Addis Ababa University

Computational Data Science Program

Course Title: Numerical Solution of DEs (CDSC 606)

Course Instructor : Hailemichael Kebede (PhD)
hailemichael.kebede@aau.edu.et
0911901997

2024

Contents

1.1 Introduction to IVP and BVP

Initial-Value Problems

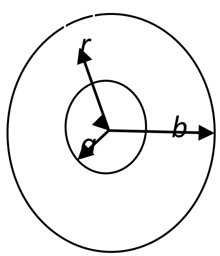
- The auxiliary conditions are at one point of the independent variable

$$\frac{dy}{dx} = x - y^2$$

$$y(0) = 1$$

Boundary-Value Problems

- The auxiliary conditions are not at one point of the independent variable
- More difficult to solve than the initial value problem



Where a = 5
and b = 8

$$\frac{d^2u}{dr^2} + \frac{1}{r}\frac{du}{dr} - \frac{u}{r^2} = 0,$$

$$u(5) = 0.0038731,$$

$$u(8) = 0.0030770$$

Initial value & BVP . . .

- Side conditions prescribing solution or derivative values at specified points are required to make solution of ODE unique
- For initial value problem, all side conditions are specified at single point, say t_0
- For *boundary value problem* (BVP), side conditions are specified at more than one point
- kth order ODE, or equivalent first-order system, requires k side conditions
- For ODEs, side conditions are typically specified at endpoints of interval $[a, b]$, so we have *two-point boundary value problem* with boundary conditions (BC) at a and b.

1.2 Numerical treatment (solution) of initial value problems

- ✓ ***Solution by Taylor series***
- ✓ ***Picard’s Method of successive approximation***
- ✓ ***Euler and Modified Euler Method***
- ✓ ***Runge Kutta Methods***

Solution by Taylor series

Consider 1st ODE : $y' = f(x,y)$ ***............ ....(2.1)***

$y(x_0) = y_0$ ***with the initial condition***

If $y(x)$ is the exact solution of (2.1), then the Taylor's series for $y(x)$ around $x = x_0$ is given by

$$y(x) = y_0 + (x - x_0)y_0' + \frac{(x - x_0)^2}{2!} y_0'' + \quad \text{............ (2.2)}$$

If the values of y_0', y_0'' ... are known, then 2.2 gives the power series for y.

Using the formula for total derivatives, we can write

$$y'' = f' = f_x + y' f_y = f_x + f f_y$$

Solution by Taylor series …

Solution in the form of power series

Where the suffixes denote partial derivatives with respect to the variable concerned. Similarly, we obtain

$$y''' = f'' = f_{xx} + f_{xy}f + f[f_{yx} + f_{yy}f] + f_y[\, f_x + f_y f]$$

$$= f_{xx} + 2\, f f_{xy} + f^2 f_{yy} + f_x\, f_y + f f_y^{\,2}$$

and other higher derivative of y.

Example: Find y(0.1) correct to four decimal places if y(x) satisfies $y' = x - y^2$ ***and y(0)=1***

Answer: $y(x) = 1 - x + \frac{3}{2}x^2 - \frac{4}{3}x^3 - \frac{17}{12}x^4 - \frac{31}{20}x^5 + \cdots$

Note: This approach requires computation of higher derivatives of Y, which can be obtained by differentiating $y' = f(x, y)$

Picard's Method of successive approximation

Integrating the differential equation in 2.1, we obtain

$$y = y_0 + \int_{x_0}^{x} f(x, y)dx \text{ ....2.3}$$

Equation 2.3 can be solved by the method of successive approximations in which the first approximation to y is obtained by putting y_0 for y on the right on the right side of 2.3, and we write

$$y^{(1)} = y_0 + \int_{x_0}^{x} f(x, y)dx$$

The integral on the right can now be solved and the resulting $y^{(1)}$ is substituted for y in the integrand of 2.3 to obtain the second approximation. $y^{(2)}$.

$$y^{(2)} = y_0 + \int_{x_0}^{x} f\left(x, y^{(1)}\right)dx$$

Picard's Method . . .

Solution in the form of power series

Proceeding in this way, we obtain $y^{(3)}, y^{(4)}, \dots y^{(n-1)}$ *and* $y^{(n)}$ *where*

$$\left.\begin{aligned} y^{(n)} &= y_0 + \int_{x_0}^{x} f(x, y^{(n-1)})\,dx \\ \text{and} \quad y^{(0)} &= y_0 \end{aligned}\right\}$$

Hence this method yields a sequence of approximations $y^{(1)}, y^{(2)}, \dots y^{(n)}$.

and this integration become more and more difficult as we proceed to higher approximations.

Example: Solve the equation $y' = x + y^2$ ***subject to the condition y=1 when x=0***

Answer : $y^{(0)}(x) = 1 \quad y^{(1)}(x) = 1 + x + \frac{1}{2}x^2 \quad y^{(2)}(x) = 1 + x + \frac{3}{2}x^2 + \frac{2}{3}x^3 + \frac{1}{4}x^4 + \frac{1}{20}x^5$

Euler’s Method

Solution in the form of tabulated values

Suppose that we wish to solve the equations 2.1 for values of y at $x = x_r = x_0 + rh$ $(r = 1, 2, \ldots)$. *Integrating 2.1 we obtain*

$$y_1 = y_0 + \int_{x_0}^{x_1} f(x, y)dx$$

Assuming that $f(x, y) = f(x_0, y_0)$ *in* $x_0 \le x \le x_1$, *this gives Euler's formula*

$$y_1 = y_0 + hf(x_0, y_0)$$

Similarly for the range $x_1 \le x \le x_2$ *we have*

$$y_2 = y_1 + \int_{x_1}^{x_2} f(x, y)dx$$

Euler's Method. . .

Substituting $f(x_1, y_1)$ *for* $f(x, y)$ *in* $x_1 \le x \le x_2$, *we obtain*

$$y_2 = y_1 + hf(x_1, y_1)$$

Proceeding in this way, we obtain the general formula

$$y_{n+1} = y_n + hf(x_n, y_n), \quad n = 0,1,2 \ldots$$

Euler’s Method . . .

