Optimizer Case Study – Working with Differential Equations

Suppose you need to solve the following differential equations for x = 0.1, 0.2, … 1.6, where y(0.1) = 0.2 and z(0.1) = 1.0:

$$y` = 10 \cdot e^{(-2.5 \cdot x)} - 2 \cdot y$$

$$z` = x - \frac{1}{y}$$

TK Solver includes several built-in functions for solving differential equations numerically. They are all used similarly. For this example, we use the ODE_RK4 function. This function requires three inputs.

The first is the function defining the differential equations. If you have simultaneous differential equations as we have in this case study, you must use a TK Procedure Function to define the set. We will use a function called test2 to hold our equations. The second input to ODE_RK4 is the reference to the list containing the values of the independent variable. These are the values at which the differential equations are to be integrated. We will the name x for this list. The last input to ODE_RK4 is the name of the list containing the solution list names. In this case, we will have two solution lists because we have two differential equations. The list Y will be used to store the two solution list names.

Rule
call ODE_RK4('test2,'x,'Y)

The apostrophes are inserted before the names to indicate that they are values and not variables.

Our next step is to create the list Y on the List Sheet and fill it with the solution list names. We will use the names y and z for the two solution lists.

Name	Elements	Unit	Comment
Y	2		master list for unknown functions

Here are the contents of the list Y.

Value
'y
'z

We can use a table to store the independent variable and solution values. Here is the name and title on the Table Sheet.

Name	Title
Solution	Solution to test2 differential equations

Here are the definitions of the content lists.

List	Format	Width	Heading
x		10	
y		10	
z		10	

We can use the table to supply the values of independent variable and the initial conditions for y and z.

Element	x	y	z
1	.1	.2	1
2	.11		
3	.12		
4	.13		
5	.14		
6	.15		

We can use TK's List Fill Command to fill the x list with values from 0.1 to 1.6 in steps of 0.1.

The next step is to create the TK procedure function test2, which defines the set of simultaneous differential equations. Here are the statements required.

Statement
y`[1] = 10*exp(-2.5*x) - 2*y[1]
y`[2] = x - 1/y[1]

The variable x is the independent variable. Variables a, b, and c, are parameters passed into the function. The variable y` represents the derivative with respect to the independent variable x. The variable y represents the function with respect to x. The expression y`[i] represents the derivative of the ith function with respect to x. Likewise, the expression y[i] represents the value of the ith function at the value of x.

The variables are mapped to the calling ODE_RK4 function as follows.

Comment:	Simultaneous Differential Equations
Parameter Variables:	
Input Variables:	y`,y,x
Output Variables:	

The input variables must be defined in the proper sequence in order for the built-in ODE_RK4 function to work properly. The sequence must represent the derivative, function, and independent variable values respectively. Because these are passed into the function they are declared as Input Variables.

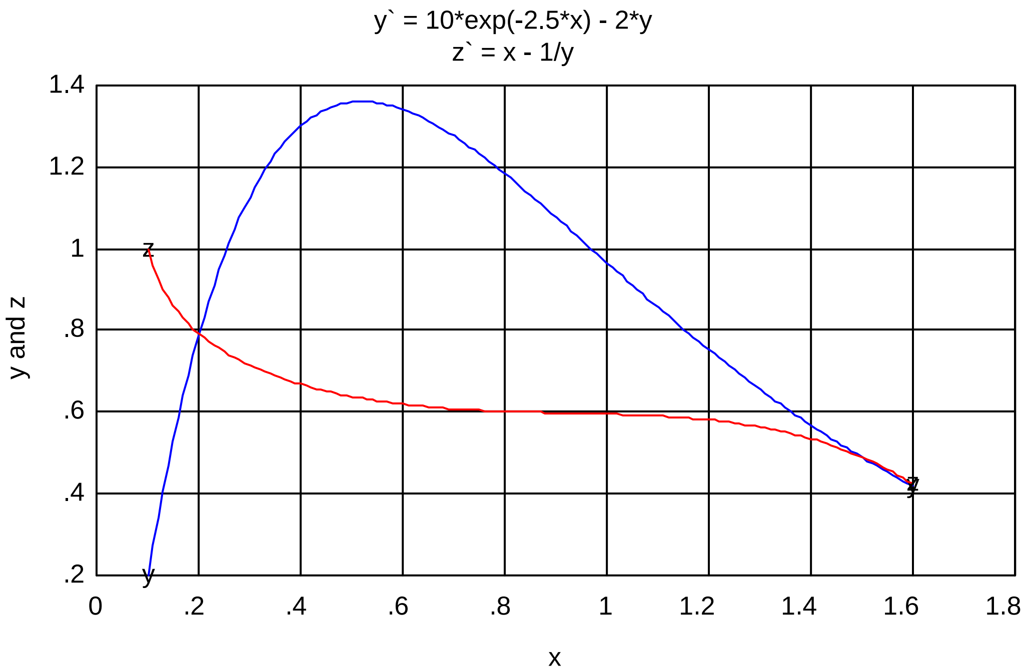
At this point, the program is ready to solve the problem. Click the solve icon and the solution values are displayed in the table. Here are the first 6 elements of the table.

