

MATHCAD'S PROGRAM FUNCTION and APPLICATION IN TEACHING OF MATH

DE TING WU

Department of Math
Morehouse College
Atlanta, GA.30314, USA
dtwu@morehouse.edu

1. Introduction

1.1 About Mathcad

Mathcad is one of popular computer algebra system (math software) in the world. Like other CAS's, it has the capabilities to perform algebraic operations, calculus operations and draw graph of 2 or 3 dimensions. We can use it to get numerical, symbolic and graphic solution of math problem. Besides above common capabilities, it has some feature: its words processor function is better than other CAS. This feature makes it look like a scientific words processor and it easy to create a worksheet.

Following current trend to add program function to math software, it added program function to itself, started in Mathcad V7. This improvement makes its function more complete and more powerful.

1.2 Role of program function in teaching of Math.

Without doubt, math softwares and technology are helpful aide in teaching of Math. They are effecting the teaching of Math profoundly and it can be expected the emerging newer technologies will penetrate and reconfigure the teaching of Math.

The program function is a new function of math software. It is natural we want to know whether or not it is useful in teaching of Math? I think: although for most problems we meet in teaching of Math it is enough to use computation ability, in certain cases the program function becomes necessary and very helpful. For example, for Numerical Analysis the program function is very useful but for Abstract Algebra it is less necessary. Another example is Differential Equations, when teaching approximate method for a solution of a differential equation the program function is necessary and teaching existence and uniqueness of the solution of a differential equation it is not necessary. Moreover, as the power of program function is increasing the more applications will be found.

1.3 About this presentation

This presentation is to introduce the program function of Mathcad and illustrate some of application in teaching of Math which is based on my teaching experience in past years.

2. Program function of Mathcad

In this part the program operators of Mathcad and the structure of Mathcad program are introduced.

2.1 Program operators of Mathcad

Mathcad program is created by using its program operators. Mathcad has 10 operators which are accessible from Program toolbar. Now, let's take a look of these operators.

1. "add line": operator to initiate a program or add a line to a program. When click it on the programming toolbar a vertical bar and 2 placeholders will be produced.
2. " \leftarrow ": local assign operator which define left holder by right holder of arrow.
3. " if ": conditional operator which can be used whenever you want a program statement to be executed only upon the occurrence of some condition.
4. " otherwise ": conditional operator which can be used together with "if" operator when you want a program to execute a statement when the condition of "if" operator is false.
5. " $\text{for } \in$ ": looping operator which can be used when you know how many times you want a program statement to be executed repeatedly.
6. " while ": looping operator which can be used when you want to stop execution of a statement upon the occurrence of a condition but don't know exactly when the condition will occur.
7. " break ": Control operator which can be used when you want to break out of a loop upon the occurrence of some condition and which moves execution to next statement outside the loop.
8. " continue ": Control operator which can be used when you want to halt current iteration of a loop upon the occurrence of a condition and force program to continue next iteration of a loop.
9. " return ": Control operator which can be used anywhere in a program when you want to interrupt a program and to return a value different from the value of last expression of a program.
10. " on error ": Operator for error control which can be used to return an alternative value when an error happened in an expression.

2.2 Structure of Mathcad program

A program is a precise sequence of instructions designed to accomplish a specific task. Mathcad program is a sequence of statements created by using program operators, too. But, it has special form and structure. Now, we use an example to illustrate.

Example of Mathcad program

$$f(x, y) := \left| \begin{array}{l} u \leftarrow \sqrt{x^2 + y^2} \\ v \leftarrow \sin(u) + e^{-u} \\ w \leftarrow 1 + u \\ \frac{v}{w} \end{array} \right.$$

Left is a Mathcad program which defines a function in 2 variable. This program consists of 2 parts. Left part is name or variable name which represent the output of last statement in the program; Right part is included in a vertical bar which is the body of the program comprised of program operators. Both parts are connected by the definition sign " $:=$ ". This means Left part is defined by right part or the program assigns the result to the name on the left. In fact, the program just is a compound expression which is represented by name on the left.

The form and structure of Mathcad program make it easy to recognize and separate from other part of worksheet. The program operators are simple so that writing a program is simple. This is very suitable for teaching purpose. This form and structure is good for revising of program too. Also, its form and structure make it convenient to interact with other parts of a worksheet.

