Henon Research

E-Mail : Info@HenonResearch.com
Http: www.HenonResearch.com

This Add-On Instruction Solve the equations System whit Gauss-Jordan Reduction

In the matrix A [i , j] put the System of N-Equation.
In the vector b put the solutions

Example 1: Linear System 3 equation (X,Y,Z)

$$\left\{\begin{matrix} 3X + 2Y - Z = 10 \\ -X + Y + Z = -2 \\ 2X - Y + 2Z = -6 \end{matrix}\right\}$$

$$\begin{vmatrix} 3 & 2 & -1 \\ -1 & 1 & 1 \\ 2 & -1 & 2 \end{vmatrix} \ldots \begin{vmatrix} 10 \\ -2 \\ -6 \end{vmatrix}$$

$$X\begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix} + Y\begin{bmatrix} 2 \\ 1 \\ -1 \end{bmatrix} + Z\begin{bmatrix} -1 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 10 \\ -2 \\ -6 \end{bmatrix}$$

Matrix A :=
Matrix[1,1]= 3 ; Matrix[1,2]= 2 ; Matrix[1,3]= -1
Matrix[2,1]= -1 ; Matrix[2,2]= 1 ; Matrix[2,3]= 1
Matrix[3,1]= 2 ; Matrix[3,2]= -1 ; Matrix[3,3]= 2

Vector b:=
Vector[1] =10 ; Vector[2] = -2 ; Vector[3] = -6 ;

Solution :=
Solution [1] := 1.0 ; Solution [2] := 2.0 ; Solution [3] := -3.0 ;
X = 1 ; Y = 2 ; Z = -3

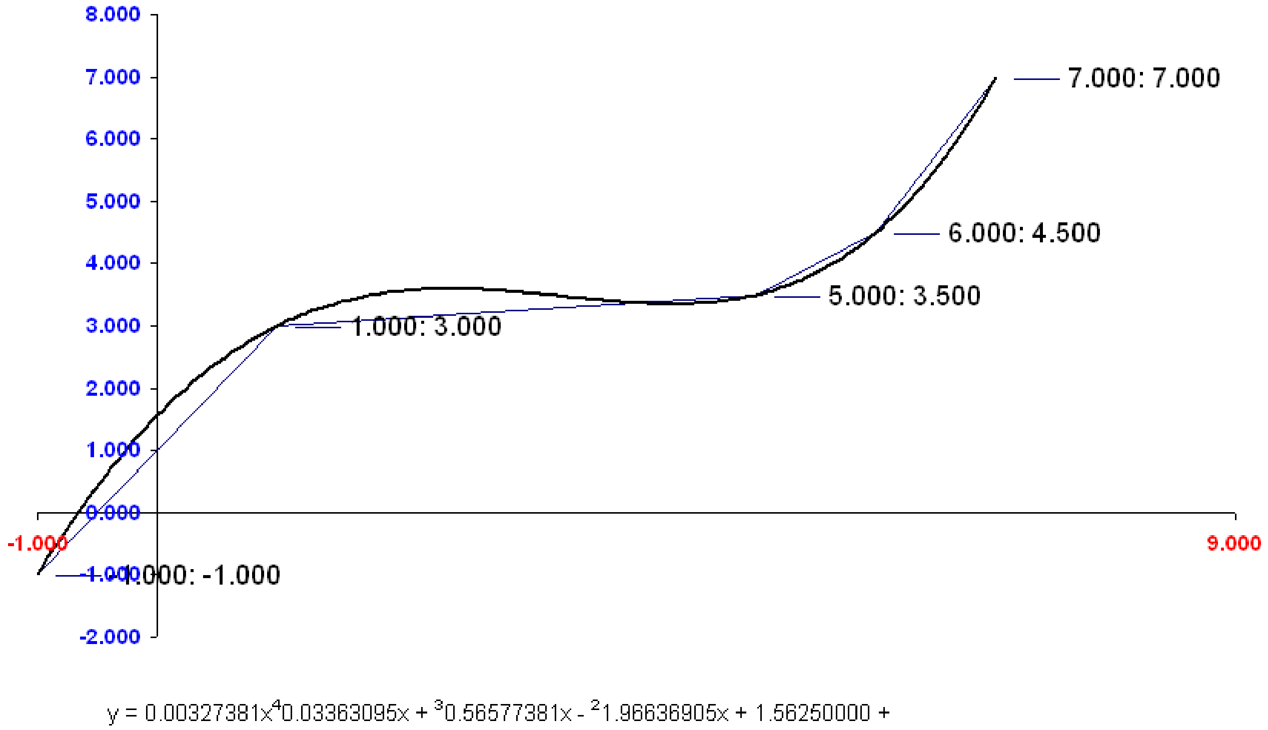
Example 2: Linear System 5 equation for resolve Polynomial 4th grade
example. Polynomial whit 5 points:

P0(-1,-1) ;
P1(1, 3) ;
P2(5, 3.5) ;
P3(6, 4.5) ;
P4(7, 7) ;