Element	x	y	z
1	.1	.2	1
2	.11	.272187157	.958406162
3	.12	.341064825	.926832997
4	.13	.406744958	.90128761
5	.14	.469336143	.87978279
6	.15	.528943701	.861184223

Here are the last few elements.

Element	x	y	z
146	1.55	.452960424	.459690414
147	1.56	.446020466	.452992197
148	1.57	.439167827	.446047312
149	1.58	.432402013	.438849371
150	1.59	.425722512	.43139187
151	1.6	.419128792	.423668185

We can easily create a TK Line Chart to plot y and z vs. x.



If these solutions are to be used in other parts of the model, they can be applied in TK List Functions.

Name	Type	Arguments	Comment
fy	List	1;1	y(x)
fz	List	1;1	z(x)

Here is the definition of the fy function. The x list is used as the domain of the function. The y list is used as the range list. Using linear interpolation, the function will be continuous over the entire domain of x. The function will be undefined for values of x outside the domain.

Comment:	y(x)
Domain List:	x
Mapping:	Linear
Range List:	y

Here are a few elements of the values used in the function.

Element	Domain	Range
1	.1	.2
2	.11	.272187156689373
3	.12	.341064825359446
4	.13	.406744957765867
5	.14	.469336142769403
6	.15	.528943701223073

The fz function is set up in the same way.

We can now reference these functions anywhere else in the model. Here are two rules on the rule sheet.

Rule
y = fy(x)
z = fz(x)

If we enter a value of 0.2135 for x on the variable sheet, we can solve for y and z.

St	Input	Name	Output	Unit	Comment
	.2135	x			
		y	.84393723		
		z	.779749799		

We can also input a value for y and solve for x.

St	Input	Name	Output	Unit	Comment
		x	.255493387		
	1	y			
		z	.744108214		

It is important to note that TK List Functions backsolve directly and always return the first solution they find. In this case, there are two solutions as can be seen from the solution plot. It looks like there should be another solution somewhere near x = 0.95.

St	Input	Name	Output	Unit	Comment
	.95	x			
		y	1.02172835		
		z	.59699697		

We can use the TK Optimizer to find this solution. Using bounds of 0.8 and 1 for x, the Optimizer displays the following solution.

St	Input	Name	Output	Unit	Comment
	.969506514	x			
		y	1		
		z	.596420823		

We can also use the Optimizer to determine an intersection point for the y and z curves. We create a new rule, d = y-z, to compute the difference between y and z. We set the target variable as d with a target value of 0. We set bounds on x of 0.1 and 0.3 and the Optimizer returns the following solution.

St	Input	Name	Output	Unit	Comment
	.201438119	x			
		y	.791987159		
		z	.791989879		

The second intersection point can be found by setting the bounds on x of 1.4 to 1.6.

St	Input	Name	Output	Unit	Comment
	1.49686809	x			
		y	.491326022		
		z	.491393944		

We can also use the Optimizer to backsolve the differential equations to solve for an initial condition that results in a desired solution at a particular value of x. To do that, we need to create a rule which maps the initial conditions to variables. The TK PLACE function performs this task, placing the values of yi and zi into the first elements of the associated lists.