Although Mathcad program is comprised of its program operators, a lot of its built-in functions can be used in the program too. For example, augment, sign, random variable and so on. This makes its program more concise and powerful. The output of Mathcad program maybe is a number, a vector, a matrix and a function, this makes its application wide and flexible.

However, the program function of Mathcad is as limited as other technologies. For example, Not all its built-in functions can be used in Mathcad program and some of them can not be used completely and symbolic operations don't work within a program.

Anyway, Program function of Mathcad is a useful tool in teaching of Math.

3. Application in teaching of Differential Equations

Mathcad's program function can be applied to the teaching of Differential Equations. Although Mathcad has several built-in functions (or routines) for solving of differential equations, such as Odesolve, rkfixed, rkadapt, etc, there're cases we need its program function. For example, when teaching approximate methods solving differential equations and they are different from those which are basis of above built-in functions, Mathcad program function becomes necessary and helpful. Especially, in comparison of different approximate methods. In this part we'll illustrate its application by examples.

3.1 Application to one differential equation

In this section we'll use Mathcad's program function to write programs to perform Euler method and Improved Euler method for a first order differential equation and compare their results with the exact solution and the results of built-in function through an example.

Let's consider following example.

Given: $y' = 1 + \frac{y}{x} + \left(\frac{y}{x}\right)^2$ and $y(1) = 0$, x in $[1, 2]$; Find: its approximate solution over given interval by Euler method and Improved Euler method with step size $h = 0.1$ and errors. To make programs of Euler and Improved Euler method

Program of Euler method

Program of Improved Euler Method

$$\begin{array}{l}
 \text{Eu}(f, a, b, c, h) := \left| \begin{array}{l}
 x_0 \leftarrow a \\
 y_0 \leftarrow b \\
 n \leftarrow \frac{c - a}{h} \\
 \text{for } i \in 0..n - 1 \\
 \left| \begin{array}{l}
 x_{i+1} \leftarrow x_i + h \\
 y_{i+1} \leftarrow y_i + h \cdot f(x_i, y_i)
 \end{array} \right. \\
 P \leftarrow \text{augment}(x, y) \\
 P
 \end{array} \right.
 \end{array}
 \qquad
 \begin{array}{l}
 \text{IEu}(f, a, b, c, h) := \left| \begin{array}{l}
 x_0 \leftarrow a \\
 y_0 \leftarrow b \\
 n \leftarrow \frac{c - a}{h} \\
 \text{for } i \in 0..n - 1 \\
 \left| \begin{array}{l}
 x_{i+1} \leftarrow x_i + h \\
 y_{i+1}^1 \leftarrow y_i + h \cdot f(x_i, y_i) \\
 y_{i+1} \leftarrow y_i + \frac{h}{2} \cdot (f(x_i, y_i) + f(x_{i+1}, y_{i+1}^1))
 \end{array} \right. \\
 P \leftarrow \text{augment}(x, y) \\
 P
 \end{array} \right.
 \end{array}
 \end{array}$$

Note: in above programs Left side is the output of the program which is a matrix of x -value and approximate solution values. Right side is the program which uses the input: 'f' function of ODE, 'a' initial x value; 'b' initial y value; 'c' end point of the interval; 'h' step size and perform approximate method to produce approximate solution.

Now, let's use these programs to get approximate solutions of above example.

In above example, $f(x,y) := 1 + \frac{y}{x} + \left(\frac{y}{x}\right)^2$ $a := 1$ $b := 0$ $c := 2$ $h := 0.1$

Let

$A := \text{Eu}(f, a, b, c, h)$ $B := \text{IEu}(f, a, b, c, h)$

then, Approximate solution produced by Euler method:

$$A^T =$$

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|-----|--------|--------|--------|--------|-------|--------|--------|--------|
| 0 | 1 | 1.1 | 1.2 | 1.3 | 1.4 | 1.5 | 1.6 | 1.7 | 1.8 | 1.9 |
| 1 | 0 | 0.1 | 0.2099 | 0.3305 | 0.4624 | 0.6063 | 0.763 | 0.9335 | 1.1185 | 1.3193 |

