

# Assignment 6 (The Gaussian Hill)

PRADEEP KUMAR BAL

April 2, 2018

## Adams-Bashforth scheme:

The Adams-Bashforth method is one of the classical methods for transient problems. This is a second-order accurate explicit method which requires evaluation of the first time derivative at two consecutive time levels. The scheme is:

$$u^{n+1} = u^n + \frac{\Delta t}{2}(3u_t^n - u_t^{n-1});$$

Where for this case:  $u_t = -au_x + \gamma u_{xx}$ .

Adams-Bashforth method is not self-starting. The first step is usually performed using the forward Euler method, preferably with a small time step. For Euler method  $w=1; W=0$ . After the first time step ( $p=1$ ),  $w$  is set to  $3/2$  and  $s(i) * (aux)$  is replaced by  $s(i) * (aux) + 0.5 * Kt * dt * Sol(:, p - 1)$  for  $p = 2$  to the final time step.

## Stability range of explicit methods (Euler and Adams-Bashforth Schemes):

In pure diffusion in 1D, the forward Euler method is stable for  $\Delta t \leq \frac{h^2}{6\gamma}$  or  $d \leq \frac{1}{6}$ . The method is unstable in pure convection if a centered (Galerkin) approximation is used for the convective term. In convection-diffusion, the stability condition depends on the value of Pe:

$$\begin{cases} \text{if } Pe \leq \sqrt{3}; & C \leq \frac{Pe}{3} \\ \text{if } Pe > \sqrt{3} & C \leq \frac{1}{Pe} \end{cases}$$

Stability of the second-order Adams-Bashforth method is also governed by the same type of criteria, but its stability range is only half that for the Euler method. A forward Euler method has been implemented. For the first time step a very small time step is chosen. Following this the Adams-Bashforth scheme has also been employed. For the first time step, It uses the solution of the Forward Euler scheme. The code is delineated in the Annex. Various cases has been observed by varying the Peclet number (Pe) and the Courant number(C).

In the first case, the Pe has been chosen as 5. The Courant number (C) has been taken as small, 0.1. The obtained result is shown in the Figure 1. For this case the method behaves well.

In the second case, Pe has been considered as 1. The C is chosen as 0.3. The obtained result has been displayed in the Figure 2. The method, for this case, becomes unstable as C is comparatively higher for the Pe=1.

In the third case, where Pe=10 and C=0.3 has been chosen. As in this case, Pe is a bit higher for the same C=0.3, the method behaves well. It can be observed from the Figure 3.

Another case Pe=0.5 has been chosen, and C is considered as 1. In this case the method becomes unstable as Pe is very low and C is high. The obtained unstable solutions have been presented in the Figure 4.

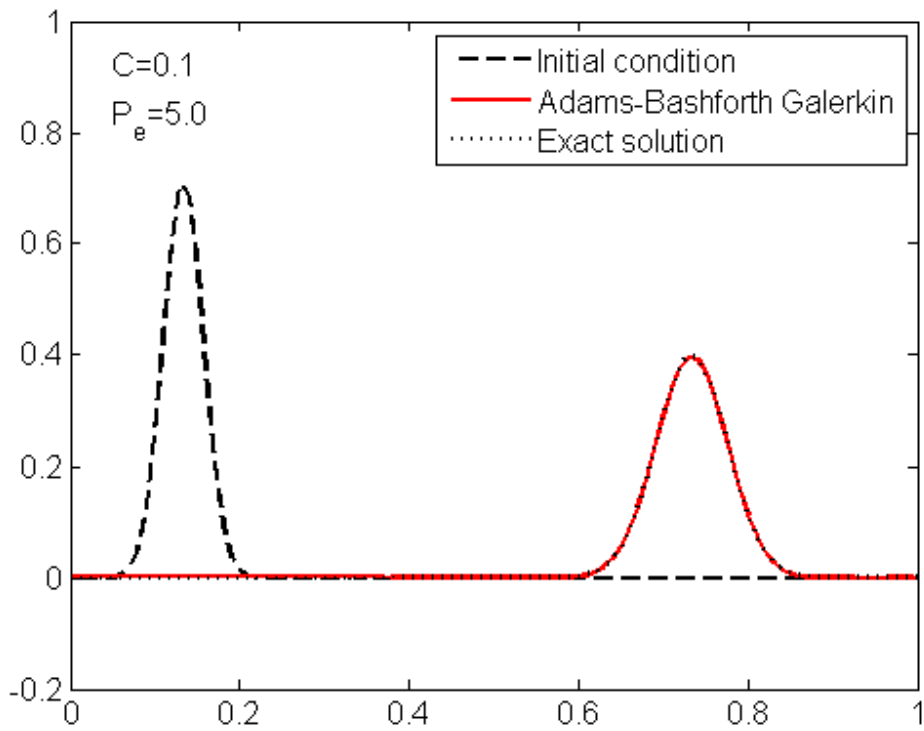


Figure 1: Comparison of the solution obtained using the Adams-Bashforth method for  $C=0.1$  and  $Pe=5$  with the exact solution

