Universitat Polytechnica de Catalunya
MSc Computational Mechanics
Spring 2018

Finite Element in Fluids

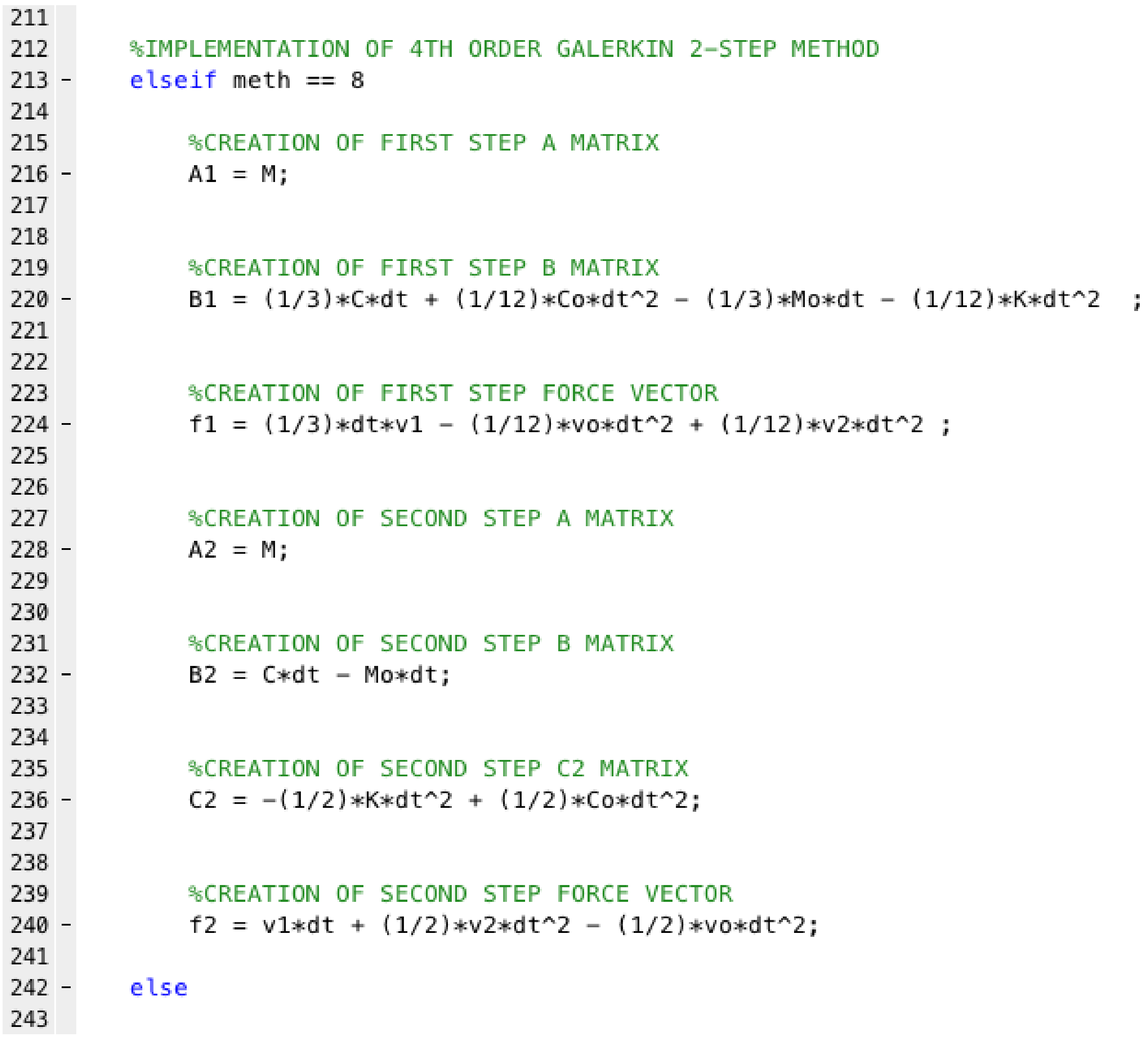
FEFmatlab4

Due 02/04/2018

Alexander Keiser

Implementation of Higher Order Method

In the first part of this assignment, we will implement a fourth order two step method to solve the classic 2D unsteady flow "Convection of a cosine hill in a pure rotation velocity field" problem. The relevant modified code can be seen below with commentary and a time-space discretization of the method from the literature can be seen on the following page.



```
    %IMPLEMENTATION OF 4TH ORDER GALERKIN 2-STEP METHOD
    elseif meth == 8

        %CREATION OF FIRST STEP A MATRIX
        A1 = M;


        %CREATION OF FIRST STEP B MATRIX
        B1 = (1/3)*C*dt + (1/12)*Co*dt^2 - (1/3)*Mo*dt - (1/12)*K*dt^2  ;


        %CREATION OF FIRST STEP FORCE VECTOR
        f1 = (1/3)*dt*v1 - (1/12)*vo*dt^2 + (1/12)*v2*dt^2 ;


        %CREATION OF SECOND STEP A MATRIX
        A2 = M;


        %CREATION OF SECOND STEP B MATRIX
        B2 = C*dt - Mo*dt;


        %CREATION OF SECOND STEP C2 MATRIX
        C2 = -(1/2)*K*dt^2 + (1/2)*Co*dt^2;


        %CREATION OF SECOND STEP FORCE VECTOR
        f2 = v1*dt + (1/2)*v2*dt^2 - (1/2)*vo*dt^2;

    else

```

Figure 1: Code implementation of the two-step fourth order method found in the literature

