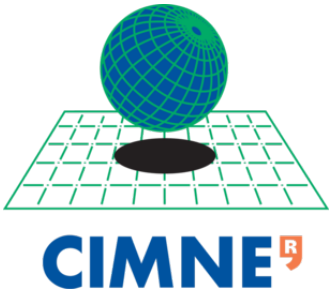
Universitat Polytechnica de Catalunya
MSc Computational Mechanics
Spring 2018

Finite Elements in Fluids

FEFmatlab5

Due 02/04/2018

Alexander Keiser



1 Propagation of a Cosine Profile

In the first part of this report we will comment on the stability and accuracy of some different implemented numerical methods to solve the problem of a classically implemented cosine profile propagation.

1.a Second-order Lax-Wendroff (TG2) Method

Here we will comment on the numerical experiments, in particular stability and accuracy, and compare with the expected results from the theory. We will first implement the Second Order Lax-Wendroff (TG2) finite element method scheme. The results can be seen below.

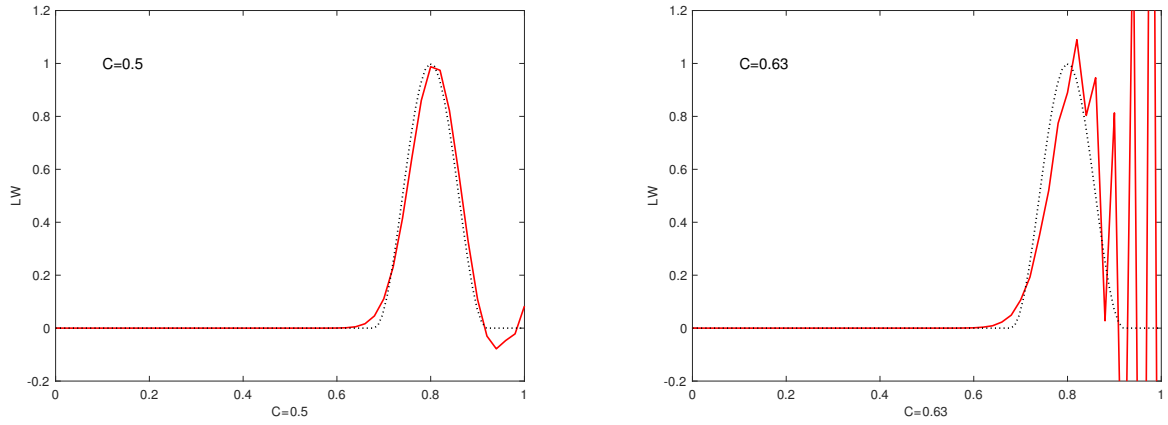


Figure 1: Propagation of Cosine profile for TG2 C=0.5 (left) C=0.63 (right)

Above we can see two different Courant number implementations. On the right hand side, we have results for an implementation with Courant number of 0.63. This value is out of the stability criterion of Second Order Lax-Wendroff (TG3) $C^2 \leq 1/3$, and therefore we expect a high degree of oscillatory behavior producing inaccurate and unstable results. The graphs confirm this phenomenon and verify from the theory. On the left hand side we see the results for a Courant number of 0.5. This is within the stability criterion of $C^2 \leq 1/3$, and as we can notice from the figure, the graph is stable with some inaccuracies. We can expect this method to be more accurate if we were to increase the time step thereby lowering the Courant number. The following graph confirms this and coincides with the theory from the literature.

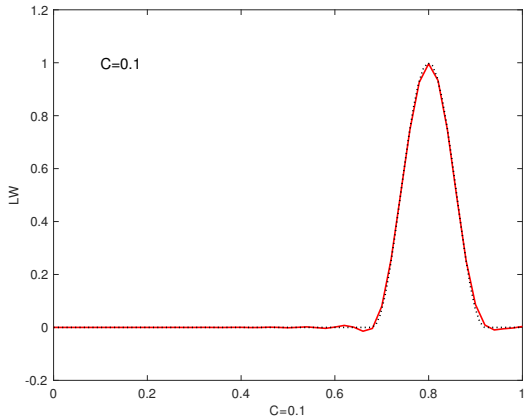


Figure 2: High accuracy propagation of Cosine profile for C=0.1

1.b Third-Order Explicit Taylor-Galerkin (TG3) Method

Here we will comment on the second of the four numerical experiments, the Third-Order Explicit Taylor-Galerkin (TG3) Method. The results for implementations of different Courant numbers can be seen below.

