

SOLVING DIFFERENTIAL EQUATIONS WITH MAPLE

VENKAT R. SUBRAMANIAN, RALPH E. WHITE
Center for Electrochemical Engineering
University of South Carolina • Columbia, SC 29208

We have developed a mathematical methods course for seniors and first-year chemical engineering graduate students that uses the matrix exponential and Maple^[1] to solve initial value problems, boundary value problems, and partial differential equations.^[2,3] We give here a brief description of some of what we cover in our course.

INITIAL VALUE PROBLEMS: ORDINARY DIFFERENTIAL EQUATIONS

The matrix exponential ($\exp[At]$) arises in the solution of a system of linear, ordinary differential equations (ODEs). For example, the time dependent concentration of species in a series reaction^[4] ($A \xrightarrow{k_1} B \xrightarrow{k_2} C$) can be obtained by writing the material balance equations in matrix form,^[2]

$$\frac{d\mathbf{Y}}{dt} = \mathbf{A}\mathbf{Y} + \mathbf{b}(t) \quad (1)$$

and solving Eq. (1) by using the matrix exponential^[3]

$$\mathbf{Y} = \exp(\mathbf{A}t)\mathbf{Y}_0 + \int_0^t \exp[-\mathbf{A}(\tau - t)]\mathbf{b}(\tau)d\tau \quad (2)$$

where \mathbf{A} has been assumed to be a constant and \mathbf{Y}_0 is the initial condition vector. Equation 2 can be thought of as an extension of the integrating factor approach for solving a single, first-order linear differential equation. Equation 2 allows one to solve coupled linear, first-order differential equations symbolically as a function of time, the initial conditions, and the parameters (k_1 and k_2 in this case).

In the Appendix, we show how to use Maple to solve for the concentration profiles for this series reaction. Note that Maple is capable of finding the exponential matrix, $\exp[At]$ (mat in the Appendix) with t as the independent variable and k_1 and k_2 as parameters. Alternatively, one can program an

algorithm in Maple to determine the matrix exponential. When the coefficient matrix \mathbf{A} is a function of time, t , the solution can be obtained by using the *matrizant*^[5-7] and not by the matrix exponential. Linear systems of ODEs can also be solved using Maple's Laplace transform commands or by using Maple's *dsolve* command.

Nonlinear, first-order ODEs can also be solved symbolically with Maple's *dsolve* command. Maple will provide either an explicit solution or an implicit solution, depending on the degree of non-linearity of the equations. In addition, one can use Maple's *dsolve* (type = series) command to obtain a series solution in terms of the independent variable and parameters for both linear and nonlinear initial value problems. Also, systems of nonlinear first-order ODEs can be solved numerically by using Maple's *dsolve* (type = numeric) command. For nonlinear systems of differential and algebraic equations (DAEs),^[8] Taylor's BESIRK^[9] can be used with Maple.

PARAMETER ESTIMATION

Given a set of data and the model equations, Maple can be used to estimate parameters such as k_1 and k_2 by nonlinear parameter estimation. The procedure consists of writing the

Venkat R. Subramanian is a doctoral student at the University of South Carolina. He received a BS in chemical and electrochemical engineering from the Central Electrochemical Research Institute in India in 1997. His research involves mathematical modeling of electrochemical systems—lithium ion batteries, metal hydride batteries, and current-density distributions. His research interest also includes symbolic/semi-analytic/numerical solution of boundary value problems using Maple, FEMLAB, etc., and nonlinear parameter estimation from experimental curves.

Ralph E. White is a professor and department chair in the Chemical Engineering Department at the University of South Carolina. He holds a BS degree from the University of South Carolina and received his MS and PhD degrees from the University of California at Berkeley.

... a mathematical methods course for seniors and first-year chemical engineering graduate students that uses the matrix exponential and Maple to solve initial value problems, boundary value problems, and partial differential equations.

sensitivity equations^[10] or equivalently, the Jacobian elements,^[11] and solving them simultaneously with the model equations. The matrix exponential method (Eq. 2) can be used easily for linear systems, and `dsolve(numeric)` or `BESIRK` can be used for nonlinear systems. Also, Maple can be used to find the Jacobians symbolically.

