# *Error Estimation*

*An interpolating polynominal, while passing through the support points used in its construction* ***does not****, in general, give exactly the correct values when used for interpolation.*

*The* ***error*** *that we make is given by*

***Error at support points is zero***

$$E(x) = f(x) - P_n(x) = (x - x_0)\cdot(x - x_1)\cdot\ldots\cdot(x - x_n)g(x)$$

$$\Rightarrow \quad f(x) - P_n(x) - (x - x_0)\cdot(x - x_1)\cdot\ldots\cdot(x - x_n)g(x) = 0$$

*Let's look at an* ***auxiliary function W(t),*** *defined as:*

$$W(t) = f(t) - P_n(t) - (t - x_0)\cdot(t - x_1)\cdot\ldots\cdot(t - x_n)g(x)$$

# *Error Estimation*

$$W(t) = f(t) - P_n(t) - (t - x_0)\cdot(t - x_1)\cdot \ldots \cdot (t - x_n) g(x)$$

*Let's examine the* ***zeros*** *(roots) of W(t):*

*For t= $x_0$, $x_1$,…,$x_n$ the function W(t) is zero: (n+1) times*

*For t= x the function W(t) is also zero.*

*--->* ***W(t) is zero at least a total of n+2 times.***

*Let's assume that W(t) is* ***continuous*** *and* ***differentiable****.*

*This assumption is true whenever the* ***original function*** *is also* ***continuous*** *and* ***differentiable****.*

# *Error Estimation*

*Between each of the n+2 zeros of W(t) we find a zero of W'(t)*

*---> W'(t) has a total of at least n+1 zeros.*

*---> W''(t) has a total of at least n zeros.*

*---> ……*

*---> $W^{n+1}(t)$ has at* ***least one zero*** *in the interval that has $x_0$, $x_n$, or x as endpoints. Let's call this* ***value*** $\xi$.

$$W^{n+1}(\xi) = 0$$

$$= \frac{d^{n+1}}{dt^{n+1}}\left[ f(t) - P_n(t) - (t - x_0)\cdot(t - x_1)\cdot \ldots \cdot(t - x_n)g(x)\right]_{t=\xi}$$

$$= f^{\,n+1}(\xi) - 0 - (n+1)!\; g(x)$$

# *Error Estimation*

$$\Rightarrow \quad g(x) = \frac{f^{n+1}(\xi)}{(n+1)!}, \quad \xi \text{ between } (x_0, x_n, x)$$

*With this our* ***error function*** *becomes:*

$$E(x) = f(x) - P_n(x) = (x - x_0)\cdot(x - x_1)\cdot \ldots \cdot(x - x_n)g(x)$$

$$= (x - x_0)\cdot(x - x_1)\cdot \ldots \cdot(x - x_n)\frac{f^{n+1}(\xi)}{(n+1)!}$$

$$\boxed{\Rightarrow \quad f(x) - P(x) = \frac{1}{(n+1)!} f^{n+1}(\xi)\prod_{i=0}^{n}(x - x_i)}$$

*with* $\xi$ *being in the interval that has* $x_0$, $x_n$, *or x as endpoints*.

# *Error Estimation*

$$\boxed{\Rightarrow \quad f(x) - P(x) = \frac{1}{(n+1)!}\, f^{n+1}(\xi) \prod_{i=0}^{n} (x - x_i)}$$

***Back to our example:***

*We had seven points ------> n+1 =7*
*The data looked suspiciously close to a sine-function*

$$\Rightarrow \quad f(x) - P(x) = \frac{1}{7!} \cdot \frac{d^7 \sin(\xi)}{dx^7} \cdot \prod_{i=0}^{6} (x - x_i) \qquad \boxed{\xi = ?}$$

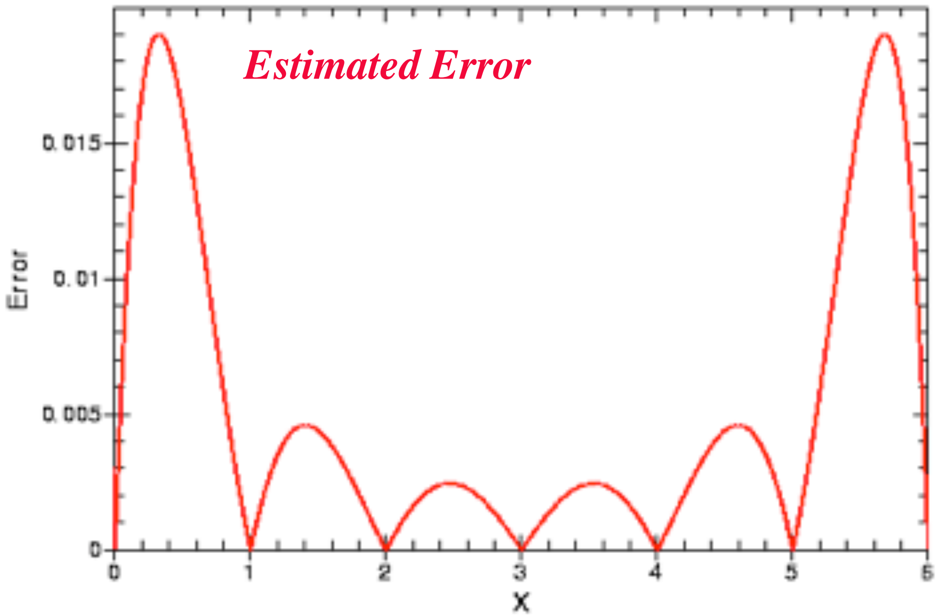
***The seventh derivative of sin(x) is bounded by ± 1***

$$\Rightarrow \quad f(x) - P(x) \le \frac{1}{7!} \cdot 1 \cdot \prod_{i=0}^{6} (x - x_i)$$