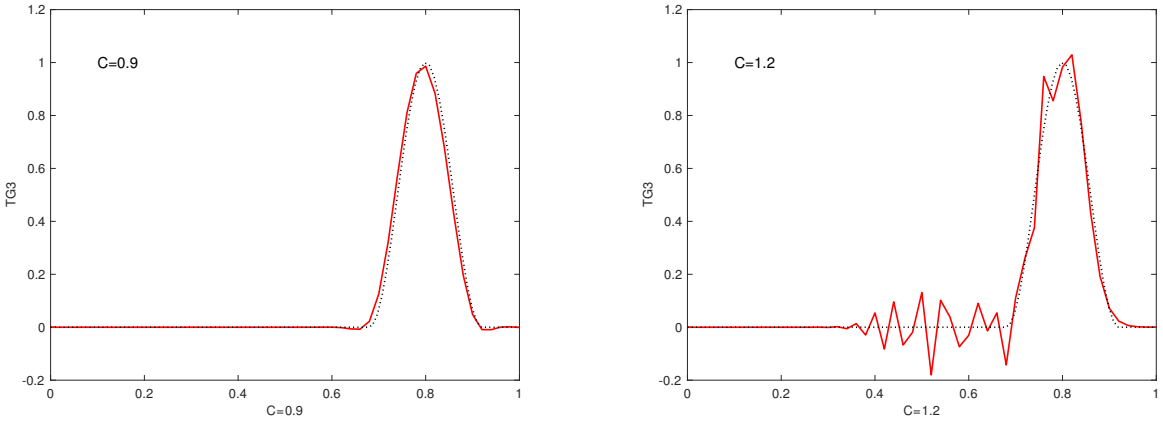


Figure 3: Propagation of Cosine profile for TG3 C=0.9 (left) C=1.2 (right)

Above we can see two different Courant number implementations. On the right hand side, we have results for an implementation with Courant number of 1.2. This value is out of the stability criterion of Second Order Lax-Wendroff (TG3) $C^2 \leq 1$, and therefore we expect oscillatory behavior producing inaccurate and unstable results. The graphs confirm this phenomenon and agree with the theory. On the left hand side we see the results for a Courant number of 0.9. This is within the stability criterion of $C^2 \leq 1$, and as we can notice from the figure, the graph is stable with very minimal inaccuracies. This method's implementation and results agree with the theory for the method.

1.c Second-Order Crank-Nicolson finite element method;

Here we will comment on the third of the four numerical experiments to be tested, the Second-Order Crank-Nicolson Method. The results for implementations of different Courant numbers can be seen below.

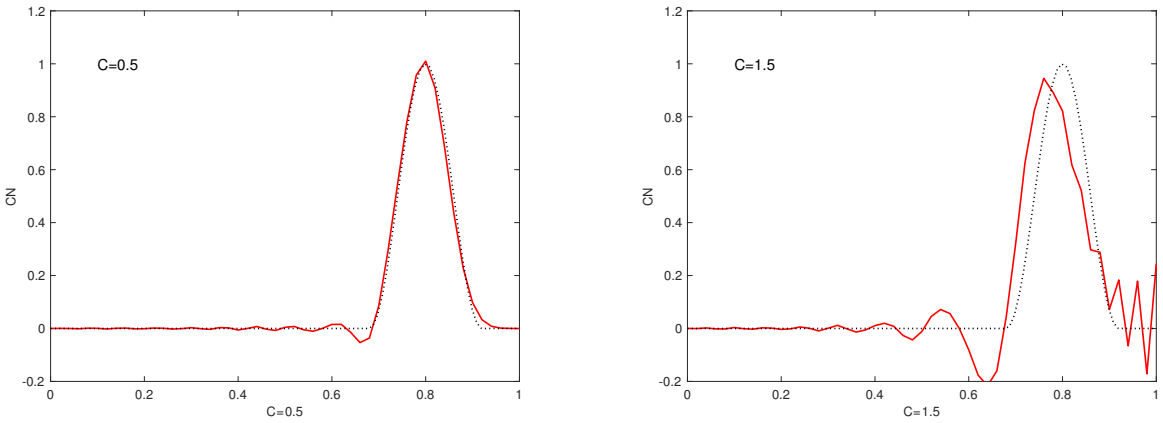


Figure 4: Propagation of Cosine profile for Crank-Nicholson C=0.5 (left) C=1.5 (right)

