ON MATLAB FINITE ELEMENT SIMULATIONS OF STEADY-STATE BURGER'S EQUATION

Rio Hirowati Shariffudin[1], Ithnin Abdul Jalil[2] and Hoy Kam Lai[3]
Institut Sains Matematik, Universiti Malaya,
50603 Kuala Lumpur, Malaysia
e-mel [1]riorhs@yahoo.com.
e-mel[2] isninj@yahoo.co.uk
e-mel[3] loy_101@hotmail.com

Abstract: Burger's equations have been used, time and again to model successfully many physical phenomena such as in shock waves, acoustic transmision, traffic flow, supersonic flow about aerofoils and wave propagations in thermoelastic medium. Via Cole-Hopf transformation, the equations are reduced to standard diffusion equations. With the advent of modern computer technology, numerical techniques are very much sought after. Two fundamental tenets of numerical methods are the methods of finite difference and the methods using finite elements. Both methods call for the solutions of large sparse matrices. In this paper, we shall present finite element simulations of the Burger's equations using MATLAB.

Keywords: MATLAB, finite element simulation, Burger's equation

Introduction

For problems of a fluid-flow nature where either shocks or viscous dissipation are important, Burger's equations are used to describe the phenomena. The one-dimensional Burger's equation is given by

$$u_t + uu_x - \mu u_{xx} = 0 \qquad (1.1)$$

where u is a velocity-like dependent variable and μ is a viscosity-like parameter. The subscript t indicates a time derivative and the subscript x is a spatial derivative. As a non-dimensional equation, μ is the inverse of the Reynold's number.

Suppose that

$$\phi_t = \mu\phi_{xx}$$

and

$$\phi = F(\psi).$$

then

$$\psi_t - \mu(\psi_x)^2 \cdot \frac{F''}{F'} = \mu\psi_{xx}.$$

On differentiating with respect to x and setting $u = \psi_x$, we get

$$u_t - 2\mu \cdot uu_x \cdot \frac{F''}{F'} - \mu u^3 \left\{ \frac{F'''}{F'} - \left(\frac{F''}{F'}\right)^2 \right\} = \mu u_{xx}$$

We see that the Burger's equation is a special case of

$$F(\psi) = \exp\left(-\psi / 2\mu \right). \qquad (1.2)$$

Cole-Hopf transformation is applicable to higher space dimensional Burger's equation. The 2-space dimensional Burger's equation is given by

$$u_t + uu_x + vu_y = \mu(u_{xx} + u_{yy}) \qquad (1.3a)$$

$$v_t + uv_x + vv_y = \mu(v_{xx} + v_{yy}) \qquad (1.3b)$$

Using the Cole-Hopf transformation in a similar fashion, equations (1.3a) and (1.3b) can also be reduced to the standard heat conduction equation

$$\phi_t = \phi_{xx} + \phi_{yy}$$

This familiar looking equation can be simulated using finite differencing or finite elementization from which systems of large sparse matrix equation have to be solved. These systems can be effectively solved using iterative methods.

The steady-state 2-space Dimensional Burger's Equation.

The 2-space dimension Burger's equation is as given in equations (1.3a) and (1.3b)
Taking $\psi = -2\mu \ln \phi$ in equation (1.2), we get

$$\frac{\partial u}{\partial t} = -2\mu \frac{\partial}{\partial x}\left\{ \frac{\partial}{\partial t} \ln \phi \right\} \qquad (2.1)$$

$$\frac{\partial u}{\partial x} = -2\mu \frac{\partial}{\partial x}\left(\frac{\partial}{\partial x} \ln \phi \right)$$

$$uu_x = \left\{ -2\mu \frac{\partial}{\partial x} (\ln \phi) \right\}\left\{ -2\mu \frac{\partial}{\partial x}\left(\frac{\partial}{\partial x} \ln \phi \right) \right\}$$

$$= 4\mu^2 \frac{\partial}{\partial x} \frac{1}{2} \left\{ \frac{\partial}{\partial x} \ln \phi \right\}^2 \qquad (2.2)$$

$$\frac{\partial u}{\partial y} = -2\mu \frac{\partial}{\partial x}\left\{ \frac{\partial}{\partial y} \ln \phi \right\}$$

$$vu_y = \left\{ -2\mu \frac{\partial}{\partial y} (\ln \phi) \right\}\left\{ -2\mu \frac{\partial}{\partial x}\left(\frac{\partial}{\partial y} \ln \phi \right) \right\}$$

$$= 4\mu^2 \frac{\partial}{\partial x} \frac{1}{2} \left\{ \frac{\partial}{\partial y} \ln \phi \right\}^2 \qquad (2.3)$$