Above we can see the implementation of the two step fourth order method found in the literature. This method was implemented similarly to the 3rd order method already given in the code. First, matricies A1, B1, and vector f1 are implemented from the time-space descretization, followed by their second step counterparts with the addition of the C2 matrix. All of these can be found at the bottom of the following page, and some sample results of the method can be found in figure 3 at the top of page 4.

Time - Space Discretization of 2-Step fourth order method

Starting with the 4th order taylor expansion

$$u(t^{n+1}) = u(t^n) + \Delta t\, u_t^n(t^n) + \frac{\Delta t^2}{2} u_{tt}^n(t^n) + \frac{\Delta t^3}{6} u_{ttt}^n(t^n) + \frac{\Delta t^4}{24} u_{tttt}^n(t^n) + \cancel{O(\Delta t^5)}^{\to 0}$$

and we can say

$$u(t^{n+1}) = 1 + \Delta t \frac{\partial}{\partial t} + \frac{\Delta t^2}{2}\frac{\partial^2}{\partial t^2} + \frac{\Delta t^3}{6}\frac{\partial^3}{\partial t^3} + \frac{\Delta t^4}{24}\frac{\partial^4}{\partial t^4}$$

$$u(t^{n+1}) = 1 + \Delta t \frac{\partial}{\partial t} + \frac{\Delta t^2}{2}\frac{\partial^2}{\partial t^2}\underbrace{\left(1 + \frac{\Delta t}{3}\frac{\partial}{\partial t} + \frac{\Delta t}{12}\frac{\partial^2}{\partial t^2}\right)}_{\tilde{U}}$$

splitting into 2-steps

$$\left.\begin{aligned} \tilde{U} &= U^n + \frac{1}{3}\Delta t U_t^n + \frac{1}{12}\Delta t^2 U_{tt}^n \\ U^{n+1} &= U^n + \Delta t U_t^n + \frac{1}{2}\Delta t^2 \tilde{U}_{tt}^n \end{aligned}\right\} \text{time discretization}$$

convection-diffusion strong form

$$\boxed{U_t + a\nabla U - \nabla(\nu \nabla U) + \sigma U = S}$$

$$U_t^n = S^n - a\cdot\nabla U^n$$

$$U_{tt}^n = \cancel{S_t}^{\to 0} - a\cdot\nabla S^n + (a\cdot\nabla)^2 U^n$$

first step space discretization...

