HOMEWORK UNSTEADY CONVECTION

-FINITE ELEMENTS IN FLUIDS-

Marcos Boniquet

For a transient advective, non-convective, non-reactive, system:

> $\boldsymbol{u_t+au_x=0}$ $x\in(0,1)$ $t\in(0,0.6)$
> $\boldsymbol{u(x,0)=u_0(x)}$ $x\in(0,1)$
> $\boldsymbol{u(0,t)=1}$ $t\in(0,0.6)$
>
> *with initial condition:*
> $\boldsymbol{u_0(x)=1}$ *if* $x\leq 0.2$
> $\boldsymbol{u_0(x)=0}$ *otherwise*

a=1;
$h=\Delta x=2*10^{-2}$
$\Delta t=1,5*10^{-2}$

Where $\boldsymbol{A*\Delta u=(-Bu^n+f)}$ is the LINEAR SYSTEM to solve.

- Compute Courant Number.
- Solve Problem using C-N in time and linear finite element for the Galerkin scheme in space. Is the solution accurate?
- Solve problem using 2nd order Lax-Wendroff method.
  Can we expect the solution to be accurate? If not, what changes are necessary?
- Solve problem with a 3d order Taylor-Galerkin method.

COURANT NUMBER

$C=|a|*\Delta t/h=1*1{,}5*10^{-2}/2*10^{-2}=0{,}75$

$C^2=0{,}5625$

- C-N (1D) unconditional stability
- TG2 (1D) stable if $C^2<⅓$→ **Possible instabilities!**
- TG3 (1D) stable If $C^2<1$

Defining the matrices: M, K, & C:

Me = Me + w_ig*(N_ig'*N_ig)
Ke = Ke + w_ig*(Nx_ig'*Nx_ig)
Ce = Ce + w_ig*(N_ig'*Nx_ig)

We choose problem type =4, which adequates program to initial conditions of this homework.

THETA METHOD+GALERKIN with C-N

Following a time stepping discretization, trucating 2nd order term $\frac{u(t^{n+1})-u(t^n)}{\triangle t}=u_t(t^n)$.

and combining with advection formulae $u_t+(\boldsymbol{a}\cdot\boldsymbol{\nabla})u=s$,

we obtain:

$$\boxed{\begin{aligned}\Big(w,\frac{\triangle u}{\triangle t}\Big)-\theta\big(\boldsymbol{\nabla}w,\boldsymbol{a}\,\triangle u\big)+\theta\big((\boldsymbol{a}\cdot\boldsymbol{n})w,\triangle u\big)_{\Gamma^{out}}\\ =\big(\boldsymbol{\nabla}w,\boldsymbol{a}\,u^n\big)-\big((\boldsymbol{a}\cdot\boldsymbol{n})w,u^n\big)_{\Gamma^{out}}\\ +\big(w,\theta h^{n+1}+(1-\theta)h^n\big)_{\Gamma_N^{in}}+\big(w,\theta s^{n+1}+(1-\theta)s^n\big)\end{aligned}}$$

<u>recall</u>: Galerkin W=N & $\Delta u=N_j*\Delta u_j$ $U=N_jU_j$, developing a bit and considering s=0 we obtain:

A = M + 1/2*a*Δt*C;
B = -a*Δt*C;

To evaluate accuracy we compare exact and calculated solution at last step and compare each point the function 'u' calculated to the exact solution.

u_{calc}=u(:,nStep+1)'
X = u_{calc} - uu;
Dif=sum(abs(X))/nPt (SQUARED DIFFERENCES)

Mean squared error=**0.0751**

THETA METHOD+GALERKIN with C-N LUMPED MASS MATRIX

Analogously,

```
% Loop on Gauss points
  for ig = 1:ngaus
    N_ig = N(ig,:);
    Nx_ig = Nxi(ig,:)*2/h;
    w_ig = wgp(ig)*h/2;

    Me = Me + w_ig*(N_ig'*N_ig);
    Ke = Ke + w_ig*(Nx_ig'*Nx_ig);
    Ce = Ce + w_ig*(N_ig'*Nx_ig);

    MLe(1,1)=w_ig*N_ig(1);
    MLe(2,2)=w_ig*N_ig(2);
```

A = ML + 1/2*a*Δt*C;
B = -a*Δt*C;

Now the Mean squared error is **0.2677,** some accuracy has been lost with the approximation of M to M_L.