Use Euler method to solve $y' = -10y \ \ for\ y(0) = 2, and\ plot\ the\ y\ trajectory\ over\ the\ range\ 0 \le t \le 0.5$ $step\ size\ \Delta t = 0.02.$

Exactly solution:

$$\frac{dy}{dt} = -10\ y$$

$$\frac{dy}{y} = -10\ dt \Rightarrow \ln\ y = -10\,t + c$$

$$\Rightarrow\ y = e^{-10\,t}C$$

$\because y(0) = 2 \Rightarrow C = 2$

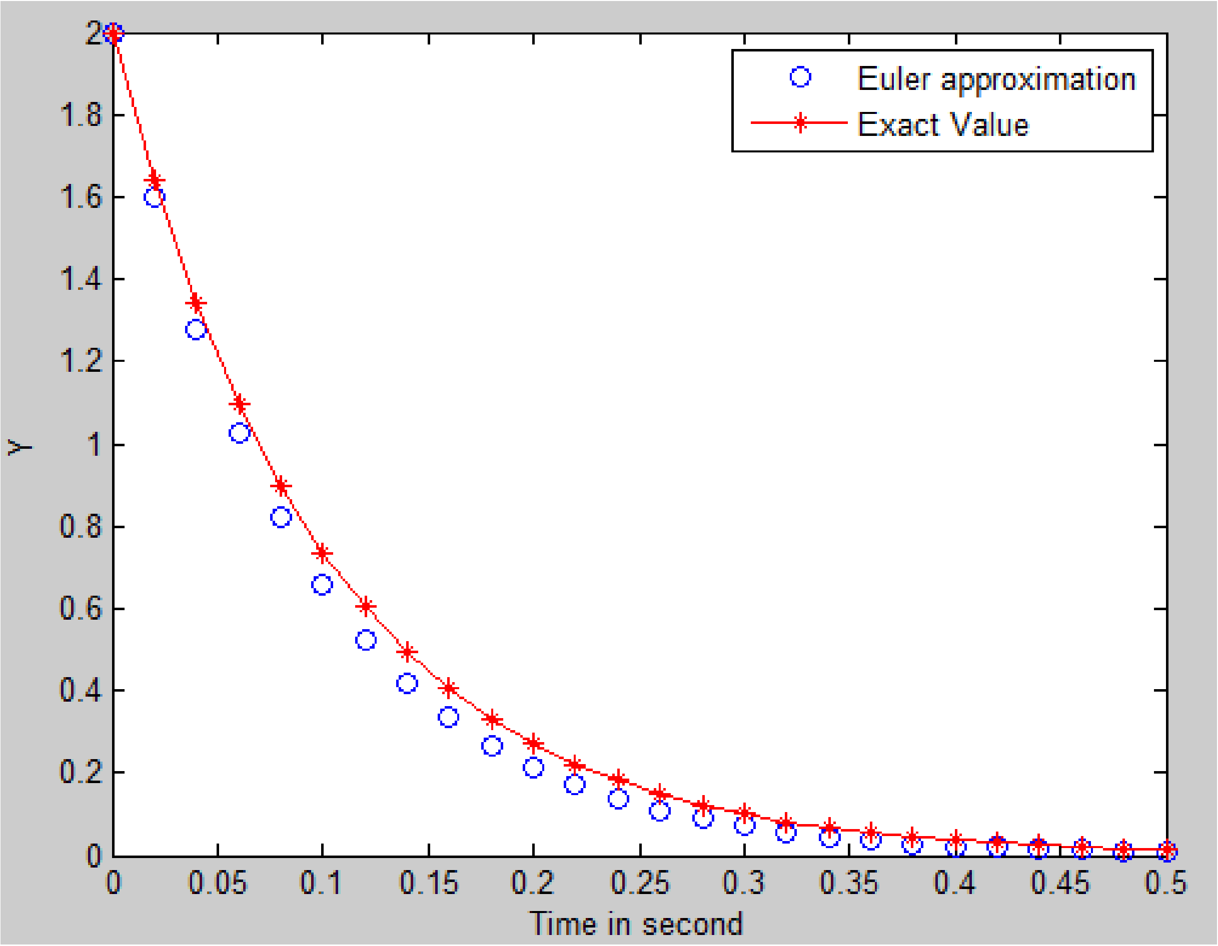
true solution is $y(t) = 2e^{-10t}$.

Numerical solution

It is obtained by coding the following

$$y_{n+1} = y_n + hf(x_n, y_n), \quad n = 0,1,2\ \ldots$$

Euler's Method . . .



MatLab code results

Time	*Euler-app.*	*True value*
0	2.0000	2.0000
0.0200	1.6000	1.6375
0.0400	1.2800	1.3406
0.0600	1.0240	1.0976
	.	
	.	
	.	
0.4400	0.0148	0.0246
0.4600	0.0118	0.0201
0.4800	0.0094	0.0165
0.5000	0.0076	0.0135

Modified Euler method (Predictor-error method)

- ***It gives more accurate results than Euler Method***

- ***One way of improving the method is to use a better approximation to the right side of the equation***

$$\frac{dy}{dt} = f(t, y)$$

The Euler approximation is

$$y(t_{k+1}) = y(t_k) + \Delta t \cdot f[t_k, y(t_k)]$$

Suppose instead we use the average of the right side of equation on the interval (t_k, t_{k+1}). This gives