Approximate solution produced by Improved Euler method:

$$B^T =$$

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|-------|--------|--------|--------|--------|--------|--------|--------|--------|
| 0 | 1 | 1.1 | 1.2 | 1.3 | 1.4 | 1.5 | 1.6 | 1.7 | 1.8 | 1.9 |
| 1 | 0 | 0.105 | 0.2208 | 0.3483 | 0.4885 | 0.6421 | 0.8104 | 0.9944 | 1.1955 | 1.4152 |

It's not hard to find its exact solution:

$g(x) := x \tan(\ln(x))$ $i := 0..10$ $x_1 := 1 + i \cdot 0.1$

Also, find its solution by built-in function Odesolve'

Given

$y'(x) = 1 + \frac{y(x)}{x} + \left(\frac{y(x)}{x}\right)^2$ $y(1) = 0$

$y := \text{Odesolve}(x, 2)$

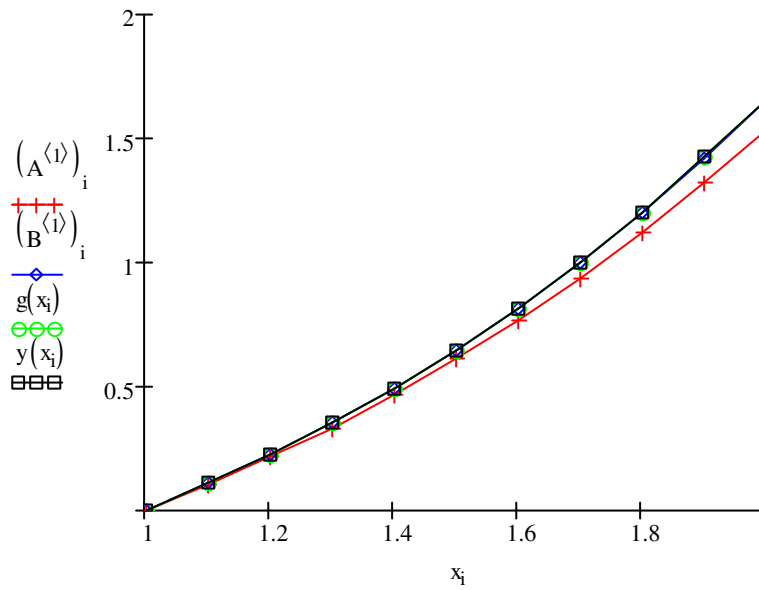
$$y(x_i) =$$

| | 0 |
|----|--------|
| 0 | 0 |
| 1 | 0.1052 |
| 2 | 0.2212 |
| 3 | 0.3491 |
| 4 | 0.4897 |
| 5 | 0.6439 |
| 6 | 0.8128 |
| 7 | 0.9975 |
| 8 | 1.1994 |
| 9 | 1.4201 |
| 10 | 1.6613 |

$$g(x_i) =$$

| | 0 |
|----|--------|
| 0 | 0 |
| 1 | 0.1052 |
| 2 | 0.2212 |
| 3 | 0.3491 |
| 4 | 0.4897 |
| 5 | 0.6439 |
| 6 | 0.8128 |
| 7 | 0.9975 |
| 8 | 1.1994 |
| 9 | 1.4201 |
| 10 | 1.6613 |

Now, we graph these 4 solutions in a picture.



Next, let's analyze their error. we define error of Euler method and Improved Euler Method as ErEu and ErIEu.