Write in the Matrix A [i, j]
Matrix A :=
Matrix[1,1]= (-1)^4 ; Matrix[1,2]= (-1)^3 ; Matrix[1,3]= (-1)^2 ; Matrix[1,4]= (-1) ; Matrix[1,5]=1;
Matrix[2,1]= (1)^4 ; Matrix[2,2]= (1)^3 ; Matrix[2,3]= (1)^2 ; Matrix[2,4]= (1) ; Matrix[2,5]=1;
Matrix[3,1]= (5)^4 ; Matrix[3,2]= (5)^3 ; Matrix[3,3]= (5)^2 ; Matrix[3,4]= (5) ; Matrix[3,5]=1;
Matrix[4,1]= (6)^4 ; Matrix[4,2]= (6)^3 ; Matrix[4,3]= (6)^2 ; Matrix[4,4]= (6) ; Matrix[4,5]=1;
Matrix[5,1]= (7)^4 ; Matrix[5,2]= (7)^3 ; Matrix[5,3]= (7)^2 ; Matrix[5,4]= (7) ; Matrix[5,5]=1;

Write in the Vector [ ]
Vector b:=
Vector[1] = -1 ; Vector[2] = 3 ; Vector[3] = 3.5 ; Vector[4] = 4.5 ; Vector[5] = 7

Solutions :=
Solution [1] := 3.27380234e-003 ; Solution [2] := 0.03363105;
Solution [3] := -0.56577414 ; Solution [4] := 1.966369 ;
Solution [5] := 1.5625005



§§§

Example 3: Linear System 6 equation for resolve Polynomial 5th grade
example. Mototion Interpolation whit Polynomial
whit 2 Points :

P0 (Time0, Position 0) Start point whit (Velocity 0, Acceleration 0)
P1 (Time1, Position 1) End point whit (Velocity 1, Acceleration 1)

Write in the Matrix A [i, j]
Matrix A :=
X0 = time0 ; X1 = time1

$$\begin{bmatrix} X0^5 & X0^4 & X0^3 & X0^2 & X0 & 1 \\ 5X0^4 & 4X0^3 & 3X0^2 & 2X0 & 1 & 0 \\ 20X0^3 & 12X0^2 & 6X0 & 2 & 0 & 0 \\ X1^5 & X1^4 & X1^3 & X1^2 & X1 & 1 \\ 5X1^4 & 4X1^3 & 3X1^2 & 2X1 & 1 & 0 \\ 20X1^3 & 12X1^2 & 6X1 & 2 & 0 & 0 \end{bmatrix} \cdots \begin{bmatrix} Pos0 \\ Vel0 \\ Acc0 \\ Pos1 \\ Vel1 \\ Acc1 \end{bmatrix} \equiv \begin{bmatrix} Position\_at\_P0 \\ Velocity\_at\_P0 \\ Acc\_at\_P0 \\ Position\_at\_p1 \\ Velocity\_at\_P1 \\ Acc\_at\_P1 \end{bmatrix}$$

Vector b:=
Vector[1] = Position P0 ; Vector[2] = Velocity P0 ; Vector[3] = Acceleration P0 ;
Vector[4] = Position P1 ; Vector[5] = Velocity P1 ; Vector[6] = Acceleration P1 ;

Interpolation Polynomial Position :=

$$Pos := a*t^5 + b*t^4 + c*t^3 + d*t^2 + e*t + f$$

AB-Logix Add-On Instruction :

Parameters

Add-On Instruction Definition - Solve_Gauss_Sys v2.1 HR

General | Parameters | Local Tags | Scan Modes | Signature | Change History | Help

Name	Usage	Data Type	Alias For	Default	Style	Req	Vis	Description
EnableIn	Input	BOOL		1	Decimal	☐	☐	Enable Input - System Defined Parameter
EnableOut	Output	BOOL		0	Decimal	☐	☐	Enable Output - System Defined Parameter
+ Num_Equ	Input	INT		2	Decimal	☑	☑	Number of equation
+ Matrix	InOut	REAL[10,10]			Float	☑	☑	Matrix input
+ Vector	InOut	REAL[10]			Float	☑	☑	Vectror input
+ Matrix_Solve	InOut	REAL[10,10]			Float	☑	☑	Matrix when finish Gauss reduction
+ Vector_Solve	InOut	REAL[10]			Float	☑	☑	Vectror when finish Gauss reduction
+ Solution	InOut	REAL[10]			Float	☑	☑	Solutions of System
						☐	☐	

Local tags

General | Parameters | Local Tags | Scan Modes | Signature | Change History | Help

Name	Data Type	Default	Style	Description
+ a	REAL[20]	{...}	Float	Aux Array for PIVOTING
+ ai	DINT	0	Decimal	Pointer of Max Value of Element on Matrix
ax	REAL	0.0	Float	Aux For PIVOTING Vector
ay	REAL	0.0	Float	Aux For PIVOTING Vector
+ b	REAL[20]	{...}	Float	Aux Array for PIVOTING
Coeff_m	REAL	0.0	Float	
+ Im	DINT	0	Decimal	Pointer Aux for Compute
+ jm	DINT	0	Decimal	Pointer Aux for Compute
KMax	REAL	0.0	Float	Max value for Element on Colum Matrix
+ N	DINT	0	Decimal	Number of Equation / 2
+ Null_Array	REAL[20]	{...}	Float	Array with every elements to Zero
+ Sm	DINT	0	Decimal	Pointer Aux for Compute
Sum	REAL	0.0	Float	Sum of Matrix * Det ǀAǀ
+ tm	DINT	0	Decimal	Pointer Aux for Compute
+ Xi	DINT	0	Decimal	Pointer Aux for Compute

