Propagation of a steep front and Burger's equation

Doremus Alexandre

June 2019

Table of contents

1 Propagation of a steep front

The unsteady convection problem to be solved is the following problem :

$$\begin{cases} u_t + au_x &= 0 \quad x \in (0,1),\ t \in (0,0.6] \\ u(x,0) &= u_0(x) \quad x \in (0,1) \\ u(0,t) &= 1 \quad t \in (0,0.6] \end{cases} \tag{1}$$

$$\text{with} \quad u_0(x) = \begin{cases} 1\ if\ x \leq 0.2 \\ 0\ otherwise \end{cases}$$

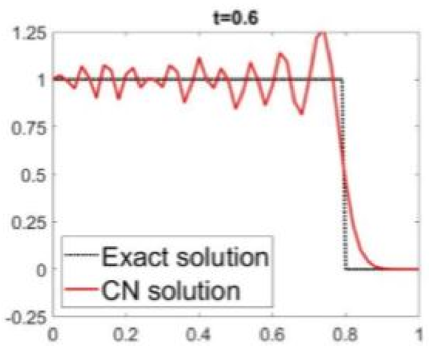
a=1, $\Delta x = 2.10^{-2}, \Delta t = 1.5.10^{-2}$
The Courant number can be obtained via the following formula :

$$C = \frac{a\Delta t}{\Delta x}$$

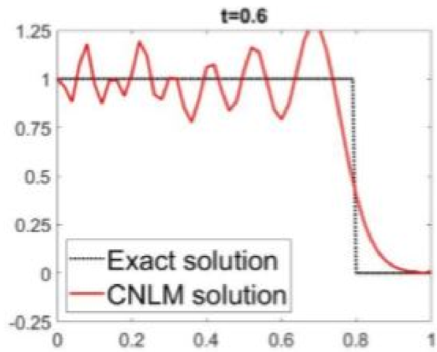
Here, we have C=0.75 and $C^2 = 0.5625$.
We will compare the Crank-Nicholson, Lax-Wendroff and third order Taylor-Galerkin scheme.
The Crank-Nicholson scheme is always stable in 1D.
TG3 is stable if $C^2 \leq 1$.
The Lax-Wendroff scheme is stable if $C^2 \leq \frac{1}{3}$.
Therefore we can expect all schemes but Lax-Wendroff's to be stable, and since TG3 has a third order precision it should give enhanced results. Let's now jump to the numerical results to see if we can confirm these conjectures.

In the code, we need the number of elements which is $\frac{1}{\Delta x}$=50 and the number of time steps which is $\frac{0.6}{\Delta t}$=40.

1.1 Crank-Nicholson



(a) Crank-Nicholson method with consistent mass matrix



(b) Crank-Nicholson method with lumped mass matrix

We can see on both cases oscillations due to the Galerkine formulation of the scheme. What's more, the lumped mass-matrix solution shows bigger oscillations, while still converging, which was to be expected as the Crank-Nicholson scheme always converges in 1D.

The lumped mass-matrix is obtained by obtaining a diagonal matrix, the diagonal terms of which are the sums of the numbers of its rows.

We can implement it in the code as follows :

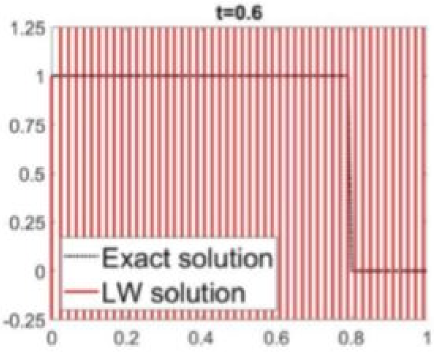
```
function ML=LumpedMatrix(M)
n=size(M)(1);
pn=size(M)(2);
ML=zeros(n,p)
for i=1:n
    Mij=0;
    for j=1:p
        Mij=Mij + M(i,j);
    end
    ML(i,j)=Mij;


end
```

(a)

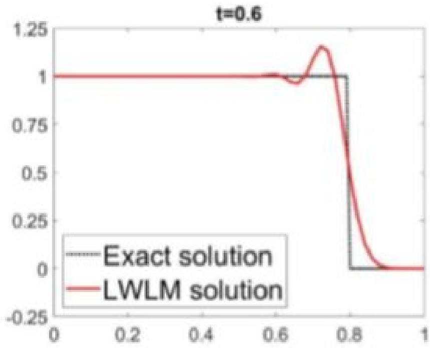
```
if method==2|| method==4
    ML=LumpedMatrix(M);
    M=ML;
end
```

(b)

1.2 Lax-Wendroff



(a) Lax-Wendroff method with consistent mass matrix



(b) Lax-Wendroff method with lumped mass matrix

As expected the Lax-Wendroff scheme performs very poorly and is not stable under normal conditions, the solution is very far from the physical one with huge oscillations. However, we can notice that by introducing lumped mass matrices we can significatively improve the scheme, as the second plot shows us that the scheme now converges, and it does so without the big oscillations inherent to the Crank-Nicholson scheme,but still with oscillations as it remains a Galerkine scheme after all.

1.3 TG3

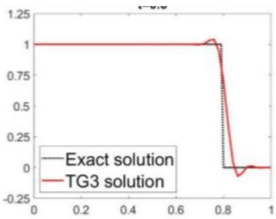


FIGURE 1 – Third order Taylor-Galerkine method

As for the TG3 method, the third order accuracy shows clear improvements on the smoothness of the solution by reducing dramatically the jump at the edges of the steep front. If we compare it with the previous second order method (Lax-Wendroff) we can see that the oscillation range is clearly reduced.