$$\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x} \right) = -2\mu \frac{\partial}{\partial x}\left(\frac{\partial^2 \ln \phi}{\partial x^2} \right) \qquad (2.4)$$

$$\frac{\partial^2 y}{\partial y^2}\left(\frac{\partial u}{\partial x} \right) = -2v \frac{\partial}{\partial x}\left(\frac{\partial^2 \ln \phi}{\partial y^2} \right) \qquad (2.5)$$

whence equation (1.3a) becomes

$$-2\mu\frac{\partial}{\partial t}\ln\phi+4\mu^2\frac{1}{2}\left(\frac{\partial}{\partial x}\ln\phi\right)^2+4\mu^2\frac{1}{2}\left(\frac{\partial}{\partial y}\ln\phi\right)^2$$

$$=-2v^2\frac{\partial^2}{\partial x^2}\ln\phi-2v^2\frac{\partial^2}{\partial y^2}\ln\phi \qquad (2.6)$$

$$\frac{\partial v}{\partial t}=-2\mu\frac{\partial}{\partial y}\left(\frac{\partial}{\partial t}\ln\phi\right) \qquad (2.7)$$

$$\frac{\partial v}{\partial x}=-2\mu\frac{\partial}{\partial y}\left(\frac{\partial}{\partial x}\ln\phi\right)$$

$$u\frac{\partial v}{\partial x}=\left\{-2\mu\frac{\partial}{\partial x}\ln\phi\right\}\left\{-2\mu\frac{\partial}{\partial y}\left(\frac{\partial}{\partial x}\ln\phi\right)\right\}$$

$$=4\mu^2\frac{\partial}{\partial y}\left\{\frac{1}{2}\left(\frac{\partial}{\partial x}\ln\phi\right)\right\} \qquad (2.8)$$

$$v\frac{\partial v}{\partial y}=\left\{-2\mu\frac{\partial}{\partial y}(\ln\phi)\right\}\left\{-2\mu\frac{\partial}{\partial y}\left(\frac{\partial}{\partial y}\ln\phi\right)\right\}$$

$$=4\mu^2\frac{\partial}{\partial y}\left\{\left(\frac{1}{2}\frac{\partial}{\partial y}\ln\phi\right)^2\right\} \qquad (2.9)$$

$$\frac{\partial^2 v}{\partial x^2}=-2\mu\frac{\partial}{\partial y}\left(\frac{\partial^2}{\partial x^2}\ln\phi\right) \qquad (2.10)$$

$$\frac{\partial^2 v}{\partial y^2}=-2\mu\frac{\partial}{\partial y}\left(\frac{\partial^2\ln\phi}{\partial y^2}\right) \qquad (2.11)$$

whence the equation (1.3b) becomes

$$-2\mu\frac{\partial}{\partial t}\ln\phi+4\mu^2\frac{1}{2}\left(\frac{\partial}{\partial x}\ln\phi\right)^2+4\mu^2\frac{1}{2}\left(\frac{\partial}{\partial y}\ln\phi\right)^2$$

$$=-2\mu^2\frac{\partial^2}{\partial x^2}\ln\phi-2\mu^2\frac{\partial^2}{\partial y^2}\ln\phi \qquad (2.12)$$

Finite Element Analog

For the steady state Burger's equation, we have

$$\phi_{xx}+\phi_{yy}=0 \qquad (3.1)$$

Equation (3.1) can be solved numerically by either finite difference modelling or using finite elements. In the method, we minimize a quadratic functional and assume an approximate solution to equation (3.1) in the form of a finite series

From the one –dimensional second-order differential equation,

$$-\frac{d}{dx}(a(x)\frac{du}{dx})-f(x)=0 \quad \text{for } x_A<x<x_B \qquad (3.2)$$

we construct the variational formulation of equation (3.2) over the entire domain. We have, over an element

$$\int_{x_e}^{x_{e+1}} \left(a(x)\frac{dv}{dx}\frac{du}{dx} - vf(x) \right) dx \; - v(x_e)P_1^{(e)} - v(x_{e+1})P_2^{(e)} \qquad (3.3)$$

where

$$P_1^{(e)} = [-a(x)\frac{du}{dx}]_{x=x_e} \qquad P_2^{(e)} = [a(x)\frac{du}{dx}]_{x=x_{e+1}}$$

and

$$K_{ij}^{(e)} = \int_{x_e}^{x_{e+1}} a(x)\frac{d\psi_i^{(e)}}{dx}\frac{d\psi_j^{(e)}}{dx}dx$$