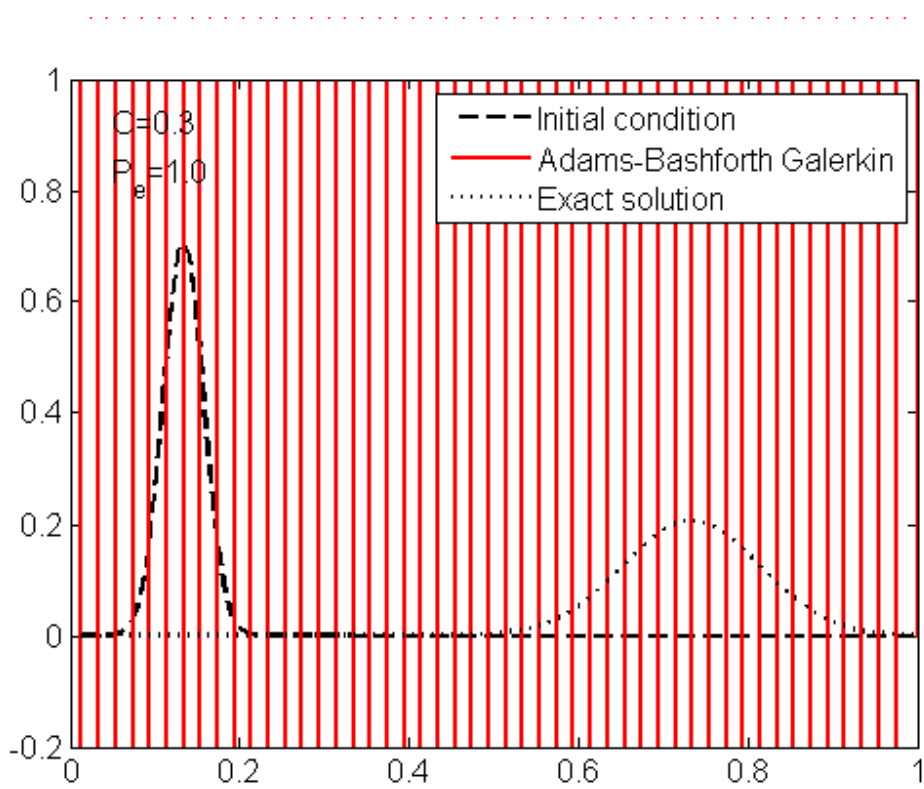


Figure 2: Comparison of the solution obtained using the Adams-Bashforth method for  $C=0.3$  and  $Pe=1$  with the Exact Solution

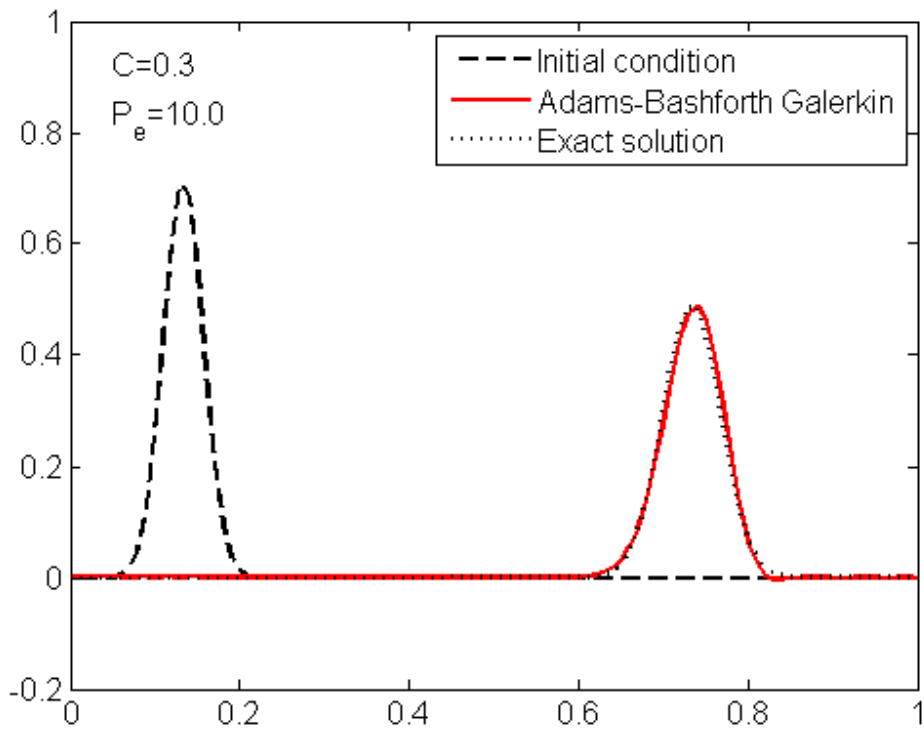


Figure 3: Comparison of the solution obtained using the Adams-Bashforth method for  $C=0.3$  and  $Pe=10$  with the exact solution

