- Polynomial evaluation
- Lagrange polynomial
- Polynomial interpolation

poly

- Given zeros, express a polynomial

```
poly([-5 0 5])
```

  - poly.m returns coefficients of a polynomial with roots –5,0 and 5

$$p(x) = (x+5)x(x-5) = x^3-25x$$

```
>> poly([-5 0 5])

ans =

     1     0   -25     0
```

$$p(x) = x^3 - 25x$$

polyval

```
p=poly([-5 0 5]);
x=linspace(-5,5);
y=polyval(p,x);
```

- Polynomial evaluation
- polyval.m substitutes elements in x to polynomial p
- p is a vector that represents a polynomial with roots 5, 0 and -5

Lagrange polynomial

- n knots, $x = [x_1, \ldots, x_n]$
- $$L_i(x) = 1, \text{ if } x = x_i$$ $$= 0, \text{ if } x = x_j, \mathrm{j} \neq \mathrm{i}$$
- L_i denotes the ith Lagrange polynomial that responds one to x_i and zeros to the other knots

Mathematical expression

$$L_i(x) = \frac{x - x_1}{x_i - x_1} \cdots \frac{x - x_{i-1}}{x_i - x_{i-1}} \frac{x - x_{i+1}}{x_i - x_{i+1}} \cdots \frac{x - x_n}{x_i - x_n}$$

$$L_i(x) = 1, \text{ if } x = x_i$$

$$= 0, \text{ if } x = x_j, \mathrm{j} \neq \mathrm{i}$$

Proof

$$L_i(x) = \frac{x - x_1}{x_i - x_1} \cdots \frac{x - x_{i-1}}{x_i - x_{i-1}} \frac{x - x_{i+1}}{x_i - x_{i+1}} \cdots \frac{x - x_n}{x_i - x_n}$$

$$L_i(x_i) = \frac{x_i - x_1}{x_i - x_1} \cdots \frac{x_i - x_{i-1}}{x_i - x_{i-1}} \frac{x_i - x_{i+1}}{x_i - x_{i+1}} \cdots \frac{x_i - x_n}{x_i - x_n} = 1$$

$$L_i(x_i) = 1$$

Proof

$$L_i(x) = \frac{x - x_1}{x_i - x_1} \cdots \frac{x - x_{i-1}}{x_i - x_{i-1}} \frac{x - x_{i+1}}{x_i - x_{i+1}} \cdots \frac{x - x_n}{x_i - x_n}$$

$$L_i(x_{j \neq i}) = \frac{x_j - x_1}{x_i - x_1} \cdots \frac{x_j - x_{i-1}}{x_i - x_{i-1}} \frac{x_j - x_{i+1}}{x_i - x_{i+1}} \cdots \frac{x_j - x_n}{x_i - x_n} = 0$$

$$L_i(x_{j \neq i}) = 0$$

Product form

$$L_i(x) = \frac{x - x_1}{x_i - x_1} \cdots \frac{x - x_{i-1}}{x_i - x_{i-1}} \frac{x - x_{i+1}}{x_i - x_{i+1}} \cdots \frac{x - x_n}{x_i - x_n}$$

$$= \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

Lagrange polynomial

Given n distinct knots, $x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n$

Let L_i denote the ith Lagrange polynomial

- L_i is a polynomial of degree n-1
- L_i has n-1 roots:

$$x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$$

- Normalization: L_i satisfies

$$L_i(x_i)=1$$

Polynomial of n-1 roots

- x is a vector that consists of n distinct knots

```
xzeros=[x(1:i-1) x(i+1:n)];
pi=poly(xzeros);
```

- pi is a polynomial whose roots are all knots except for x_i

Normalization

```
c=polyval(pi,x(i));
pi=pi/c;
```

Normalization condition

$$L_i(x_i) = 1$$

Implementation I

- Apply poly.m and polyval.m

```
xzeros=[x(1:i-1) x(i+1:n)];
pi=poly(xzeros);
```

```
c=polyval(pi,x(i));
pi=pi/c;
```