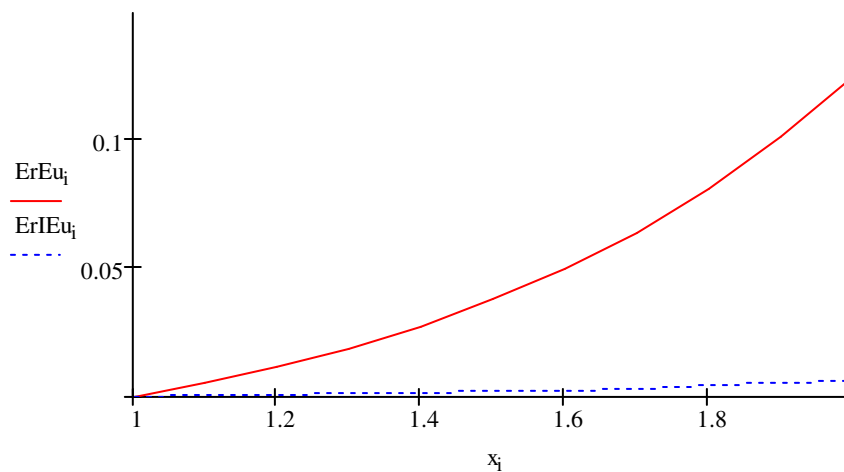
$$\text{ErEu}_i := \left| g(x_i) - (A^{(l)})_i \right| \qquad \text{ErIEu}_i := \left| g(x_i) - (B^{(l)})_i \right|$$

$$\text{ErEu}^T =$$

| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|------------------------|--------|--------|--------|--------|
| 0 | 0 | $5.1598 \cdot 10^{-3}$ | 0.0113 | 0.0187 | 0.0273 | 0.0376 |

$$\text{ErIEu}^T =$$

| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|------------------------|-----------------------|------------------------|------------------------|------------------------|
| 0 | 0 | $2.0114 \cdot 10^{-4}$ | $4.714 \cdot 10^{-4}$ | $8.1176 \cdot 10^{-4}$ | $1.2286 \cdot 10^{-3}$ | $1.7322 \cdot 10^{-3}$ |



3.2 Application to system of differential equations

In this section we'll write a program to implement Improved Euler method to find the approximate solutions of a system of 3 differential equations and analyze the error. First, write the program

```

IEu(f1,f2,f3,a,b,c,d,l,h) :=
  t0 ← a
  u10 ← b
  u20 ← c
  u30 ← d
  n ← (l - a) / h
  for i ∈ 0..n - 1
    ti+1 ← ti + h
    u1i+1 ← u1i + h·f1(ti, u1i, u2i, u3i)
    u2i+1 ← u2i + h·f2(ti, u1i, u2i, u3i)
    u3i+1 ← u3i + h·f3(ti, u1i, u2i, u3i)
    u1i+1 ← u1i + (h/2)·(f1(ti, u1i, u2i, u3i) + f1(ti+1, u1i+1, u2i+1, u3i+1))
    u2i+1 ← u2i + (h/2)·(f2(ti, u1i, u2i, u3i) + f2(ti+1, u1i+1, u2i+1, u3i+1))
    u3i+1 ← u3i + (h/2)·(f3(ti, u1i, u2i, u3i) + f3(ti+1, u1i+1, u2i+1, u3i+1))
  P ← augment(t, u1)
  P ← augment(P, u2)
  P ← augment(P, u3)
  P
  
```

Now, we use this program to find approximate solutions of following example.

$$\begin{aligned}
 \text{Given: } u_1' &= u_2 & u_1(0) &= 1 \\
 u_2' &= -u_1 - 2e^t + 1 & u_2(0) &= 0 & t \text{ in } [0, 1] \\
 u_3' &= -u_1 - e^t + 1 & u_3(0) &= 1
 \end{aligned}$$

Find: its approximate solutions by Improved Euler method with step size $h=0.1$

In this example

$$\begin{aligned}
 f_1(t, u_1, u_2, u_3) &:= u_2 & f_2(t, u_1, u_2, u_3) &:= -u_1 - 2e^t + 1 & f_3(t, u_1, u_2, u_3) &:= -u_1 - e^t + 1 \\
 a &:= 0 & b &:= 1 & c &:= 0 & d &:= 1 & l &:= 1 & h &:= 0.1
 \end{aligned}$$

Let $A := \text{IEu}(f1, f2, f3, a, b, c, d, l, h)$

then its approximate solution is shown in the right matrix.