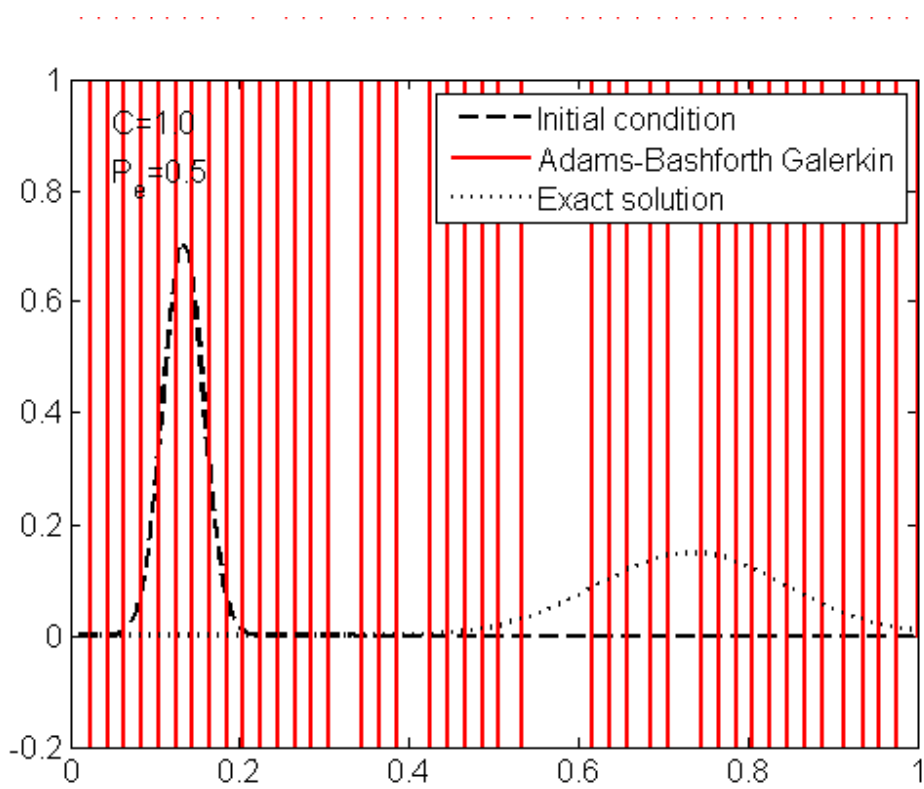


Figure 4: Comparison of the solution obtained using the Adams-Bashforth method for  $C=1$  and  $Pe=0.5$  with the Exact Solution

Another last two cases have been considered where a very small time step has been chosen.  $Pe$  is chosen as 0.5 and  $C=0.1$  in the first case, and  $Pe=10$  and  $C=0.1$  in the second case. In the first case we obtain an unstable solution as the  $Pe$  is low instead of low  $C$ . In the second case, we obtain more improved accurate solution as  $Pe$  is comparatively higher and  $C$  is low. The results has shown in the Figure 5 and in the Figure 6. When  $C$  is increased, Adams-Basforth method does not behave well. A few more cases have been considered for  $C=3, Pe=1; C=3, Pe=5; C=3, Pe=100; C=4, Pe=100$ . For all these cases Adams-Basforth method gives unstable solutions as  $C$  is very high; instead of values of  $Pe$ .

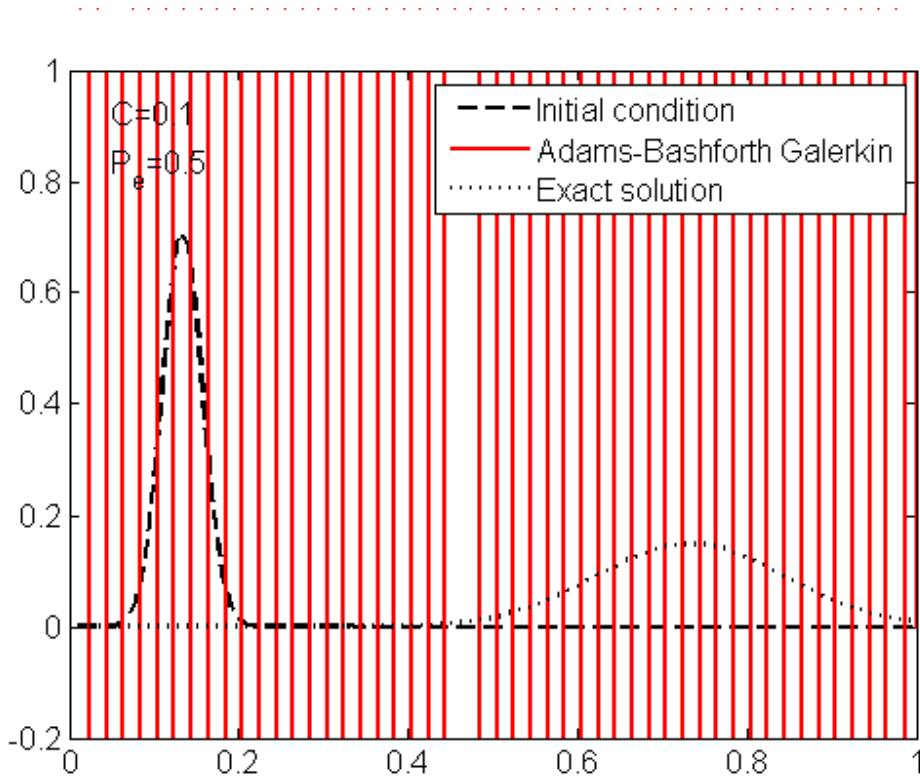


Figure 5: Comparison of the solution obtained using the Adams-Bashforth method for  $C=0.1$  and  $Pe=0.5$  with the exact solution

### Padé $R_{2,2}$ scheme:

This is an implicit fourth order two step method, which is unconditionally stable. It allows us to use high-order time stepping schemes. Thus, accuracy can be improved drastically (obviously, for the solvable frequencies  $(0 \leq \xi \leq \pi/4)$ ). The behaviour of the method has been studied for 1:  $C=3, Pe=1$ ; 2:  $C=3, Pe=5$ ; 3:  $C=3, Pe=100$ ; 4:  $C=4, Pe=100$ . At high  $C$  values, this method behaves well in-comparison to the Adams-Bashforth method. It can be observed from the Figures 7,8,9,10; where for these cases the Adams-Bashforth was unstable for the same  $Pe$ , and  $C$  values. Figure 9, 10 shows relative errors in amplitude and phase for the fourth-order method,  $R_{2,2}$ . It is concluded that a Galerkin formulation presents spatial instabilities in the presence of boundary layers. They need to be stabilised. At the end two other cases ( $Pe=1, C=0.3; Pe=0.5, C=1$ ) have been studied for which Adams-Bashforth scheme was unstable. The  $R_{2,2}$  scheme gives improved accurate results for these two cases. It can be observed from the Figure 12 and Figure 14.

