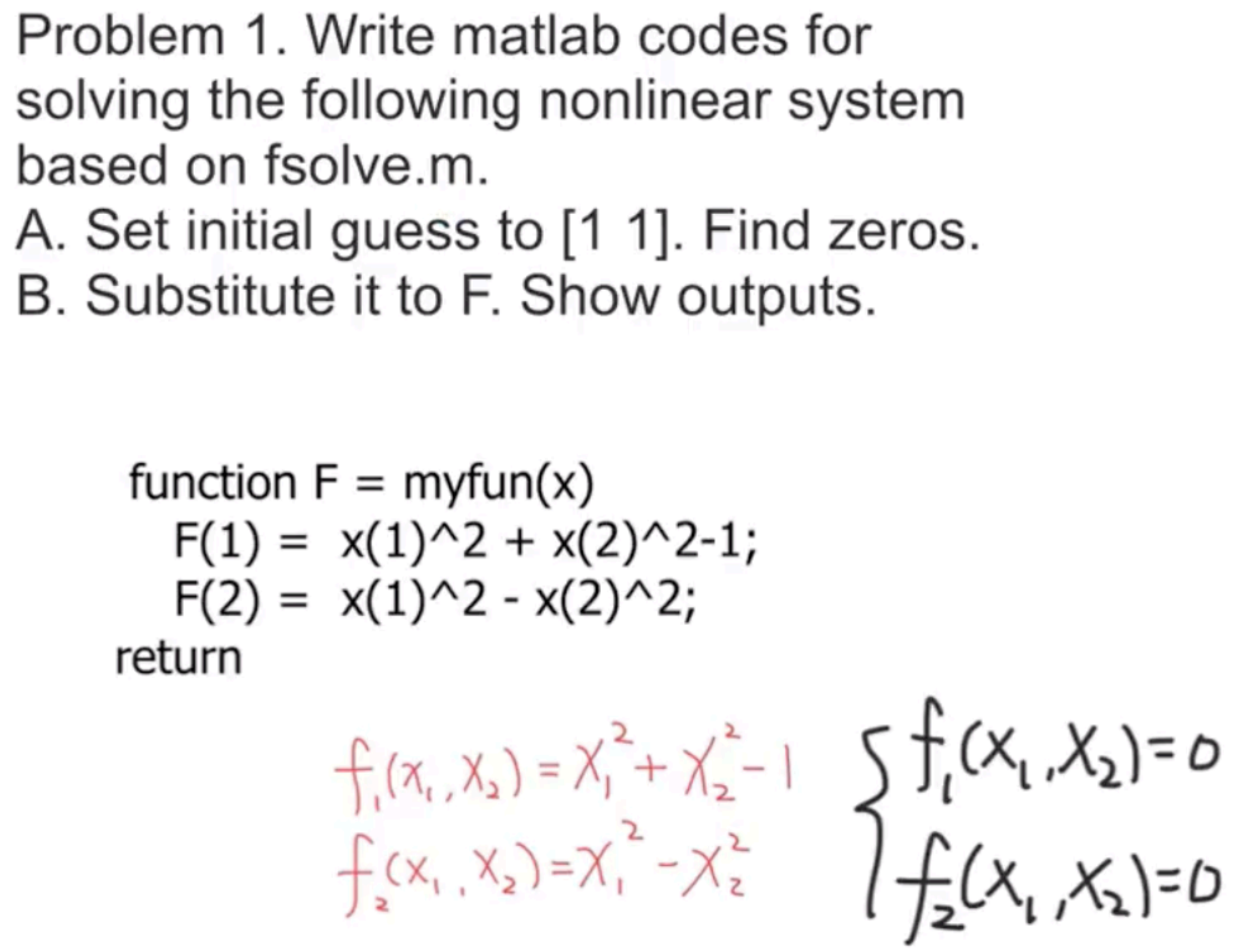
Use it to eliminate other elements in col j

```
function B=my_rref(A)
B=A;
[M N]=size(B);
pivot_col = 0;
for i = 1:M
    for j=pivot_col+1:N

            w=0;
            for k=i:M
                if B(k,j) ~= 0
                    w=B(k,j);
                    break;
                end
            end
            if w~=0
                temp=B(i,:);
                B(i,:)=B(k,:);
                B(k,:)=temp;
                r=B(i,j);
               B(i,:)=B(i,:)/r;
               for p=1:M
                    if (p~= i) & B(p,j)~=0
                    B(p,:)=B(p,:)-B(i,:)*B(p,j);
                    end
               end
               pivot_col = j;
               break
        end
    end
end
```