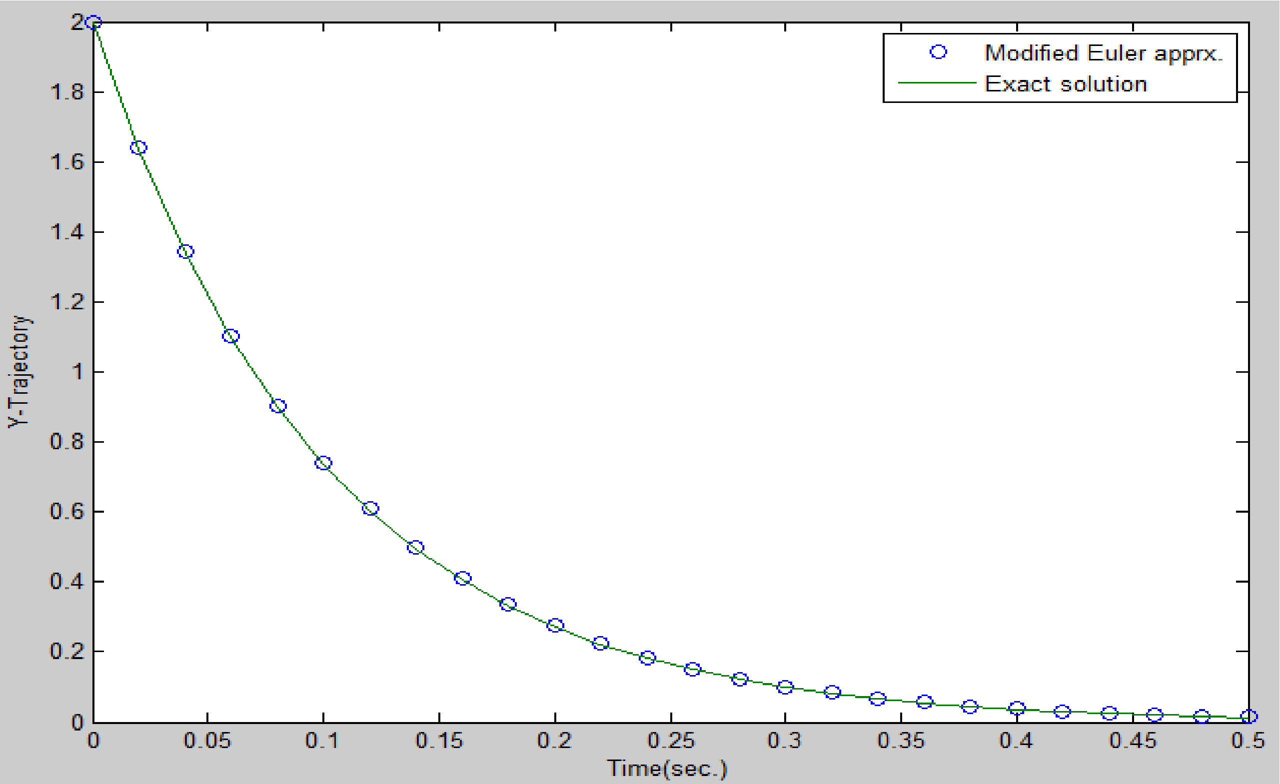
$$y(t_{k+1}) = y(t_k) + \frac{\Delta t}{2} \cdot (f_k + f_{k+1})$$

where
$$f_k = f(t_k, y(t_k))$$
$$f_{k+1} = f(t_{k+1}, y(t_{k+1}))$$

Let $x_{k+1} = y_k + h \cdot f(t_k, y_k)$

Thus, we have $$y_{k+1} = y_k + \frac{h}{2} \cdot \left[f(t_k, y_k) + f(t_{k+1}, x_{k+1}) \right]$$

Modified Euler method . . .



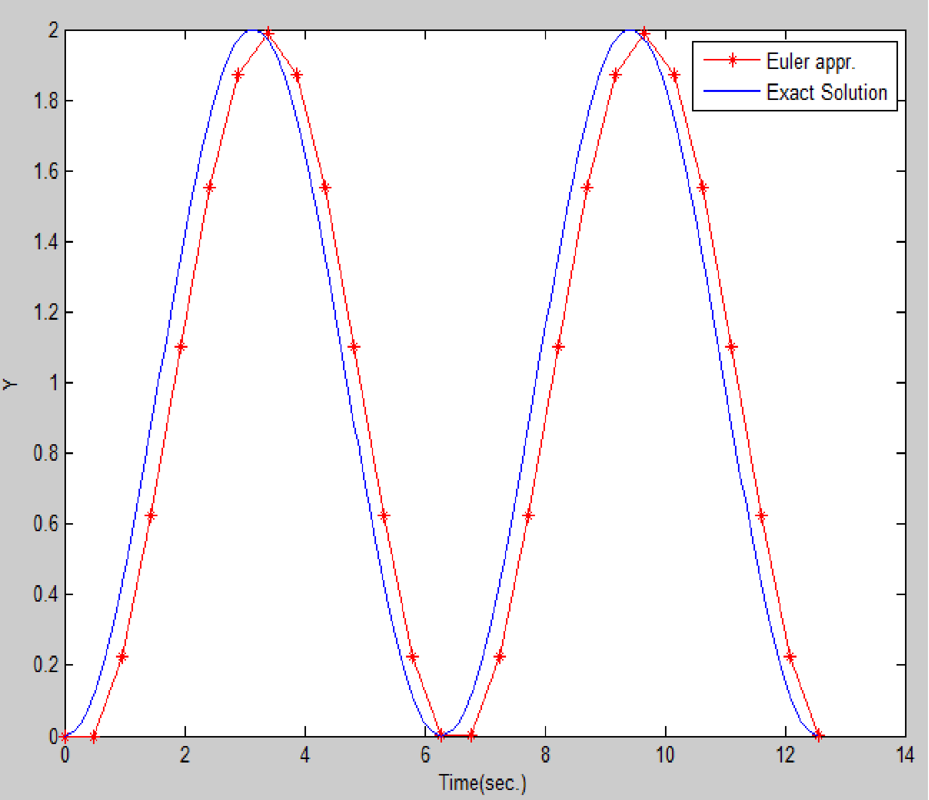
➢***Observe that there is less error than with the Euler method using the same step size.***

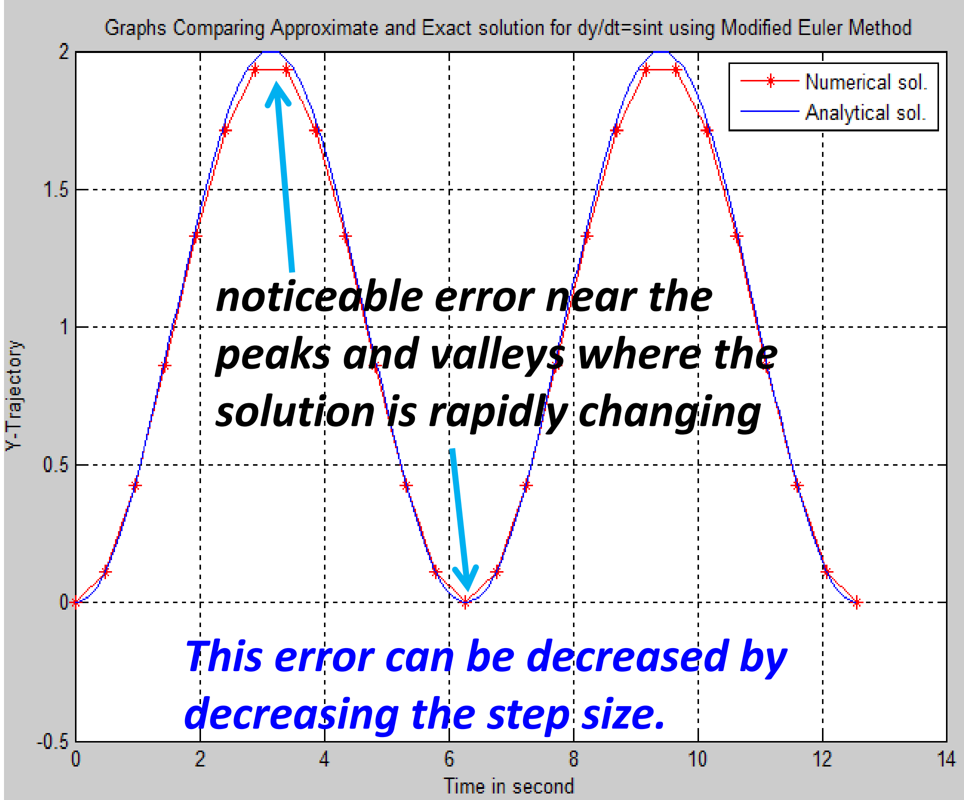
➢ Conclusion: Comparison between Euler and modified Euler