Logic Source:

```
(*      ###################################################
        #    SOLVE    MATRIX-SYSTEM   GAUSS  REDUCTION    #
        ###################################################
*)

// Save the Original Matrix

COP(Matrix[0,0],Matrix_Solve[0,0],100);
COP(Vector[0],Vector_Solve[0],10);

N := Num_Equ;  // Number of Equations

tm := 1;

While tm <= N do


KMax := -1.0e+015;   (*  Reset KMax *)

 (* Find max Row value *)
    FOR Sm := tm TO N DO;
      if Matrix_Solve[Sm,tm] > KMax then
                                    KMax := Matrix_Solve[Sm,tm];
                                 end_If;
    END_FOR;

    (* if the max find value is Zero, search the min value *)
    if KMax = 0 then
      FOR Sm := tm TO N DO;
       if abs(Matrix_Solve[Sm,tm]) > KMax then
                                          KMax := -Abs(Matrix_Solve[Sm,tm]);
                                       end_If;
      END_FOR;
    end_if;

    (*if in the row all values are zero By-pass *)
    if KMax <>0 then
                 Xi :=tm;
            (* Mark this position *)
            if KMax > 0 then
              FOR Xi := 1 TO N DO;
                      if Matrix_Solve[Xi,tm] = KMax then
                                                  ai := Xi;
                                                end_If;
              END_FOR;
            end_if;

            if KMax = 0 then
                While KMax=Abs(Matrix_Solve[Xi,tm]) do
                     if abs(Matrix_Solve[Xi,tm]) = KMax then
                                                       ai := Xi;
                                                   end_If;

                     Xi := Xi + 1;
                End_While;
            end_if;

            if ai > tm then   (* Pivoting Matrix Rows *)
                    FOR jm := 1 to N Do;
                        a[jm]:=Matrix_Solve[tm,jm];
                        b[jm]:=Matrix_Solve[ai,jm];
                        Matrix_Solve[tm,jm] := b[jm];
                        Matrix_Solve[ai,jm] := a[jm];
                    end_for;


           (*  Pivoting vectors  *)
                    ax := Vector_Solve[tm];
                    ay := Vector_Solve[ai];
                    Vector_Solve[tm] := ay;
                    Vector_Solve[ai] := ax;

                    COP(Null_Array[0],a[0],12);
                    COP(Null_Array[0],b[0],12);
```

```
            end_if;

        (*  Reduce th-Step  *)


        FOR Im := (tm+1) TO N DO;
            Coeff_m := Matrix_Solve[Im,tm] / Matrix_Solve[tm,tm];
            Vector_Solve[Im] := Vector_Solve[Im] - (Coeff_m * Vector_Solve[tm]);
            FOR jm := tm TO N DO
                Matrix_Solve[Im,jm] := Matrix_Solve[Im,jm] - (Coeff_m * Matrix_Solve[tm,jm]);
            end_For;
        END_FOR;

    end_If;     (* Label By-Pass *)


    tm := tm+1; (* Increase Exam Row *)


end_while;



(*  Calcolate Solutions *)


 tm := N ;
 Solution[N] := Vector_Solve[N]/Matrix_Solve[N,N];

 While tm >= 1 do
    Sum := 0;
    For jm := (tm+1) to N do
        Sum := Sum + (Matrix_Solve[tm,jm] * Solution[jm]);
    end_For;
    Solution[tm] := (Vector_Solve[tm]- Sum) / Matrix_Solve[tm,tm];
    tm := tm - 1;
 end_while;
```

Siemens S7-300 SCL Source:

Test whit linear System 5 equation for resolve Polynomial 4th grade

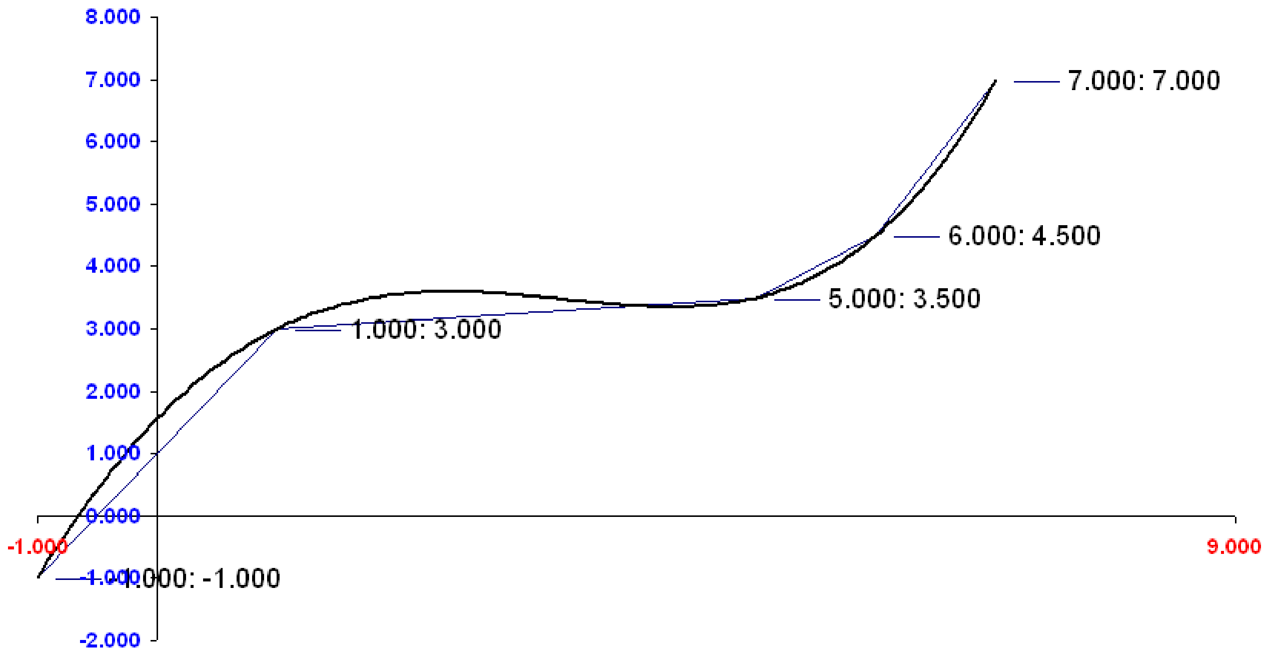
example. Polynomial whit 5 points:

P0(-1,-1) ;
P1(1, 3) ;
P2(5, 3.5) ;
P3(6, 4.5) ;
P4(7, 7) ;

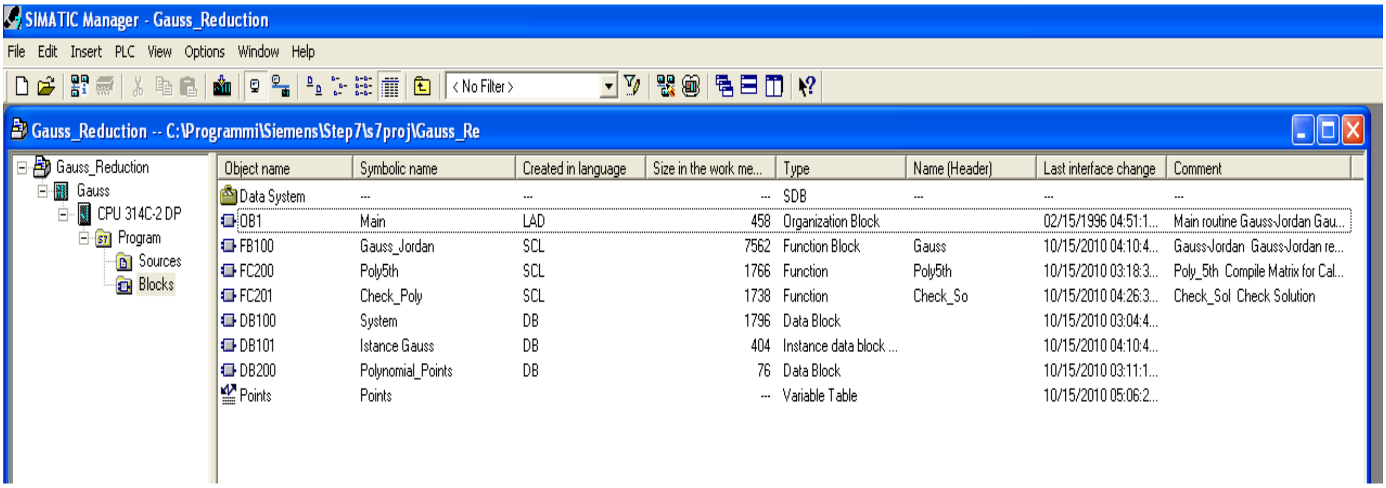
Write in the Matrix A [i, j]
Matrix A :=
Matrix[1,1]= (-1)^4 ; Matrix[1,2]= (-1)^3 ; Matrix[1,3]= (-1)^2 ; Matrix[1,4]= (-1) ; Matrix[1,5]=1;
Matrix[2,1]= (1)^4   ; Matrix[2,2]= (1)^3 ; Matrix[2,3]= (1)^2 ; Matrix[2,4]= (1) ; Matrix[2,5]=1;
Matrix[3,1]= (5)^4   ; Matrix[3,2]= (5)^3 ; Matrix[3,3]= (5)^2 ; Matrix[3,4]= (5) ; Matrix[3,5]=1;
Matrix[4,1]= (6)^4   ; Matrix[4,2]= (6)^3 ; Matrix[4,3]= (6)^2 ; Matrix[4,4]= (6) ; Matrix[4,5]=1;
Matrix[5,1]= (7)^4   ; Matrix[5,2]= (7)^3 ; Matrix[5,3]= (7)^2 ; Matrix[5,4]= (7) ; Matrix[5,5]=1;