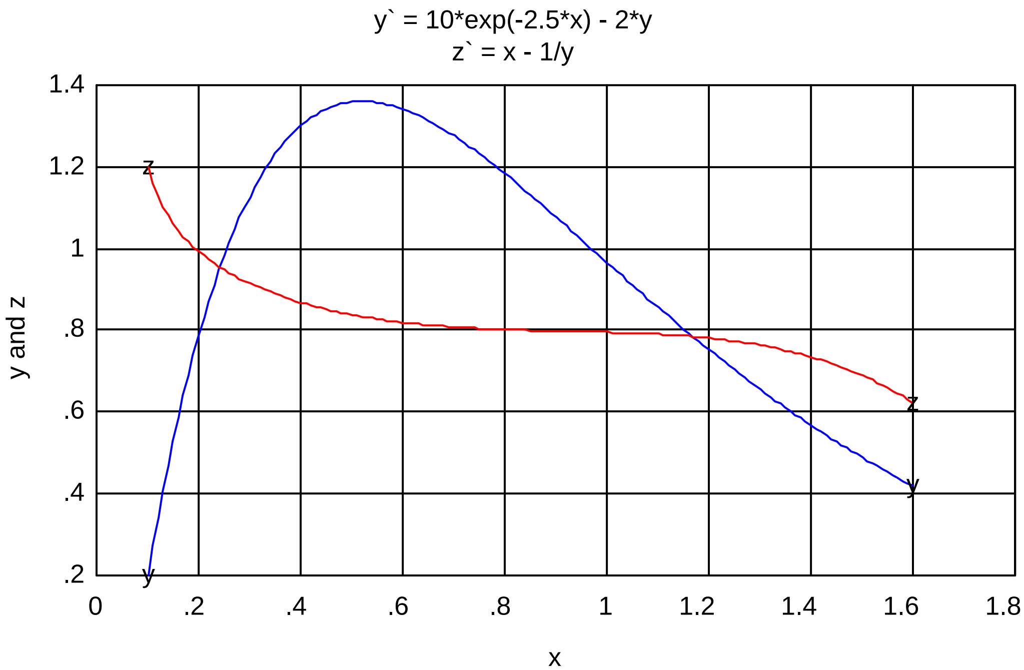
Rule
place('y,1) = yi
place('z,1) = zi

Be sure to insert these rules *above* the ODE_RK4 function call. Otherwise, they will not be applied until after the integration has been completed using the prior values. This is one of those rare times when the sequence of entries on the rule sheet is important. This is something to keep in mind whenever you are processing lists in TK as we are here.

Now we can input the initial conditions on the variable sheet.

St	Input	Name	Output	Unit	Comment
	1.49686809	x			
		y	.491326022		
		z	.491393944		
	.2	yi			
	1	zi			

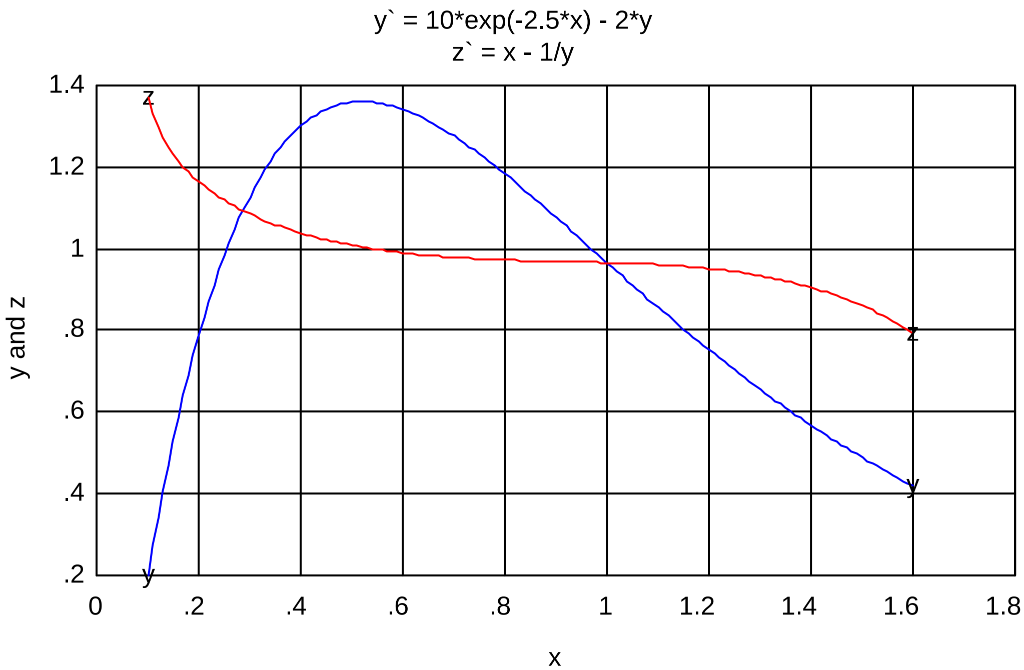
Try changing zi from 1 to 1.2. The effect can be seen on the plot. The z function is raised.



We can use the Optimizer to determine the value of zi which forces the y and z curves to intersect at x = 1. We set the target variable as d with a target value of 0 again. This time, we use zi as the decision variable, with bounds of 1 and 1.5. Here is the solution.

St	Input	Name	Output	Unit	Comment
	1	x			
		y	.966057876		
		z	.966057876		
	.2	yi			
	1.37063173	zi			

The plot confirms the solution.



y` = 10*exp(-2.5*x) - 2*y
z` = x - 1/y
y and z
1.4
1.2
1
.8
.6
.4
.2
0
.2
.4
.6
.8
1
1.2
1.4
1.6
1.8
x
z
z
y
y