Universitat Politècnica de Catalunya
Master of Science in Computational Mechanics
Finite Element in Fluids
FEF
Spring Semester 2017/2018

MatLab Session 3
2D Unsteady Advection equation

Luan Malikoski Vieira

April 2, 2018

1 Introduction

This report will cover the 2D *usteady state advection reaction* equation, seen in (1), solution with WRM using Galerkin approach for *FEM* spatial discretization and time discretization uisng different methods based on Cartesian coordinates. The first part will briefly explain the code changes in order to implement each integration method: Fourth order Taylor Galerkin(TG4) Two steps fourth order Taylor Galerkin (TG4-2s).

$$u_t + a \cdot \nabla u = s \tag{1}$$

2 Changes in MatLab routines

The implemented changes in MatLab routines are explained in the following subsections.

2.1 Implementation of TG4

The TG4 time integration method starts from:

$$\frac{\Delta u}{\Delta t} = \frac{1}{2}\left(u_t^{n+1} + u_t^n\right) - \frac{\Delta t}{12}\left(u_{tt}^{n+1} - u_{tt}^n\right) = \frac{1}{2}\Delta u_t - \frac{\Delta t}{12}\Delta u_{tt} + u_t^n \tag{2}$$

Where, from governing equation (1), $u_t = s - \mathbf{a} \cdot \nabla u$ and $u_{tt} = s_t - \mathbf{a} \cdot \nabla s + (\mathbf{a} \cdot \nabla)^2 u$. Plugging those results in equation (2) we obtain:

$$\frac{\Delta u}{\Delta t} = \frac{1}{2}\left(\Delta s - \mathbf{a} \cdot \nabla \Delta u\right) - \frac{\Delta t}{12}\left(\Delta s_t - \mathbf{a} \cdot \nabla \Delta s + (\mathbf{a} \cdot \nabla)^2 \Delta u\right) + s^n - \mathbf{a} \cdot \nabla u^n \tag{3}$$

If the source term s is such that $s(x, y, t) = s(x, y)$, we have $s_t = \frac{ds}{dt} = 0$ and $\Delta s = s^{n+1} - s^n = 0$. Thus, after rearranging, equation (3) becomes:

$$\frac{\Delta u}{\Delta t} + \frac{1}{2}\mathbf{a} \cdot \nabla \Delta u + \frac{\Delta t}{12}(\mathbf{a} \cdot \nabla)^2 \Delta u = -\mathbf{a} \cdot \nabla u^n + s^n \tag{4}$$

After time discretization, WRM using Galerkin approach is employed for FEM spatial discretion. For the code implementation purposes, first and second order derivatives terms in u are integrated by parts after applying the weight function w. This leads to the final time-space discretized form of the governing equation, for case where homogeneous Dirichlet B.C. are imposed at inlet boundary (Γ_{in}).

The left side of equation (4) reads:

$$(w, \Delta u) + \frac{\Delta t}{2}\left[(\mathbf{a}\cdot\mathbf{n}w, \Delta u)_{\Gamma_{out}} - (\mathbf{a}\cdot\nabla w, \Delta u)\right] + \frac{\Delta t^2}{12}\left[(\mathbf{a}\cdot\mathbf{n}w, \mathbf{a}\cdot\nabla\Delta u)_{\Gamma_{out}} - (\mathbf{a}\cdot\nabla w, \mathbf{a}\cdot\nabla\Delta u)\right] \tag{5}$$

The right side of equation (4) reads:

$$-\Delta t\left[(\mathbf{a}\cdot\mathbf{n}w, u^n)_{\Gamma_{out}} - (\mathbf{a}\cdot\nabla w, u^n)\right] + \Delta t(w, s^n) \tag{6}$$

After identifying each matrix and vector which results from spatial discretization of left side, equation (5) and right side, equation (6), as per provided code notation we obtain:

$$\left[\mathbf{M} + \frac{\Delta t}{2}(\mathbf{M_o} - \mathbf{C}) + \frac{\Delta t^2}{12}(\mathbf{C_o} - \mathbf{K})\right]\Delta u = -\Delta t(\mathbf{M_o} - \mathbf{C})u^n + \Delta t\mathbf{V_1} \tag{7}$$