**BOUNDARY VALUE PROBLEMS:
ORDINARY DIFFERENTIAL EQUATIONS**

Linear, two-point boundary value problems, which consist of a second-order differential equation subject to two boundary conditions at different points, can be solved by using the matrix exponential (Eq. 2) and Maple. One simply changes the second-order differential equation into a system of first-order ODEs and solves the system with an unknown condition at $x=0$. This unknown condition at $x=0$ is then determined directly by applying the outer boundary condition at $x=1$ to the solution, as illustrated by the solution of the heat transfer from a rectangular fin^[12] in the Appendix.

The direct method we have developed has apparently been overlooked as a method for solving two-point boundary value problems. Perhaps the reason for this is that chemical engineers are not familiar with the matrix exponential (as a function of the independent variable, t) and do not use Eq. 2 to solve equations. Also, before we had software such as Maple, evaluation of the matrix exponential was difficult and time consuming. In addition, for problems that yield a coefficient matrix A that depends on the independent variable, one must use the matrixant,^[5-7] which is difficult to do without software such as Maple. It is important to note that this direct method (Eq. 2) can be used instead of the method of undetermined coefficient, variations of parameters, etc., that are commonly used for solving second-order linear BVPs.

Maple can also be used to obtain approximate solutions in symbolic form for linear, second-order BVPs. The process for doing this consists of using finite difference expressions to represent the derivatives in the governing equation and the boundary conditions, which results in an equation of the form^[13]

$$AX = b \quad (3)$$

The solution to Eq. 3 can be found by using Maple to find

the inverse of A symbolically, so that the solution is then

$$X = A^{-1}b \quad (4)$$

This method has been applied to the heat transfer from a rectangular fin example in the Appendix, where it is shown that the solution consists of expressions for the values of the dependent variables at the node points in terms of the parameters of the problem (H in this case). Alternatively, Maple's `fsolve` command can be used to solve linear/nonlinear algebraic equations arising during discretization of linear/nonlinear ODEs. Alternatively, one can use `NEWTON`, developed by Taylor,^[9] to solve a system of nonlinear equations. The efficiency of `fsolve` compared with `NEWTON` depends on the problem.

PARTIAL DIFFERENTIAL EQUATIONS (PDEs)

Time dependent heat/mass transfer in a semi-infinite domain occurs frequently in chemical engineering.^[14-16] The parabolic PDEs that describe these processes are often solved by a similarity variable transformation. The solutions obtained usually involve error functions or complimentary error functions.^[2,3,15,16] Instead of using a similarity variable approach, one can apply the Laplace transformation technique for the time variable and solve the resulting ODE in the Laplace domain using Eqs. (1) and (2). The solution once obtained in the Laplace domain can be inverted back to the time domain. We illustrate (in the Appendix) how this can be done by using Maple's built-in Laplace transformation commands. We used this method to solve many of the problems given in References 15 and 16 for various boundary/initial conditions and source terms in the governing equation.

Linear, parabolic PDEs in a finite domain can also be solved by this method by applying the Laplace transformation in time, solving the resulting second-order differential equation with the boundary conditions in the Laplace domain, using the matrix exponential approach presented above, followed by using the Heaviside expansion theorem^[3] to invert the solution back into the time domain. Maple is especially useful for this purpose when multiple poles are involved in the inversion^[3,17,18] because Maple's limit command can be used to find the residues.

Parabolic PDEs in a finite domain can also be solved by separation of variables. Maple can be used to separate the variables and integrate the resulting ODEs in time and x variable. Oftentimes, the eigenvalues are given by a nonlinear transcendental equation. If one were using FORTRAN, one would have to find the eigenvalues by writing a Newton-Raphson code, or refer to tables.^[15,16] Maple can be used to find the eigenvalues and store them in an array. The solution obtained from this method is usually in the form of an infinite series^[3,4,17,18] whose coefficients are found out by applying the Sturm-Liouville theorem. Maple is extremely useful in evaluating the definite integrals involved in this tedious process. Since the final solution is stored symbolically by Maple, many case studies can be done for different parameters and the results can be plotted easily by using Maple. This method can be extended to treat Laplace's equation in two spatial coordinates (x and y). This type of equation occurs in steady-state heat/mass transfer in a rectangular plate/cylinder^[15,16] and potential distributions in electrochemical cells.^[19,20]

Linear parabolic PDEs (*e.g.*, heat transfer in a rod) can be solved by method of lines.^[21,22] The derivatives in x are replaced by finite differences. The resulting set of coupled linear first-order ODEs are usually numerically integrated in time.^[21,22] We represent the resulting first-order ODEs in the matrix form (Eq. 1) and the solution is given by Eq. 2. This allows one to solve for the dependent variable (*e.g.*, temperature, concentration) as a function of time, as shown in the Appendix. We call this the semi-analytical method.^[23] We have found it to be an extremely useful method for solving linear PDEs. Nonlinear PDEs can also be solved this way, by quasi-linearizing the nonlinear term and iterating for convergence.^[24] Another important aspect is that it allows us to solve Laplace's equation (two dimensions in x and y), which involves solving simultaneously coupled linear BVPs.^[18]

Finally, for coupled nonlinear PDEs (parabolic, elliptic,^[25] or any kind), finite differences can be used in both x and y (or t) and solved simultaneously using Maple's `fsolve` command.