Write in the Vector [ ]
Vector b:=
Vector[1] = -1 ; Vector[2] = 3 ; Vector[3] = 3.5 ; Vector[4] = 4.5 ; Vector[5] = 7

Solutions :=
Solution [1] := 3.27380234e-003 ; Solution [2] := 0.03363105;
Solution [3] := -0.56577414 ; Solution [4] := 1.966369 ;
Solution [5] := 1.5625005



y = 0.00327381x⁴0.03363095x + ³0.56577381x - ²1.96636905x + 1.56250000 +

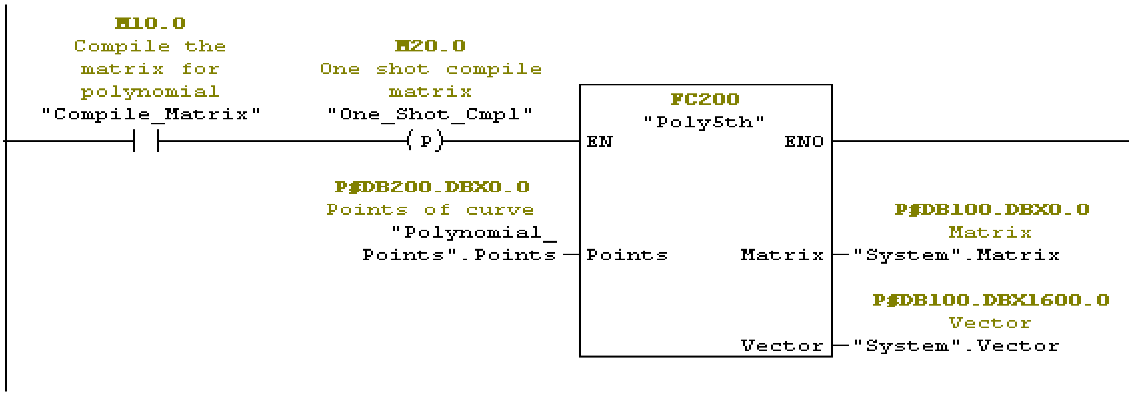


OB1 : Main routine Gauss-Jordan

Gauss Jordan reduction system

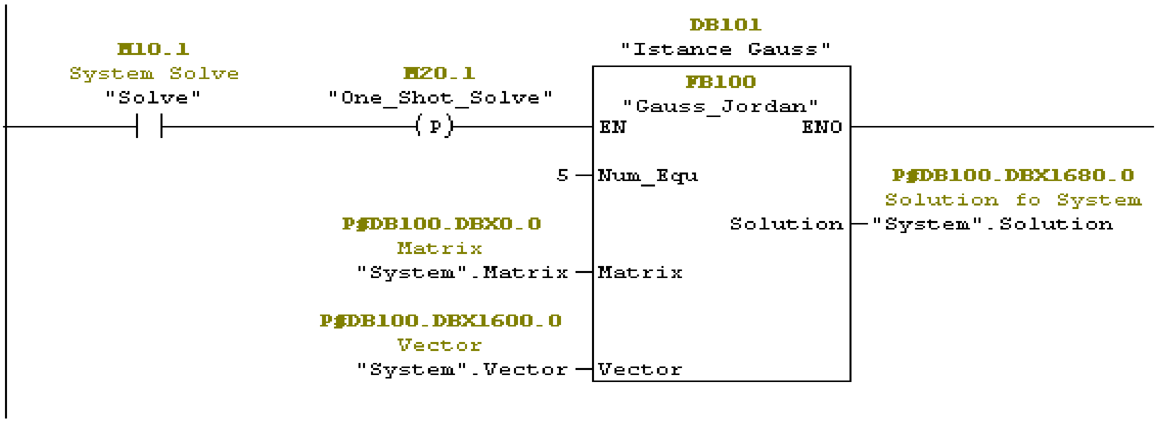
Network 1 : Compile Matrix & vector for Polynomial curve

Comment:



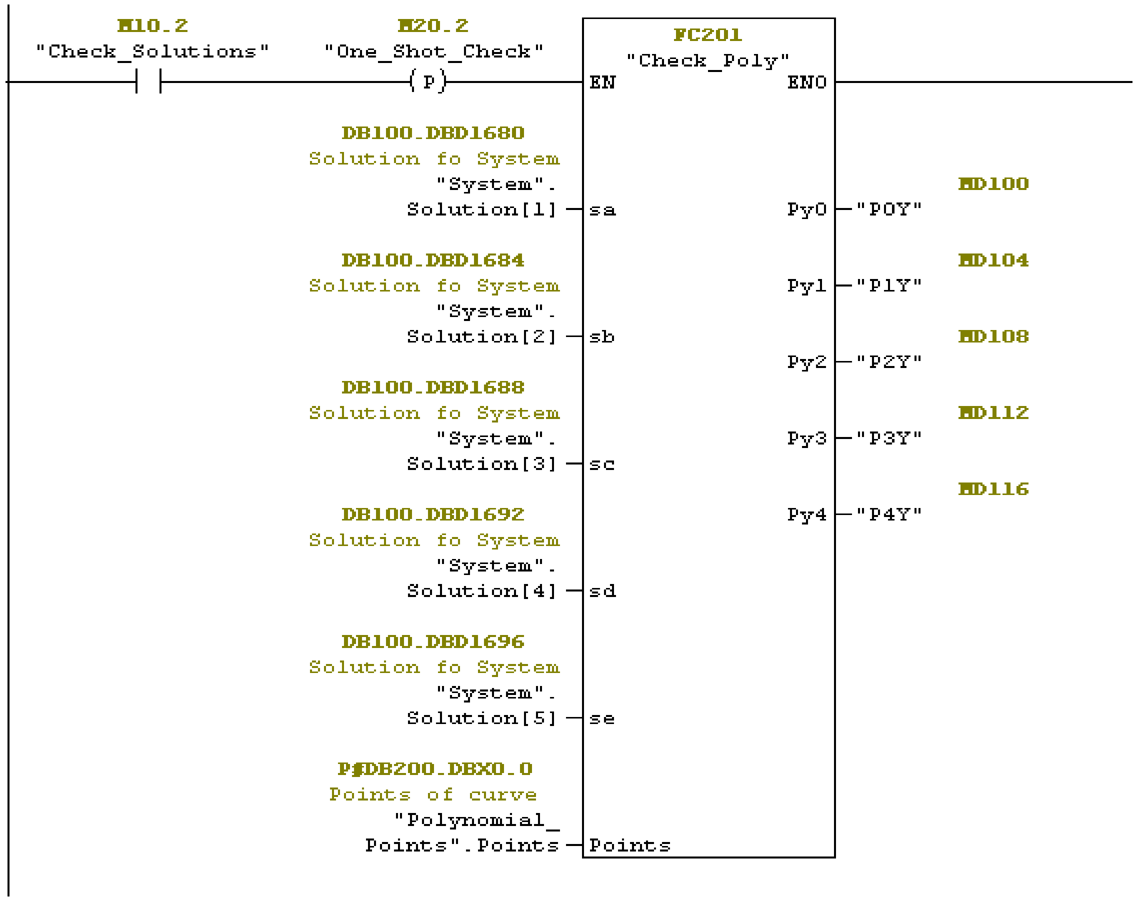
Network 2 : System Solve

Comment:



Network 3 : Check Solutions

Comment:



DB 100

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	Matrix	ARRAY[1..20,1..20]		Matrix
*4.0		REAL		
+1600.0	Vector	ARRAY[1..20]		Vector
*4.0		REAL		
+1680.0	Solution	ARRAY[1..20]		Solution fo System
*4.0		REAL		
=1760.0		END_STRUCT		

DB 200

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	Points	ARRAY[0..4,0..1]	0.000000e+000	Points of curve
*4.0		REAL		
=40.0		END_STRUCT		

Gauss-Jordan Reduction S7 SCL-Source:

```
FUNCTION_BLOCK FB100

TITLE = 'Gauss-Jordan'
//
// Gauss-Jordan reduction system
//
VERSION: '1.1'
AUTHOR: Beaty
NAME:   Gauss
FAMILY: System

VAR_INPUT
     Num_Equ:       INT;                        // Number of equation
end_var

VAR_IN_OUT
    Matrix:         ARRAY[1..20,1..20] OF REAL;      // Matrix input
    Vector:         ARRAY[1..20] OF REAL;            // Vector input
end_var

VAR
    Null_Array:     ARRAY[1..20] OF REAL;            // Array with every elements to Zero
    av,bv:          ARRAY[1..20] OF REAL;
    ax,ay:          REAL;                            // Aux For PIVOTING Vector
    Sum:            REAL;                            // Sum of Matrix * Det |A|
    KMax:           REAL;                            // Max value for Element on Colum Matrix
    Coeff_m:        REAL;                            //
    N:              INT;                             // Number of Equations
    tm,Sm,Xi,ai:    INT;
    jm,Im:          INT;
end_var

VAR_OUTPUT
    Solution:       ARRAY[1..20] OF REAL;            // Solutions of System
end_var



(*     ##########################################################
       #    SOLVE    MATRIX-SYSTEM   GAUSS-JORDAN  REDUCTION    #
       ##########################################################
*)

// Save the Original Matrix


N := Num_Equ;  // Number of Equations

tm := 1;

While tm <= N do


KMax := -1.0e+015;    (*  Reset KMax *)

 (* find max Row value *)
    FOR Sm := tm TO N DO;
      if Matrix[Sm,tm] > KMax then
         KMax := Matrix[Sm,tm];
      end_If;
    END_FOR;

    (* if the max find value is Zero, search the min value*)
    if KMax = 0 then
      FOR Sm := tm TO N DO;
       if Abs(Matrix[Sm,tm]) > KMax then
         KMax := -Abs(Matrix[Sm,tm]);
       end_If;
      END_FOR;
    end_if;
```

```
    (* if in the row all values are zero By-pass*)
    if KMax <>0 then
               Xi :=tm;
            (* Mark this position*)
            if KMax > 0 then
              FOR Xi := 1 TO N DO;
                   if Matrix[Xi,tm] = KMax then
                ai := Xi;
               end_If;
              END_FOR;
            end_if;

            if KMax = 0 then
                While KMax=Abs(Matrix[Xi,tm]) do
                     if Abs(Matrix[Xi,tm]) = KMax then
                      ai := Xi;
                     end_If;
                     Xi := Xi + 1;
                End_While;
            end_if;

            if ai > tm then  (* Pivoting Matrix Rows *)
                    FOR jm := 1 TO N DO;
                        av[jm]:=Matrix[tm,jm];
                        bv[jm]:=Matrix[ai,jm];
                        Matrix[tm,jm] := bv[jm];
                        Matrix[ai,jm] := av[jm];
                    end_for;
                (* Pivoting vactors *)

                ax := Vector[tm];
                ay := Vector[ai];
                Vector[tm] := ay;
                Vector[ai] := ax;

            end_if;


        (*  Reduction th-step*)

        FOR Im := (tm+1) TO N DO;
            Coeff_m := Matrix[Im,tm] / Matrix[tm,tm];
            Vector[Im] := Vector[Im] - (Coeff_m * Vector[tm]);
            FOR jm := tm TO N DO
                Matrix[Im,jm] := Matrix[Im,jm] - (Coeff_m * Matrix[tm,jm]);
            end_For;
        END_FOR;

    end_If;    (* Label By-pass *)


    tm := tm+1; (* Increase Exam Row *)

end_while;


(*  Calculate Solutions *)

 tm := N ;
 Solution[N] := Vector[N]/Matrix[N,N];

 While tm >= 1 do
    Sum := 0;
    For jm := (tm+1) to N do
        Sum := Sum + (Matrix[tm,jm] * Solution[jm]);
    end_For;
    Solution[tm] := (Vector[tm]- Sum) / Matrix[tm,tm];
    tm := tm - 1;
 end_while;


END_FUNCTION_BLOCK
```