Lagrange polynomial evaluation

```
y=lagrange_poly(v,x,i)
```

% x contains n knots

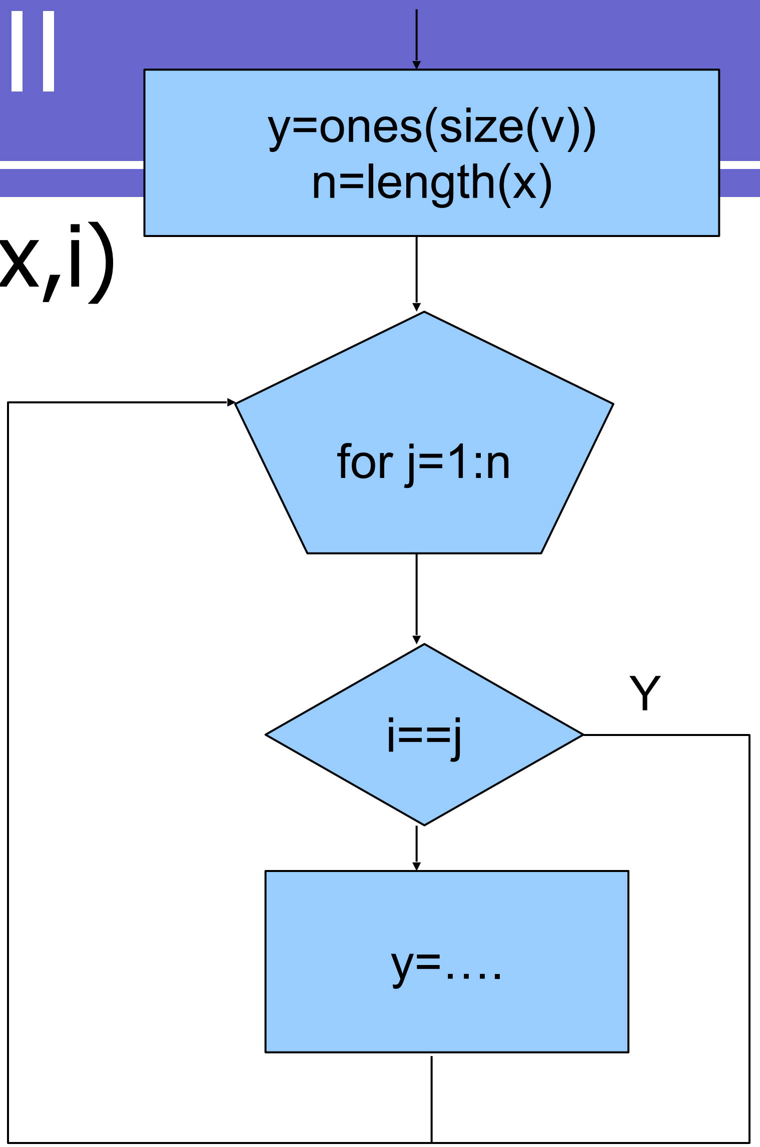
% Substitute elements in v to L_i defined by given knots

% y= $L_i(v)$

Implementation II

y=lagrange_poly(v,x,i)