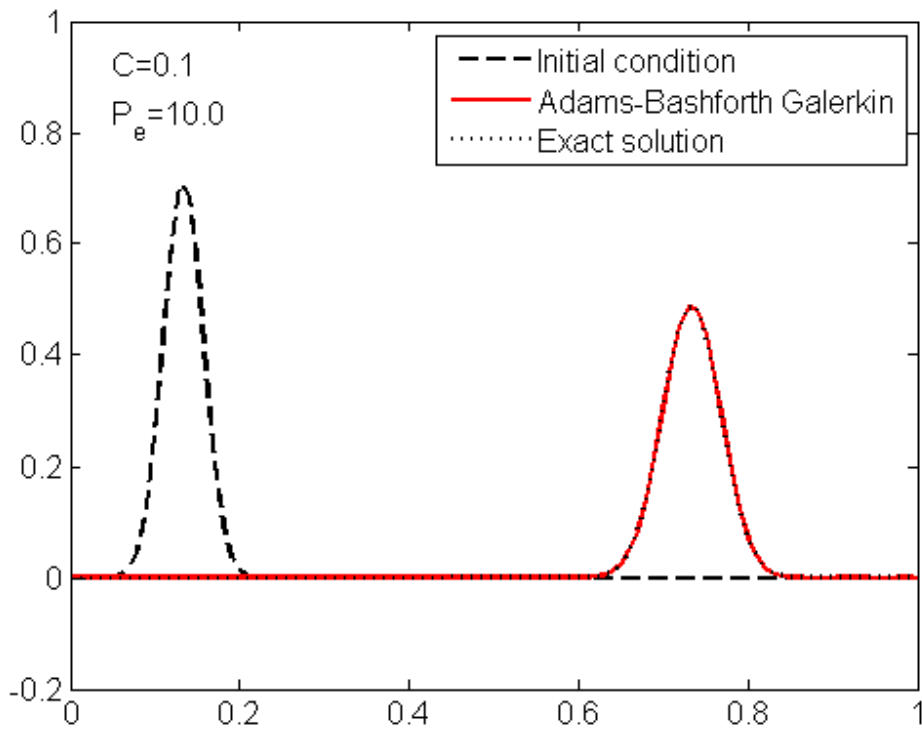


Figure 6: Comparison of the solution obtained using the Adams-Bashforth method for  $C=0.1$  and  $Pe=10$  with the exact solution

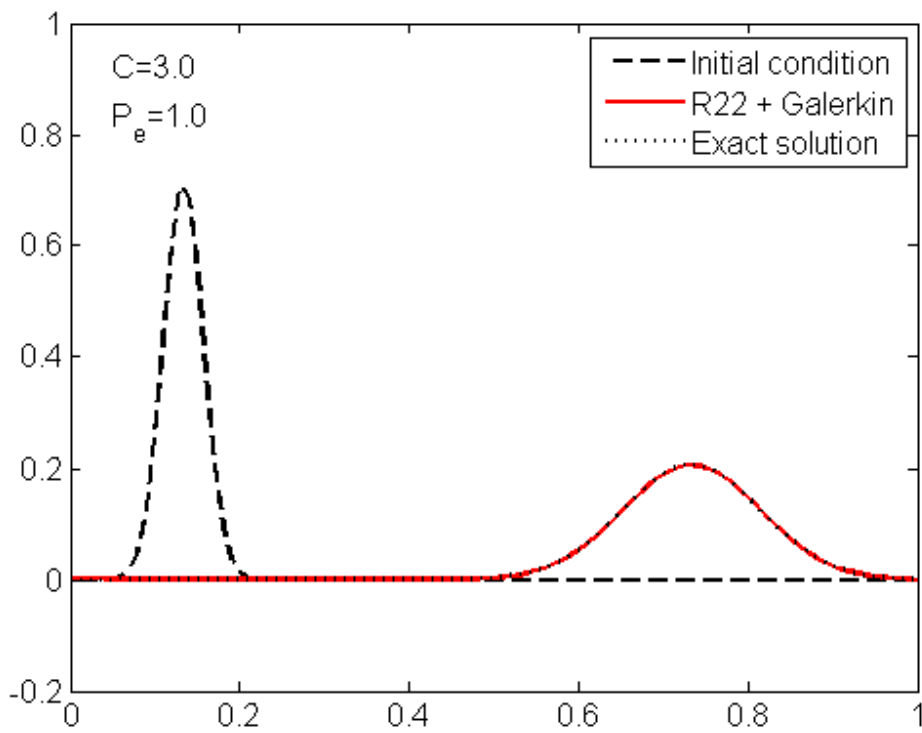


Figure 7: Comparison of the solution obtained using the  $R_{2,2}$  for  $C=3$  and  $Pe=1$  with the exact solution