# Error Estimation

***Error for our approximation of the 7 sine-function data points***

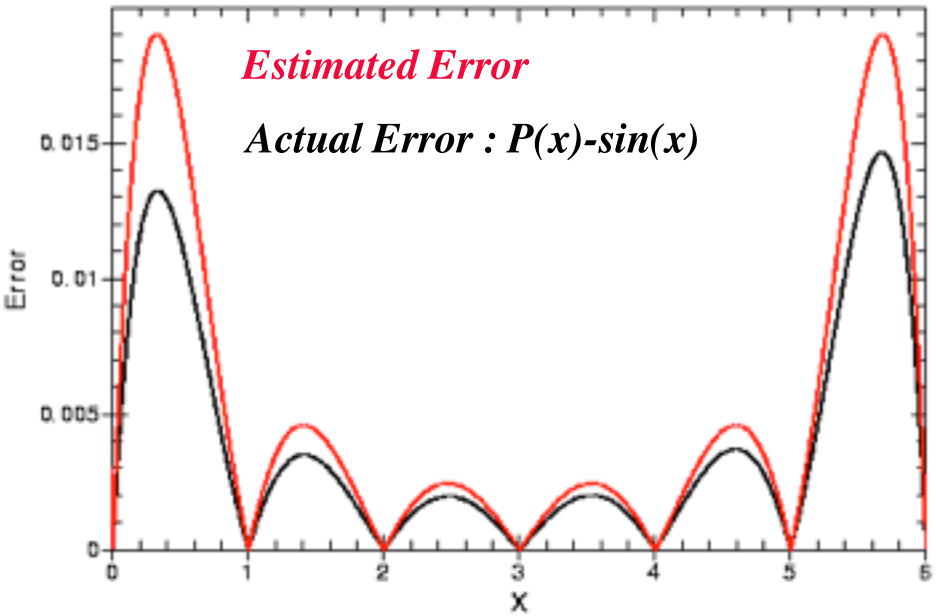
$$Error = \frac{1}{7!} \cdot \prod_{i=0}^{6} (x - x_i)$$



# Error Estimation

***Error for our approximation of the 7 sine-function data points***

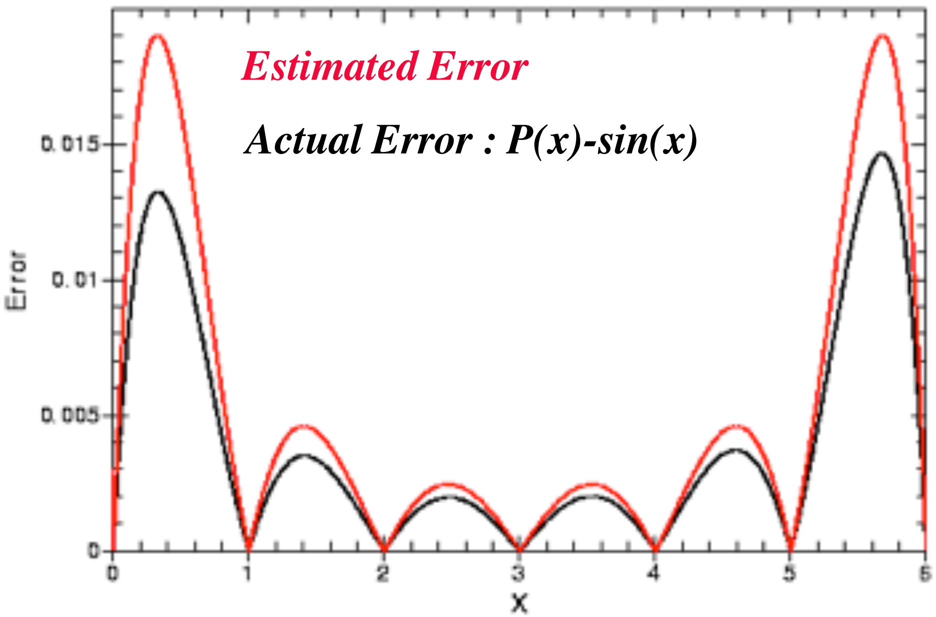
$$Error = \frac{1}{7!} \cdot \prod_{i=0}^{6} (x - x_i)$$