$$L_i(v) = \prod_{\substack{j=1 \\ j\neq i}}^{n} \frac{v - x_j}{x_i - x_j}$$



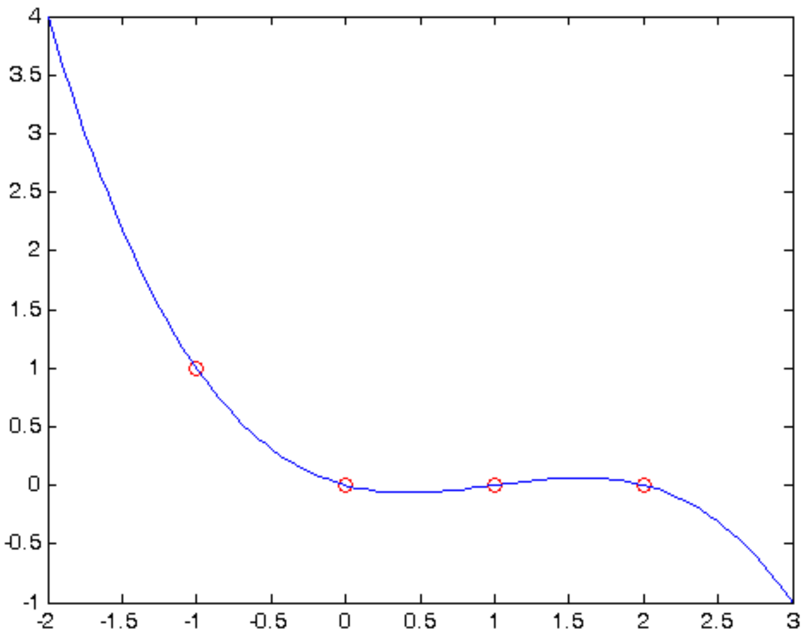
Evaluation of product form by for-looping

```
function y=lagrange_poly(v,x,i)
% evaluation of Li defined by knots in x
y=ones(size(v));
n=length(x);
for j=1:n

return
```

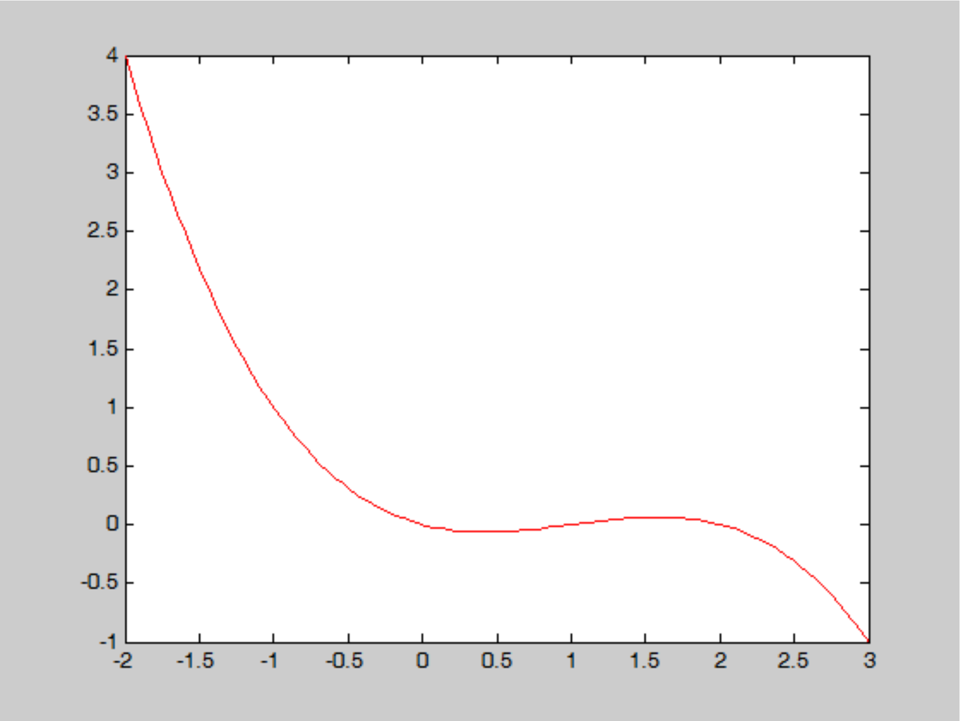
n=4,i=1

```
v=linspace(-2,3);
y=lagrange_poly(v,[-1 0 1 2],1);
plot(v,y);
```