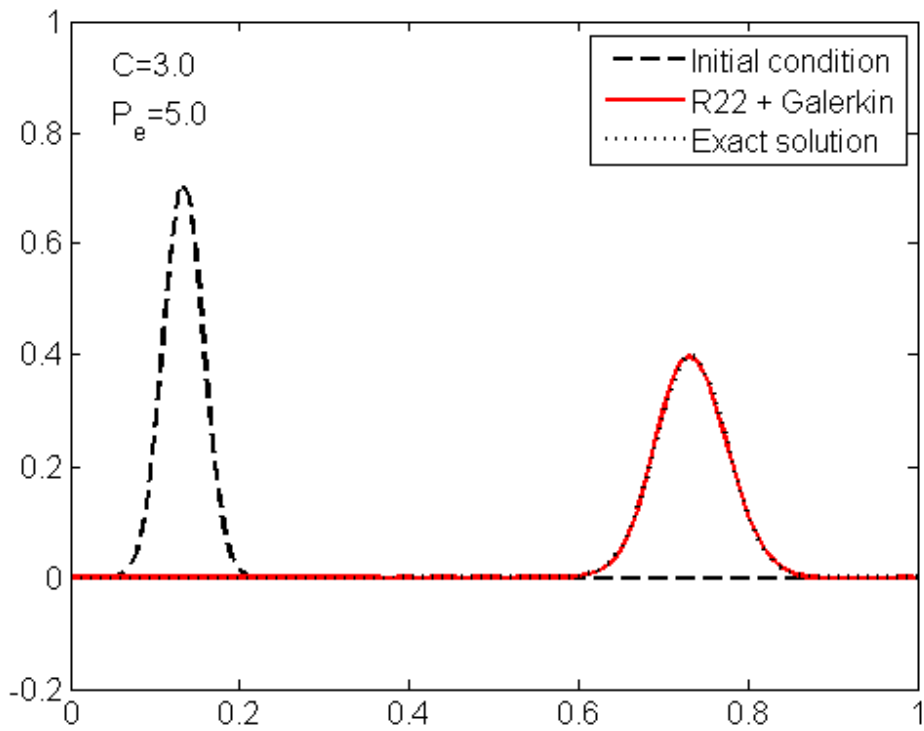


Figure 8: Comparison of the solution obtained using the  $R_{2,2}$  for  $C=3$  and  $Pe=5$  with the exact solution

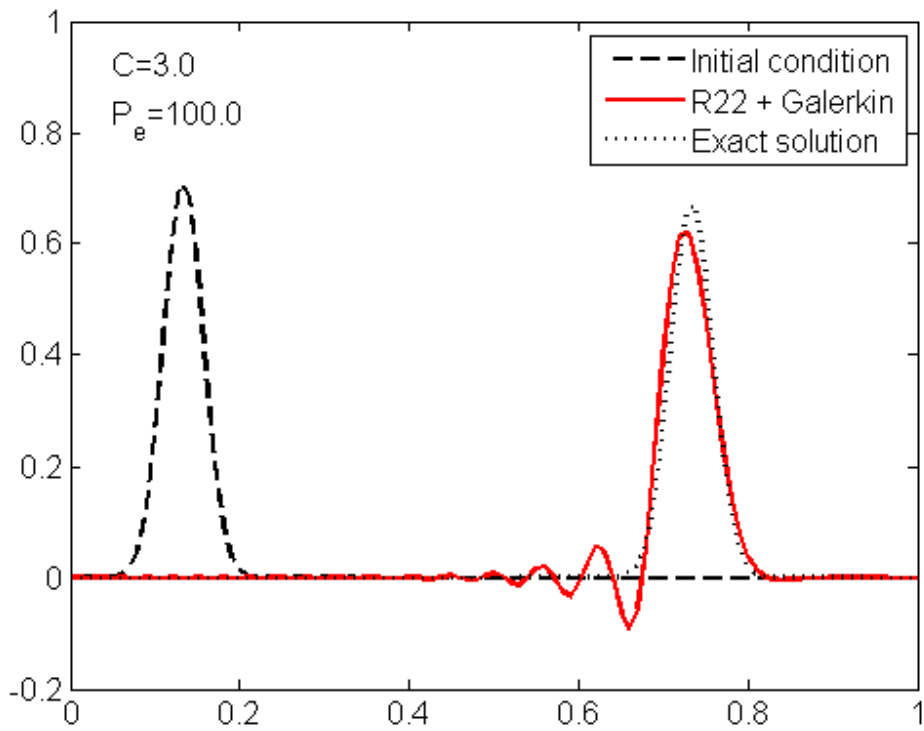


Figure 9: Comparison of the solution obtained using the  $R_{2,2}$  for  $C=3$  and  $Pe=100$  with the exact solution