$$F_i^{(e)} = \int_{x_e}^{x_{e+1}} a(x)\psi_i^{(e)}dx + P_i^{(e)}$$

Using $\psi_1^{(e)} = \dfrac{x_{e+1} - x}{x_{e+1} - x_e}$ and $\psi_2^{(e)} = \dfrac{x - x_e}{x_{e+1} - x_e}$ $\quad x_e \le x \le x_{e+1}$, we get

$$\frac{d\psi_i^e}{\text{dx}} = \frac{-1}{h_e} \qquad \frac{d\psi_2^e}{\text{dx}} = \frac{1}{h_e}$$

$$K_{11}^{(e)} = \int_{x_e}^{x_{e+1}} a(x)\left(\frac{-1}{h_e}\right)^2 dx$$

$$K_{12}^{(e)} = \int_{x_e}^{x_{e+1}} a(x)\left(\frac{-1}{h_e}\right)\frac{1}{h_e} dx$$

$$K_{22}^{(e)} = \int_{x_e}^{x_{e+1}} a(x)\left(\frac{-1}{h_e}\right)^2 dx$$

$$F_1^{(e)} = \int_{x_e}^{x_{e+1}} f(x)\left(\frac{x_{e+1} - x}{h_e}\right)dx \; + P_1^{(e)}$$

$$F_2^{(e)} = \int_{x_e}^{x_{e+1}} a(x)\left(\frac{x - x_e}{h_e}\right)dx \; + P_2^{(e)}$$

Taking into account of the boundary conditions which are usually known or specified U_i and P_i, we have finally to solve an algebraic linear system of equations. The assembled matrix equation is given by

$$[K]\{U\} = \{F\}$$

For the two space dimensions problems , we subdivide the domain of definition equation (3.1) into triangles with no edges on the boundaries, triangles with at least on edge on the Neumann boundary and triangles with neither of the first two properties. The solution to equation (3.1) is approximated by a combination of a finite number of functions called the basis or trial functions. For the complete problem, suppose that the two dimensional partial differential equation is given by

$$\frac{\partial}{\partial x}(p(x,y)\frac{\partial u}{\partial x}) + \frac{\partial}{\partial y}(q(x,y)\frac{\partial u}{\partial y}) + r(x,y)u(x,y) = f(x,y) \qquad (x,y) \in D$$

with the boundary conditions

$$u(x,y) = g(x,y) \qquad \text{on } S_1$$

$$p(x,y)\frac{\partial u}{\partial x}\cos\theta_1 + q(x,y)\frac{\partial u}{\partial x}\cos\theta_2 + g_1(x,y)u(x,y) = g_2(x,y) \qquad \text{on } S_2$$

Note that the cosines of the angles have been taken into consideration when considering the normal derivatives.

Just as in the one dimensional problem, we shall minimize the functional

$$I[\phi] = \iint_D [\frac{1}{2}\{p(x,y)[\sum_i^m \gamma_i \frac{\partial \phi_i}{\partial x_i}]^2 + q(x,y)[\sum_1^m \gamma_i \frac{\partial \phi_i}{\partial x_i}]^2 - r(x,y)[\sum_1^m \gamma_i \phi_i(x,y)]^2$$

$$+ f(x,y)\sum_1^m \gamma_i \phi_i(x,y)]dydx + \int_{S_2} \{-g_2(x,y)\sum_1^m \gamma_i \phi_i(x,y)] + \frac{1}{2} g_1(x,y)[\sum_1^m \gamma_i \phi_i(x,y)]^2\} dS$$

where

$$\phi(x,y) = \sum_1^m \gamma_i \phi_i(x,y)$$

and $\phi_1, \phi_2, \ldots\ldots, \phi_m$ are piecewise polynomials.
For a minimum, we will get

$$\alpha_{ij} = \iint_D [p(x,y)\frac{\partial \phi_i}{\partial x}\frac{\partial \phi_j}{\partial x} + q(x,y)\frac{\partial \phi_i}{\partial y}\frac{\partial \phi_j}{\partial y} - r(x,y)\phi_j]dxdy + \int_{S_2} g_1(x,y)\phi_i \phi_j dS$$

$$\beta_i = -\iint_D f(x,y)\phi_i dxdy + \int_{S_2} g_2 \phi_i dS - \sum_{n+1}^m \alpha_{ik}\gamma_k$$

With these data, we solve iteratively (or directly) the matrix equation $A\mathbf{c} = \mathbf{b}$ where $A = (\alpha_{ij}), \mathbf{b} = (\beta_1, \beta_2, \ldots\ldots, \beta_n)^T$ and $\mathbf{c} = (\gamma_1, \gamma_2, \ldots\ldots, \gamma_n)^T$. Note that we work with double integrals over triangles. We also need to compute all line integrals. Finally, we assemble the integrals over each triangle into a linear system.