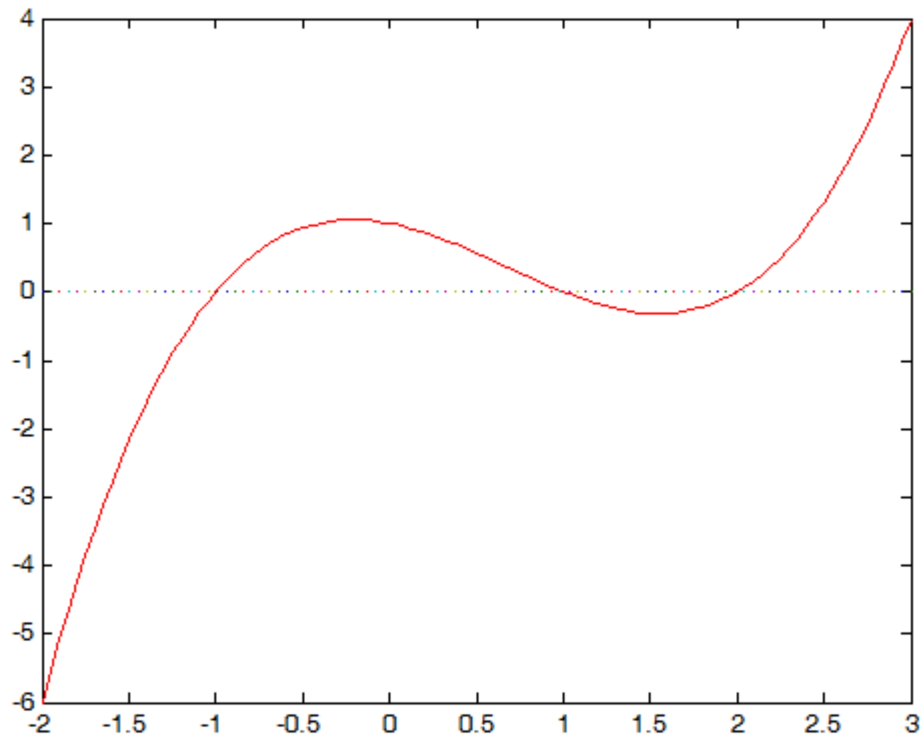
n=4,i=1

```
x=[-1 0 1 2];i=1;n=length(x);
xzeros=[x(1:i-1) x(i+1:n)];
pi=poly(xzeros);
pi=pi/polyval(pi,x(i));
v=linspace(-2,3);
plot(v,polyval(pi,v),'r');
```



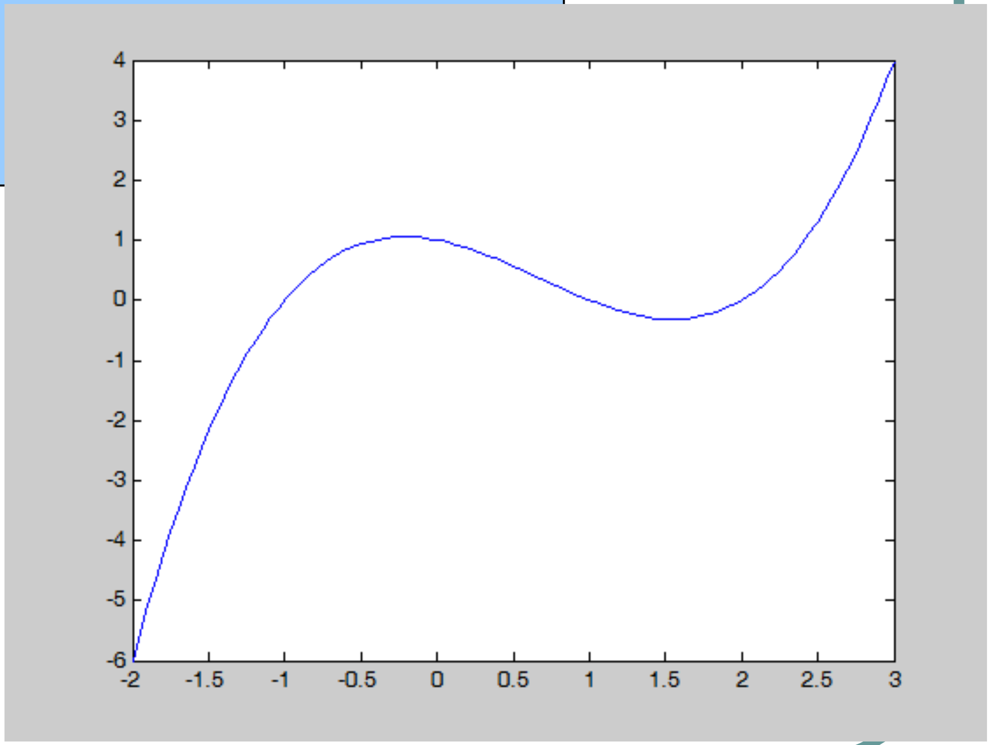
n=4,i=2 (Implementation I)

```
x=[-1 0 1 2];i=2;n=length(x);
xzeros=[x(1:i-1) x(i+1:n)];
pi=poly(xzeros);
pi=pi/polyval(pi,x(i));
v=linspace(-2,3);
plot(v,polyval(pi,v),'r');
plot(v,0);
```