In summary, we have found that it is possible to solve old problems in new ways by using the power of Maple. We suggest that the readers consider trying Maple and contact us for help with their problems if necessary. We are planning to write a book based on the Maple worksheets that one of us (VRS) has prepared for our course and this paper.

REFERENCES

1. Waterloo Maple Inc., <<http://www.maplesoft.com>>
2. Rice, Richard G., and Duong D. Do, *Applied Mathematics and Modeling for Chemical Engineers*, John Wiley & Sons, Inc., New York, NY (1995)
3. Varma, Arvind, and Massimo Morbidelli, *Mathematical Methods in Chemical Engineering*, Oxford University Press, New York, NY (1997)
4. Fogler, H. Scott, *Elements of Chemical Reaction Engineering*, Prentice Hall, Inc., Englewood Cliffs, NJ (1999)
5. Amundson, N.R., *Mathematical Methods in Chemical Engineering. I: Matrices and Their Applications*, Prentice Hall, Inc., Englewood Cliffs, NJ (1966)
6. Taylor, R., and R. Krishna, *Multi-Component Mass Transfer*, John Wiley & Sons, Inc., New York, NY (1993)
7. Subramanian, V.R., B.S. Haran, and R.E. White, *Comp. and Chem. Eng.*, **23**(3), 287 (1999)
8. Holland, C.D., and A.I. Liapis, *Computer Methods for Solving Dynamic Separation Problems*, McGraw-Hill, Inc., New York, NY (1983)
9. Schwalbe, D., H. Kooijman, and R. Taylor, *Mapletech*, **3**(2), 47 (1996)
10. Constantinides, A., and N. Mostoufi, *Numerical Methods for Chemical Engineers with Matlab Applications*, Prentice Hall, Englewood Cliffs, NJ (1999)
11. Bard, Y., *Nonlinear Parameter Estimation*, Academic Press, New York, NY (1974)
12. Davis, M.E., *Numerical Methods and Modeling for Chemical Engineers*, John Wiley & Sons, New York, NY (1984)
13. Subramanian, V.R., and R.E. White, "Symbolic Solutions for Boundary Value Problems with Maple," accepted for publication in *Computers and Chemical Engineering*
14. Schiesser, W.E., and C.A. Silebi, *Computational Transport Phenomena*, Cambridge University Press, New York, NY (1997)
15. Carslaw, H.S., and J.C. Jaeger, *Conduction of Heat in Solids*, Oxford University Press, London (1973)
16. Crank, J., *Mathematics of Diffusion*, Oxford University Press, New York, NY (1975)
17. Subramanian, V.R., and R.E. White, in *Tutorials in Electrochemical Engineering—Mathematical Modeling*, R.F. Savinell, J.M. Fenton, A.C. West, S.L. Scanlon, and J. Weidner, eds., PV99-14, p. 100, The Electrochemical Society Proceedings Series, Pennington, NJ (1999)
18. Subramanian, V.R., and R.E. White, "New Separation of Variables Method for Composite Electrodes with Galvanostatic Boundary Conditions," submitted to *J. of the Electrochem. Soc.*
19. Subramanian, V.R., and R.E. White, *J. of the Electrochem. Soc.*, **147**(5), 1636 (2000)
20. Newman, J.S., *Electrochemical Systems*, Chapter 18, Prentice Hall, Englewood Cliffs, NJ (1991)
21. Cutlip, M.B., *CACHE News*, **18**, Fall (1998)
22. Taylor, R., *CACHE News*, **5**, Fall (1999)
23. De Vidts, P., and R.E. White, *Comp. and Chem. Eng.*, **16**(10/11), 1007 (1992)
24. Haran, B.S., and R.E. White, *Comp. Appl. in Eng. Ed.*, **4**(3), 229 (1996)
25. Nguyen, T.V., and R.E. White, *Comp. and Chem. Eng.*, **11**(5), 543 (1987)