TG2 (OR L-W) +GALERKIN (2ND ORDER)

Following a time stepping discretization, trucating 3nd order terms,

$$\frac{u(t^{n+1})-u(t^n)}{\triangle t} = u_t(t^n) + \frac{1}{2}\triangle t\, u_{tt}(t^n)$$

and combining with advection formulae $u_t + (\boldsymbol{a}\cdot\boldsymbol{\nabla})u = s$,

we obtain:

$$\boxed{\begin{aligned}\left(w, \frac{\triangle u}{\triangle t}\right) &= \left(\boldsymbol{a}\cdot\boldsymbol{\nabla} w, u^n + \frac{\triangle t}{2}[s^n - (\boldsymbol{a}\cdot\boldsymbol{\nabla})u^n]\right) \\ &\quad - \left((\boldsymbol{a}\cdot\boldsymbol{n})w, u^n + \frac{\triangle t}{2}[s^n - (\boldsymbol{a}\cdot\boldsymbol{\nabla})u^n]\right)_{\Gamma^{out}} \\ &\quad + \left(w, h^{n+1/2}\right)_{\Gamma_N^{in}} + \left(w, s^n + \frac{\triangle t}{2} s_t^n\right)\end{aligned}}$$

recall: Galerkin W=N & $\Delta u = N_j * \Delta u_j$ $U = N_j U_j$, developing a bit and considering s=0 we obtain:

A = M;
B =- a*C*Δt-K*Δt^2*a^2/2;

Mean squared error makes no sense, it's **10^{18}** **NOT STABLE**

C^2=0,5625 , so it's bigger than ⅓. In order to make it a stable method whe should change Courant number, diminish it. ***This could be done by decreasing Δt (e.g.1,5*10^{-2}-->1*10^{-2}) or increasing h (e.g. 2*10^{-2}-->4*10^{-2}).*** For example doing the latter, the number of elements is 25, -the half- , and it becomes stable: 0,0542.

TG2 (OR L-W) +GALERKIN (2ND ORDER) LUMPED MASS MATRIX

Recall:

```
MLe(1,1)=w_ig*N_ig(1);
MLe(2,2)=w_ig*N_ig(2);
```

A = ML;
B =- a*C*Δt-K*Δt^2*a^2/2;

Considering the new Courant number from the beginning, Mean squared error is , **0.9128.** some accuracy has been lost with the approximation of M to M_L.

Again**,** In order to make it a stable method whe should change Courant number, diminish it.

TG3 + GALERKIN (3D ORDER)

Following a time stepping discretization, trucating 4th order terms,

$$\frac{u(t^{n+1}) - u(t^n)}{\Delta t} = u_t(t^n) + \frac{1}{2}\Delta t\, u_{tt}(t^n) + \frac{1}{6}\Delta t^2 u_{ttt}(t^n)$$

and combining with advection formulae $u_t + (\boldsymbol{a}\cdot\boldsymbol{\nabla})u = s$,

we obtain:

$$\begin{aligned}(w, \frac{\Delta u}{\Delta t}) + \boxed{\frac{\Delta t^2}{6}(\boldsymbol{a}\cdot\boldsymbol{\nabla}w, \boldsymbol{a}\cdot\boldsymbol{\nabla}\frac{\Delta u}{\Delta t})} - \boxed{\frac{\Delta t^2}{6}((\boldsymbol{a}\cdot\boldsymbol{n})w, \boldsymbol{a}\cdot\boldsymbol{\nabla}\frac{\Delta u}{\Delta t})_{\Gamma^{out}}} \\ = (\boldsymbol{a}\cdot\boldsymbol{\nabla}w, u^n - \frac{\Delta t}{2}(\boldsymbol{a}\cdot\boldsymbol{\nabla}u^n)) - ((\boldsymbol{a}\cdot\boldsymbol{n})w, u^n - \frac{\Delta t}{2}(\boldsymbol{a}\cdot\boldsymbol{\nabla}u^n))_{\Gamma^{out}} \\ + \frac{\Delta t}{2}(\boldsymbol{a}\cdot\boldsymbol{\nabla}w, s^{n+1/3}) - \frac{\Delta t}{2}((\boldsymbol{a}\cdot\boldsymbol{n})w, s^{n+1/3}) \\ + (w, \frac{3}{4}s^{n+2/3} + \frac{1}{4}s^n) + (w, \frac{3}{4}h^{n+2/3} + \frac{1}{4}h^n)_{\Gamma_N^{in}}\end{aligned}$$