Above we can see two different Courant number implementations. On the right hand side, we have results for an implementation with Courant number of 1.5, and on the left hand side we have results for Courant number of 0.5. Both methods exhibit stability but it is clear from the graph that the implementation with a Courant number of 0.5 is much more accurate than the implementation with Courant number of 1.5. These results are consistent with the theory as a implementing the method with a lower time step will produce higher accuracy results reflected in the lower Courant number as it is directly proportional to the value of the time step.

1.d Fourth-Order Implicit Taylor-Galerkin (TG4) and Comparisons

Here we will comment on the last of the four numerical experiments to be tested, the Fourth-Order Implicit Taylor-Galerkin (TG4) finite element Method. The results for implementations of different Courant numbers can be seen below.

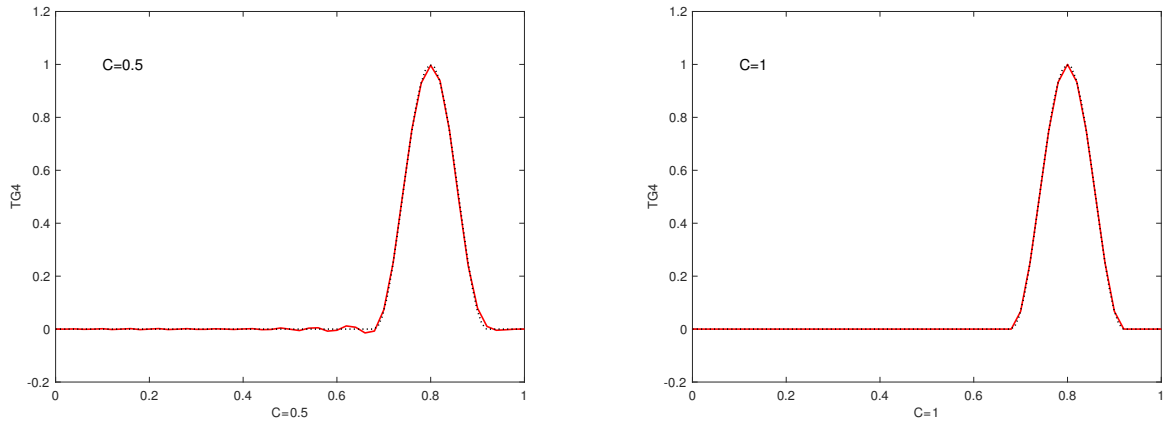


Figure 5: Propagation of Cosine profile for TG4 C=0.5 (left) C=1 (right)

Above we can see two different Courant number implementations. On the right hand side, we have results for an implementation with Courant number of 1. And on the left we have an implementation with a Courant number equal to 0.5. Both of these implementations produce nearly identical results with the exception of some slight inaccuracies in the graph with Courant number equal to 0.5.

Based on all the previous results, it can be seen that this method (TG4) has more accurate solutions at lower Courant numbers when compared to the Crank-Nicholson method. It can also be concluded that the TG3 method is inferior to the Crank-Nicholson method at higher Courant numbers. The Lax-Wendroff second order method appears to be the least accurate and stable of the bunch. These observations from the graphs coincide with the theory and this indicated that the implementations are correct.

2 Propagation of a Steep Front

2.a Various Crank-Nicholson Solutions

We will first solve the problem using the Crank-Nicholson scheme in time and linear finite element for the Galerkin scheme in space. We will then comment on the accuracy of these solutions. The relevant figures for this can be seen below.