APPENDIX: Solving Differential Equations with Maple

[Appendix

[Initial Value Problems (Series Reactions, reference 4)

[> restart:with(linalg):

[The governing equations for the concentrations of species involved in a series reaction can be written as (reference 4)

[> eq1:=diff(Ca(t),t)=-k1*Ca(t);eq2:=diff(Cb(t),t)=k1*Ca(t)-k2*Cb(t);

$$eq1 := \frac{\partial}{\partial t} Ca(t) = -k1 Ca(t)$$

$$eq2 := \frac{\partial}{\partial t} Cb(t) = k1 Ca(t) - k2 Cb(t)$$

[with the initial conditions

[> Ca(0):=Ca0;Cb(0):=0;

$$Ca(0) := Ca0$$

$$Cb(0) := 0$$

[These two equations can be written in matrix form (see equation 1 in the text, note that in this case b is zero).

[> A:=matrix(2,2,[coeff(rhs(eq1),Ca(t)),coeff(rhs(eq1),Cb(t)),coeff(rhs(eq2),Ca(t)),coeff(rhs(eq2),Cb(t))]);Y0:=matrix(2,1,[Ca0,0]);

$$A := \begin{bmatrix} -k1 & 0 \\ k1 & -k2 \end{bmatrix}$$

$$Y0 := \begin{bmatrix} Ca0 \\ 0 \end{bmatrix}$$

[The solution is given by equation 2 with b = 0.

[> mat:=exponential(A,t);sol:=evalm(mat*Y0);

$$mat := \begin{bmatrix} e^{(-k1 t)} & 0 \\ k1 (e^{(-k2 t)} - e^{(-k1 t)}) & e^{(-k2 t)} \\ -k2 + k1 & \end{bmatrix}$$

$$sol := \begin{bmatrix} e^{(-k1 t)} Ca0 \\ k1 (e^{(-k2 t)} - e^{(-k1 t)}) Ca0 \\ -k2 + k1 \end{bmatrix}$$

[> Ca:=sol[1,1];Cb:=sol[2,1];

$$Ca := e^{(-k1 t)} Ca0$$

$$Cb := \frac{k1 (e^{(-k2 t)} - e^{(-k1 t)}) Ca0}{-k2 + k1}$$

[From a material balance

[> Cc:=simplify(Ca0-Ca-Cb);

$$Cc := - \frac{Ca0 (k2 - k1 - e^{(-k1 t)} k2 + k1 e^{(-k2 t)})}{-k2 + k1}$$

[Boundary Value Problems-ODEs (Heat transfer in a Rectangular Fin, reference 12)

[> restart:with(linalg):

[Consider the heat transfer in a rectangular fin (see references 7,12,13)

[> eq:=diff(theta(x),x\$2)=H^2*theta(x);

$$eq := \frac{\partial^2}{\partial x^2} \theta(x) = H^2 \theta(x)$$

[where H is the dimensionless heat transfer coefficient and θ is the dimensionless temperature. The boundary conditions are

[> bc1:=theta(0)=1;bc2:=D(theta)(1)=0;

$$bc1 := \theta(0) = 1$$

$$bc2 := D(\theta)(1) = 0$$

[We can solve this BVP by using the matrix exponential method (equations 1,2) by first converting the second order ODE

[into a system of first order ODEs.

```
[ > Y[1]=theta(x):Y[2]=diff(theta(x),x):Y(x)=matrix(2,1,[Y[1],Y[2]]):
> A:=matrix(2,2):A[1,1]:=0:A[1,2]:=1:A[2,1]:=H^2:A[2,2]:=0:A:=evalm(A);
```

$$A := \begin{bmatrix} 0 & 1 \\ H^2 & 0 \end{bmatrix}$$

[The initial condition vector is

```
[ > Y0:=matrix(2,1,[1,Y[20]]);
```

$$Y0 := \begin{bmatrix} 1 \\ y_{20} \end{bmatrix}$$

[The solution for the matrix differential equation is (equation 2 with $\mathbf{b} = 0$).

```
[ > mat:=evalm(exponential(A,x)):mat:=map(convert,mat,trig):Y:=evalm(mat*Y0);
```

$$Y := \begin{bmatrix} \cosh(Hx) + \frac{\sinh(Hx)y_{20}}{H} \\ \frac{1}{2}H(\cosh(Hx) + \sinh(Hx)) - \frac{1}{2}H(\cosh(Hx) - \sinh(Hx)) + \cosh(Hx)y_{20} \end{bmatrix}$$