The 1st column is t-value ;
 The 2nd column is u1-value;
 The 3rd column is u2-value;
 The 4th column is u3-value.

$$A =$$

| | 0 | 1 | 2 | 3 |
|----|-----|---------|---------|---------|
| 0 | 0 | 1 | 0 | 1 |
| 1 | 0.1 | 0.99 | -0.2105 | 0.8947 |
| 2 | 0.2 | 0.9579 | -0.4411 | 0.7805 |
| 3 | 0.3 | 0.9018 | -0.6918 | 0.6583 |
| 4 | 0.4 | 0.8196 | -0.9627 | 0.5295 |
| 5 | 0.5 | 0.7093 | -1.2539 | 0.3953 |
| 6 | 0.6 | 0.5689 | -1.5657 | 0.2571 |
| 7 | 0.7 | 0.3963 | -1.8983 | 0.1163 |
| 8 | 0.8 | 0.1893 | -2.2524 | -0.0258 |
| 9 | 0.9 | -0.0541 | -2.6286 | -0.1678 |
| 10 | 1 | -0.3363 | -3.0278 | -0.3081 |

Its exact solutions are :

$$g1(t) := \cos(t) + \sin(t) - e^t + 1$$

$$g2(t) := -\sin(t) + \cos(t) - e^t$$

$$g3(t) := -\sin(t) + \cos(t)$$

$$i := 0..10$$

$$t_i := 0.1 \cdot i$$

$$g1(t_i) =$$

| | 0 |
|----|---------|
| 0 | 1 |
| 1 | 0.9897 |
| 2 | 0.9573 |
| 3 | 0.901 |
| 4 | 0.8187 |
| 5 | 0.7083 |
| 6 | 0.5679 |
| 7 | 0.3953 |
| 8 | 0.1885 |
| 9 | -0.0547 |
| 10 | -0.3365 |

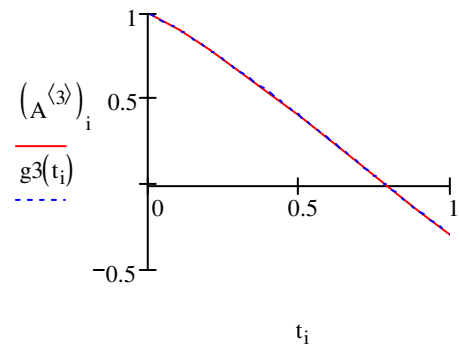
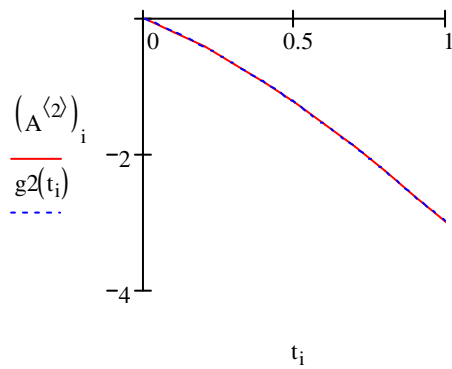
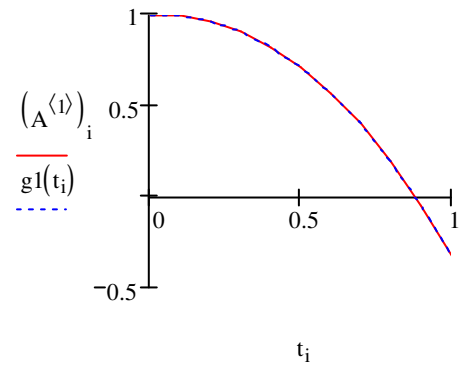
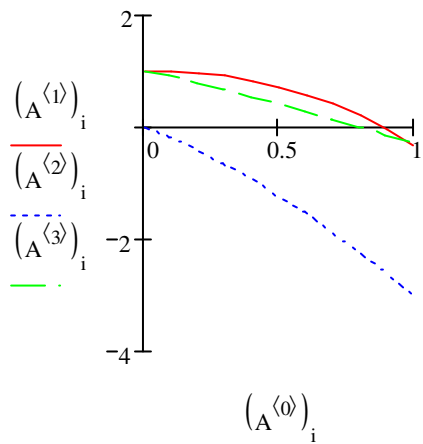
$$g2(t_i) =$$

| | 0 |
|----|---------|
| 0 | 0 |
| 1 | -0.21 |
| 2 | -0.44 |
| 3 | -0.69 |
| 4 | -0.9602 |
| 5 | -1.2506 |
| 6 | -1.5614 |
| 7 | -1.8931 |
| 8 | -2.2462 |
| 9 | -2.6213 |
| 10 | -3.0195 |

$$g3(t_i) =$$

| | 0 |
|----|---------|
| 0 | 1 |
| 1 | 0.8952 |
| 2 | 0.7814 |
| 3 | 0.6598 |
| 4 | 0.5316 |
| 5 | 0.3982 |
| 6 | 0.2607 |
| 7 | 0.1206 |
| 8 | -0.0206 |
| 9 | -0.1617 |
| 10 | -0.3012 |