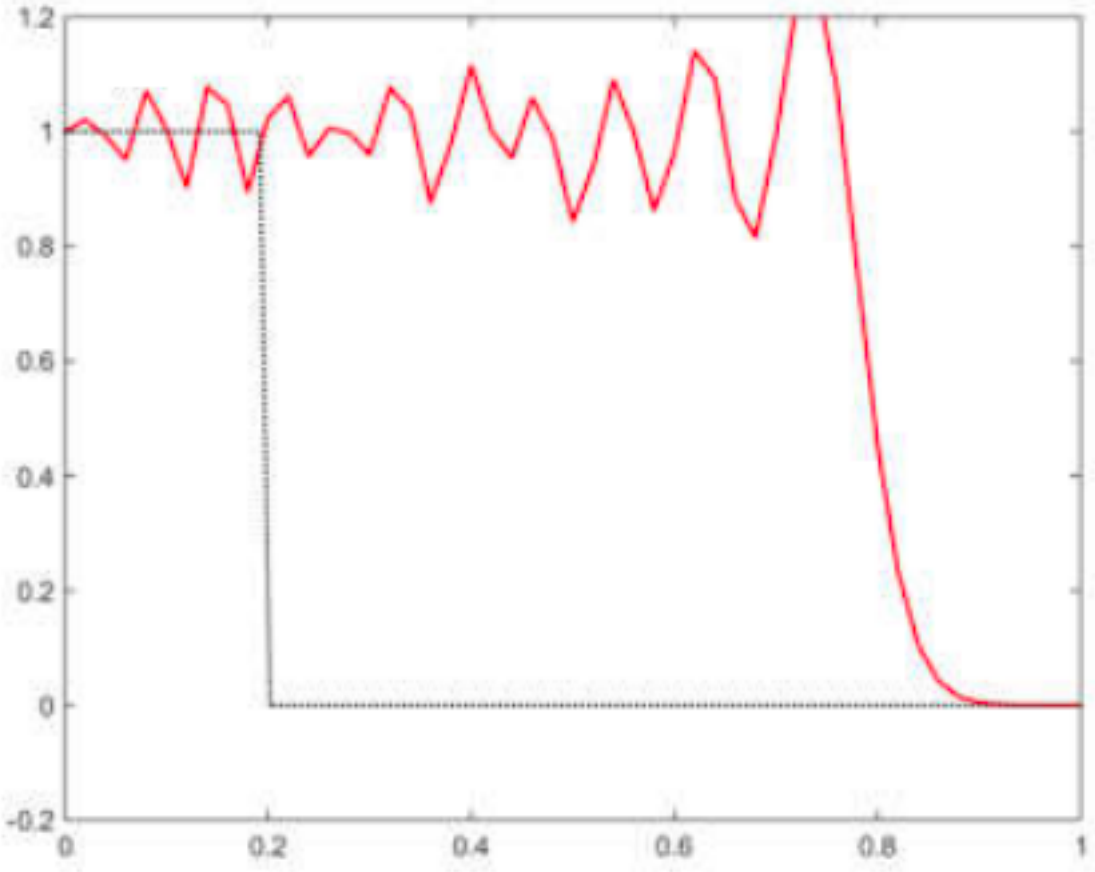


Figure 6: Crank Nicholson with Galerkin Solution with Courant Number of 0.75

Above we can see the results of the Crank-Nicholson in time and Galerkin in space implementation. This graph exhibits some level of stability with a high degree of inaccurate oscillatory behavior trailing the steep front. Since the Courant number has a value of 0.75, these results are fairly consistent with the theory presented and available in the literature.

Now that we have these results, we can compare to the same Crank-Nicholson time discretization but with a different space discretization. This space discretization will be the Galerkin Least Square (GLS) discretization and can be seen on the next page by hand.

Crank-Nicholson GLS

$$\left(\frac{w}{\Delta t} + \theta a\cdot\nabla w,\ \frac{\Delta U}{\Delta t} + \theta a\cdot\nabla \Delta U\right) = \left(\frac{w}{\Delta t} + \theta a\cdot\nabla w,\ -a\cdot\nabla U^{n}\right)$$

(3 4 5 6 1 2)

and simplifying the terms...