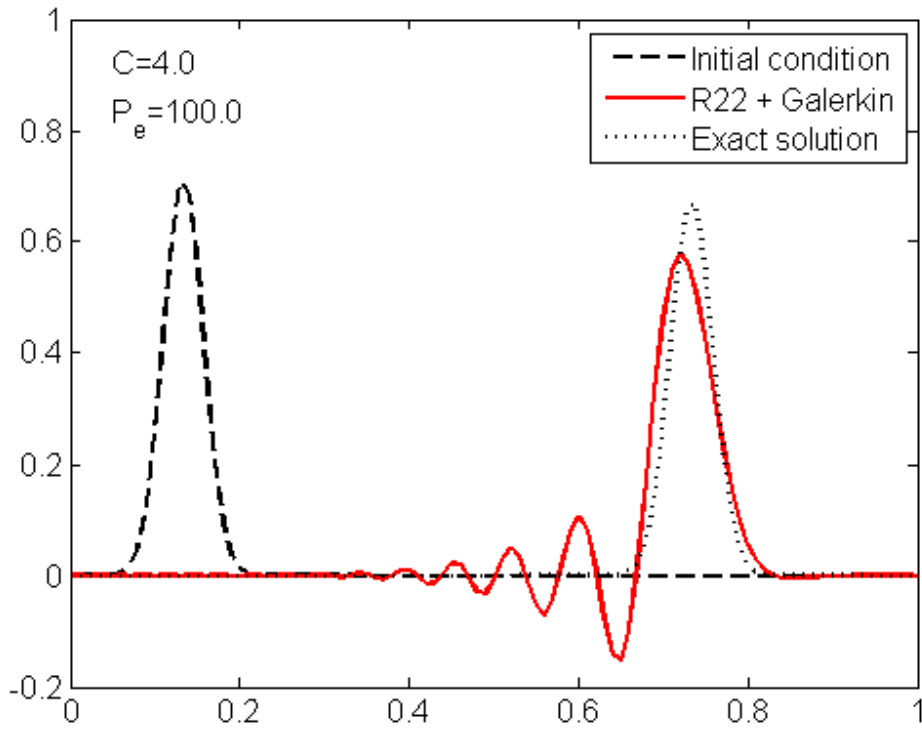


Figure 10: Comparison of the solution obtained using the  $R_{2,2}$  for  $C=4$  and  $Pe=100$  with the exact solution

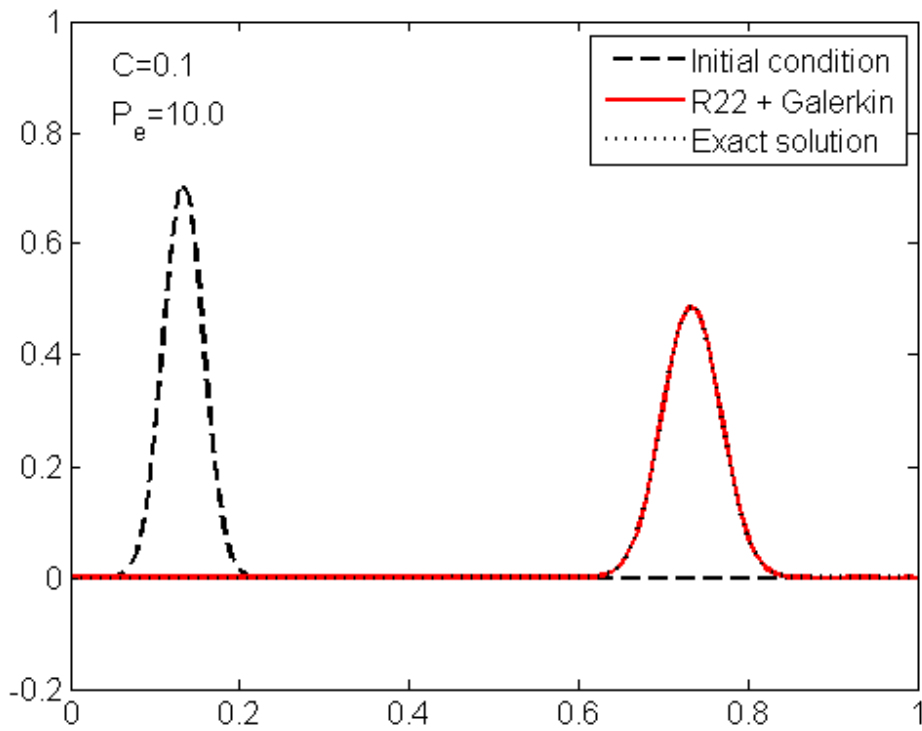


Figure 11: Comparison of the solution obtained using the  $R_{2,2}$  for  $C=0.1$  and  $Pe=10$  with the exact solution

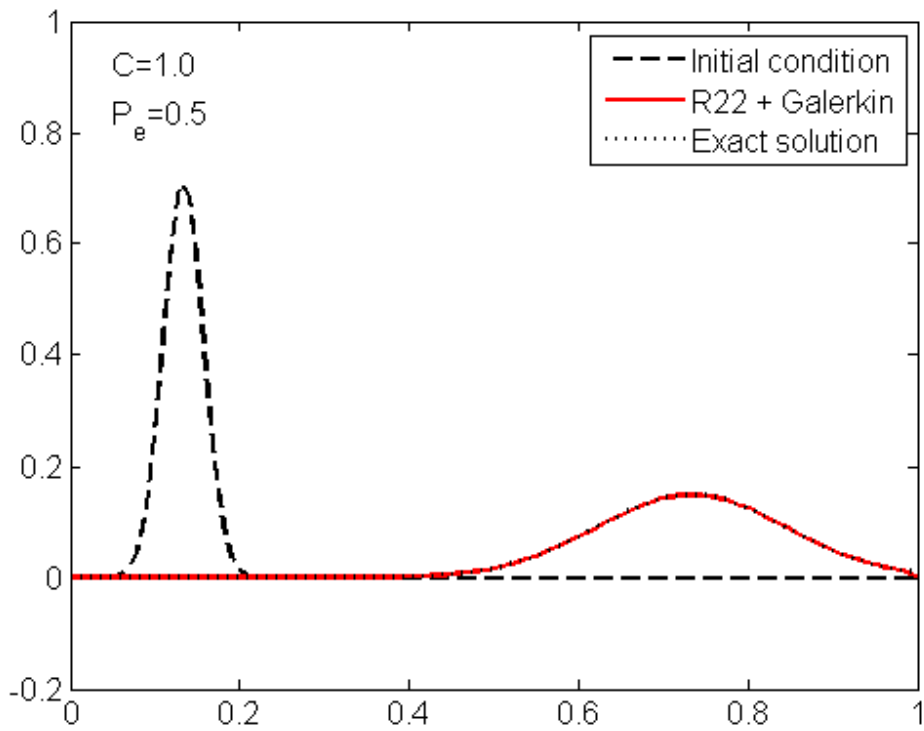


Figure 12: Comparison of the solution obtained using the  $R_{2,2}$  for  $C=1$  and  $Pe=0.5$  with the exact solution

