Finite Elements for Fluids - Coding

Arthur Lustman

March 20, 2019

This report for the Finite Element for Fluids class is split in two sections : the unsteady convection problems and the implementation of a solution of a non-linear system of the Burger's equation.

1 Unsteady convection problems

1.1 Crank-Nicolson

$$\left(w, \frac{\Delta u}{\Delta t}\right) + \theta(w, (a \cdot \nabla)\Delta u) = (w, \theta s^{n+1} + (1-\theta)s^n) - (w, a \cdot \nabla u^n) \tag{1}$$

$$\left(\boldsymbol{M} + \frac{\Delta t}{2}\boldsymbol{C}\right)\Delta u = \Delta t\,(f - \boldsymbol{C}u^n) \tag{2}$$

The equations are translated to a simple problem $\boldsymbol{A}u = b$. In here there are no source terms so f in equal to zero.

```
case 3 % Crank−Nicolson + Galerkin
    A = M + 1/2∗a∗dt∗C;
    B = −a∗dt∗C;
    methodName = 'CN';
```

1.2 Lax-Wendroff

$$\frac{\Delta u}{\Delta t} = -(a \cdot \nabla)u^n + \frac{\Delta t}{2}(a \cdot \nabla)^2 u^n \tag{3}$$

Which translated to the weak form

$$\frac{1}{\Delta t}\boldsymbol{M}\Delta u = \left(-a\boldsymbol{C} - \frac{\Delta t}{2}a^2\boldsymbol{K}\right)u^n \tag{4}$$

```
case 1 % Lax−Wendroff + Galerkin
    A = M;
    B = − a∗dt∗C − 1/2∗K∗a∗a∗dt∗dt;
    methodName = 'LW';
```

1.3 Lumped Mass matrix

The definition of the Lumped mass matrix can be found on page 39 of the reference book, which resolve in a simple change in the terms of the mass matrix. The following piece of code is necessary.

```
for i = 1:length(M)
    abba = 0;
    for j = 1:length(M)
        abba = abba + M(i,j);
        M(i,j) = 0;
    end
    M(i,i) = abba;
end
```

1.4 Third-order Taylor Galerkin

$$\left(1 - \frac{\Delta t^2}{6}(a \cdot \nabla)^2\right)\frac{\Delta u}{\Delta t} = -(a \cdot \nabla)u^n + \frac{\Delta t}{2}(a \cdot \nabla)^2 u^n \tag{5}$$

Which translated to the weak form

$$\left(\boldsymbol{M} - \frac{\Delta t^2}{6}a^2\boldsymbol{K}\right)\frac{\Delta u}{\Delta t} = \left(-a\boldsymbol{C} - \frac{\Delta t}{2}a^2\boldsymbol{K}\right)u^n \tag{6}$$

```
case 5 % TG3
    A = M + dt*dt*a*a*K/6;
    B = -dt*a*C-1/2*dt*dt*a*a*K;
    methodName = 'TG3';
```

1.5 Computational Results

The courant number is 0.75 with the chosen parameters. The LW method, which is stable up until $C^2 < 1/3$ is unstable in this case. Using the lumped mass matrix actually raises the stability up to $C^2 < 1$, which is displayed in the following figure. The CN method is unconditionally stable, but using the lumped mass matrix means solving a different problem which resolve in a bigger error on the solution. TG3 is stable for this problem parameters as its stability is $C^2 < 1$.