① $\left(\frac{w}{\Delta t}, -a\cdot\nabla U^{n}\right) = \frac{-1}{\Delta t} a C U^{n}$

② $\left(\theta a\cdot\nabla w, -a\cdot\nabla U^{n}\right) = -\theta a^{2} K U^{n}$

③ $\left(\frac{w}{\Delta t}, \frac{\Delta U}{\Delta t}\right) = \frac{M}{(\Delta t)^{2}} \Delta U$

④ $\left(\frac{w}{\Delta t}, \theta a\cdot\nabla \Delta U\right) = \frac{\theta}{\Delta t} C$

⑤ $\left(\theta a\cdot\nabla w, \frac{\Delta U}{\Delta t}\right) = \frac{-\theta}{\Delta t} C$

⑥ $\left(\theta a\cdot\nabla w, \theta a\cdot\nabla \Delta U\right) = \theta^{2} a^{2} K$

And finally, rearranging and substituting $\theta = \frac{1}{2}$ for CN we can write...

$$M\Delta U + \frac{a^{2} K (\Delta t)^{2} \Delta U}{4} = -\left(\Delta t\, a C U^{n} + \frac{1}{2} a^{2} K U^{n}\right)$$

Figure 7: Crank Nicholson with GLS on-paper development

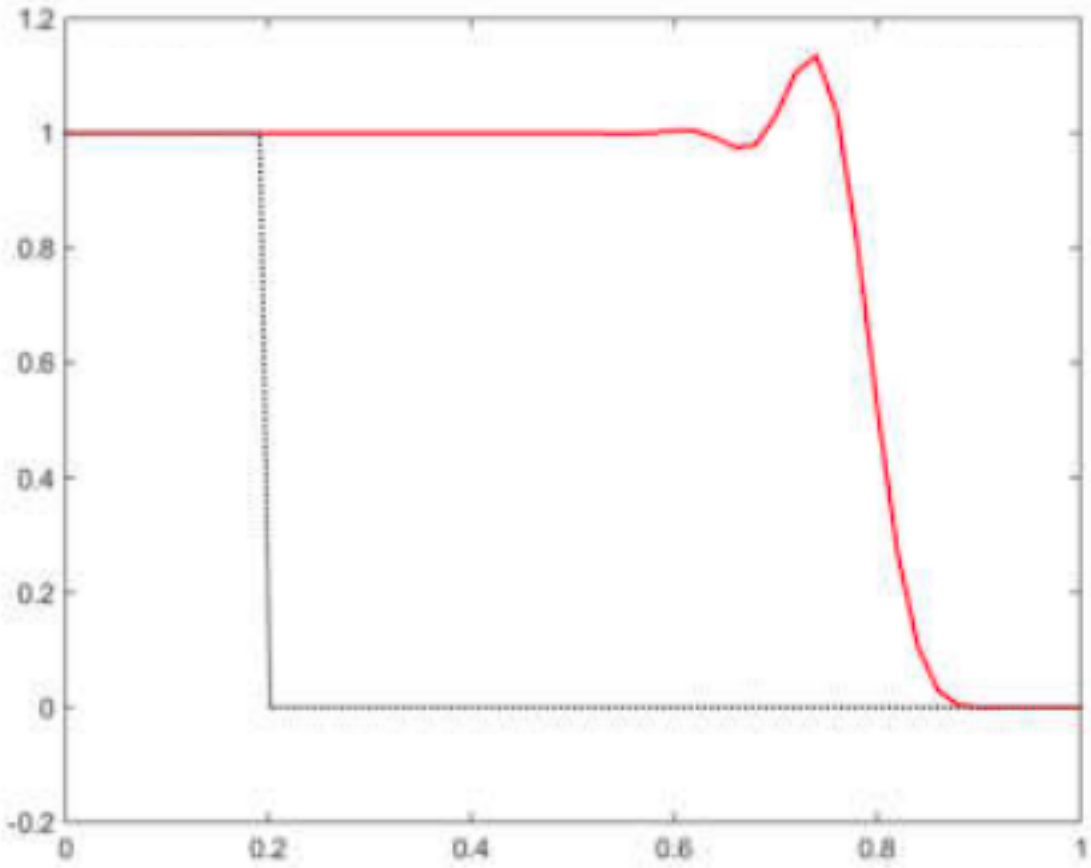


Figure 8: Crank Nicholson with GLS with Courant Number of 0.75

Above we can see the results for the Crank Nicholson in time with Galerkin Least Squares space discretization. It is very important to notice that the Crank-Nicolson with least-squares is successful in removing the unwanted oscillations caused by the Galerkin only approach over the entire computational domain. And because Crank-Nicolson is not a monotone scheme, there are some residual oscillations remaining at the steep front.

3 The Gaussian Hill

3.a Implememtation of Adams-Bashforth

Here we will first develop the Adams-Bashforth method from the the literature before implementing it in the codes. Below we can find the development of this method by hand.