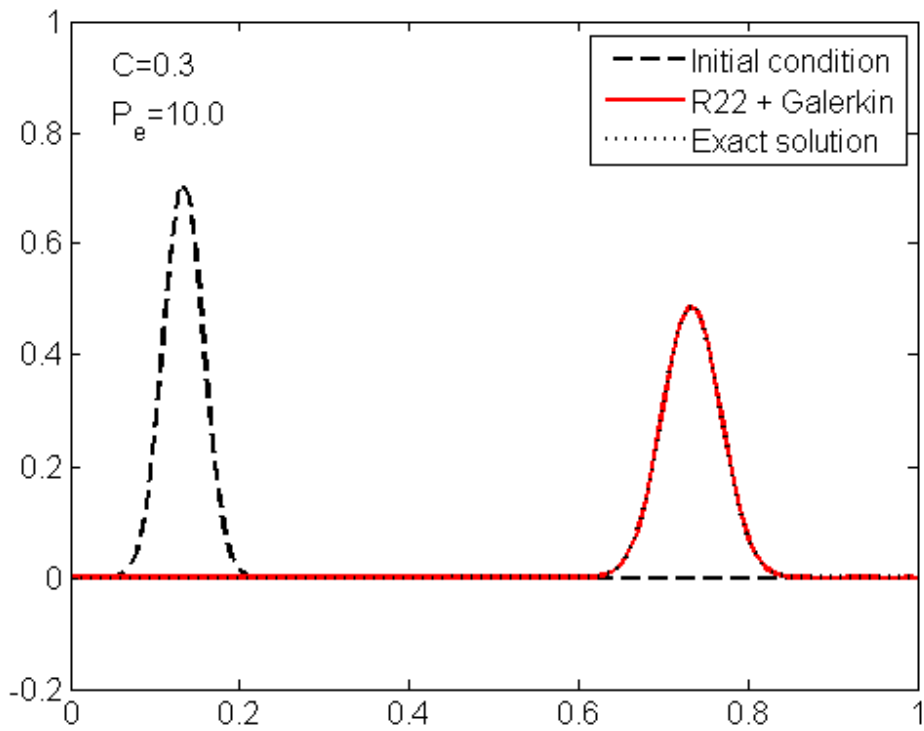


Figure 13: Comparison of the solution obtained using the  $R_{2,2}$  for  $C=0.3$  and  $Pe=10$  with the exact solution



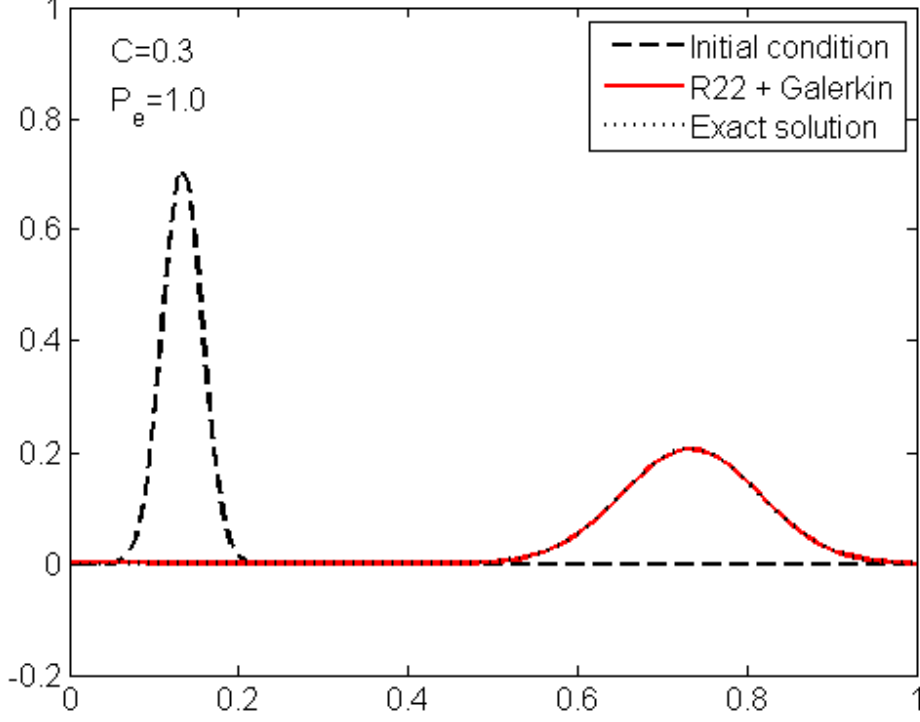


Figure 14: Comparison of the solution obtained using the  $R_{2,2}$  for  $C=0.3$  and  $Pe=1$  with the exact solution

### Time-discontinuous Galerkin:

Since the finite element interpolation is discontinuous at the space-time slab interfaces, it is useful to employ the notation  $u^h(t_{\pm}^n) = \lim_{\epsilon \rightarrow 0^+} u^h(t^n \pm \epsilon)$ . The weighted residual formulation of the homogeneous linear convection-diffusion equation:

$$\iint_{Q^n} w^h (u_t^h + \mathbf{a} \cdot \nabla u^h - \nabla \cdot (\nu \nabla u^h)) d\Omega dt + \int_{\Omega} w^h(t_+^n) (u^h(t_+^n) - u^h(t_-^n)) d\Omega = 0;$$