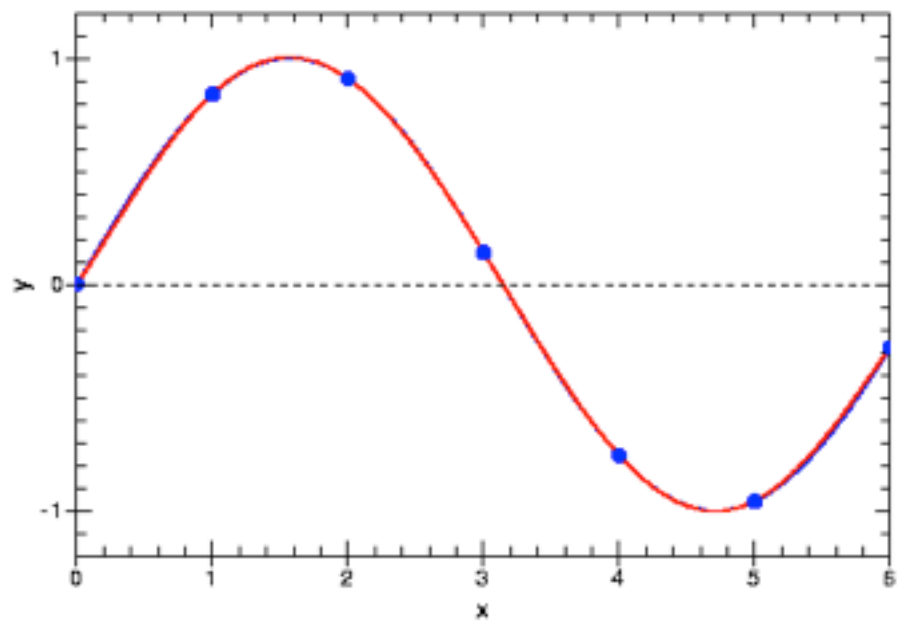


# Error Estimation

***Error for our approximation of the 7 sine-function data points***

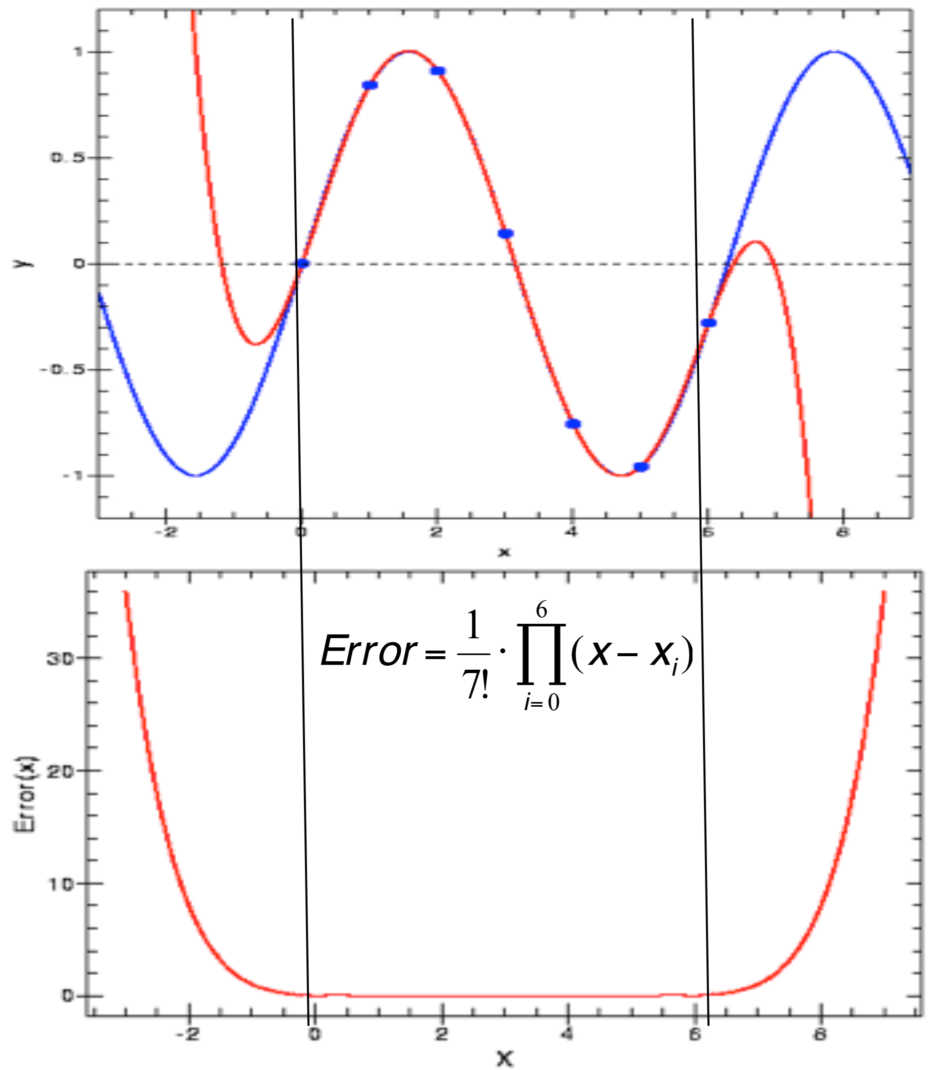
$$Error = \frac{1}{7!} \cdot \prod_{i=0}^{6} (x - x_i)$$