$$\tilde{U}^n = U^n + \frac{1}{3}\Delta t\left[S^n - a\cdot\nabla U^n\right] + \frac{1}{12}\Delta t^2\left[-a\cdot\nabla S^n + (a\cdot\nabla)^2 U^n\right]$$

applying galerkins...

$$(w,\tilde{U}^n) = (w,U^n) + \frac{1}{3}\Delta t\left(w, S^n - a\cdot\nabla U^n\right) + \frac{1}{12}\Delta t^2\left(w, -a\cdot\nabla S^n + (a\cdot\nabla)^2 U^n\right)$$

integrating by parts gives us...

$$(w,\tilde{U}^n) = (w,U^n) + \frac{1}{3}\Delta t(w,S^n) - \frac{1}{3}\Delta t\, a\left(wU^n\big|_{\Gamma_{out}} - U^n\nabla w\right) - \frac{1}{12}\Delta t^2 a\left(wS^n\big|_{\Gamma_{out}} - S^n\nabla w\right)\ldots$$

$$\ldots + \frac{1}{12}\Delta t^2 a^2\left(w\nabla U^n\big|_{\Gamma_{out}} - \nabla w\nabla U^n\right)$$

and rearranging returns...

$$(w,\Delta\tilde{U}^n) = \frac{1}{3}\Delta t(w,S^n) + \frac{1}{3}\Delta t\, a U^n\nabla w + \frac{1}{12}\Delta t^2 a S^n\nabla w - \frac{1}{12}\Delta t^2 a^2\nabla w\nabla U^n\ldots$$

$$\ldots - \left[\frac{1}{3}\Delta t\, a w U^n + \frac{1}{12}\Delta t^2 a w S^n - \frac{1}{12}\Delta t^2 a^2 w\nabla U^n\right]_{\Gamma_{out}}$$

second step space discretization

$$U^{n+1} = U^n + \Delta t\left[S^n - a\cdot\nabla U^n\right] + \frac{1}{2}\Delta t^2\left[-a\cdot\nabla S^n + (a\cdot\nabla)^2 U^n\right]$$ & applying galerkins gives us...

$$(w,U^{n+1}) = (w,U^n) + \Delta t\left(w, S^n - a\nabla U^n\right) + \frac{1}{2}\Delta t^2\left(w, -a\cdot\nabla S^n + (a\cdot\nabla)^2 U^n\right)$$ & integrating by parts...

$$(w,U^{n+1}) = (w,U^n) + \Delta t(w,S^n) - a\Delta t\left(wU^n\big|_{\Gamma_{out}} - U^n\nabla w\right) - \frac{1}{2}\Delta t^2 a\left(wS^n\big|_{\Gamma_{out}} - S^n\nabla w\right) + \frac{1}{2}\Delta t^2 a^2\left(w\nabla U^n\big|_{\Gamma_{out}} - \nabla w\nabla U^n\right)$$

and rearranging gives us..

$$(w,\Delta U) = \Delta t(w,S^n) + a\Delta t\, U^n\nabla w + \frac{1}{2}\Delta t^2 a S^n\nabla w - \frac{1}{2}\Delta t^2 a^2\nabla w\nabla U^n - \left[a\Delta t\, wU^n + \frac{1}{2}\Delta t^2 a w S^n - \frac{1}{2}\Delta t^2 a^2 w\nabla U^n\right]_{\Gamma_{out}}$$