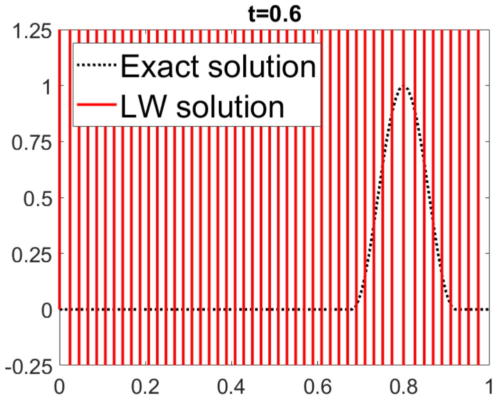

Figure 1: Lax-Wendroff

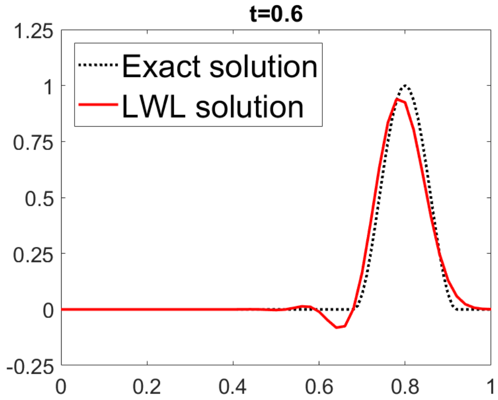

Figure 2: Lumped Lax-Wendroff

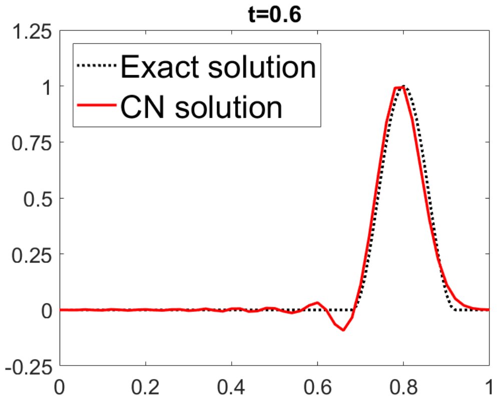

Figure 3: Crank-Nicolson

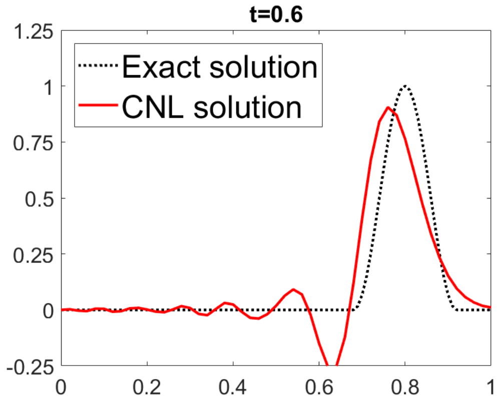

Figure 4: Lumped Crank-Nicolson

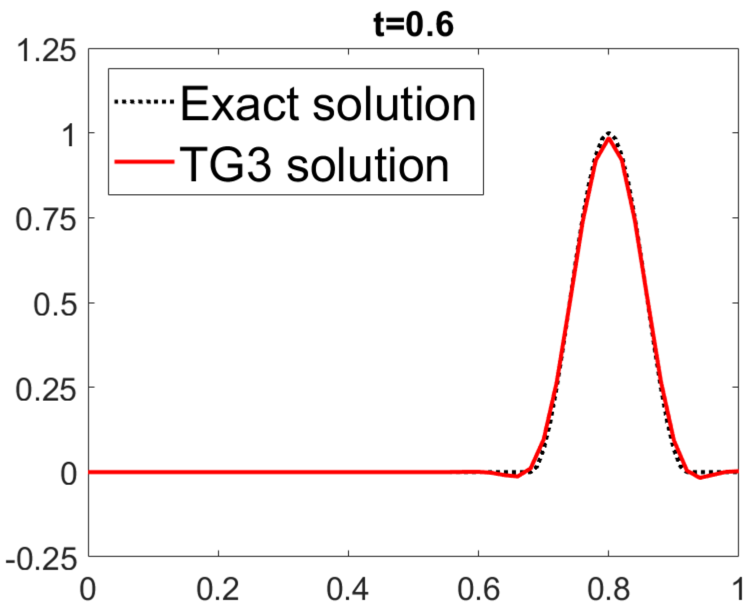

Figure 5: Taylor Galerkin 3rd order

2 Newton-Raphson method

In order to implement the Newton-Raphson method we need to build a new function that takes exactly the same arguments as for the Picard method. The soltion is build by iteration where the important components are the Jacobian matrix $\boldsymbol{J}$ which is the derivative of the f(U) function

$$f(U) = A(U)U - MU^n \tag{7}$$

$$J = \frac{df}{dU} = A(U) + \frac{\partial A}{\partial U}U \tag{8}$$

$$= A(U) + \Delta t\frac{\partial C}{\partial U}U \tag{9}$$

$$= A(U) + \Delta tC \tag{10}$$

T

```
for n = 1:nTimeSteps
    U0 = U(:,n);
    error_U = 1; k = 0;
    while (error_U > 0.5e-5) && k < 20
        C = computeConvectionMatrix(X,T,U0);
        A = M + At*C + At*E*K;
        f = A*U0 - M*U(:,n);
        df = A + C*At;
        sol = U0 - df\f;
        U1 = sol(1:m+1);
        error_U = norm(U1-U0)/norm(U1);
        fprintf('\t Iteration %d, error_U=%e\n',k,error_U);
        U0 = U1; k = k+1;
    end
    U(:,n+1) = U1;
end
```

Each time step took in average 3 iterations to compute. The solution is displayed in the following figure.

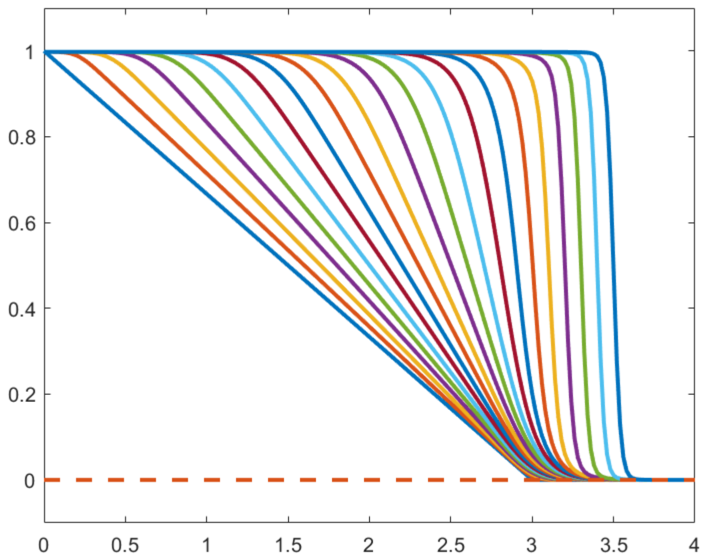

Figure 6: Newton-Raphson