Polynomial 4th Matrix Compile S7 SCL-Source:

```
FUNCTION FC200 : Void


TITLE = 'Poly_5th'
//
// Compile Matrix
// for Calculate Polynomial 5th curve
//
VERSION: '1.1'
AUTHOR: Beaty
NAME:   Poly5th
FAMILY: System


VAR_INPUT
  Points:                 ARRAY[0..4,0..1] OF REAL;       //Points of Polynomial 5th grade
End_var

VAR_OUTPUT
  Matrix:                 ARRAY[1..20,1..20] OF REAL;     // Matrix Compile
  Vector:                 ARRAY[1..20] OF REAL;           // Vector Compile
End_var

VAR_TEMP
  P0_Xm,P1_Xm,P2_Xm,
  P3_Xm,P4_Xm           :     REAL;
End_var


P0_Xm := Points[0,0];
P1_Xm := Points[1,0];
P2_Xm := Points[2,0];
P3_Xm := Points[3,0];
P4_Xm := Points[4,0];



(*   1st Row *)
Matrix[1,1]  := P0_Xm*P0_Xm*P0_Xm*P0_Xm;
Matrix[1,2]  := P0_Xm*P0_Xm*P0_Xm;
Matrix[1,3]  := P0_Xm*P0_Xm;
Matrix[1,4]  := P0_Xm ;
Matrix[1,5]  := 1 ;
Matrix[1,6]  := 0 ;
Matrix[1,7]  := 0 ;
Matrix[1,8]  := 0 ;

(*   2nd Row *)
Matrix[2,1]  := P1_Xm*P1_Xm*P1_Xm*P1_Xm;
Matrix[2,2]  := P1_Xm*P1_Xm*P1_Xm;
Matrix[2,3]  := P1_Xm*P1_Xm;
Matrix[2,4]  := P1_Xm ;
Matrix[2,5]  := 1 ;
Matrix[2,6]  := 0 ;
Matrix[2,7]  := 0 ;
Matrix[2,8]  := 0 ;

(*    3rd Row *)
Matrix[3,1]  := P2_Xm*P2_Xm*P2_Xm*P2_Xm;
Matrix[3,2]  := P2_Xm*P2_Xm*P2_Xm;
Matrix[3,3]  := P2_Xm*P2_Xm;
Matrix[3,4]  := P2_Xm ;
Matrix[3,5]  := 1 ;
Matrix[3,6]  := 0 ;
Matrix[3,7]  := 0 ;
Matrix[3,8]  := 0 ;

(*    4th Row *)
Matrix[4,1]  := P3_Xm*P3_Xm*P3_Xm*P3_Xm;
Matrix[4,2]  := P3_Xm*P3_Xm*P3_Xm;
Matrix[4,3]  := P3_Xm*P3_Xm;
Matrix[4,4]  := P3_Xm ;
Matrix[4,5]  := 1 ;
Matrix[4,6]  := 0 ;
Matrix[4,7]  := 0 ;
Matrix[4,8]  := 0 ;
```

```
(*    5th Row *)
Matrix[5,1] := P4_Xm*P4_Xm*P4_Xm*P4_Xm;
Matrix[5,2] := P4_Xm*P4_Xm*P4_Xm;
Matrix[5,3] := P4_Xm*P4_Xm;
Matrix[5,4] := P4_Xm ;
Matrix[5,5] := 1 ;
Matrix[5,6] := 0 ;
Matrix[5,7] := 0 ;
Matrix[5,8] := 0 ;

(*  Vector Compile  *)
Vector[1] := Points[0,1];
Vector[2] := Points[1,1];
Vector[3] := Points[2,1];
Vector[4] := Points[3,1];
Vector[5] := Points[4,1];
Vector[6] := 0.0;
Vector[7] := 0.0;
Vector[8] := 0.0;


END_FUNCTION
```

