Computing With COMSOL Script™: Differential Equations

Edward M. Rosen - EMR Technology Group

There are a number of systems for carrying out numerical computations. Each has its limitations, features and capabilities. The use of a particular system to solve a problem often depends on the user's familiarity with the system, its accessibility, capabilities, and the cost. Edgar [1] discusses many of the systems in use in academia.

COMSOL Script provides a large number of capabilities. It can run MATLAB® m-files, interface to COMSOL Multiphysics™ (for partial differential equations) and can carry out a large variety of mathematical calculations. It is very interactive. Having a single system to carry out such a variety of computations makes it an attractive platform.

In order to become familiar with the system, the author decided to explore COMSOL Script's capability to solve ordinary differential equations. A User's Guide [2] is available that describes the system.

Some Aspects of COMSOL Script

The results of commands entered into the command window are immediately displayed unless the command is followed by a ';'.

There is an '**edit**' command that allows editing as well as a '**type**' command. These are very convenient commands when checking the contents of an m-file while remaining in the command window. The ↑ key and ↓ key are used to recall previously entered commands.

To solve sets of differential equations:

1. Invoke COMSOL Script command window. This can be done from the COMSOL Multiphysics menu file.

2. Store any m-files (functions) in a folder that is on the search path. If the folder is named MyCOMSOLFcns then:

   path ('C:\MyCOMSOLFcns', path)

   places MyCOMSOLFcns on the top of the search path

3. Invoke the differential equation solver **daspk** . If there are no parameters:

   [t,y] = daspk ('fname', [0 5], [0;1])

   If there are parameters, b and c to be passed:

[t,y] = daspk (‘fname’, [0 5], [0;1], opts, b, c)

where fname is the name of the m-file

[0 5] is the lower and upper bounds of time (i.e independent variable)

[0;1] is the initial values for the dependent variables (for two equations)

opts is a vector of options (to set tolerances for example)
= **odeset (‘RelTol’, 1e-8)**

b and c are parameters in the differential equations

4. The ouptut is placed into the t and y vectors

For a test of daspk, three intial value differential equations were studied. The results are compared to two other systems for solving differential equations, PLAS [3] and Polymath [4].

The White-dwarf Equation

Davis [5} describes the equation which S. Chandrasekhar [6] introduced in his study of the gravitational potential of degenerate (white-dwarf) stars:

$$x\frac{d^2y}{dx^2} + 2\frac{dy}{dx} + x(y^2 - c)^{3/2} = 0 \qquad (1)$$

y(0) = 1

y’(0) = 0

Rewriting Equation (1) in terms of y_2 and t results in::

$$t\frac{d^2y_2}{dt^2} + 2\frac{dy_2}{dt} + t({y_2}^2 - c)^{3/2} = 0 \qquad (2)$$

Let

$$y_1 = \frac{dy_2}{dt}$$

Then

$$y_1^{'} = -2\frac{y_1}{t} - (y_2^2 - c)^{3/2} \qquad (3)$$

and

$$y_2^{'} = y_1$$

with boundary conditions:

$$y_1(0) = 0$$

$$y_2(0) = 1$$

1) The following m-file is stored in the folder MyCOMSOLFcns with the name **dwarf.m** (using any text editor)

```
function ydot = dwarf(t, y, c)
ydot = zeros(2, 1);
ydot(1) = -2*y(1)/t- (y(2)^2-c) ^1.5;
ydot(2) = y(1);
ydot(3) = 1;
```

Since the output [t,y] comes out sequentially, first t and then y, it is sometimes convenient to add an extra equation (of form $dy_3/dt = 1$) to record the independent variable along with the two dependent variables on output.

2) In the command window the 'path' command is used to add MyCOMSOLFcns to the search path:

path ('C:\MyCOMSOLFcns', path)

3) The function **daspk** is invoked:

[t,y] = daspk ('dwarf', [0.000001 4], [0;1;0.000001], odeset ('RelTol', 1e-8) , 0.1)

In order to avoid division by 0, the initial time is set to 1e-6. Final time is 4.
The function **odeset** is used to set the **Relative Tolerance** to 1.e-8 (default is 1e-3)
The 0.1 is the value of c.

.

The output is placed into t and y and displayed in the command window.

Figure 1 indicates the results of the integration and compares it to the results obtained from PLAS and Polymath.

White Dwarf Equation Value of c= 0.1			
t	<------------------------Value of y_2--------------------------------->		
	COMSOL Script	PLAS version 1.2	POLYMATH
	RelTol = 1e-8	hr = 0.001	Stiff
1.00E-06	1	1	1
0.5	0.9656	0.9656475	0.965655
1	0.8755	0.875471	0.875471
1.5	0.7579	0.7578576	0.7578571
2	0.6388	0.6387517	0.6387511
2.5	0.533	0.5330401	0.53304
3	0.4458	0.4458161	0.4458162
3.5	0.3766	0.3766133	0.3766158
4	0.3227	0.3226673	0.322668

Figure 1. Solution of the White Dwarf Equation

The Generalized Equation of Blasius

As described in Davis [5] the generalized equation of Blasius is:

$$\frac{d^3y}{dx^3} + ay\frac{d^2y}{dx^2} = b\left[\left(\frac{dy}{dx}\right)^2 - 1\right] \qquad (3)$$

where

$y(0) = 0$
$y'(0) = 0$
$y'(x) \to k$ as $x \to \infty$ (k is a constant)

When $b \geq 0$ a unique solution exists.