Finite Element Simulations For Steady-State Burger's equation

Our reports constitute simulation work for the steady-state one dimensional as well as the two dimensional Burger's equation. This work is being carried out using MATLAB. Simulations for the one dimensional problem are summarized in table 1 while those involving two dimensional problems are given in table 2. In table 2, we present the matrix involving the interior points. We also present the basis functions. We give a graphical form of the solution to one-dimensional problem in figure 1. Note that the problems given have been added some spices.

Discussions and Conclusions

We have attempted in essence to solve partial differential equations of the type

$$\frac{\partial}{\partial x}(p(x,y)\frac{\partial u}{\partial x}) + \frac{\partial}{\partial y}(q(x,y)\frac{\partial u}{\partial y}) + r(x,y)u(x,y) = f(x,y) \qquad (x,y)\in D$$

subject to

$$u(x,y) = g(x,y) \ \text{ on } S_1$$

$$p(x,y)\frac{\partial u}{\partial x}\cos\theta_1 + q(x,y)\frac{\partial u}{\partial y}\cos\theta_2 + g_1(x,y)u(x,y) = g_2(x,y) \text{ on } S_2$$

by taking an approximation to the solutions of the form

$$\phi(x,y) = \sum_1^m \gamma_i \phi_i(x,y)$$

where $\phi_1, \phi_2, \ldots\ldots, \phi_m$ are piecewise polynomials. In the spirit of the technique given in Rio [1], we shall present the results which we have obtained. Table 3 gives the matrix obtained involving the interior nodes while the basis functions are given in table 4. Note that the matrix involved is of the form which warrants to be accelerated by the conjugate gradient method. This work is totally new and provides an avenue for further pursuant. The algebraic structure we obtain is an insight into accelerating the simulations using the conjugate gradient method. We observed that the solution sets can be divided into two independent subsets, thus providing natural environment for parallel computing. Where parallel computing is concerned, we discover that we can leverage on the number of processors. Note that the concept of leveraging which has been applied in the brilliant compensation for network marketing is perceived to be relevant.

References

[1] Rio Hirowati Shariffudin & Abdul Rahman Abdullah, " Hamiltonian Circuited Simulations of Elliptic Partial Differential Equatbz lah jangan kacauions Using A Spark", *Applied Mathematics Letters 14(2001)*, pp 413-418