[The unknown constant y_{20} can be obtained by using the boundary condition at $x = 1$ (equate the second row in the above equation for Y to with $x = 1$).

```
[ > Y1:=evalm(subs(x=1,evalm(Y)));Y[20]:=solve(Y1[2,1]=0,Y[20]);
```

$$Y1 := \begin{bmatrix} \cosh(H) + \frac{\sinh(H)y_{20}}{H} \\ \frac{1}{2}H(\cosh(H) + \sinh(H)) - \frac{1}{2}H(\cosh(H) - \sinh(H)) + \cosh(H)y_{20} \end{bmatrix}$$

$$y_{20} := -\frac{H \sinh(H)}{\cosh(H)}$$

[Substitution of y_{20} back into the solution followed by simplification yields the desired solution.

```
[ > Y:=simplify(evalm(Y));Ym:=combine(Y[1,1]);
```

$$Y := \begin{bmatrix} \frac{\cosh(Hx)\cosh(H) - \sinh(Hx)\sinh(H)}{\cosh(H)} \\ \frac{H(\cosh(H)\sinh(Hx) - \cosh(Hx)\sinh(H))}{\cosh(H)} \end{bmatrix}$$

$$Ym := \frac{\cosh(Hx - H)}{\cosh(H)}$$

[The solution to this ODE can also be obtained easily by using Maple's dsolve command.

```
[ > Ya:=rhs(dsolve({eq,bc1,bc2},theta(x)));
```

$$Ya := \cosh(Hx) - \frac{\sinh(H)\sinh(Hx)}{\cosh(H)}$$

[Maple solves the given ODE subject to the boundary conditions. However, it does not give the solution in the desired (more familiar) form. Fortunately, Maple's combine command can be used to obtain the desired form (see reference 12, p. 73).

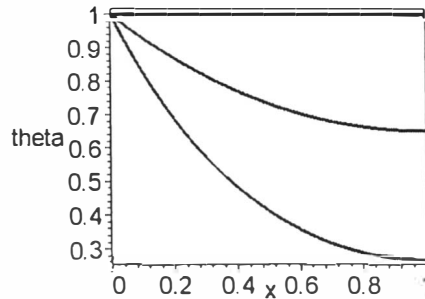
```
[ > Ya:=combine(Ya);
```

$$Ya := \frac{\cosh(-H + Hx)}{\cosh(H)}$$

[This solution can be plotted easily vs x for particular values of H .

```
[ > with(plots):p[1]:=plot(subs(H=0,Ym),x=0..1,thickness=4,color=red,axes=boxed):p[2]
:=plot(subs(H=1,Ym),x=0..1,thickness=4,color=red):p[3]:=plot(subs(H=2,Ym),x=0..
1,thickness=4,color=red):
[ > display(seq(p[i],i=1..3),labels=[x,theta]);
```

APPENDIX: Solving Differential Equations with Maple



The same problem can be solved symbolically by applying finite differences in x . For illustrations two interior node points are used here.

```
> N:=2:Eq[0]:=y[0]=1:for i from 1 to N do
  Eq[i]:=subs(diff(theta(x),x$2)=(y[i+1]-2*y[i]+y[i-1])/h^2,theta(x)=y[i],eq)=0;od
  Eq[N+1]:=(y[i-2]-4*y[i-1]+3*y[i])/2/h=0:
```

```
> eqns:=[seq((Eq[j]),j=0..N+1)];Y:=[seq(Y[i],i=0..N+1)];
```

$$eqns := \left[y_0 = 1, \frac{y_2 - 2y_1 + y_0}{h^2} - H^2 y_1 = 0, \frac{y_3 - 2y_2 + y_1}{h^2} - H^2 y_2 = 0, \frac{1}{2} \frac{y_1 - 4y_2 + 3y_3}{h} = 0 \right]$$

$$Y := [y_0, y_1, y_2, y_3]$$

Maple is used to generate the A matrix and b (equations 3 and 4).

```
> h:=eval(1/(N+1)):A:=genmatrix(eqns,Y,'b'):evalm(b):
> sol:=evalm(inverse(A)&*b);
```

$$sol := \left[1, 9 \frac{6 + H^2}{54 + 24H^2 + H^4}, \frac{54}{54 + 24H^2 + H^4}, \frac{1}{54 + 24H^2 + H^4}, -3 \frac{-18 + H^2}{54 + 24H^2 + H^4} \right]$$

which shows that linear ODEs can be solved symbolically. For more details see reference 13.