The equation arises when one considers the flow of a fluid which streams past a very thin flat plate placed edgeways in it (Schlichting [7]).

Rewriting Equation (3) in terms of y_3 and t results in

$$\frac{d^3y_3}{dt^3} + ay\frac{d^2y_3}{dt^2} = b\left[\left(\frac{dy_3}{dt}\right)^2 - 1\right] \tag{4}$$

Let $y_2 = \dfrac{dy_3}{dt}$ or $y_3' = y_2$

$y_1 = \dfrac{dy_2}{dt}$ or $y_2' = y_1$

Then substituting in Equation (4) results in:

$$y_1' = -ay_3y_1 - b\,(y_2^2 - 1)$$

With boundary conditions

$y_1\,(0) = 1.32824$ (Taken from Davis [5] p 404
for k = 2 as given by Howarth [8]} $\rightarrow y''$

$y_2\,(0) = 0$ $\rightarrow y'$
$y_3\,(0) = 0$ $\rightarrow y$

A file **<u>blasius.m</u>** is placed into the folder C:\MyCOMSOLFcns:

```
function ydot = blasius(t, y, a, b)
ydot = zeros(3,1);
ydot(1) = -a*y(3)*y(1) - b*(y(2)^2-1);
ydot(2) = y(1);
ydot(3) = y(2);
```

In the command window the folder MyCOMSOLfcns is placed on the path:

path (‘C:\MyCOMSOLfcns’, path)

Then the following is entered into the command window:

[t,y] = daspk(‘blasius’, [0 4.4], [1.32824; 0; 0], odeset (‘RelTol’, 1e-8) , 1, 0)

Alternately:

opts = odeset ('RelTol', 1e-8)
[t,y] = daspk('blasius', [0 4.4], [1.32824; 0; 0], opts, 1, 0)

The integration is carried out between 0 and 4.4, the initial values of the dependent variables are 1.32824, 0. and 0 and the value of a = 1 and b= 0..

The results of the integration is shown in Figure (2). The results of Howarth is taken from Davis [5] is included for comparison

Blasius Equation				
a = 1 b = 0				
y(0) = 0 y'(0) = 0 y''(0) = 1.32824				
Values at t = 4.4	**POLYMATH**	**PLAS**	**COMSOL Script**	**Howarth**
	Stiff			
y3 --> y	7.7079255	7.079242	7.0792	7.07923
y2 --> y'	2.00001	2.000007	2	2
y1 --> y''	3.38E-06	3.38E-06	3.70E-06	0.00E+00

Figure 2. Solution to the Blasius Equation

Originally Blasius solved Equation (4) with a = 0.5, b= 0 and k = 1. Howarth (in Schlicting [7], p 107) indicated that setting y''(0) = 0.33206 in this case leads to a solution (i.e. $y'(x) \to 1$ as $x \to \infty$).

<u>The Thomas-Fermi Equation</u>

In 1927 L. H. Thomas and E. Fermi independently gave a method of studying the electron distribution in an atom using statistics for a degenerate gas. This led to the Thomas-Fermi equation:

$$\frac{d^2y}{dx^2} = \frac{1}{\sqrt{x}}\, y^{3/2} \qquad (5)$$

$y(0) = 1$
$y(x) \to 0 \quad \text{as } x \to \infty$

Rewriting Equation (5) in terms of y_2 and t results in

$$\frac{d^2 y_2}{dt^2} = \frac{1}{\sqrt{t}} y_2^{\ 3/2} \qquad (6)$$

Letting

$y_1 = \dfrac{dy_2}{dt}$ or $y_2' = y_1$

Substituting into Equation (6)

$$y_1' = \frac{1}{\sqrt{t}} y_2^{\ 3/2}$$

with

$y_1(0) = -1.58807102$ (Taken from Lee and Wu [9]) $\rightarrow y'$

$y_2(0) = 1$ $\rightarrow y$

To solve Equation (6) an m-file (fermi.m) is developed and placed into folder MyCOMSOLFcns.

```
function ydot = fermi(t, y)
ydot = zeros(2,1);
ydot(1) = (t^-0.5)*(y(2)^1.5);
ydot(2) = y(1);
```

In the command window the path is set (as above). Then the following is entered into the command window (to set the parameters):

opts = odeset (‘AbsTol’, 1e-8, ‘RelTol’, 1e-8)

Then to integrate from 1e-11 to (say) 5

[t,y] = daspk (‘fermi’, [1e-11, 5], [-1.58807102;1], opts)

The results are shown in Figure (3).

The solution is **very** sensitive to the initial value of $y_1(0)$. The value chosen (call it ω) such that $y(x) \rightarrow 0$ as $x \rightarrow \infty$ corresponds to a neutral atom. Values smaller (more negative than ω) result in the value of y to dropping to 0. Values greater than ω result in an unbounded solution.