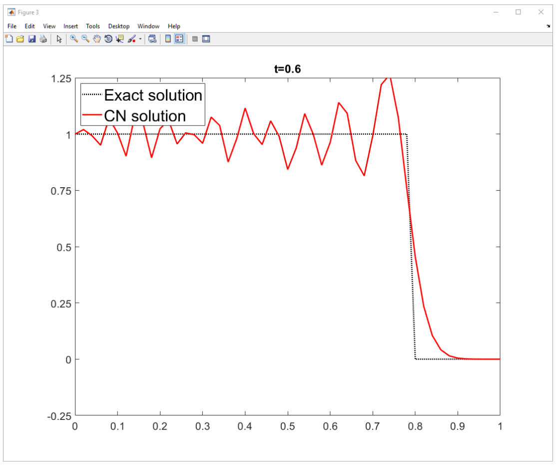
<u>recall</u>: Galerkin W=N & $\Delta u=N_j{}^*\Delta u_j$ $U=N_jU_j$, developing a bit and considering s=0 we obtain:

$$\mathbf{A = M + \Delta t^2{*}a^2{*}K/6;}$$
$$\mathbf{B = -a{*}C{*}\Delta t+\Delta t^2{*}a^2{*}K/2;}$$

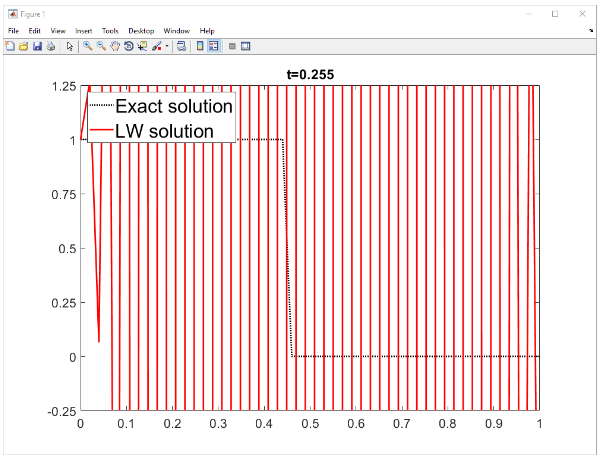
Mean squared error is **0.0265**.

<u>RESULTS</u>

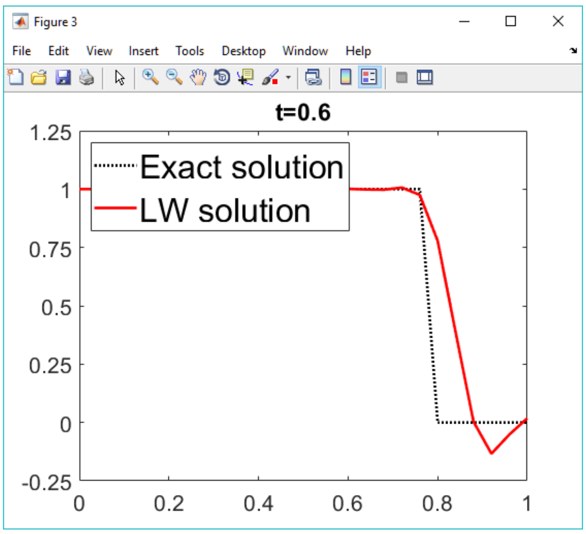
Lumped matrix comparison and change of Courant for L-W case in or to stabilize it. TW3 appears to be the most precise method.