# *Extrapolation*

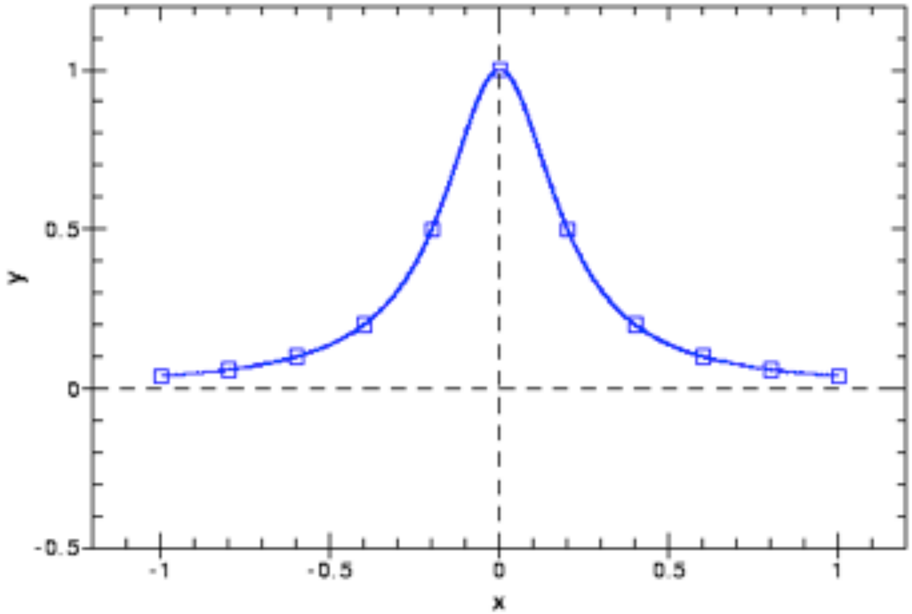
***We can use the interpolating polynominals to extrapolate outside of our original data area***




$$Error = \frac{1}{7!} \cdot \prod_{i=0}^{6} (x - x_i)$$

# Difficult Data

***The Runge Function:*** $$f(x) = \frac{1}{1+25x^2}$$

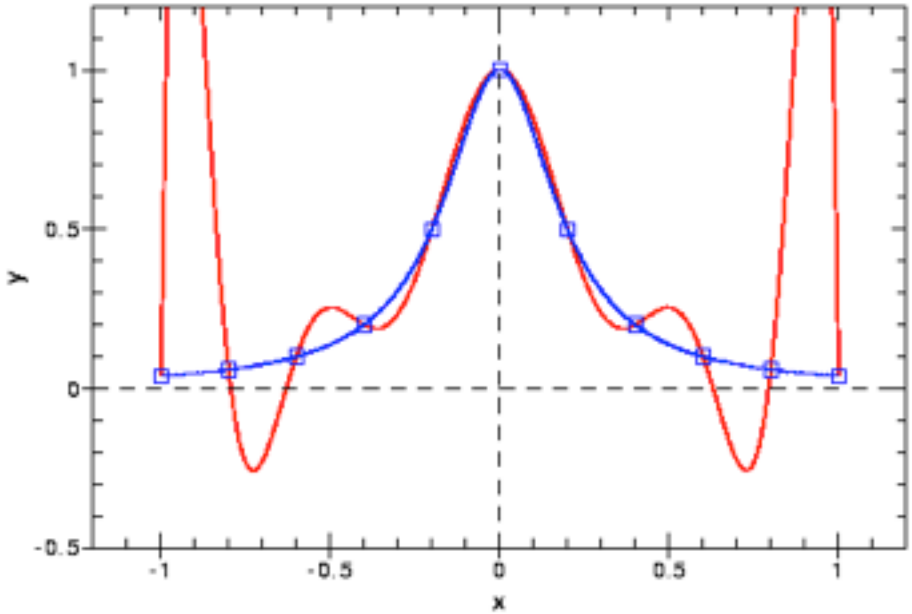



***11 Points***

***The Runge function is a smooth function.***

# *Difficult Data*

***The Runge Function:*** $$f(x) = \frac{1}{1+25x^2}$$




***11 Points***

***The interpolating polynominal oscillates toward the end of the interpolation interval.***

# *Computer Program*

$$\mathrm{P}(x) = \sum_{i=0}^{n} L_i^{(n)}(x)\, y_i = L_0^{(n)} y_0 + L_1^{(n)} y_1 + \ldots + L_n^{(n)} y_n$$

$$L_k^{(n)} = \frac{(x - x_0)(x - x_1)\ldots(x - x_{k-1})(x - x_{k+1})\ldots(x - x_n)}{(x_k - x_0)(x_k - x_1)\ldots(x_k - x_{k-1})(x_k - x_{k+1})\ldots(x_k - x_n)}$$