Lee and Wu[9] discuss these numerical problems and generate a solution shown in Figure (3). Each of the solvers (Script, Polymath, PLAS) had to have their parameters adjusted so as to match the Lee and Wu solution. up to about x = 5. However, at large values of x (greater than 5) the solvers begin to deviate and finally "blow-up".

A detailed discussion of the equation from a detailed physics point of view can be found in References [10] and [11].

Thomas-Fermi Potential of a Neutral Atom: $y_1(0) = -1.58807102$				
	Value of y_2 ---> y			
t	Lee/Wu ([9]	COMSOL Script[1]	POLYMATH[2]	PLAS[3]
		RelTol = 1e-8	STIFF	Adams /BDF
1.00E-11	1	1	1	1
0.004	0.994	0.994	0.994	0.994
0.008	0.9882	0.9882	0.9882	0.9882
0.01	0.9854	0.9854	0.9854	0.9854
0.02	0.9721	0.972	0.972	0.972
0.05	0.9352	0.9352	0.9352	0.9352
0.08	0.9022	0.9022	0.9022	0.9022
0.1	0.8818	0.8817	0.8817	0.8817
0.4	0.6596	0.6595	0.6595	0.6595
0.8	0.4849	0.4849	0.4849	0.4849
1	0.424	0.424	0.424	0.424
1.5	0.3148	0.3148	0.3148	0.3148
2.5	0.193	0.1929	0.1929	0.1929
3.5	0.1294	0.1293	0.1293	0.1293
4.5	9.19E-02	9.18E-02	9.18E-02	9.18E-02
5	7.88E-02	7.86E-02	7.86E-02	7.86E-02
6	5.94E-02	5.91E-02	5.91E-02	5.91E-02
7	4.61E-02	4.56E-02	4.56E-02	4.56E-02
8	3.66E-02	3.59E-02	3.59E-02	3.59E-02
10	2.43E-02	2.30E-02	2.30E-02	2.30E-02
15	1.08E-02	5.80E-02	6.27E-03	
50	6.32E-04			
1000	1.65E-07			
1	Script:	AbsTol = 1.e-8, RelTol = 1e-8		
2	POLYMATH:	Independent Variable Accuracy = 1e-5		
		Initial Step Size =1e-5		
		Minimum Allowed StepSize 1e-9		
3	PLAS:	hr = 1e-6		

Figure 3. Solution of the Thomas-Fermi Equation

Conclusions

For the beginner there is a learning curve to effectively work in the COMSOL Script environment However, for the solution of ordinary differential equations, the system offers a reliable routine, **daspk,** which duplicates the results obtained from other routines**.** Adding an extra equation to record the independent variable is a convenient method to output the dependent and independent variables together.

Nomenclature

1. *White Dwarf Equation*

y	= dependent variable
x	= t, the independent variable
c	= a constant
y_1	= y'
y_2	= y, dependent variable
y_3	= dependent variable equal to t ($dy_3/dt = 1$)

2 *The Generalized Equation of Blasius*

y	= dependent variable
x	= t, the independent variable
a	= constant, set to 1
b	= constant, set to 0
k	= constant , equal to 2.0
y_1	= y''
y_2	= y'
y_3	= y, the dependent variable

3. *Thomas-Fermi Equation*

y	= dependent variable
x	= t, the independent variable
y_1	= y'
y_2	= y, dependent variable
ω	= value of $y_1(0)$ which satisfies boundary condition $y(x) \rightarrow 0$ as $x \rightarrow \infty$

References

1. Edgar, T. E. “Computing Through the Curriculum: An Integrated Approach for Chemical Engineering”, http://www.CACHE.org/

2. COMSOL Script User’s Guide, Version 1.0 , COMSOL AB 1.

3. Rosen, E. M., “Solving Differential Equations Using PLAS”, *CACHE News*, No 55 Fall 2002

4. Cutlip , M.B. and M. Shacham, “Polymath”, http://www.polymat-software.com/

5. Davis, Harold T., “Introduction to Nonlinear Differential and Integral Equations”, *United States Atomic Energy Commission* Sept 1960

6. Chandrasekhar, S. *An Introduction to Stellar Structure*, 1958

7. Schlichting, H., *Boundary Layer Theory*, McGraw-Hill, New York (1955), pp 103, 118

8. Howarth, L., “On the solution of the laminar boundary layer equations”, *Proc. Roy. Soc., London, A*, **164**, 547 (1938).

9. Lee, P.S. and Ta-You Wu, “Statistical Potential of Atomic Ions”, Chinese Journal of Physics, Vol 35, No 6-11, Dec 1997 http://psroc.phys.ntu.edu.tw/cjp/find_content.php?year=1997&vol=35&no=6-II

10. Physics 221B Spring 1997 Notes 23 The Thomas-Fermi Model http://socrates.berkeley.edu/~tplagge/221-littlejohn/23-thomas-fermi-model.pdf

11. Hille, E. “On the Thomas-Fermi Equation” http://www.pnas.org/cgi/reprint/62/1/7