Next, we make 4 graphes. 1st one is graph of 3 approximate solutions; 2nd one is the graph of u1 and g1; 3rd one is the graph of u2 and g2; 4th is graph of u3 and g3.



Now, let's analyze the error. We define error in u_1 , u_2 , u_3 as Eru_1 , Eru_2 , Eru_3 .

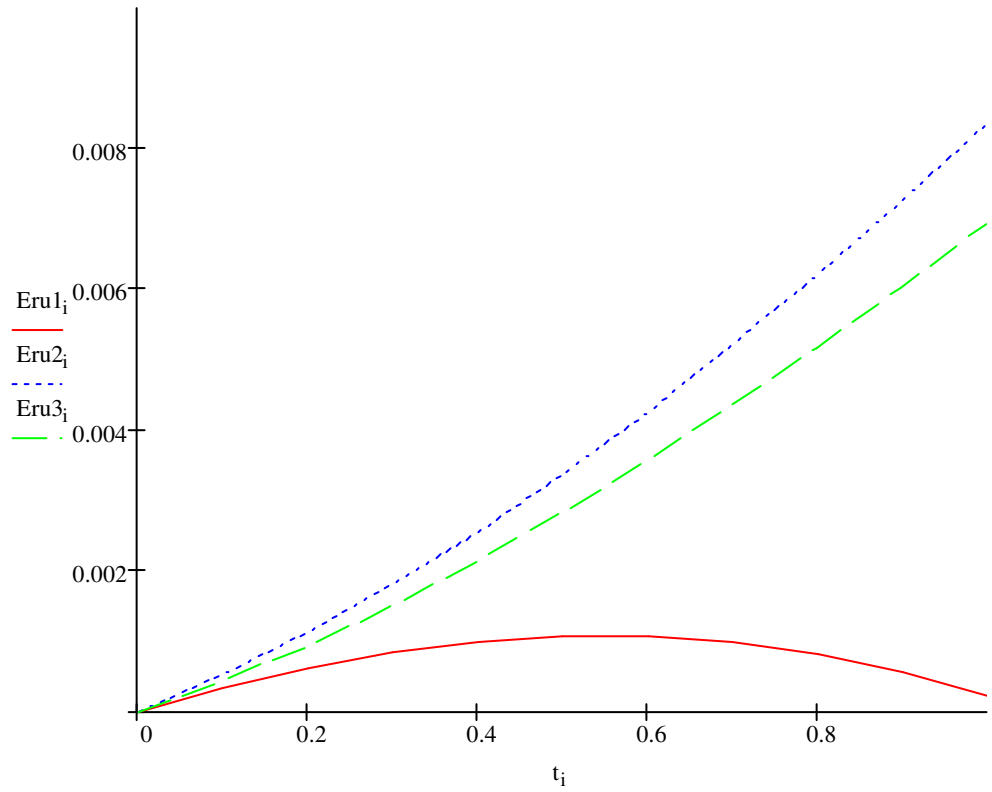
$$Eru_1 := \left| (A^{(1)})_i - g1(t_i) \right| \quad Eru_2 := \left| (A^{(2)})_i - g2(t_i) \right| \quad Eru_3 := \left| (A^{(3)})_i - g3(t_i) \right|$$

$$Eru_1^T = \begin{array}{c|cccccc} & 0 & 1 & 2 & 3 & 4 & 5 \\ \hline 0 & 0 & 3.3334 \cdot 10^{-4} & 6.1343 \cdot 10^{-4} & 8.3275 \cdot 10^{-4} & 9.8455 \cdot 10^{-4} & 1.063 \cdot 10^{-3} \end{array}$$