*The **numerator** is the product*

$$N_k(x) = (x - x_0)(x - x_1)\ldots(x - x_{k-1})(x - x_{k+1})\ldots(x - x_n)$$

*And the **denominator** is the product*

$$D_k = (x_k - x_0)(x_k - x_1)\ldots(x_k - x_{k-1})(x_k - x_{k+1})\ldots(x_k - x_n)$$

$$\Rightarrow\ \mathrm{P}(x) = \frac{y_0}{D_0} N_0 + \frac{y_1}{D_1} N_1 + \ldots + \frac{y_n}{D_n} N_n \ = c_0 N_0(x) + c_1 N_1(x) + \ldots + c_n N_n(x)$$

# Computer Program

$$P(x) = c_0 N_0(x) + c_1 N_1(x) + \ldots + c_n N_n(x)$$

$$N_k(x) = (x - x_0)(x - x_1)\ldots(x - x_{k-1})(x - x_{k+1})\ldots(x - x_n)$$

***<u>Tasks:</u>***

- ***Our Program should read n+1 data pairs from a file***
- ***The Coefficients $c_n$ need to be calculated and stored (in a vector)***
- ***The Polynominal P(x) needs to be calculated at 101 points between $x_0$ and $x_n$***
- ***The Results should be written out on the screen***

```
PROGRAM LAGRANGE_INTERPOLATION
IMPLICIT NONE

INTEGER, PARAMETER :: MAXPOINTS=20
INTEGER                        :: ISTEP, NPOINTS

REAL    :: X(MAXPOINTS), Y(MAXPOINTS)     !ORIGINAL DATA SET
REAL    :: XP, POLY                        !INTERPOLATED DATA
REAL    :: XMIN, XMAX, DX, STEP

REAL, ALLOCATABLE :: LCOEF (:)             !LAGRANGE COEFFICIENTS

INTERFACE
  …..
END INTERFACE

!READ DATA FROM FILE DATA.DAT
 ……
!END READ DATA FRM FILE

!GET COEFFICIENTS FOR LAGRANGE POLYNOMINALS

!USE ARRAY_VALUED FUNCTION
   ALLOCATE (LCOEF(NPOINTS))

   LCOEF = CK (X, Y, NPOINTS)

!GET THE MINIMUM AND MAXIMUM X-VALUE
   XMIN=MINVAL(X(1:NPOINTS))
   XMAX=MAXVAL(X(1:NPOINTS))

!CALCULATE THE LAGRANGE POLYNOMINAL FROM
!XMIN TO XMAX IN 101 STEPS
  …….
!END CALCULATE THE LAGRANGE POLYNOMINALS

STOP
END PROGRAM LAGRANGE_INTERPOLATION
```