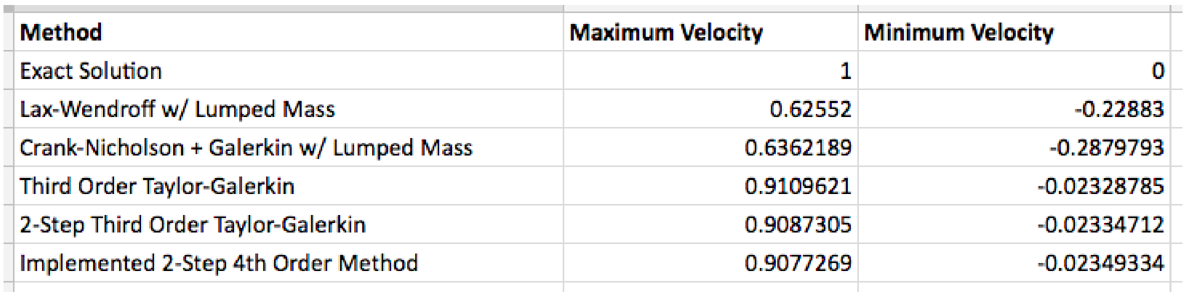
And substituting in terms of code gives us..

STEP I: $\underbrace{M}_{A1}\Delta\tilde{U}^n = \underbrace{\frac{1}{3}\Delta t C - \frac{1}{12}\Delta t^2 K - \frac{1}{3}\Delta t M_0 + \frac{1}{12}\Delta t^2 C_0}_{B1}$ $\Big\}$ $F_1 = \frac{1}{3}\Delta t V_1 + \frac{1}{12}\Delta t^2 V_2 - \frac{1}{12}\Delta t^2 V_0$

STEP II: $\underbrace{M}_{A2}\Delta U^{n+1} = \underbrace{\Delta t C - \Delta t M_0}_{B2}$ $\Big\}$ $C2 = -\frac{1}{2}\Delta t^2 K$ $\Big($ $F2 = \Delta t V_1 + \frac{1}{2}\Delta t^2 V_2 - \frac{1}{2}\Delta t^2 V_0$

Discussion of Method Behavior

After studying the theory and the problem type from the literature, it becomes apparent that a good measure of the accuracy of each method is the by measuring the maximum and minimum velocities and comparing them to the exact solutions of $U_{max} = 1.0 \quad U_{min} = 0.0$. This will be done and shown below for each method.



Method	Maximum Velocity	Minimum Velocity
Exact Solution	1	0
Lax-Wendroff w/ Lumped Mass	0.62552	-0.22883
Crank-Nicholson + Galerkin w/ Lumped Mass	0.6362189	-0.2879793
Third Order Taylor-Galerkin	0.9109621	-0.02328785
2-Step Third Order Taylor-Galerkin	0.9087305	-0.02334712
Implemented 2-Step 4th Order Method	0.9077269	-0.02349334