For a particular value of H , these 4 equations can be solved using Maple's `fsolve` command

```
> fsolve({seq(subs(H=1.,h=1./(N+1),Eq[i]),i=0..N+1)},{seq(Y[i]=0..1.0,i=0..N+1)});
{y_0 = 1., y_1 = .7974683557, y_3 = .6455696221, y_2 = .6835443055}
```

```
> fsolve({seq(subs(H=2.,h=1./(N+1),Eq[i]),i=0..N+1)},{seq(Y[i]=0..1.0,i=0..N+1)});
{y_0 = 1., y_1 = .5421686749, y_3 = .2530120485, y_2 = .3253012050}
```

Note that same command can be used for nonlinear ODEs with non-linear boundary conditions.

Partial Differential Equations (Heat conduction in a semi-infinite domain)

```
> restart:with(linalg):
```

Consider the transient state heat conduction equation (references 2,3, 14-16)

```
> eq:=diff(u(x,t),t)-alpha*diff(u(x,t),x$2)=0;
```

$$eq := \left(\frac{\partial}{\partial t} u(x, t) \right) - \alpha \left(\frac{\partial^2}{\partial x^2} u(x, t) \right) = 0$$

where α is the thermal diffusivity ($\frac{m^2}{sec}$) and u is the dimensionless temperature. The initial and boundary conditions are

```
> u(x,0):=1;bc1:=u(0,t)=0;bc2:=u(infinity,t)=defined;
```

$$u(x, 0) := 1$$

$$bc1 := u(0, t) = 0$$

$$bc2 := u(\infty, t) = \text{defined}$$

Now the Laplace transformation package is called and the governing equation is converted into Laplace domain.

```
> with(inttrans):eqs:=laplace(eq,t,s);
```

$$eqs := s \text{laplace}(u(x, t), t, s) - 1 - \alpha \left(\frac{\partial^2}{\partial x^2} \text{laplace}(u(x, t), t, s) \right) = 0$$

Note that the initial condition is included. For brevity let's use the variable U.

```
> eqs:=subs(laplace(u(x,t),t,s)=U(x),eqs);
```

$$eqs := s U(x) - 1 - \alpha \left(\frac{\partial^2}{\partial x^2} U(x) \right) = 0$$

Next transform the boundary conditions into the Laplace domain

```
> bc1:=laplace(bc1,t,s):bc1:=subs(laplace(u(0,t),t,s)=U(0),bc1);
```

$$bc1 := U(0) = 0$$

Next, the eqs equation given above is solved by using the matrix exponential method with x as the independent variable instead of t (s is a parameter in this case).

```
> A:=matrix(2,2,[0,1,s/alpha,0]):b:=matrix(2,1,[0,-1/alpha]):y0:=matrix(2,1,[0,c2]):
```

```
> mat:=exponential(A,x):invmat:=evalm(subs(x=x-x1,evalm(mat))):mat1:=evalm(invmat&*evalm(subs(x=x1,evalm(b)))):mat2:=map(int,mat1,x1=0..x):mat3:=evalm(subs(x1=x,mat2)):
```

```
> sol:=evalm(evalm(mat*y0)+mat3);
```

$$sol := \begin{bmatrix} -\frac{1}{2} \frac{\alpha \left(-e^{\frac{\sqrt{\alpha s} x} \alpha} + e^{-\frac{\sqrt{\alpha s} x} \alpha} \right) c2 - \frac{1}{2} - 2 + e^{-\frac{\sqrt{\alpha s} x} \alpha} \left(2 \frac{\sqrt{\alpha s} x} \alpha + e^{-\frac{\sqrt{\alpha s} x} \alpha} \right)}{\sqrt{\alpha s}}}{\left(\frac{1}{2} e^{-\frac{\sqrt{\alpha s} x} \alpha} + \frac{1}{2} e^{\frac{\sqrt{\alpha s} x} \alpha} \right) c2 - \frac{1}{2} \left(-1 + e^{\frac{\sqrt{\alpha s} x} \alpha} \right) e^{-\frac{\sqrt{\alpha s} x} \alpha}} \end{bmatrix}$$