Linear finite element approximations are employed in both space and time, giving a third-order accurate and unconditionally stable method. Adapting the developments in Section 3.10.1 (pure convection) to the present convection-diffusion case, The following partitioned matrix system is obtained for the nodal unknowns  $u^{n+1}$  and  $u^{n+}$ :

$$(\mathbf{M} + \frac{2}{3}\Delta t \mathbf{C} + \frac{2}{3}\nu \Delta t \mathbf{K}) \mathbf{u}^{n+1} - (\mathbf{M} - \frac{1}{3}\Delta t \mathbf{C} - \frac{1}{3}\nu \Delta t \mathbf{K}) \mathbf{u}^{n+} = 0$$

$$(\mathbf{M} + \frac{1}{3}\Delta t \mathbf{C} + \frac{1}{3}\nu \Delta t \mathbf{K}) \mathbf{u}^{n+1} + (\mathbf{M} + \frac{2}{3}\Delta t \mathbf{C} + \frac{2}{3}\nu \Delta t \mathbf{K}) \mathbf{u}^{n+} = 2M u^{n-}$$

The matrices  $\mathbf{M}$  (consistent mass),  $\mathbf{C}$  (convection), and  $\mathbf{K}$  (diffusion) are defined as:

$$M_{ab}^e = \sum_e \int_{\Omega^e} N_a N_b d\Omega \quad \mathbf{M} = \mathbf{A}^e M^e \quad (\mathbf{A} \text{ DENOTES THE ASSEMBLE OPERATOR})$$

$$C_{ab}^e = \sum_e \int_{\Omega^e} N_a (\mathbf{a} \cdot \nabla N_b) d\Omega \quad \mathbf{C} = \mathbf{A}^e C^e$$

$$K_{ab}^e = \sum_e \int_{\Omega^e} \nabla N_a (\nu \cdot \nabla N_b) d\Omega \quad \mathbf{K} = \mathbf{A}^e K^e$$

Where  $1 \leq a, b \leq n_{en}$ ;  $n_{en}$  is the number of element nodes.

The condition  $u^{n+1}(1) = u^{n+}(1) = 1$  is then enforced to satisfy the inlet condition. This third-order accurate and unconditionally stable method requires the solution of an algebraic system double the size of usual time-stepping algorithms. It is observed that, oscillations extending over the whole computational domain characterize the Galerkin formulation. It is investigated that the level of the spurious oscillations appears to decrease when the Courant number is increased.

## Adams Bash-forth Scheme:-

```
disp('There are four integration schemes available ')
disp('      [0] = Crank-Nicolson');
disp('      [1] = R22');
disp('      [2] = R33');
disp('      [4] = Adam-Basforth');
%disp('      [4] = Forward-Euler');
disp('and five methods to perform spatial discretization')
disp('      [0] = Galerkin');
disp('      [1] = Least-Squares');
disp('      [2] = Streamline-Upwind Petrov-Galerkin (SUPG)');
disp('      [3] = Galerkin Least-Squares (GLS)');
disp('      [4] = Sub-Grid Scale (SGS)');
disp(' ')
d_temp = input('Choose a method to perform time integration = ');
d_esp = input('and another one for the spatial discretization = ');

-----

elseif d_temp == 4
    method = 'Adams-Bashforth ';
    W = 0;
    w = 1;%%% Start with Euler; Will change to Adams-Basforth in Galerkin1
else
    error('Unavailable time integration scheme')
end
-----
% Solution

if d_esp==0 && d_temp==4
    Sol = Galerkin1(W,w,a,nu,f,K,M,G,xnode,dt,nstep,c,Accd,bccd);
elseif d_esp == 0 && d_temp~=4
    Sol = Galerkin(W,w,a,nu,f,K,M,G,xnode,dt,nstep,c,Accd,bccd);
elseif d_esp == 1
    Sol = ILS(W,w,a,nu,f,K,M,G,xnode,dt,nstep,c,Accd,bccd);
elseif d_esp == 2
    Sol = SUPG(W,w,tau,a,nu,f,K,M,G,xnode,dt,nstep,c,Accd,bccd);
elseif d_esp == 3
    Sol = GLS(W,w,tau,a,nu,f,K,M,G,xnode,dt,nstep,c,Accd,bccd);
elseif d_esp == 4
    Sol = SGS(W,w,tau,a,nu,f,K,M,G,xnode,dt,nstep,c,Accd,bccd);
end
-----

function Sol = Galerkin1(T,s,a,nu,f,K,M,G,xnode,dt,nstep,c,Accd1,bccd1)
Time step loop:

% Loop to compute the transient solution
for p=1:nstep
if p==1 %%%Forward Euler method loop
    aux = dt*(-Kt*c + Mf);
    F = [];
    for i =1:n
        F = [F; s(i)*aux];
    end
    F = [F;bccd*0];
    dc = U\ (L\F);
    dc = reshape(dc(1:n*npoin),npoin,n);
    c = c + sum(dc,2);
    Sol = [Sol c];
end
end
```

```

s=1.5*s; %%%w=1.5 for Adam basforth

else

aux = dt*(-Kt*c+ Mf);
F = [];
for i =1:n
    F = [F; s(i)*(aux)+0.5*Kt*dt*Sol(:,p-1)];%% for Adam basforth
end
F = [F;bccd*0];
dc = U\ (L\F);
dc = reshape(dc(1:n*npoin),npoin,n);
c = c + sum(dc,2);
Sol = [Sol c];
end

```