$$\dot{y} = \sin t \quad \text{for } y(0) = 0 \text{ and } 0 \le t \le 4\pi$$

Euler method

modified Euler method (less error)





Runge-Kutta methods . . .

➢ *There are many variants of the Runge-Kutta method, but the most widely used one is the following.*

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

where $k_1 = f(t_n, y_n)$

$$k_2 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right)$$

$$k_3 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right)$$

$$k_4 = f(t_n + h, y_n + hk_3)$$

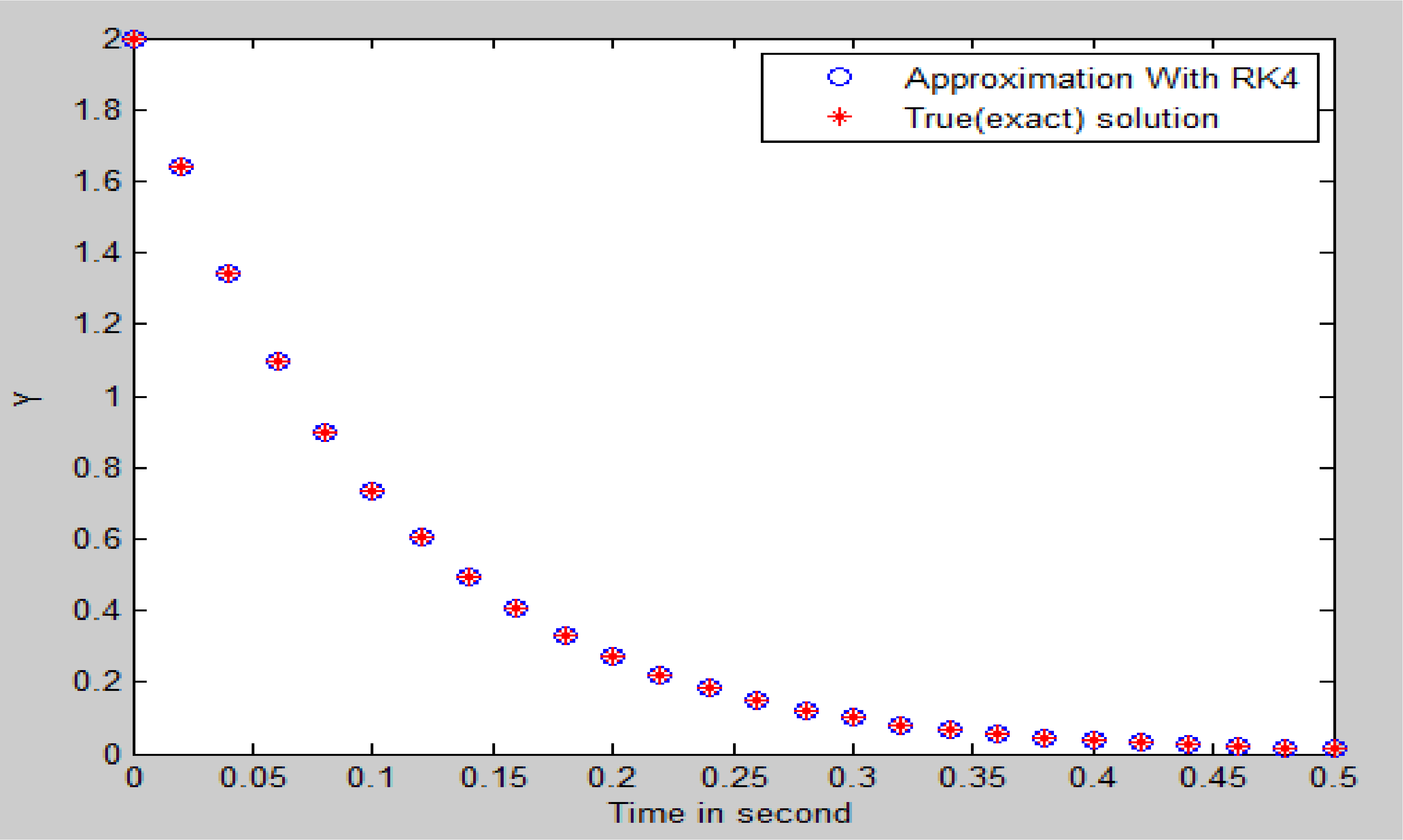
➢ *To run the simulation, we simply start with y_0 and find y_1 using the formula above. Then we plug in y_1 to find y_2 and so on.*

Runge-Kutta methods . . .

➢ ***Some of the results of the Matlab Code is as follows***

Time	*Runge-Kutta appr.*	*Analytical(exact solution)*
0	*2.000000000000000*	*2.000000000000000*
0.020000000000000	*1.637466666666667*	*1.637461506155964*
0.040000000000000	*1.340648542222222*	*1.340640092071279*
0.060000000000000	*1.097633649802074*	*1.097623272188053*
0.080000000000000	*0.898669256881285*	*0.898657928234443*
0.100000000000000	*0.735770476250604*	*0.735758882342885*
0.120000000000000	*0.602399814588911*	*0.602388423824404*
	.	
	.	
	.	
0.400000000000000	0.036633586738749	0.036631277777468
0.420000000000000	0.029993138582572	0.029991153640955
0.440000000000000	0.024556382328837	0.024554679806137
0.460000000000000	0.020105128758697	0.020103671489267
0.480000000000000	0.016460739085704	0.016459494098040
0.500000000000000	0.013476955780768	0.013475893998171

Runge-Kutta methods . . .