Our solution is given by the first row of sol

```
> uu:=expand(sol[1,1]);
```

$$uu := \frac{1}{2} \frac{\alpha c2 e^{\frac{\sqrt{\alpha s} x} \alpha}}{\sqrt{\alpha s}} - \frac{1}{2} \frac{\alpha c2}{\sqrt{\alpha s} e^{\frac{\sqrt{\alpha s} x} \alpha}} + \frac{1}{s} - \frac{1}{2} \frac{e^{\frac{\sqrt{\alpha s} x} \alpha}}{s} - \frac{1}{2} \frac{1}{s e^{\frac{\sqrt{\alpha s} x} \alpha}}$$

Since we are dealing with the semi-infinite domain, from the boundary condition at the infinity (bc2), the solution has to be finite when x goes to ∞. Consequently, the coefficient of $e^{\frac{\sqrt{\alpha s} x} \alpha}$ in the equation uu above must be zero. This requirement yields an expression for c2 and a simplified uu equation.

```
> eqc:=coeff(uu,exp(sqrt(s*alpha)*x/alpha));c2:=solve(eqc,c2);
```

$$eqc := \frac{1}{2} \frac{\alpha c2}{\sqrt{\alpha s}} - \frac{1}{2} \frac{1}{s}$$

$$c2 := \frac{\sqrt{\alpha s}}{\alpha s}$$

```
> uu;
```

$$-\frac{1}{s e^{\frac{\sqrt{\alpha s} x} \alpha}} + \frac{1}{s}$$

Now inverting uu back to time domain yields a solution.

```
> Ut:=invlaplace(uu,s,t);
```

$$Ut := - \left(\lim_{u1 \rightarrow 0^+} \operatorname{erf} \left(\frac{1}{2} \frac{x}{\sqrt{\alpha} \sqrt{-u1}} \right) - \operatorname{erf} \left(\frac{1}{2} \frac{x}{\sqrt{t} \sqrt{\alpha}} \right) \right) + 1$$

Maple does not give the simplified solution since we haven't specified the signs of x and α. We just have to specify that they are positive. We use dummy variables (x1 and α1) for this purpose.

APPENDIX: Solving Differential Equations with Maple

```
> uu:=subs(alpha=alpha1,x=x1,uu):assume(x1>0,alpha1>0);Ut:=invlaplace(uu,s,t);
```

$$U_t := -\operatorname{erfc}\left(\frac{1}{2} \frac{x/\sim}{\sqrt{\alpha}1\sim\sqrt{t}}\right) + 1$$

Once the inverse is obtained, the dummy variables can be changed back to the original variables.

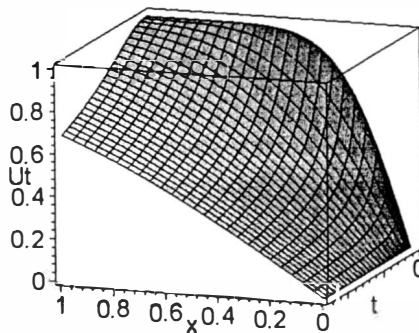
```
> Ut:=subs(x1=x,alpha1=alpha,Ut);Ut:=convert(Ut,erf);
```

$$U_t := -\operatorname{erfc}\left(\frac{1}{2} \frac{x}{\sqrt{t}\sqrt{\alpha}}\right) + 1$$

$$U_t := \operatorname{erf}\left(\frac{1}{2} \frac{x}{\sqrt{t}\sqrt{\alpha}}\right)$$

Ut is the expected solution (reference 2,3,14-16). For a given value of α , a three dimensional plot of Ut as a function of x and t can be obtained.

```
> with(plots):plot3d(subs(alpha=0.001,Ut),x=1..0,t=500..0,axes=boxed,labels=[x,t,"Ut"]);
```



Analytic method of lines for PDEs (Heat transfer in a finite medium)

```
> restart;with(linalg):
```

Consider heat transfer in a finite medium (references 14, 21, 22)

```
> ge:=diff(T(x,t),t)=alpha*diff(T(x,t),x$2):
```