Adams-Bashforth Scheme

$$U^{n+1} = U^n + \frac{\Delta t}{2}\left(3U_t^n - U_t^{n-1}\right)$$

$$U_t = -a\cdot\nabla U + \nabla\cdot(\nu\nabla U) + 0 + 0$$

$$U^{n+1} = U^n + \frac{\Delta t}{2}\left(3(-a\cdot\nabla U + \nabla\cdot(\nu\nabla U)) - (-a\cdot\nabla U + \nabla\cdot(\nu\nabla U))\right)$$

$$\frac{U^{n+1}-U^n}{\Delta t} = \frac{\Delta U}{\Delta t} = \frac{1}{2}\left(-3a\cdot\nabla U^n + 3\nabla\cdot(\nu\nabla U^n) + a\cdot\nabla U^{n-1} - \nabla(\nu\nabla U^{n-1})\right)$$

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

now integrating by parts...

$$\left(w, \frac{3}{2}\nu\nabla\cdot(\nabla U^n)\right) = -\frac{3}{2}\nu\left(\nabla w, \nabla U^n\right) + 0$$

$$-\left(w, \frac{\nu}{2}\nabla\cdot(\nabla U^{n-1})\right) = \frac{\nu}{2}\left(\nabla w, \nabla U^{n-1}\right) + 0$$

And finally rearranging...

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

$$M\frac{\Delta U}{\Delta t} = -\frac{3}{2}aC - \frac{3}{2}\nu K + \frac{1}{2}aC + \frac{\nu}{2}K$$

U^n U^{n-1}

Figure 9: By Hand Development of Adams-Bashforth Method

Now that we have developed the Adams-Bashforth we will now implement it into the codes. This code implementation and results can be seen on the following page.

```
30
31          % ADAMS-BASHFORTH METHOD
32 -        case 5
33
34              %LEFT HAND SIDE OF ADAMS BASHFORTH
35 -            A= M;
36
37              %TERMS RIGHT HAND SIDE OF ADAMS BASHFORTH AT TIME STEP N
38 -            B= -(3/2)*a*dt*C + (3/2)*dt*nu*K;
39
40              %TERMS RIGHT HAND SIDE OF ADAMS BASHFORTH AT TIME STEP N-1
41 -            B_B= (1/2)*a*dt*C - (1/2)*dt*nu*K;
42
43 -            methodName = 'Adams-Bashforth Method';
44
45
```

Figure 10: Code implementation of Adams-Bashforth Method

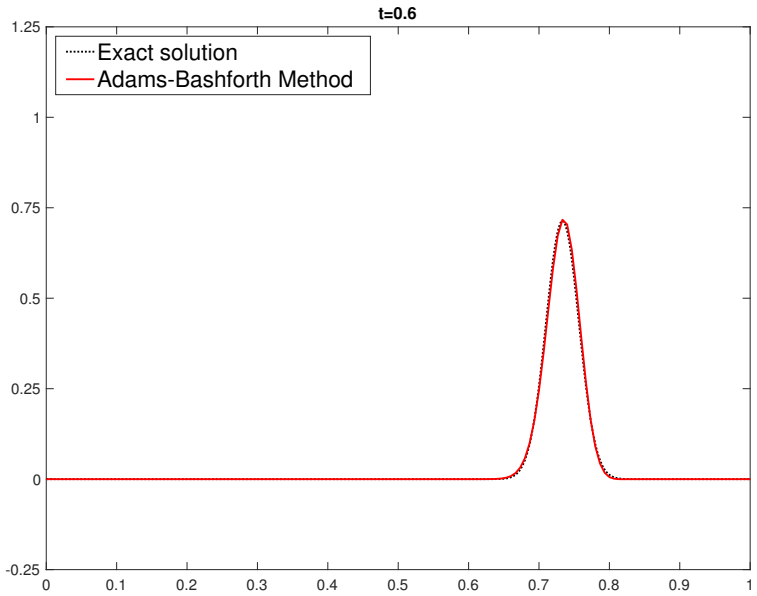


Figure 11: Adams-Bashforth with Courant number of 0.1875

Above we can see the results for the Adams-Bashforth implementation with low Courant number and low Peclet number. The results are almost perfect with very small variations on the cosine hill. This corresponds to the theory and signals that the implementation is behaving correctly.