Check Results for Polynomial 4th S7 SCL-Source:

```
FUNCTION FC201 : VOID


TITLE = 'Check_Sol'
//
// Check Solution
//
VERSION: '1.1'
AUTHOR:  Beaty
NAME:    Check_Solution
FAMILY:  System

VAR_INPUT
 sa,sb,sc,sd,se:            REAL;
 Points:                    ARRAY[0..4,0..1] OF REAL;
End_var

VAR_OUTPUT
 Py0,Py1,Py2,Py3,Py4:       REAL;
end_var

VAR
 P0_Xm,P1_Xm,P2_Xm,P3_Xm,
 P4_Xm:                     REAL;
end_var


P0_Xm:= Points[0,0];
P1_Xm:= Points[1,0];
P2_Xm:= Points[2,0];
P3_Xm:= Points[3,0];
P4_Xm:= Points[4,0];



// Check the Solution for matrix 5Poly

Py0 := sa * P0_Xm * P0_Xm * P0_Xm * P0_Xm +
       sb * P0_Xm * P0_Xm * P0_Xm +
       sc * P0_Xm * P0_Xm +
       sd * P0_Xm +
       se;

Py1 := sa * P1_Xm * P1_Xm * P1_Xm * P1_Xm +
       sb * P1_Xm * P1_Xm * P1_Xm +
       sc * P1_Xm * P1_Xm +
       sd * P1_Xm +
       se;

Py2 := sa * P2_Xm * P2_Xm * P2_Xm * P2_Xm +
       sb * P2_Xm * P2_Xm * P2_Xm +
       sc * P2_Xm * P2_Xm +
       sd * P2_Xm +
       se;

Py3 := sa * P3_Xm * P3_Xm * P3_Xm * P3_Xm +
       sb * P3_Xm * P3_Xm * P3_Xm +
       sc * P3_Xm * P3_Xm +
       sd * P3_Xm +
       se;

Py4 := sa * P4_Xm * P4_Xm * P4_Xm * P4_Xm +
       sb * P4_Xm * P4_Xm * P4_Xm +
       sc * P4_Xm * P4_Xm +
       sd * P4_Xm +
       se;

END_FUNCTION
```

Variables Table for S7 program:

Var - Points

Table Edit Insert PLC Variable View Options Window Help

Points -- @Gauss_Reduction\Gauss\CPU 314C-2 DP\Program ONLINE

	Address	Symbol	Display format	Status value	Modify value
1	DB200.DBD 0	"Polynomial_Points".Points[0, 0]	FLOATING_POINT	-3.0	-3.0
2	DB200.DBD 4	"Polynomial_Points".Points[0, 1]	FLOATING_POINT	-5.67	-5.67
3	DB200.DBD 8	"Polynomial_Points".Points[1, 0]	FLOATING_POINT	-1.456	-1.456
4	DB200.DBD 12	"Polynomial_Points".Points[1, 1]	FLOATING_POINT	-2.34567	-2.34567
5	DB200.DBD 16	"Polynomial_Points".Points[2, 0]	FLOATING_POINT	5.0	
6	DB200.DBD 20	"Polynomial_Points".Points[2, 1]	FLOATING_POINT	3.5	
7	DB200.DBD 24	"Polynomial_Points".Points[3, 0]	FLOATING_POINT	6.0	
8	DB200.DBD 28	"Polynomial_Points".Points[3, 1]	FLOATING_POINT	4.5	
9	DB200.DBD 32	"Polynomial_Points".Points[4, 0]	FLOATING_POINT	7.0	
10	DB200.DBD 36	"Polynomial_Points".Points[4, 1]	FLOATING_POINT	7.0	
11					
12	// Compile				
13	M 10.0	"Compile_Matrix"	BOOL	false	
14					
15	// Solve				
16	M 10.1	"Solve"	BOOL	false	
17					
18	// Solution				
19	DB100.DBD 1680	"System".Solution[1]	FLOATING_POINT	0.006845836	
20	DB100.DBD 1684	"System".Solution[2]	FLOATING_POINT	-0.02606254	
21	DB100.DBD 1688	"System".Solution[3]	FLOATING_POINT	-0.2664205	
22	DB100.DBD 1692	"System".Solution[4]	FLOATING_POINT	1.708763	
23	DB100.DBD 1696	"System".Solution[5]	FLOATING_POINT	0.5958714	
24					
25	// Control				
26	M 10.2	"Check_Solutions"	BOOL	false	
27					
28	// Check				
29	MD 100	"P0Y"	FLOATING_POINT	-5.670001	
30	MD 104	"P1Y"	FLOATING_POINT	-2.34567	
31	MD 108	"P2Y"	FLOATING_POINT	3.5	
32	MD 112	"P3Y"	FLOATING_POINT	4.500002	
33	MD 116	"P4Y"	FLOATING_POINT	7.000002	
34					