**lagrange_interp.f90**

```
PROGRAM LAGRANGE_INTERPOLATION
IMPLICIT NONE

INTEGER, PARAMETER :: MAXPOINTS=20
INTEGER            :: ISTEP,NPOINTS

REAL    :: X(MAXPOINTS),Y(MAXPOINTS)  !ORIGINAL DATA SET
REAL    :: XP,POLY                     !INTERPOLATED DATA
REAL    :: XMIN,XMAX,DX,STEP

REAL, ALLOCATABLE :: LCOEF(:)

INTERFACE
   …..
END INTERFACE

!READ DATA FROM FILE DATA.DAT
 ……
!END READ DATA FRM FILE

!GET COEFFICIENTS FOR LAGRANGE POLYNOMINALS

!USE ARRAY_VALUED FUNCTION
   ALLOCATE (LCOEF(NPOINTS))

   LCOEF = CK (X, Y, NPOINTS)

!GET THE MINIMUM AND MAXIMUM X-VALUE
   XMIN=MINVAL(X(1:NPOINTS))
   XMAX=MAXVAL(X(1:NPOINTS))

!CALCULATE THE LAGRANGE POLYNOMINAL FROM
!XMIN TO XMAX IN 101 STEPS
  …….
!END CALCULATE THE LAGRANGE POLYNOMINALS

STOP
END PROGRAM LAGRANGE_INTERPOLATION
```

```
INTERFACE
 FUNCTION CK(X, Y, NPOINTS)
   IMPLICIT NONE
   REAL, INTENT (IN) :: X(:), Y(:)
   INTEGER, INTENT(IN) :: NPOINTS
   REAL, DIMENSION(NPOINTS) :: CK
 END FUNCTION CK

 FUNCTION LAGRANGE(XP, X, LCOEF, NPOINTS)
    IMPLICIT NONE
    REAL :: LAGRANGE
    REAL, INTENT(IN) :: XP, X(:), LCOEF(:)
    INTEGER, INTENT(IN) :: NPOINTS
 END FUNCTION LAGRANGE
END INTERFACE
```

```
PROGRAM LAGRANGE_INTERPOLATION
IMPLICIT NONE

INTEGER, PARAMETER :: MAXPOINTS=20
INTEGER            :: ISTEP,NPOINTS

REAL    :: X(MAXPOINTS),Y(MAXPOINTS)  !ORIGINAL DATA SET
REAL    :: XP,POLY                     !INTERPOLATED DATA
REAL    :: XMIN,XMAX,DX,STEP

REAL, ALLOCATABLE :: LCOEF(:)

INTERFACE
  …..
END INTERFACE

!READ DATA FROM FILE DATA.DAT
 ……
!END READ DATA FRM FILE

!GET COEFFICIENTS FOR LAGRANGE POLYNOMINALS

!USE ARRAY_VALUED FUNCTION
   ALLOCATE (LCOEF(NPOINTS))

   LCOEF = CK (X, Y, NPOINTS)

!GET THE MINIMUM AND MAXIMUM X-VALUE
   XMIN=MINVAL(X(1:NPOINTS))
   XMAX=MAXVAL(X(1:NPOINTS))

!CALCULATE THE LAGRANGE POLYNOMINAL FROM
!XMIN TO XMAX IN 101 STEPS
  …….
!END CALCULATE THE LAGRANGE POLYNOMINALS

STOP
END PROGRAM LAGRANGE_INTERPOLATION
```

```
!READ DATA FROM FILE DATA.DAT
    OPEN(10, FILE = 'DATA.DAT', STATUS='OLD')
    NPOINTS = 0
    DO
      NPOINTS = NPOINTS+1
      IF (NPOINTS .GT. MAXPOINTS) THEN
        WRITE(*,*) 'TOO MANY POINTS'
        STOP
      END IF
      READ(10,*, END = 99) X(NPOINTS), Y(NPOINTS)
    END DO
99  NPOINTS = NPOINTS - 1
    CLOSE(10)
```

```
PROGRAM LAGRANGE_INTERPOLATION
IMPLICIT NONE

INTEGER, PARAMETER :: MAXPOINTS=20
INTEGER          :: ISTEP,NPOINTS

REAL   :: X(MAXPOINTS),Y(MAXPOINTS)  !ORIGINAL DATA SET
REAL   :: XP,POLY                     !INTERPOLATED DATA
REAL   :: XMIN,XMAX,DX,STEP

REAL, ALLOCATABLE :: LCOEF(:)

INTERFACE
  …..
END INTERFACE

!READ DATA FROM FILE DATA.DAT
 ……
!END READ DATA FRM FILE

!GET COEFFICIENTS FOR LAGRANGE POLYNOMINALS

!USE ARRAY_VALUED FUNCTION
   ALLOCATE ( LCOEF(NPOINTS) )

   LCOEF = CK (X,Y,NPOINTS)

!GET THE MINIMUM AND MAXIMUM X-VALUE
   XMIN=MINVAL(X(1:NPOINTS))
   XMAX=MAXVAL(X(1:NPOINTS))

!CALCULATE THE LAGRANGE POLYNOMINAL FROM
!XMIN TO XMAX IN 101 STEPS
  …….
!END CALCULATE THE LAGRANGE POLYNOMINALS

STOP
END PROGRAM LAGRANGE_INTERPOLATION
```

```
!CALCULATE THE COEFFICIENTS
FUNCTION CK(X,Y,N)
IMPLICIT NONE

INTEGER,INTENT(IN)  :: N
REAL,INTENT(IN)       :: X(:),Y(:)
REAL,DIMENSION(N)   :: CK

INTEGER :: I,K
REAL       :: DK

DO K = 1,N
  DK = 1.
  DO I = 1,N
    IF(I .NE. K) THEN
      DK = DK*(X(K)-X(I))
    END IF
  END DO
 CK(K) = Y(K)/DK
END DO
RETURN
END FUNCTION CK
```

```
PROGRAM LAGRANGE_INTERPOLATION
IMPLICIT NONE

INTEGER, PARAMETER :: MAXPOINTS=20
INTEGER          :: ISTEP,NPOINTS

REAL    :: X(MAXPOINTS),Y(MAXPOINTS)  !ORIGINAL DATA SET
REAL    :: XP,POLY                     !INTERPOLATED DATA
REAL    :: XMIN,XMAX,DX,STEP

REAL, ALLOCATABLE :: LCOEF(:)

INTERFACE
  .....
END INTERFACE

!READ DATA FROM FILE DATA.DAT
 ......
!END READ DATA FRM FILE

!GET COEFFICIENTS FOR LAGRANGE POLYNOMINALS

!USE ARRAY_VALUED FUNCTION
   ALLOCATE (LCOEF(NPOINTS))

   LCOEF = CK (X, Y, NPOINTS)

!GET THE MINIMUM AND MAXIMUM X-VALUE
   XMIN  = MINVAL(X(1:NPOINTS))
   XMAX = MAXVAL(X(1:NPOINTS))

!CALCULATE THE LAGRANGE POLYNOMINAL FROM
!XMIN TO XMAX IN 101 STEPS
  .......
!END CALCULATE THE LAGRANGE POLYNOMINALS

STOP
END PROGRAM LAGRANGE_INTERPOLATION
```

```
!CALCULATE THE INTERPOLATING POLYNOMINAL
FUNCTION LAGRANGE (T, X, COEF, NPOINTS)
IMPLICIT NONE

REAL                      :: LAGRANGE
INTEGER, INTENT(IN) :: NPOINTS
REAL, INTENT(IN)       :: T, X(:),COEF(:)

INTEGER     :: J, K
REAL          :: NK

LAGRANGE = 0.   ! INITIALIZE VALUE

DO J = 1, NPOINTS
 NK = 1.
 DO K = 1, NPOINTS
  IF (J. NE. K) THEN
   NK = NK*(T-X(K))
  END IF
 END DO
 LAGRANGE = LAGRANGE + NK * COEF(J)
END DO
RETURN
END FUNCTION LAGRANGE
```

```
DX = (XMAX - XMIN)
STEP = DX/100.
XP = XMIN
DO ISTEP = 0, 100
  POLY = LAGRANGE (XP, X, LCOEF, NPOINTS)
  WRITE(*,*) XP, POLY
  XP = XP + STEP
END DO
```

```
PROGRAM LAGRANGE_INTERPOLATION
IMPLICIT NONE

INTEGER, PARAMETER :: MAXPOINTS=20
INTEGER            :: ISTEP,NPOINTS

REAL    :: X(MAXPOINTS),Y(MAXPOINTS)  !ORIGINAL DATA SET
REAL    :: XP,POLY                     !INTERPOLATED DATA
REAL    :: XMIN,XMAX,DX,STEP

REAL, ALLOCATABLE :: LCOEF(:)

INTERFACE
   …..
END INTERFACE

!READ DATA FROM FILE DATA.DAT
 ……
!END READ DATA FRM FILE

!GET COEFFICIENTS FOR LAGRANGE POLYNOMINALS

!USE ARRAY_VALUED FUNCTION
   ALLOCATE (LCOEF(NPOINTS))

   LCOEF = CK (X, Y, NPOINTS)

!GET THE MINIMUM AND MAXIMUM X-VALUE
   XMIN=MINVAL(X(1:NPOINTS))
   XMAX=MAXVAL(X(1:NPOINTS))

!CALCULATE THE LAGRANGE POLYNOMINAL FROM
!XMIN TO XMAX IN 101 STEPS
   …….
!END CALCULATE THE LAGRANGE POLYNOMINALS

STOP
END PROGRAM LAGRANGE_INTERPOLATION
```

```
INTERFACE
  FUNCTION CK(X, Y, NPOINTS)
    IMPLICIT NONE
    REAL, INTENT (IN) :: X(:), Y(:)
    INTEGER, INTENT(IN) :: NPOINTS
    REAL, DIMENSION(NPOINTS) :: CK
  END FUNCTION CK

  FUNCTION LAGRANGE(XP, X, LCOEF, NPOINTS)
    IMPLICIT NONE
    REAL :: LAGRANGE
    REAL, INTENT(IN) :: XP, X(:), LCOEF(:)
    INTEGER, INTENT(IN) :: NPOINTS
  END FUNCTION LAGRANGE
END INTERFACE
```