Figure 2a: Chart comparing the results of the 5 methods and exact solution

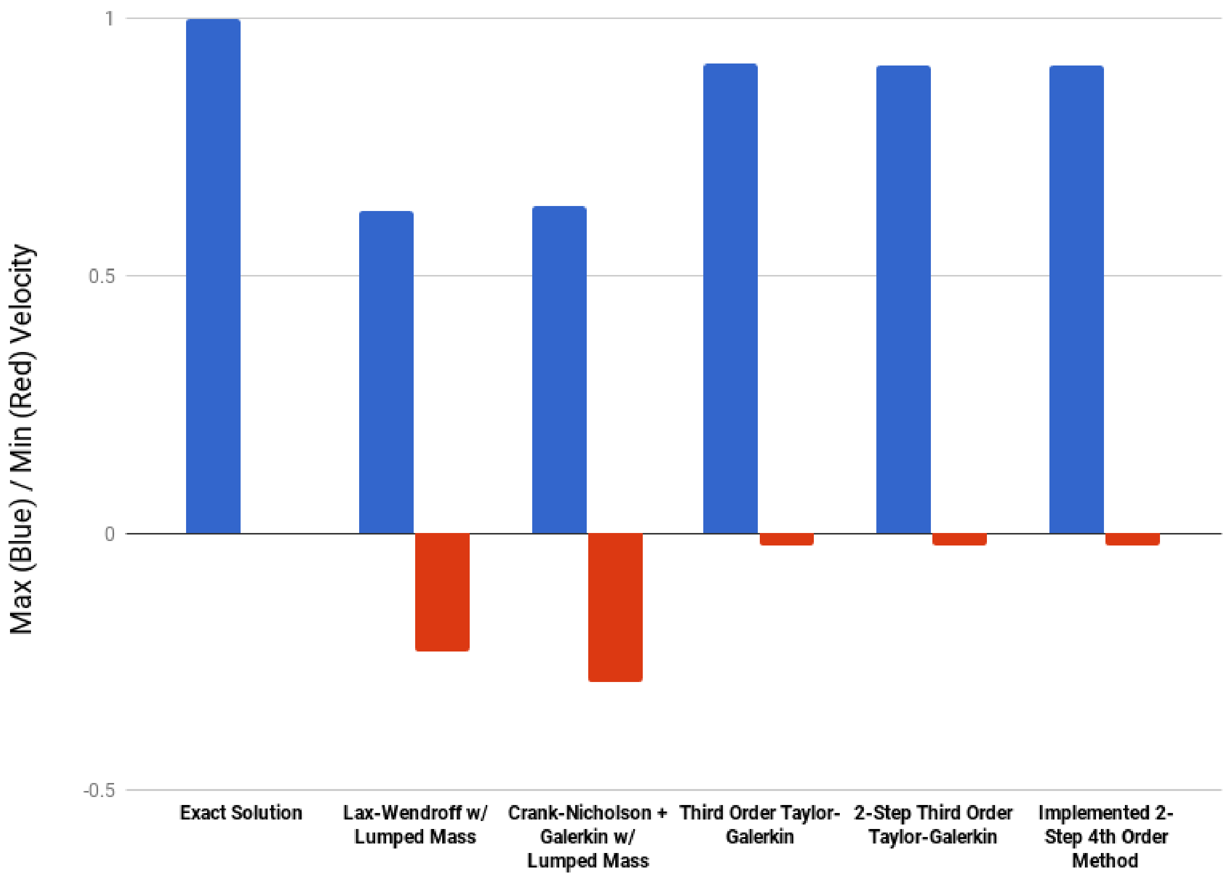


Figure 2b: Bar graph comparing the results of the 5 methods and exact solution

Above we can see the results of the varying methods performance. It becomes obvious that both the Lax-Wendroff with Lumped Mass and Crank-Nicholson with Lumped Mass are simply outperformed by the third and fourth order methods. The minimums of the third and fourth order methods are an order of magnitude closer to the exact solution than the Lax-Wendroff w/ Lumped Mass and Crank-Nicholson w/ Lumped Mass methods. And the maximums are around 30% of an order of magnitude closer to the exact solution. It is worth noting that there is no significant difference in performance between the third and implemented fourth order methods. Some sample results of this fourth order method, along with some others, can be seen on the following page.

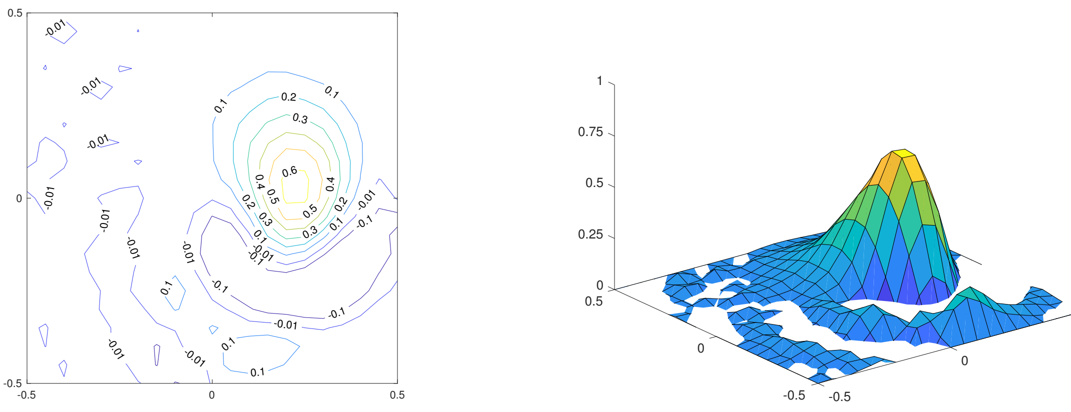


Figure 3a: Sample results of the Crank-Nicholson + Galerkin w/ Lumped Mass method