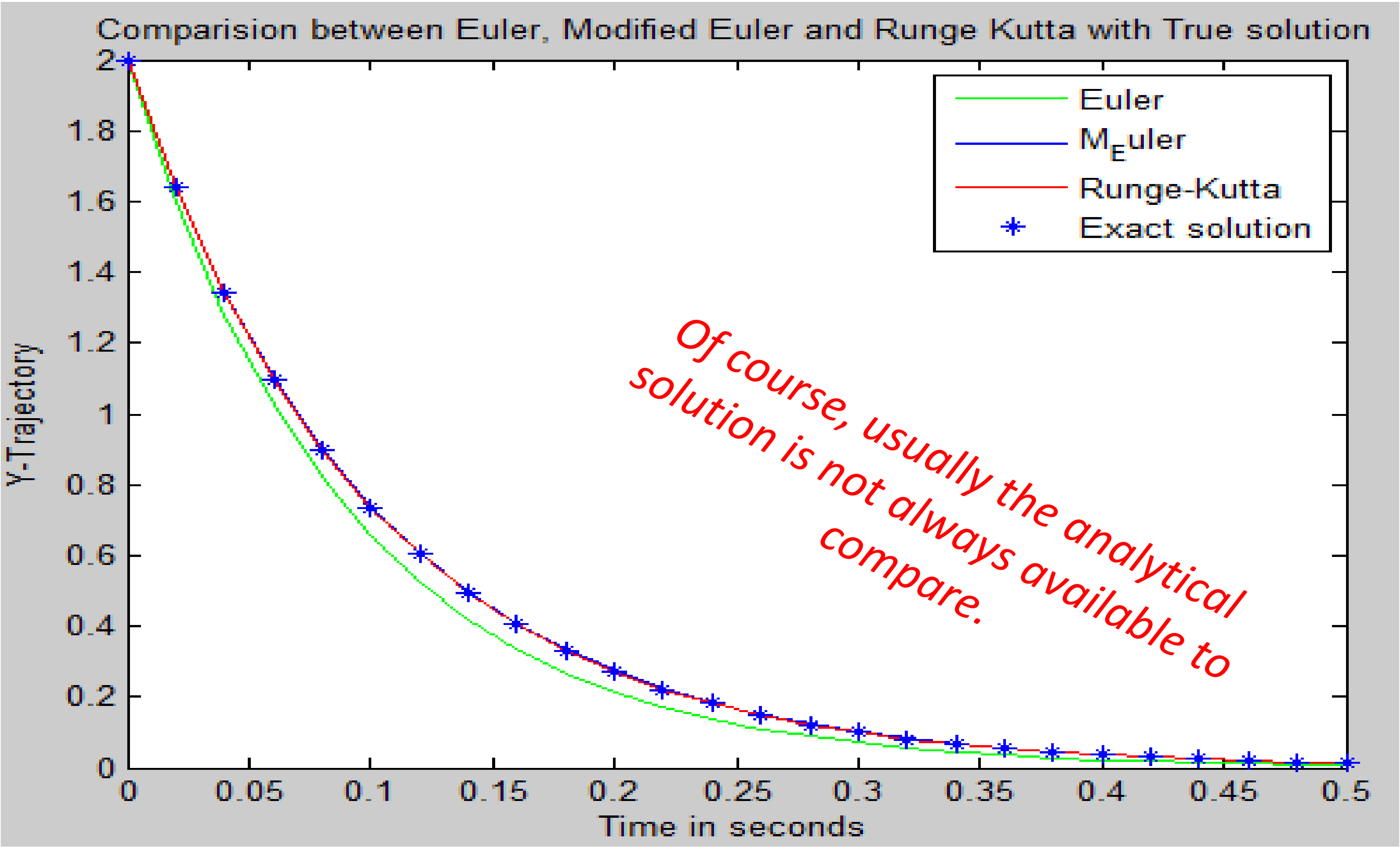


Comparison between Euler, Modified Euler, Runge Kutta and Exact solution

time	Euler	Modified Euler	Runge-Kutta	Exact
0.00	2.000000000000000	2.000000000000000	2.000000000000000	2.000000000000000
0.02	1.600000000000000	1.640000000000000	1.637466666666660	1.637461506155960
0.04	1.280000000000000	1.344800000000000	1.340648542222220	1.340640092071270
0.06	1.024000000000000	1.102736000000000	1.097633649802070	1.097623272188050
0.08	0.819200000000000	0.904243520000000	0.898669256881285	0.898657928234443
0.10	0.655360000000000	0.741479686400000	0.735770476250604	0.735758882342885
0.12	0.524288000000000	0.608013342848000	0.602399814588911	0.602388423824404
0.14	0.419430400000000	0.498570941135360	0.493204808197761	0.493193927883213
0.16	0.335544320000000	0.408828171730995	0.403803216631780	0.403793035989311
		.		
		.		
		.		
0.42	0.018446744073710	0.030982816561735	0.029993138582572	0.029991153640955
0.44	0.014757395258968	0.025405909580623	0.024556382328837	0.024554679806137
0.46	0.011805916207174	0.020832845856111	0.020105128758697	0.020103671489267
0.48	0.009444732965739	0.017082933602011	0.016460739085704	0.016459494098040
0.50	0.007555786372591	0.014008005553649	0.*01347*6955780768	0.*01347*5893998171

Graphical Comparison

The plots show the results obtained from deferent algorithms.