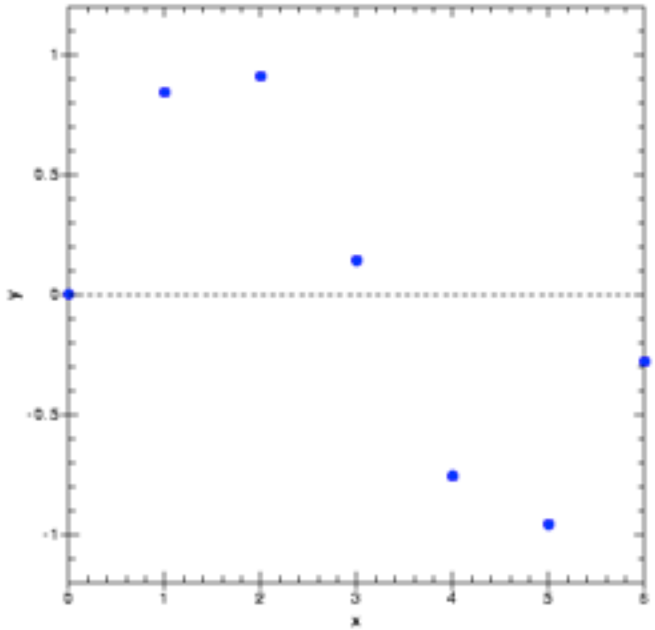
```
PROGRAM LAGRANGE_INTERPOLATION
USE LAGRANGE_INTERFACE
IMPLICIT NONE

INTEGER, PARAMETER :: MAXPOINTS=20
INTEGER            :: ISTEP,NPOINTS

REAL    :: X(MAXPOINTS),Y(MAXPOINTS)  !ORIGINAL DATA SET
REAL    :: XP,POLY                      !INTERPOLATED DATA
REAL    :: XMIN,XMAX,DX,STEP

REAL, ALLOCATABLE :: LCOEF(:)

! INTERFACE
   .....
! END INTERFACE

!READ DATA FROM FILE DATA.DAT
 ......
!END READ DATA FRM FILE

!GET COEFFICIENTS FOR LAGRANGE POLYNOMINALS

!USE ARRAY_VALUED FUNCTION
   ALLOCATE (LCOEF(NPOINTS))

   LCOEF = CK (X, Y, NPOINTS)

!GET THE MINIMUM AND MAXIMUM X-VALUE
   XMIN=MINVAL(X(1:NPOINTS))
   XMAX=MAXVAL(X(1:NPOINTS))

!CALCULATE THE LAGRANGE POLYNOMINAL FROM
!XMIN TO XMAX IN 101 STEPS
  .......
!END CALCULATE THE LAGRANGE POLYNOMINALS

STOP
END PROGRAM LAGRANGE_INTERPOLATION
```

```
MODULE LAGRANGE_INTERFACE

INTERFACE
  FUNCTION CK(X, Y, NPOINTS)
    IMPLICIT NONE
    REAL, INTENT (IN) :: X(:), Y(:)
    INTEGER, INTENT(IN) :: NPOINTS
    REAL, DIMENSION(NPOINTS) :: CK
  END FUNCTION CK

  FUNCTION LAGRANGE(XP, X, LCOEF, NPOINTS)
    IMPLICIT NONE
    REAL :: LAGRANGE
    REAL, INTENT(IN) :: XP, X(:), LCOEF(:)
    INTEGER, INTENT(IN) :: NPOINTS
  END FUNCTION LAGRANGE
END INTERFACE

END MODULE LAGRANGE_INTERFACE
```