The boundary and initial conditions are

```
> bc1:=H*(T0-T(0,t))=-k*Diff(T(0,t),x);bc2:=Diff(T(1,t),x)=0;IC:=T(x,0)=100;
```

$$bc1 := H(T_0 - T(0, t)) = -k \left(\frac{\partial}{\partial x} T(0, t) \right)$$

$$bc2 := \frac{\partial}{\partial x} T(1, t) = 0$$

$$IC := T(x, 0) = 100$$

The governing equation is discretized along the x-axis from i = 1 to N interior node points. N = 2 node points are used for illustration

```
> N:=2:eq[0]:=H*(T0-y[0]) = -k*(-y[2]+4*y[1]-3*y[0])/(2*h):
```

```
eq[N+1]:=(3*y[N+1]-4*y[N]+y[N-1])/(2*h)=0:
```

```
> for i from 1 to N do eq[i]:=Diff(y[i],t)= subs(diff(T(x,t),x$2) = (y[i+1]-2*y[i]+y[i-1])/(h^2),rhs(ge));od:
```

Now the boundary conditions are used to eliminate the end values

```
> y[0]:=solve(eq[0],y[0]);y[N+1]:=solve(eq[N+1],y[N+1]);
```

$$y_0 := \frac{2 H T_0 h - k y_2 + 4 k y_1}{2 H h + 3 k}$$

APPENDIX: Solving Differential Equations with Maple

```

[

$$y_3 := \frac{4}{3}y_2 - \frac{1}{3}y_1$$

> for i from 1 to N do eq[i]:=eval(eq[i]);od;

$$eq_1 := \frac{\partial}{\partial t} y_1 = \frac{\alpha \left( y_2 - 2 y_1 + \frac{2 H T_0 h - k y_2 + 4 k y_1}{2 H h + 3 k} \right)}{h^2}$$

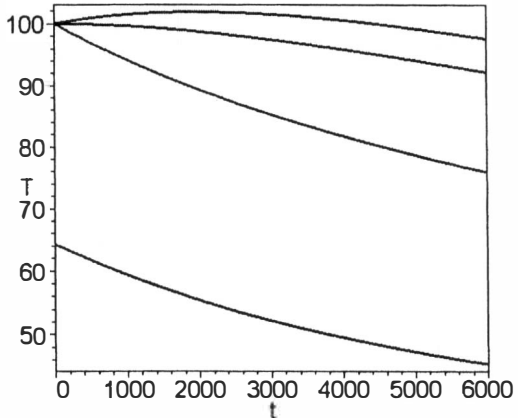

$$eq_2 := \frac{\partial}{\partial t} y_2 = \frac{\alpha \left( -\frac{2}{3} y_2 + \frac{2}{3} y_1 \right)}{h^2}$$

[ The governing equations at the interior node points ( i = 1.. N) can be expressed in matrix form (equation 1) and solved using equation 2.
[ > eqns:=[seq(rhs(eq[j]),j=1..N)]:Y:=[seq(y[i],i=1..N)]:A:=genmatrix(eqns,Y,'b1'):
[ > b:=matrix(N,1):for i from 1 to N do b[i,1]:=-b1[i];od:evalm(b):
[ Note that the matrix is not printed for saving space. The parameters are entered here.
[ > L:=1.00:h:=evalf(L/(N+1)):alpha:=2*10.^(-5):H:=25.0:k:=10.0:T0:=0:
[ > A:=map(eval,A):b:=map(eval,b):
[ The initial condition is entered now
[ > Y0:=matrix(N,1):for i from 1 to N do Y0[i,1]:=evalf(subs(x=i*h,rhs(IC)));od:
[ The solution is found out now (equation 2)
[ > mat:=exponential(A,t):invmat:=evalm(subs(t=t-t1,evalm(mat))):mat1:=evalm(invmat&
*evalm(subs(t=t1,evalm(b)))):mat2:=map(int,mat1,t1=0..t):mat3:=evalm(subs(t1=t,m
at2)):Y:=evalm(evalm(mat&*Y0)+mat3);

$$Y := \begin{bmatrix} 14.06250002 e^{(-.0003000000000 t)} + 85.93749998 e^{(-.00002571428576 t)} \\ 109.3750000 e^{(-.00002571428576 t)} - 9.37500002 e^{(-.0003000000000 t)} \end{bmatrix}$$

[ Note that we obtain the temperature at different node points as a function of time (analytical in time and numerical in x).
[ The values at the node points can be plotted.
[ > with(plots):for i from 1 to N do y[i]:=simplify(eval(Y[i,1]));od:
[ > for i from 0 to N+1 do p[i]:=plot(y[i],t=0..6000,thickness=3);od:
[ > display({seq(p[i],i=0..N+1)},axes=boxed,labels=[t,T]);

```



[By increasing N, more accurate solutoins can be obtained. N = 6 is found to be sufficient for this case.