$$Eru_2^T = \begin{array}{c|cccccc} & 0 & 1 & 2 & 3 & 4 & 5 \\ \hline 0 & 0 & 5.1692 \cdot 10^{-4} & 1.1164 \cdot 10^{-3} & 1.7946 \cdot 10^{-3} & 2.5473 \cdot 10^{-3} & 3.3699 \cdot 10^{-3} \end{array}$$

$$Eru_3^T = \begin{array}{c|cccccc} & 0 & 1 & 2 & 3 & 4 & 5 \\ \hline 0 & 0 & 4.2929 \cdot 10^{-4} & 9.3189 \cdot 10^{-4} & 1.5031 \cdot 10^{-3} & 2.1375 \cdot 10^{-3} & 2.8294 \cdot 10^{-3} \end{array}$$

The graph of these errors as follows:



4. Application in teaching of Numerical Analysis

Numerical Analysis is another area in which Mathcad's program function can display its full potential. Since Numerical Analysis often needs massive, repeated and tedious computation the program function can best meet this requirement. In this part we just illustrate its application in finding approximate solution of non-linear equations and Newton's interpolation polynomial.

4.1 Application of finding approximate solution of non-linear equations

In this section we'll use program function of Mathcad to write a program to perform Secant method to find approximate solution of non-linear equations. And, we'll compare the result with built-in function's result.

Program for Secant method.

```

ApS(f, a, b, r) :=
  p0 ← a
  p1 ← b
  while |p1 - p0| ≥ r
    p ← (p0·f(p1) - p1·f(p0)) / (f(p1) - f(p0))
    p0 ← p1
    p1 ← p
  p
  
```

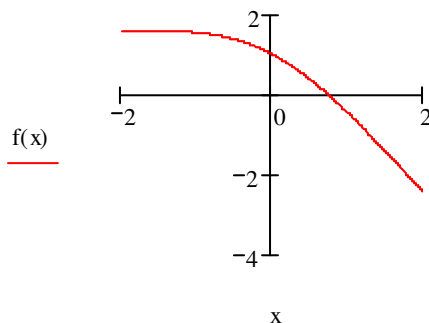
Note:

Input: $f(x)=0$ - given equation
 a, b -endpoints of $[a, b]$
 that bracket the solution
 r -desired accuracy
 Output: Approximate solution
 satisfying accuracy

Now, let's see an example: Find approximate solution of $\cos(x)-x=0$ with accuracy 10^{-5} .

$$f(x) := \cos(x) - x \quad f(0) = 1 \quad f(1) = -0.45969769 \quad a := 0 \quad b := 1 \quad r := 10^{-5}$$

Use graph to locate solution in $[0, 1]$



The desired approximate solution:

$$x := \text{ApS}(f, a, b, r) \quad x = 0.73908513$$

Let's compare with result from built-in function.

$$r := \text{root}(f(x), x, 0, 1) \quad r = 0.73908513$$

Let's check the error

$$\text{er} := |x - r| \quad \text{er} = 1.5931700 \times 10^{-13}$$

4.2 Application in finding Newton interpolation polynomial

In this section we'll use the program function to make a program to produce the coefficients of Newton interpolation polynomial then use symbolic operation to form Newton interpolation polynomial. Also, the graph of Newton polynomial is provided. We illustrate by below example.

Example. Given: $f(1)=0.7651977$, $f(1.3)=0.620086$, $f(1.6)=0.4554022$, $f(1.9)=0.2818186$, $f(2.2)=0.1103623$. Construct interpolating polynomial of degree 4 by Newton's interpolatory divided difference formula and use it to approximate $f(2)$.