**modules/lagrange_interface.f90**

# *Polynominal Interpolation*

## *Linear Interpolation*




***Linear interpolation*** *between the first two values can be written as:*

$$P_{0,1}(x) = y_0 \frac{x - x_1}{x_0 - x_1} + y_1 \frac{x - x_0}{x_1 - x_0} = \frac{y_0(x - x_1) + y_1(x_0 - x)}{x_0 - x_1}$$

# *Polynominal Interpolation*

***Linear interpolation*** *between the first two values can be written as:*

$$P_{0,1}(x) = y_0 \frac{x - x_1}{x_0 - x_1} + y_1 \frac{x - x_0}{x_1 - x_0} = \frac{P_0(x - x_1) + P_1(x_0 - x)}{x_0 - x_1}$$

$$P_{1,2}(x) = y_1 \frac{x - x_2}{x_1 - x_2} + y_2 \frac{x - x_1}{x_2 - x_1} = \frac{P_1(x - x_2) + P_2(x_1 - x)}{x_1 - x_2}$$

***Let's look at the following term:***

$$\frac{x - x_2}{x_0 - x_2} \cdot P_{0,1}(x) + \frac{x_0 - x}{x_0 - x_2} \cdot P_{1,2}(x) = y_0 \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + y_1 \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_0 - x_2)}$$

$$+ y_1 \frac{(x_0 - x)(x - x_2)}{(x_0 - x_2)(x_1 - x_2)} + y_2 \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

# *Polynominal Interpolation*

$$\frac{x-x_2}{x_0-x_2}\cdot P_{0,1}(x)+\frac{x_0-x}{x_0-x_2}\cdot P_{1,2}(x)=y_0\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}+y_1\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_0-x_2)}$$

$$+y_1\frac{(x_0-x)(x-x_2)}{(x_0-x_2)(x_1-x_2)}+y_2\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$

$$=y_0\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}+y_1\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}+y_2\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$

$$=P_{0,1,2}(x)$$

$$\boxed{P_{0,1,2}(x)=\frac{(x-x_2)P_{0,1}(x)+(x_0-x)P_{1,2}(x)}{x_0-x_2}}$$

# *Polynominal Interpolation*

$$P_{0,1}(x) = \frac{P_0(x - x_1) + P_1(x_0 - x)}{x_0 - x_1}$$

$$P_{0,1,2}(x) = \frac{(x - x_2)P_{0,1}(x) + (x_0 - x)P_{1,2}(x)}{x_0 - x_2}$$

$$P_{0,1,2,3}(x) = \frac{(x - x_3)P_{0,1,2}(x) + (x_0 - x)P_{1,2,3}(x)}{x_0 - x_3}$$

***Interpolating Polynominals can be recursively calculated***

# *Neville's Algorithm*

$x_0 \qquad y_0 = P_0(x)$

# *Neville's Algorithm*

$x_0$ $\qquad y_0 = P_0(x)$

$$P_{12}(x) = \frac{(x - x_2)P_1(x) + (x_1 - x)P_2(x)}{x_1 - x_2}$$

***1st Order Polynominal***

# *Neville's Algorithm*

$x_0$ $\quad y_0 = P_0(x)$

$$P_{123}(x) = \frac{(x - x_3)P_{12}(x) + (x_1 - x)P_{23}(x)}{x_1 - x_3}$$

$\boxed{\textit{2}^{nd}\textit{ Order Polynominal}}$

# *Neville's Algorithm*

$x_0 \qquad y_0 = P_0(x)$

$$P_{1234}(x) = \frac{(x - x_4)P_{123}(x) + (x_1 - x)P_{234}(x)}{x_1 - x_4}$$

$\boxed{\textbf{\textit{3}}^{\textbf{\textit{rd}}}\ \textbf{\textit{Order Polynominal}}}$

# *Neville's Algorithm*

$x_0 \qquad y_0 = P_0(x)$

$$P_{01234}(x) = \frac{(x - x_4)P_{0123}(x) + (x_0 - x)P_{1234}(x)}{x_0 - x_4}$$

$\boxed{\textbf{\textit{4}}^{\textbf{\textit{th}}}\ \textbf{\textit{Order Polynominal}}}$