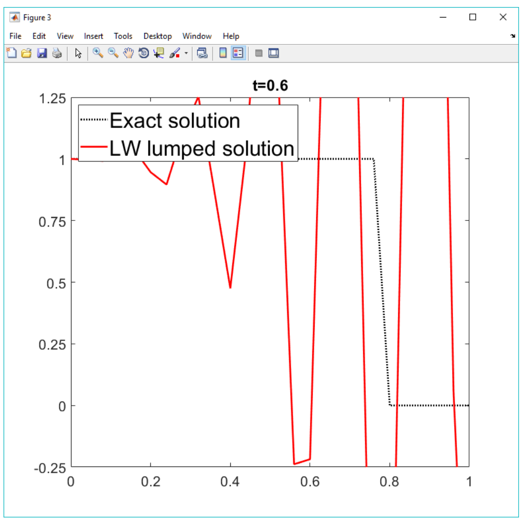


CN



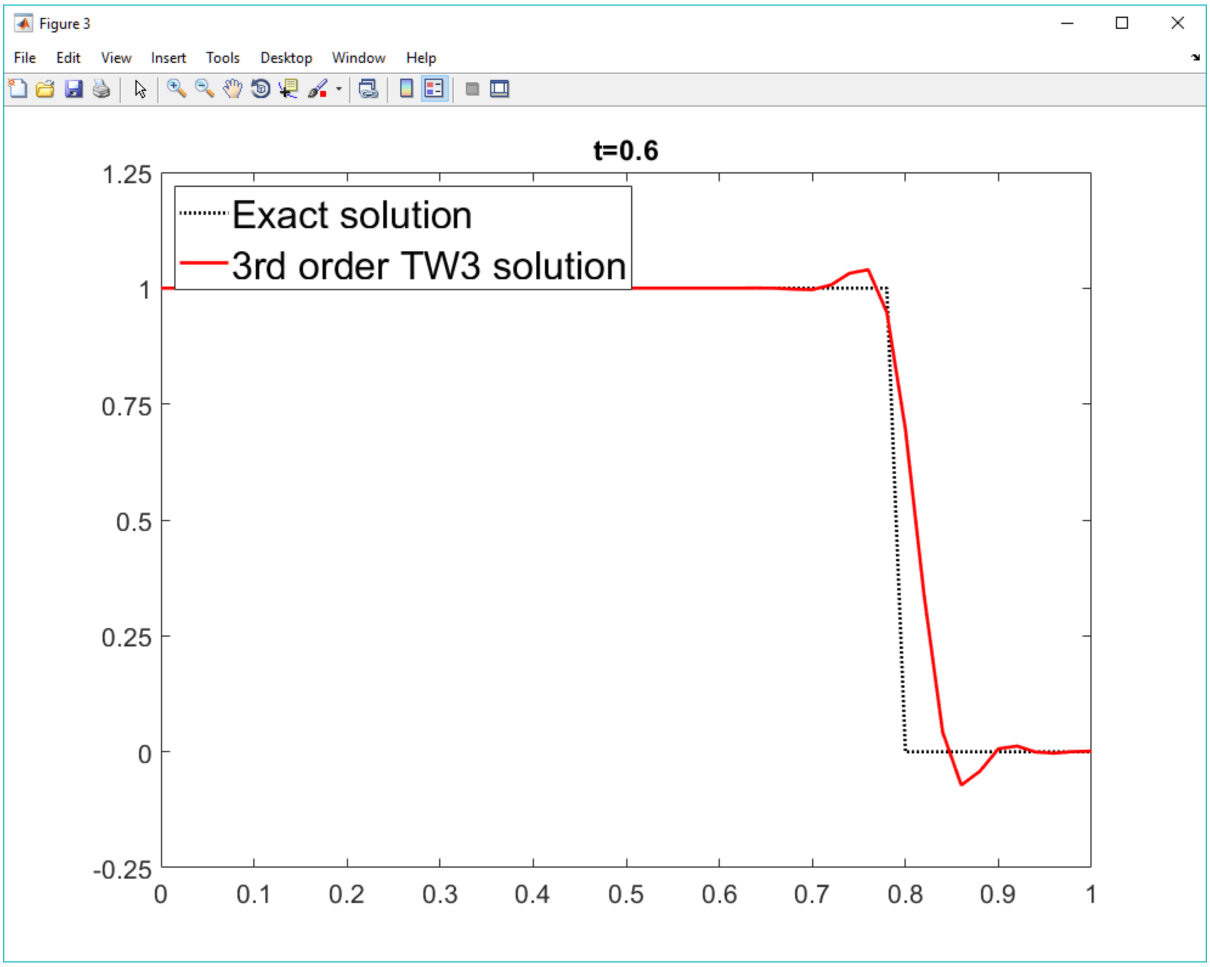
LW WITH COURANT>1/3





LW WITH COURANT<⅓

LW LUMPED MASS MATRIX WITH COURANT<⅓



TW3