Sol'n: First, enter the data:

$$a_0 := 1 \quad a_1 := 1.3 \quad a_2 := 1.6 \quad a_3 := 1.9 \quad a_4 := 2.2$$

$$b_0 := 0.7651977 \quad b_1 := 0.620086 \quad b_2 := 0.4554022 \quad b_3 := 0.2818186 \quad b_4 := 0.1103623$$

Second, use following program to figure out divided-differences. In following program, use a for-loop to enter above set of data, and then use double for-loop to figure out the divided-differences. The output of the program is a matrix and the elements on the diagonal are the divided-differences needed in Newton's formula.

$$f := \left(\begin{array}{l} \text{for } i \in 0..4 \\ \quad \left| \begin{array}{l} x_i \leftarrow a_i \\ f_{i,0} \leftarrow b_i \end{array} \right. \\ \text{for } i \in 1..4 \\ \quad \text{for } j \in 1..i \\ \quad \quad f_{i,j} \leftarrow \frac{f_{i,j-1} - f_{i-1,j-1}}{x_i - x_{i-j}} \\ f \end{array} \right. \quad f = \begin{pmatrix} 0.7651977 & 0 & 0 & 0 & 0 \\ 0.620086 & -0.48370567 & 0 & 0 & 0 \\ 0.4554022 & -0.548946 & -0.10873389 & 0 & 0 \\ 0.2818186 & -0.578612 & -0.04944333 & 0.0658784 & 0 \\ 0.1103623 & -0.571521 & 0.01181833 & 0.06806852 & 1.82510288 \times 10^{-3} \end{pmatrix}$$

Then, use Newton's interpolatory divided-difference formula to construct polynomial.

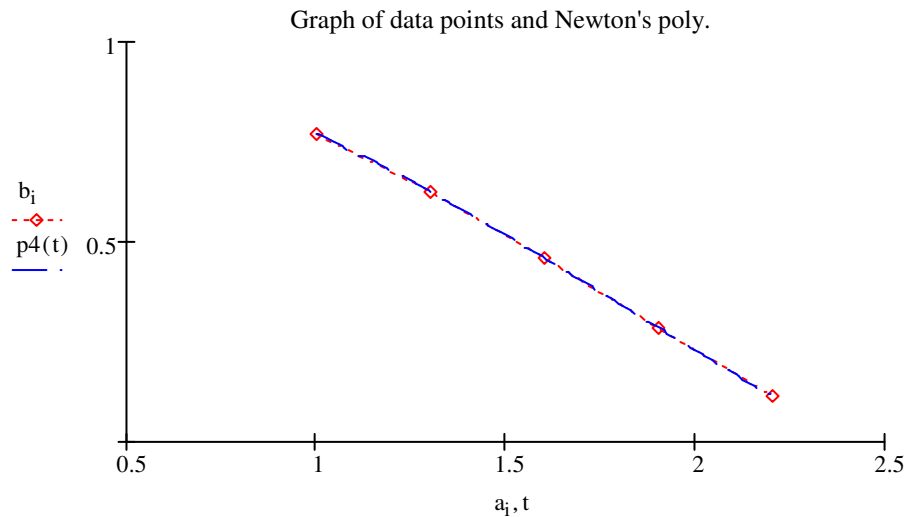
$$p4(t) := f_{0,0} + \sum_{i=1}^4 \left[f_{i,i} \prod_{k=0}^{i-1} (t - a_k) \right]$$

$$p4(t) \begin{array}{l} \text{simplify} \\ \text{float,6} \end{array} \rightarrow .977735 + 7.3391310^{-2} \cdot t - .343047t^2 + 5.5292810^{-2} \cdot t^3 + 1.8251010^{-3} \cdot t^4$$

Then, find the approximate value of $f(2)$ is $p_4(2) = 0.22387536$

Finally, Plot the data points and Newton's interpolation polynomial in a graph below:

$i := 0..4$ $t := 1, 1.01.. 2.2$



5. Reference

- [1] R.L.Burden & J.D.Faires:"Numerical Analysis" 7th edition, Brooks/Cole ,2001.
- [2] L.Fausett: "Numerical Methods Using Mathcad", Prentice Hall, 2002.
- [3] R.K.Nagle, E.B.Staff & A.D.Snide: "Fundamentals of Differential Equations" 6th edition , Addition Wesley, 2004
- [4] D.T.Wu: "Using Mathcad in Teaching Differential Equations" , Electronic Proceedings of ICTMT4, 1999.
Web site: www.tech.plym.ac.uk/math/CTMHOME/ictmt4.html
- [5] D.T.Wu: "Teaching Numerical Analysis with Mathcad", "International Journal of Computer Algebra in Mathematics Education" Vol 10 No.1, 2003.