1.3 Boundary value problem for ODE

- For IVP, initial data supply all information necessary to begin numerical solution method at initial point and step forward from there
- For BVP, we have insufficient information to begin step-by-step numerical method, so numerical methods for solving BVPs are more complicated than those for solving IVPs
- We will consider four types of numerical methods for two-point BVPs
  - Shooting
  - Finite difference

Shooting Methods

1. *Guess a value for the auxiliary conditions at one point of time.*
2. *Solve the initial value problem using Euler, Runge-Kutta, …*
3. *Check if the boundary conditions are satisfied, otherwise modify the guess and resolve the problem*

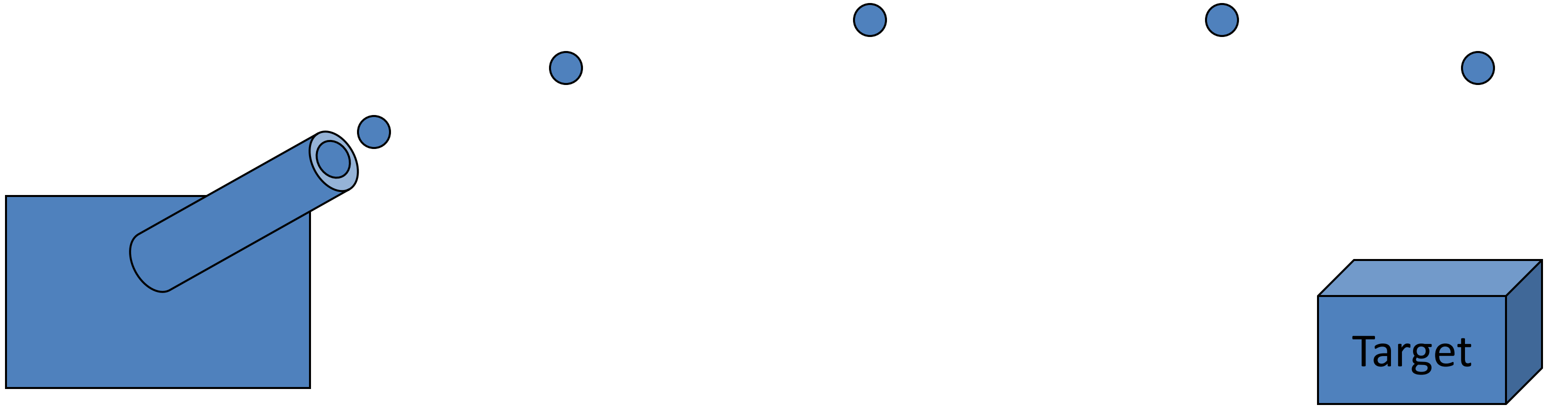
- *Use interpolation in updating the guess.*
- *It is an iterative procedure and can be efficient in solving the BVP.*

BVP → convert → IVP

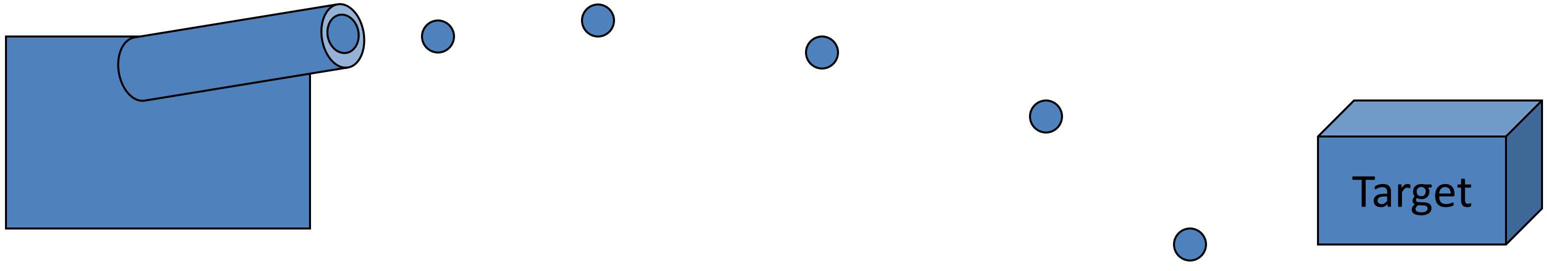
In detail, the steps on Shooting Method

1. ***Convert the ODE to a system of first order ODEs.***
2. ***Guess the initial conditions that are not available.***
3. ***Solve the Initial-value problem.***
4. ***Check if the known boundary conditions are satisfied.***
5. ***If needed modify the guess and resolve the problem again.***

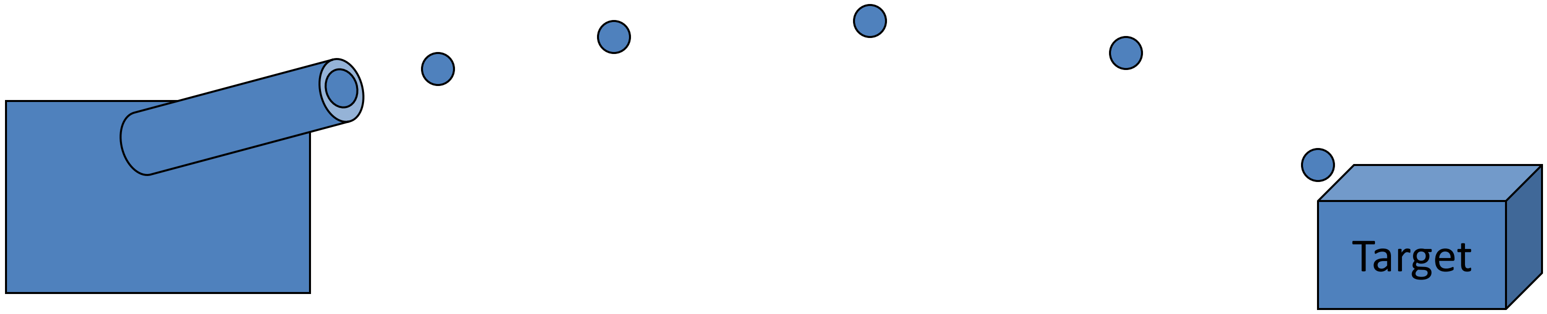
The Shooting Method



The Shooting Method

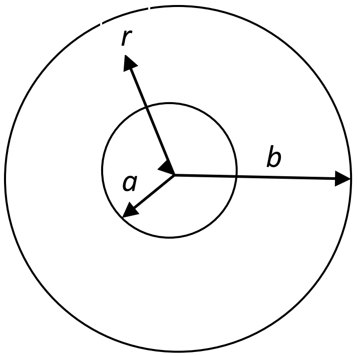


The Shooting Method



Example

$$\frac{d^2u}{dr^2} + \frac{1}{r}\frac{du}{dr} - \frac{u}{r^2} = 0,$$
$$u(5) = 0.0038731,$$
$$u(8) = 0.0030770$$