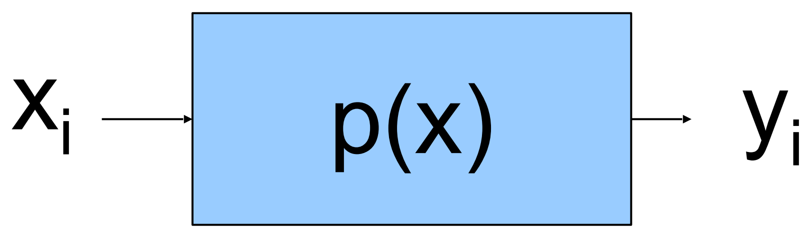
n=4,i=2

```
v=linspace(-2,3);
y=lagrange_poly(v,[-1 0 1 2],2);
plot(v,y);
plot(v,0);
```



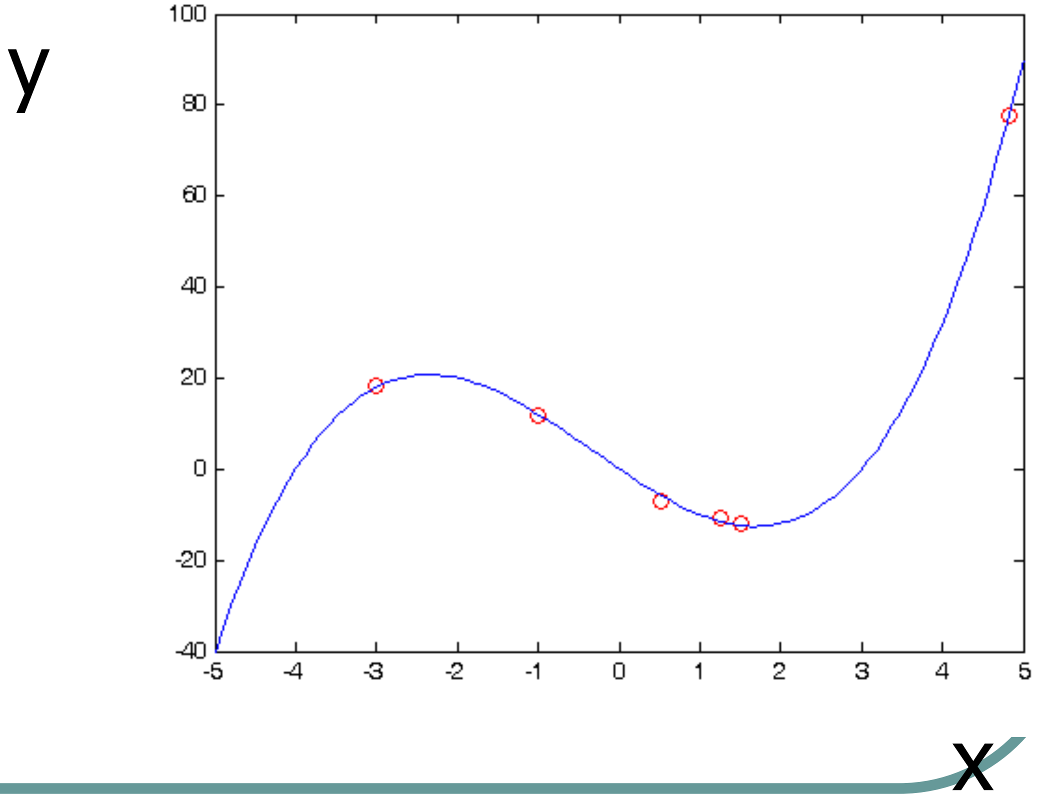
Polynomial interpolation

- Find a polynomial well interpolating given points
- Input: paired data, $S=\{(x_i,y_i)\}_i$
- Output: a polynomial $p(x)$ that passes all points in S



Sampling

A sample from an unknown target function



Polynomial interpolation

- Given $(x_i, y_i), i = 1, \ldots, n$
- Find a polynomial that satisfies

$$f(x_i) = y_i$$

for all i

Interpolating polynomial

- A linear combination of n Lagrange polynomials defined by n knots

$$f(v) = \sum_i y_i L_i(v)$$

- f is a polynomial of degree n-1
- L_i denotes the ith Lagrange polynomial

Verification

$$f(x_i) = \sum_k y_k L_k(x_i)$$

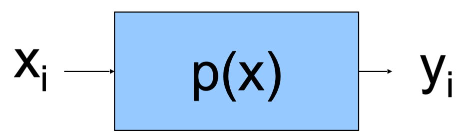
$$= y_i L_i(x_i) + \sum_{k \neq i} y_k L_k(x_i)$$