Newton’s method for solving nonlinear systems

Problem 1



Problem 1. Write matlab codes for solving the following nonlinear system based on fsolve.m.
A. Set initial guess to [1 1]. Find zeros.
B. Substitute it to F. Show outputs.

```
function F = myfun(x)
   F(1) =  x(1)^2 + x(2)^2-1;
   F(2) =  x(1)^2 - x(2)^2;
return
```

$$f_1(x_1, x_2) = x_1^2 + x_2^2 - 1$$
$$f_2(x_1, x_2) = x_1^2 - x_2^2$$

$$\begin{cases} f_1(x_1, x_2) = 0 \\ f_2(x_1, x_2) = 0 \end{cases}$$

Apply fsolve to a nonlinear system

Problem 2

Problem 2. Apply jacobian.m to find the Jacobian of the following coordinate functions

A. Derive Jacobian

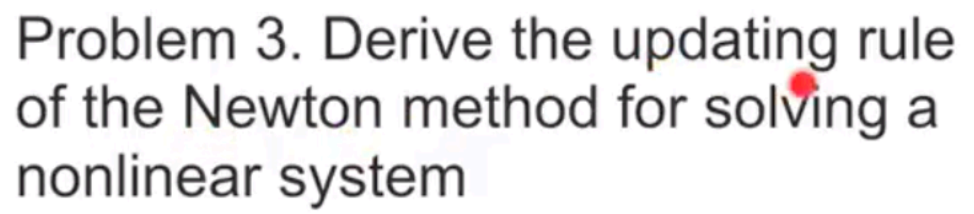
B. Write matlab codes to obtain an inline function for calculation of Jacobian

$$f_1(x_1, x_2) = x_1^2 + x_2^2 - 1$$

$$f_2(x_1, x_2) = x_1^2 - x_2^2$$

Apply matlab jacobian.m to have an inline function for calculation of Jacobian

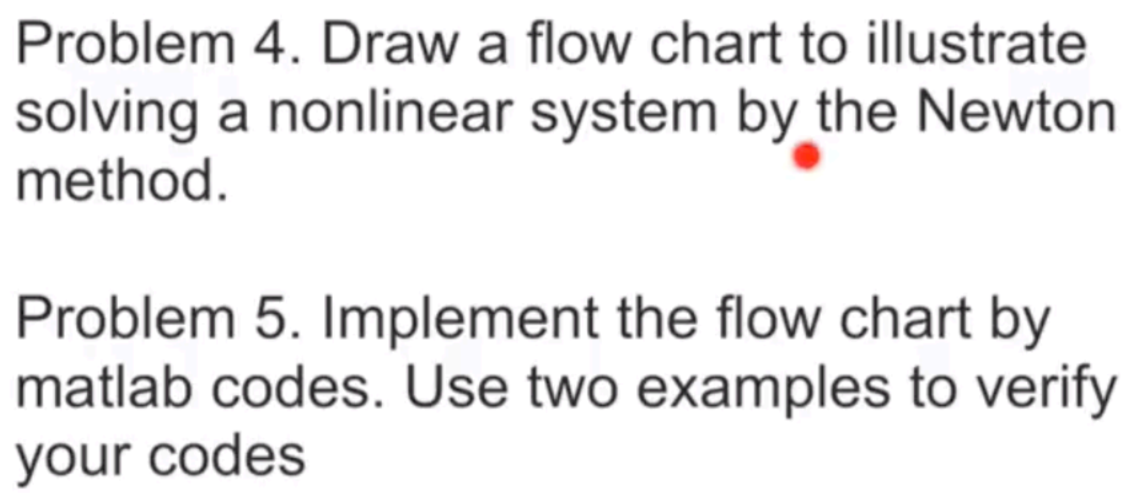
Problem 3



Problem 3. Derive the updating rule of the Newton method for solving a nonlinear system

Derive the updating rule of the Newton method for solving a nonlinear system

Problem 4 & 5



Problem 4. Draw a flow chart to illustrate solving a nonlinear system by the Newton method.

Problem 5. Implement the flow chart by matlab codes. Use two examples to verify your codes

Flow chart and implementation of the Newton method for solving a nonlinear system