Where a = 5
and b = 8

Let

$$\frac{du}{dr} = w$$

Then

$$\frac{dw}{dr} + \frac{1}{r}w - \frac{u}{r^2} = 0$$

Solution

- *Two first order differential equations are given as*

$$\frac{du}{dr} = w, \quad u(5) = 0.0038371$$

$$\frac{dw}{dr} = -\frac{w}{r} + \frac{u}{r^2}, \quad w(5) = not\ known$$

Let us assume

$$w(5) = \frac{du}{dr}(5) \approx \frac{u(8) - u(5)}{8 - 5} = -0.00026538$$

To set up initial value problem

$$\frac{du}{dr} = w = f_1(r, u, w), u(5) = 0.0038371$$

$$\frac{dw}{dr} = -\frac{w}{r} + \frac{u}{r^2} = f_2(r, u, w), w(5) = -0.00026538$$

Solution Cont

Using Euler's method,

$$u_{i+1} = u_i + f_1(r_i, u_i, w_i)h$$

$$w_{i+1} = w_i + f_2(r_i, u_i, w_i)h$$

Let us consider 4 segments between the two boundaries, $r = 5$ and then, $r = 8$

$$h = \frac{8-5}{4} = 0.75$$

Solution Continue

For $i=0, r_0=5, u_0=0.0038371, w_0=-0.00026538$

$$
\begin{aligned}
u_1 &= u_0 + f_1(r_0, u_0, w_0)h \\
&= 0.0038371 + f_1(5, 0.0038371, -0.00026538)(0.75) \\
&= 0.0038371 + (-0.00026538)(0.75) \\
&= 0.0036741
\end{aligned}
$$

$$
\begin{aligned}
w_1 &= w_0 + f_2(r_0, u_0, w_0)h \\
&= -0.00026538 + f_2(5, 0.0038371, -0.00026538)(0.75) \\
&= -0.00026538 + \left(-\frac{-0.00026538}{5} + \frac{0.0038371}{5^2}\right)(0.75) \\
&= -0.00010938
\end{aligned}
$$

Solution Cont

For $i=1, r_1 = r_0 + h = 5 + 0.75 = 5.75,\ u_1 = 0.0036741, w_1 = -0.00010940$

$$
\begin{aligned}
u_2 &= u_1 + f_1(r_1, u_1, w_1)h \\
&= 0.0036741 + f_1(5.75, 0.0036741, -0.00010938)(0.75) \\
&= 0.0036741 + (-0.00010938)(0.75) \\
&= 0.0035920
\end{aligned}
$$

$$
\begin{aligned}
w_2 &= w_1 + f_2(r_1, u_1, w_1)h \\
&= -0.00010938 + f_2(5.75, 0.0036741, -0.00010938)(0.75) \\
&= -0.00010938 + (0.00013015)(0.75) \\
&= -0.000011769
\end{aligned}
$$

Solution Cont

For $i = 2, r_2 = r_1 + h = 5.75 + 0.75 = 6.5$ $\quad u_2 = 0.0035920, w_2 = -0.000011785$

$$
\begin{aligned}
u_3 &= u_2 + f_1(r_2, u_2, w_2)h \\
&= 0.0035920 + f_1(6.5, 0.0035920, -0.000011769)(0.75) \\
&= 0.0035920 + (-0.000011769)(0.75) \\
&= 0.0035832
\end{aligned}
$$

$$
\begin{aligned}
w_3 &= w_2 + f_2(r_2, u_2, w_2)h \\
&= -0.000011769 + f_2(6.5, 0.0035920, -0.000011769)(0.75) \\
&= -0.000011769 + (0.000086829)(0.75) \\
&= 0.000053352
\end{aligned}
$$

Solution Cont

For $i = 3, r_3 = r_2 + h = 6.50 + 0.75 = 7.25 \quad u_3 = 0.0035832, w_3 = 0.000053332$

$$\begin{aligned}u_4 &= u_3 + f_1(r_3, u_3, w_3)h \\ &= 0.0035832 + f_1(7.25, 0.0035832, 0.000053352)(0.75) \\ &= 0.0035832 + (0.000053352)(0.75) \\ &= 0.0036232\end{aligned}$$

$$\begin{aligned}w_4 &= w_3 + f_2(r_3, u_3, w_3)h \\ &= -0.000011785 + f_2(5.75, 0.0035832, -0.000053352)(0.75) \\ &= 0.000053352 + (0.000060811)(0.75) \\ &= 0.000098961\end{aligned}$$

So at $r = r_4 = r_3 + h = 7.25 + 0.75 = 8$

$$u(8) \approx u_4 = 0.0036232$$

Solution Cont

Let us assume a new value for $\frac{du}{dr}(5)$

$$w(5) = 2\frac{du}{dr}(5) \approx 2\frac{u(8) - u(5)}{8-5} = 2(-0.00026538) = -0.00053076$$

Using $h = 0.75$ and Euler's method, we get

$$u(8) \approx u_4 = 0.0029665''$$

While the given value of this boundary condition is

$$u(8) \approx u_4 = 0.0030770$$

Solution Cont

Using linear interpolation on the obtained data for the two assumed values of

$\frac{du}{dr}(5)$ we get

$$u(8) = 0.00030770$$

$$\frac{du}{dr}(5) \approx \frac{-0.00053076 - (-0.00026538)}{0.0029645 - 0.0036232}(0.0030770 - 0.0036232) + (-0.00026538)$$

$$= -0.00048611$$

Using $h = 0.75$ and repeating the Euler's method with $w(5) = -0.00048611$

$$u(8) \approx u_4 = 0.0030769$$

Solution Cont.