$$= y_i L_i(x_i)$$

$$= y_i$$

$$\therefore f(x_i) = y_i \ \forall \mathrm{i}$$

$$f(v) = \sum_i y_i L_i(v)$$



Evaluation of interpolating polynomial

- $$f(v) = \sum_i y_i L_i(v)$$

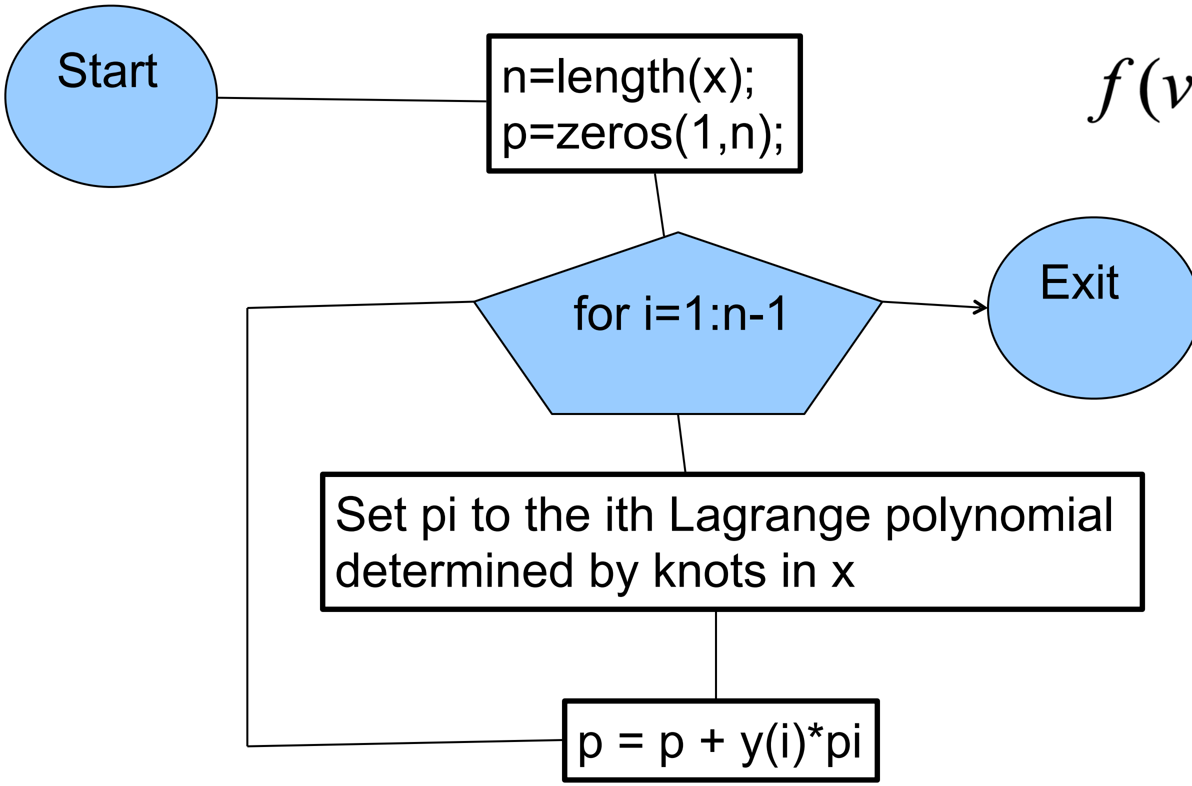
  ```
  function z=int_poly(v,x,y)

  % x contains n knots
  % y contains desired targets
  % substitute elements in v to f
  ```

Polynomial interpolation