Writing the system of equations as $\mathbf{A}\Delta u = \mathbf{B}u^n + \mathbf{f}$. The matrices $\mathbf{A}$, $\mathbf{B}$ and vector $\mathbf{f}$ are identified. Consequently, the code implementation becomes

```
elseif meth == 9 %TG4
    A = M+(dt/2)*(Mo-C)+(dt^2/12)*(Co-K);
    B = -dt*(Mo-C);
    f = dt*v1;
```

2.2 Implementation of TG4-2s

The implementation of the TG4-2s starts from the following set of equations:

$$\tilde{u} = u^n + \frac{1}{3}\Delta t u_t^n + \frac{1}{12}\Delta t^2 u_{tt}^n \tag{8}$$

$$u^{n+1} = u^n + \Delta t u_t^n + \frac{1}{2}\Delta t^2 \tilde{u}_{tt}^n \tag{9}$$

Where the only noticed difference from the implementation of the TG3-2s is the α coefficient which changes form 1/9 to 1/12. Consequently the code implementation becomes:

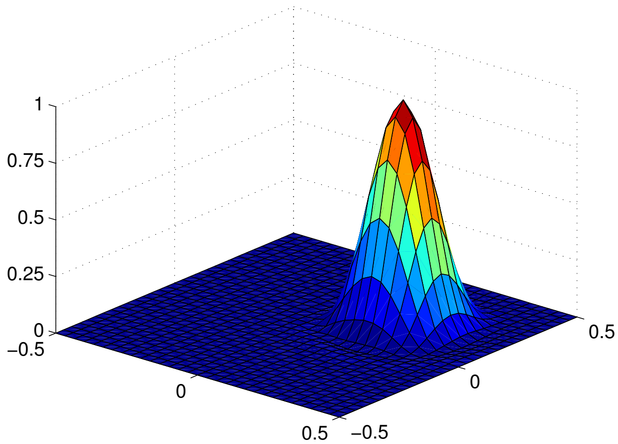
```
elseif meth == 8 %TG4-2s
    alpha = 1/12;
    A1 = M;
    B1 = -(dt/3)*C'- alpha*dt^2*(K - Co);
    f1 = (dt/3)*v1 + alpha*dt^2*(v2 - vo);
    A2 = M;
    B2 = -dt*C';
    C2 = - (dt^2/2)*(K-Co);
    f2 = dt*v1 + (dt^2/2)*(v2 - vo);
```

3 Results

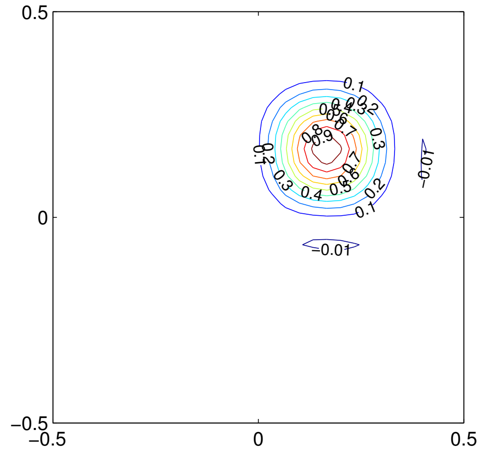
In order to asses those methods the Cosine Hill problem will be run with the following conditions: 30x30 Bilinear elements mesh. Total time $T = 2*\pi$ for a complete revolution. The number of time-steps $n = 200$. The advection velocity field is $\mathbf{a} = (-y, x)$.

In order to have a rough idea about the accuracy of those methods (TG4 and TG4-2s), the maximum and minimum amplitude of the solution at the last time step is compared with the same values for the TG3 solution.

Figure (1) shows the results for the TG3 method.