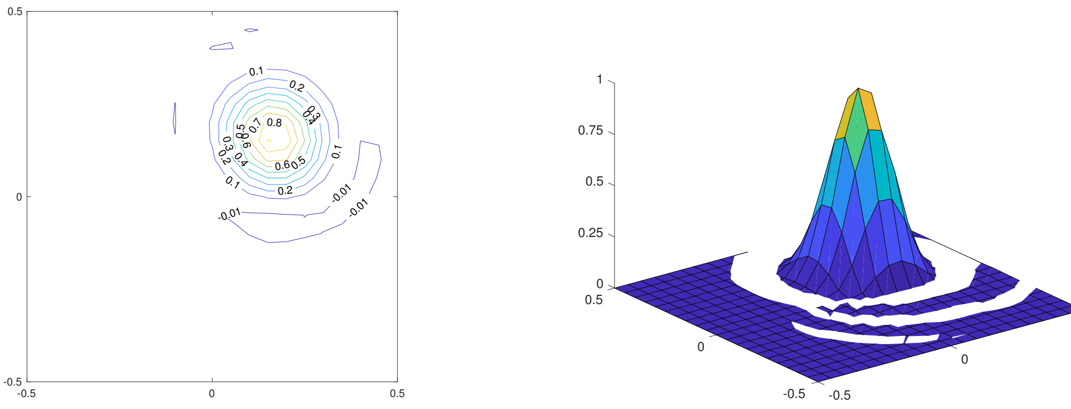


Figure 3b: Sample results of the 2-Step Third Order Taylor-Galerkin method

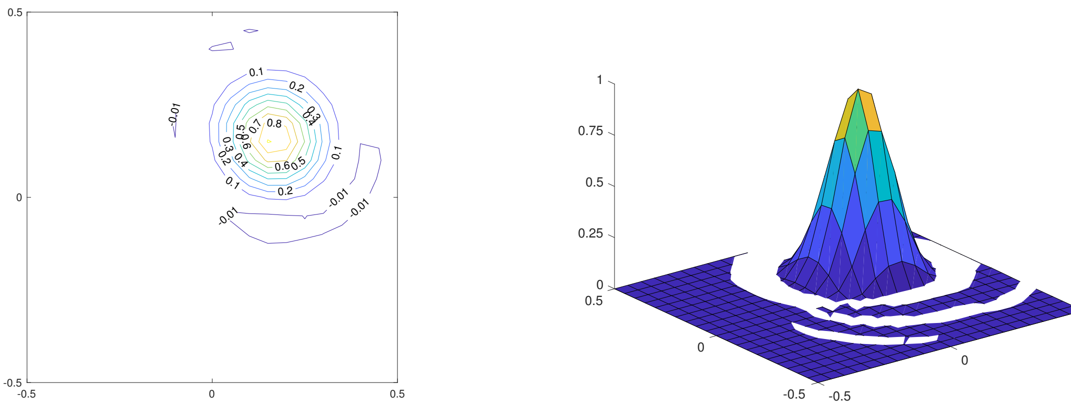


Figure 3c: Sample results of the Implemented 2-Step Fourth Order method

Above we have some visualizations of the velocity plots at the final time steps. From these figures it is also apparent that the 3rd/4th order methods are much more accurate than the second order Crank-Nicholson + Galerkin w/ Lumped Mass method. We can say this because the higher order methods exhibit much less dissipating oscillatory behavior when compared to the Crank-Nicholson method. The above graphs also reinforce the claim that there is no significant difference in performance between the third and implemented fourth order methods.