1.4 Conclusion

While all these methods show oscillations due to their Galerkine formulation of the problem, we can see that TG3 is the best of three with its third-order accuracy which helps smoothing the numerical solution. What's more, it is a stable solution. Crank-Nicholson's unconditional stability comes at the cost of a less precise method with big oscillations, even increased in the case of lumped mass matrix. On the other hand, Lax-Wendroff (TG2) is dramatically improved simply by using lumped mass matrix.

2 Burger's equation

The unsteady convection problem to be solved is the following problem :

$$\begin{cases} u_t + uu_x & = & 0 & x \in (0,4),\ t \in [0,4] \\ u(x,0) & = & u_0(x) \end{cases} \quad (2)$$

with :

$$u_0(x) = \begin{cases} \frac{1-x}{3}\ if\ x \leq 3 \\ 0\ otherwise \end{cases}$$

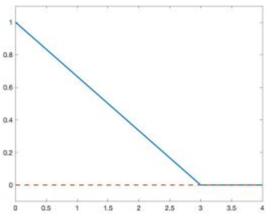


FIGURE 2 – Initial condition

We are going to solve this non-linear hyperbolic problem with a decreasing initial condition using the Newton-Raphson method. Two other methods are already coded, namely the Picards's implicit and explicit method. We will compare these three methods' results. To solve this problem numerically it is necessary to use the vanishing viscosity method which is to look for the solution of the following equation and use the solution when $\epsilon \to 0$:u$_t + uu_x = u_{xx}$.

The Galerkine discrete formulation of the equation is as follows :

$$M\frac{\Delta U}{\Delta T} + C(U)U + \epsilon KU = 0$$

.

To solve this numerically, the Newton-Raphson method consists in solving at each step :

$$f(U_{n+1}) = 0$$

where $f(U) = (M + \Delta t C(U) + \epsilon \Delta K)U - MU_n$

The iteration is computed using the Jacobian which is the main difference with the Picards's method as follows :

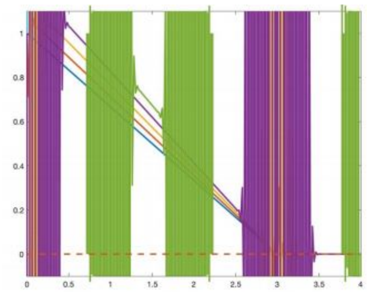
$$U_{n+1}^{k+1} = U_{n+1}^{k} - J(U_{n+1}^{k})f(U_{n+1}^{k})$$

until covergence with $U_{n+1}^0 = U_n$

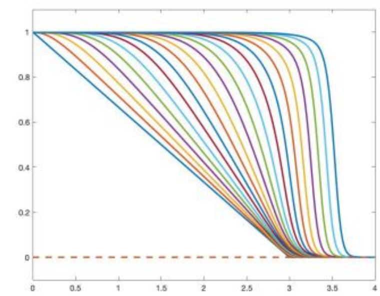
The Jacobian J is $J = \frac{df}{dU} = M + 2\Delta t C(U) + \Delta t \epsilon K$

2.1 Comparison of the three methods

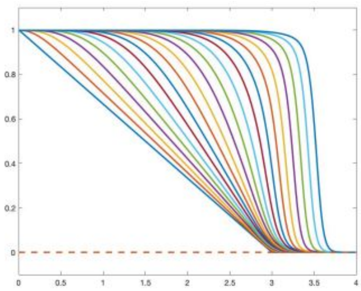
2.1.1 Results for $\epsilon = 0.1$



(a) Explicit Picard's method



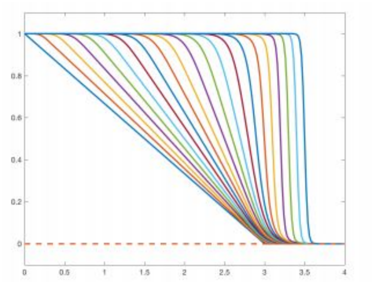
(b) Implicit Picard's method



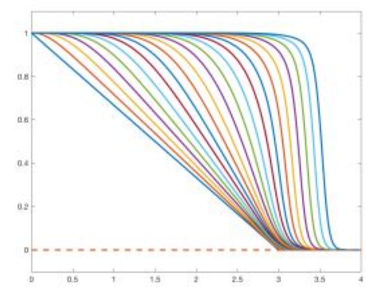
(c) Newton-Raphson's method

With a high ϵ (considering that our goal is to make it as small as possible to get as close as we can to $\epsilon = 0$) we can see that the explicit method performs very poorly with large oscillations. The implicit methods perform well and in a similar way.

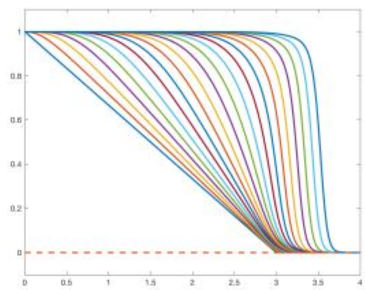
2.1.2 Results for $\epsilon = 0.01$



(a) Explicit Picard's method



(b) Implicit Picard's method



(c) Newton-Raphson's method

We can see that these three methods perform similarly with this smaller ϵ and pretty well (no big oscillations). But the explicit method seems less smooth by looking at the right upper edge of the plot.

2.1.3 Conclusion

We can see that while the three methods perform in a similar way in a specific range of parameters, in other cases the explicit method performs very poorly with big oscillations appearing, which are even different than Galerkine oscillations for $P_e > 1$, which means that in these cases it would be even worse. However this extra stability comes at the extent of computational cost which is much higher for implicit methods.
What's more, the Newton-Raphson method converges faster (quadratic convergence while Picard's convergence is linear) but in some cases if the initial guess of the algorithm is too far from the physical solution it may diverge.