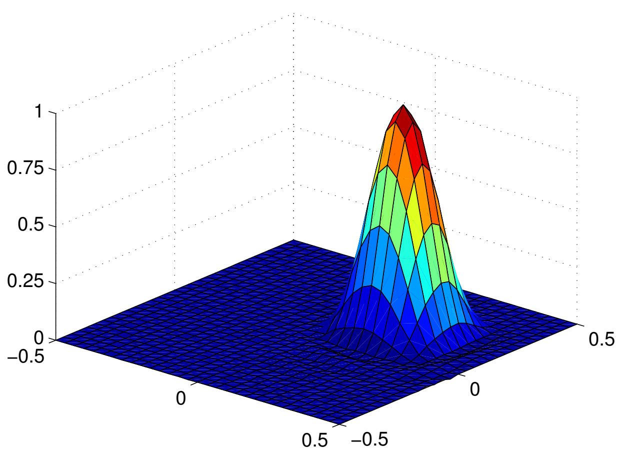
(a)



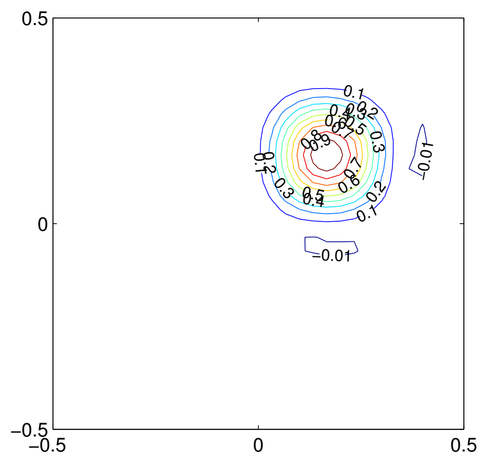
(b)

Figure 1: TG3 - Solution at $t = 2\pi$:(a) 3D view;(b) Chart hill

Figure (2) shows the results for the TG4 method.



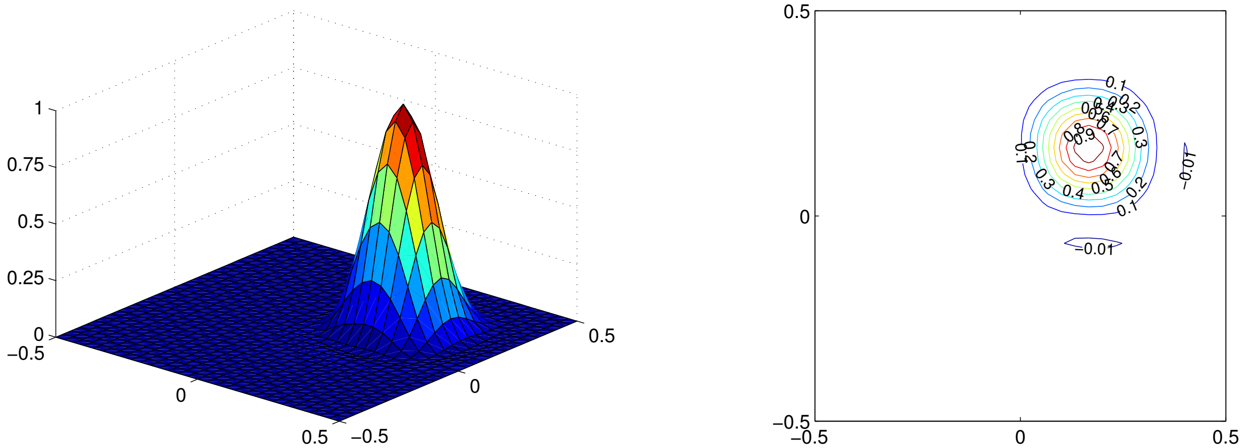
(a)



(b)

Figure 2: TG4 - Solution at $t = 2\pi$:(a) 3D view;(b) Chart hill

Figure (3) shows the results for the TG4-2s method.



(a) (b)

Figure 3: TG4-2s - Solution at $t = 2\pi$:(a) 3D view;(b) Chart hill

It can be seen in table (1) that the TG4 method presents a higher amplitude, closet than 1 which is the initial maximum amplitude. This gives an idea about how accurate and dissipative the method is. The TG3 and TG4-2s behaved similarly. Probably the Courant number C for this test was such that the TG4-2s losses its accuracy and becomes more dissipative, given the maximum amplitude being lower.

Method	**Maximum**	**Maximum**
TG3	9.835E-1	-1.484E-2
TG4	9.924E-1	-1.728E-2
TG4-2s	9.825E-1	-1.497E-2

Table 1: Amplitudes at last time step