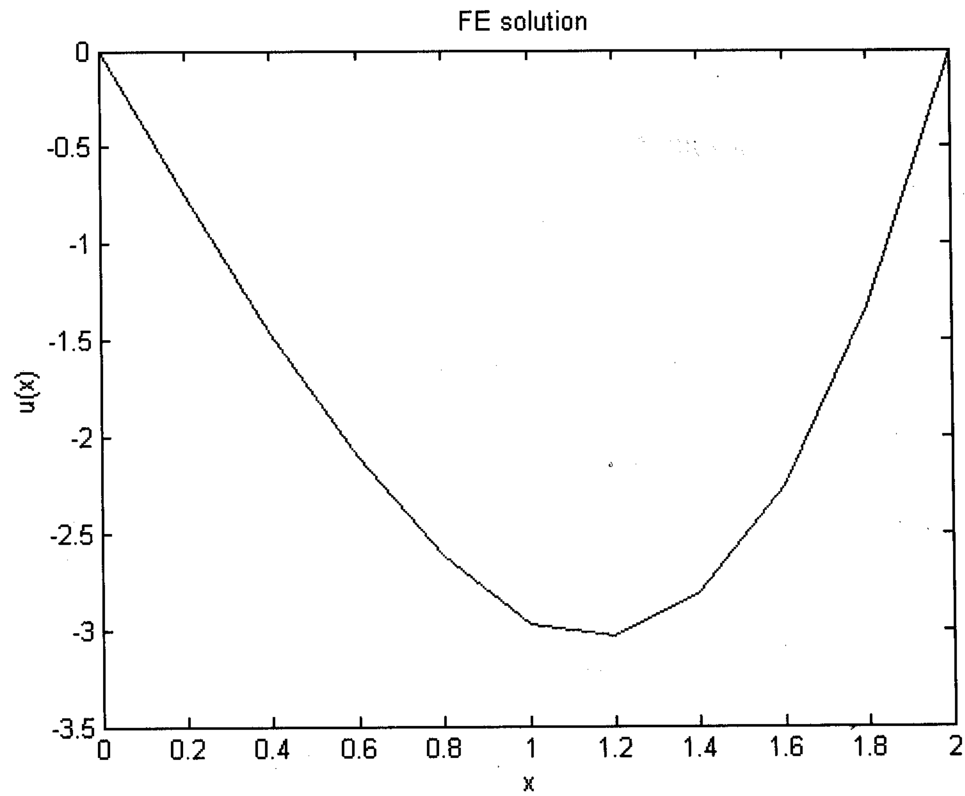

Figure 1 Finite Element Solutions For One dimensional Problem

Table 1 Simulations For 2-Dimensional Problem(non-Hamilton)

$$\begin{bmatrix} 410 & -407 & & -1 & & & & & \\ -407 & 470 & & -1 & & -1 & & & \\ & -1 & 4 & & & -1 & & & \\ -1 & & & 4 & -1. & & -1 & & \\ & -1 & & -1. & 4. & -1 & & -1. & \\ & & -1. & & -1 & 4 & & & -1 \\ & & & -1 & & & 4. & -1 & \\ & & & & -1 & & -1 & 4 & -1 \\ & & & & & -1 & & -1 & 4 \end{bmatrix}$$

Table 2 Some Basis Functions(non-Hamilton)

$$\phi_1(x,y) = 1.00234869(-1+10x+10y)+(1-10y)+(-1+10x)$$
$$\phi_2(x,y) = 1.00234869(30-10x-280y)+1.002325424(-30+10x+290y)+(1-10y)$$
$$\phi_3(x,y) = 1.00232524(3-10x)+(1-10y)+1.02560438(-3+10x+10y)$$
$$\phi_4(x,y) = 1.00234869(2-10x)+1.016591254(-2+10x+10y)+(1-10x)$$
$$\phi_5(x,y) = 1.00234869(3-10x-10y)+1.00232524(-1+10x)$$
$$\phi_6(x,y) = 1.00232524(2-10y)+1.01659125(2-10x)+1.034607645(-3+10x+10y)$$
$$\phi_7(x,y) = 1.00232524(4-10x-10y)+1.02560438(-2+10x)+1.034607645(-1+10y)$$
$$\phi_8(x,y) = 1.02560438(2-10y)+1.05928151(-4+10x+10y)+1.034607645(3-10x)$$
$$\phi_9(x,y) = 1.02560438(5-10x-10y)+1.059281516(-1+10y)+1.04081077(-3+10x)$$

Table 3 Simulations For 2-Dimensional Problem(Hamilton)

$$\begin{bmatrix} 4 & & & & & & -1 & & -1 \\ & 4 & & & & & & -1 & -1 \\ & & 4 & & & -1 & -1 & -1 & -1 \\ & & & 4 & & -1 & -1 & & \\ & & & & 4 & -1 & & -1 & \\ & & -1 & -1 & -1 & 4 & & & \\ -1 & & -1 & -1 & & & 4 & & \\ & -1 & -1 & & -1 & & & 4 & \\ -1 & -1 & -1 & & & & & & 4 \end{bmatrix}$$

Tabre 4 Some Basis Functions(Hamilton)

$$\phi_1(x,y) = 1.03113280(2-10x) + 1.041860223(3-10y) + 1.062865299(-4+10x+10y)$$

$$\phi_2(x,y) = 1.095181072\ (-5+10x+10y) + 1.062865298(3-10y) + 1.062865299(3-10x)$$

$$\phi_3(x,y) = 1.041860223\ (-3+10x+10y) + 1.020855156(2-10y) + 1.020855157(2-10x)$$

$$\phi_4(x,y) = 1.041860223(3-10x) + 1.031132805(2-10y) + 1.062865298(-4+10x+10y)$$

$$\phi_5(x,y) = 1.031132801(-2+10y) + 1.041860223\ (-1+10x) + 1.020855157(4\text{-}10x\text{-}10y)$$

$$\phi_6(x,y) = 1.041860223(5-10x-10y) + 1.062865298(-2+10x) + 1.062865299(-2+10y)$$

$$\phi_7(x,y) = 1.010427574(3-10x-10y) + 1.020855156(-1+10x) + 1.020855157(-1+10y)$$

$$\phi_8(x,y) = 1.041860223(-1+10y) + 1.031132805(-2+10x) + 1.020855156(4-10x-10y)$$

$$\phi_9(x,y) = 1.031132801(10x) + (-3+10y) + (4\text{ - }10x\text{ - }10y)$$