function p=poly_interpolation(x,y)

$$f(v) = \sum_i y_i L_i(v)$$

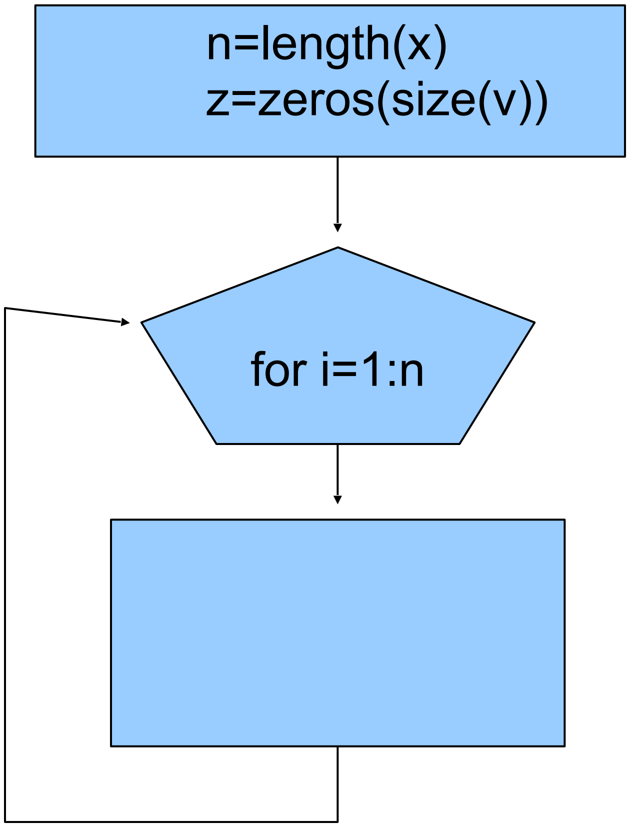


Evaluation of interpolating polynomial

- $$f(v) = \sum_i y_i L_i(v)$$

function z=int_poly(v,x,y)

% call lagrange_poly(v,x,i) to evaluate $L_i(v)$



```
function z=int_poly(v,x,y)
    z=zeros(size(v));
    n=length(x);
    for i=1:n

    end
return
```