Using linear interpolation to refine the value of u_4

till one gets close to the actual value of $u(8)$ which gives you,

$$u_1 = u(5) = 0.0038731$$

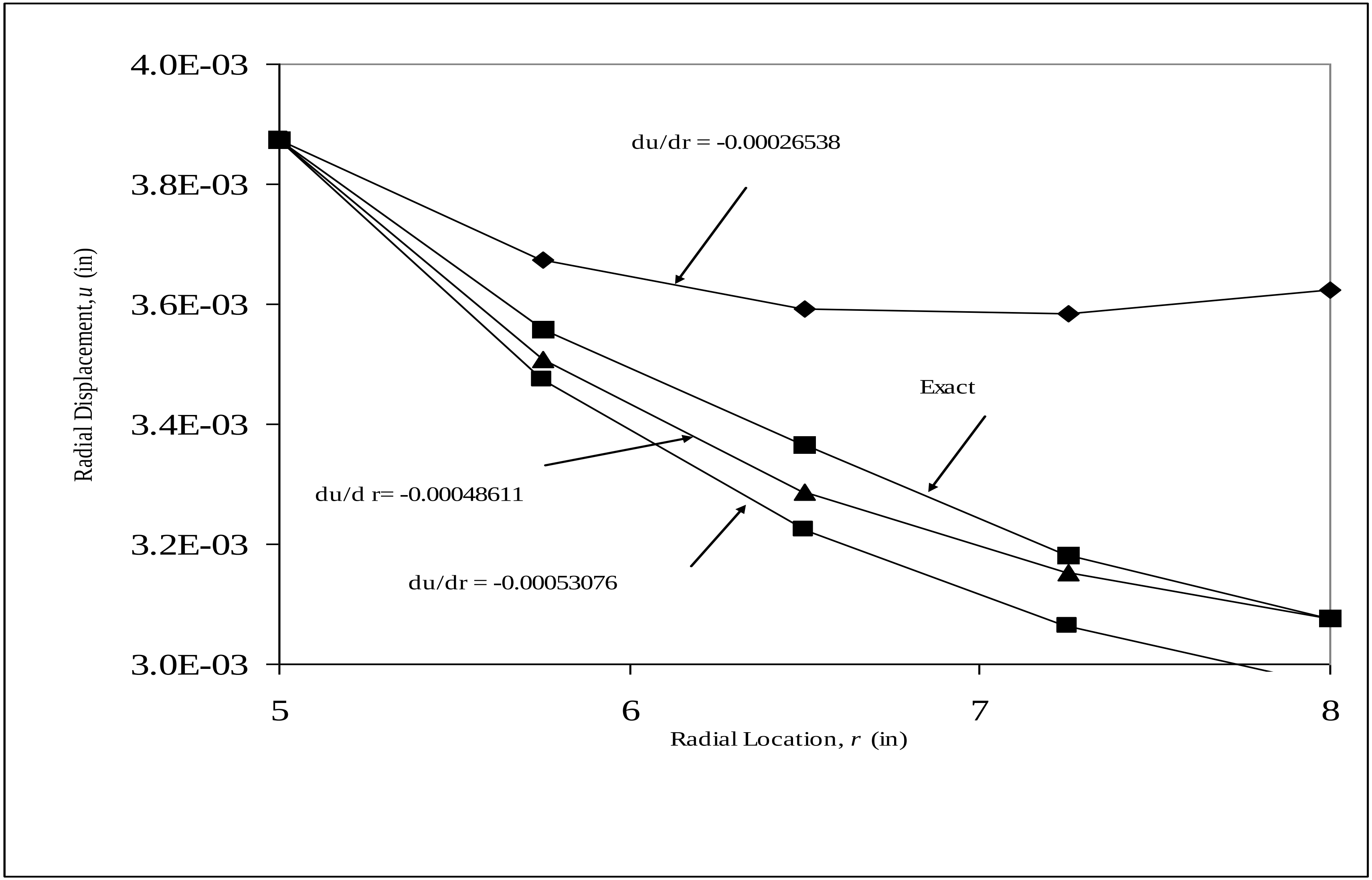
$$u(5.75) \approx u_2 = 0.0035085$$

$$u(6.50) \approx u_3 = 0.0032858$$

$$u(7.25) \approx u_4 = 0.0031518$$

$$u(8.00) \approx u_5 = 0.0030770$$

Comparisons of different initial guesses



Comparison of Euler and Runge-Kutta Results with exact results

Table 1 Comparison of Euler and Runge-Kutta results with exact results.

r (in)	Exact (in)	Euler (in)	$\|\in_t\|\%$	Runge-Kutta (in)	$\|\in_t\|\%$
5	3.8731×10^{-3}	3.8731×10^{-3}	0.0000	3.8731×10^{-3}	0.0000
5.75	3.5567×10^{-3}	3.5085×10^{-3}	1.3731	3.5554×10^{-3}	3.5824×10^{-2}
6.5	3.3366×10^{-3}	3.2858×10^{-3}	1.5482	3.3341×10^{-3}	7.4037×10^{-2}
7.25	3.1829×10^{-3}	3.1518×10^{-3}	9.8967×10^{-1}	3.1792×10^{-3}	1.1612×10^{-1}
8	3.0770×10^{-3}	3.0770×10^{-3}	1.9500×10^{-3}	3.0723×10^{-3}	1.5168×10^{-1}

Solutions to BVP: Finite Difference Method (Discretization method)

- ***Select the base points.***
- ***Divide the interval into n sub-intervals.***
- ***Use finite approximations to replace the derivatives.***
- ***This approximation results in a set of algebraic equations.***
- ***Solve the equations to obtain the solution of the BVP.***

Remarks

Finite Difference Method :

- *Different formulas can be used for approximating the derivatives.*
- *Different formulas lead to different solutions. All of them are approximate solutions.*
- *For linear second order cases, this reduces to tri-diagonal system.*

Finite difference method

- For these particular finite difference formulas, system to be solved is tridiagonal, which saves on both work and storage compared to general system of equations
- This is generally true of finite difference methods: they yield sparse systems because each equation involves few variables

Example

- Consider again two-point BVP

$$u'' = 6t, \qquad 0 < t < 1$$

  with BC

$$u(0) = 0, \qquad u(1) = 1$$

- To keep computation to minimum, we compute approximate solution at one interior mesh point, $t = 0.5$, in interval $[0, 1]$
- Including boundary points, we have three mesh points, $t_0 = 0$, $t_1 = 0.5$, and $t_2 = 1$
- From BC, we know that $y_0 = u(t_0) = 0$ and $y_2 = u(t_2) = 1$, and we seek approximate solution $y_1 \approx u(t_1)$

Example

- Replacing derivatives by standard finite difference approximations at t_1 gives equation

$$\frac{y_2 - 2y_1 + y_0}{h^2} = f\left(t_1, y_1, \frac{y_2 - y_0}{2h}\right)$$

- Substituting boundary data, mesh size, and right hand side for this example we obtain

$$\frac{1 - 2y_1 + 0}{(0.5)^2} = 6t_1$$

or

$$4 - 8y_1 = 6(0.5) = 3$$

so that

$$y(0.5) \approx y_1 = 1/8 = 0.125$$

Example (cont.)

- In a practical problem, much smaller step size and many more mesh points would be required to achieve acceptable accuracy
- We would therefore obtain *system* of equations to solve for approximate solution values at mesh points, rather than single equation as in this example

End of Chapter 1