

Discontinuous Galerkin (DG)

for linear advection 1D

In these short notes I explain how to adopt the Finite Element Method to hyperbolic equations by adding upwinding / Riemann Solvers to classical FEM. This is based on extensive notes by Sandra May.

I will not discuss elliptic/parabolic problems since this is not (IMO) a strength of DG methods, and classical FEM works better.

So we focus on simple advection

$$u_t + c u_x = 0 \quad \text{in 1D periodic domain.} \quad (1)$$

Discontinuous Galerkin (DG)

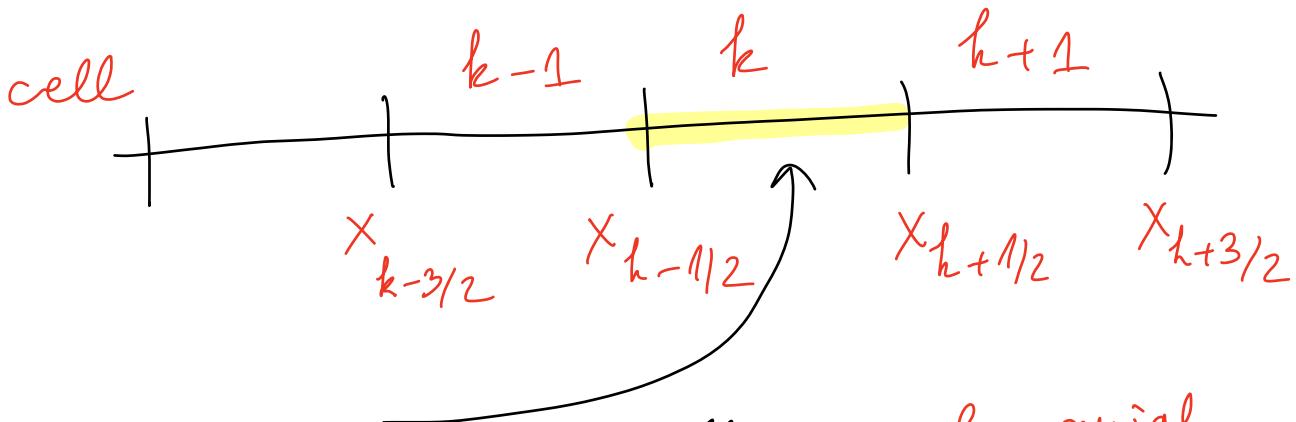
can be seen in two ways:

- 1) A generalization of FEM to allow discontinuous solutions. The main advantage is computational: cells become weakly coupled so parallel execution improves markedly.
- 2) A "better" way to look at DG is as

$$\text{FV} + \text{FE} = \text{DG}$$

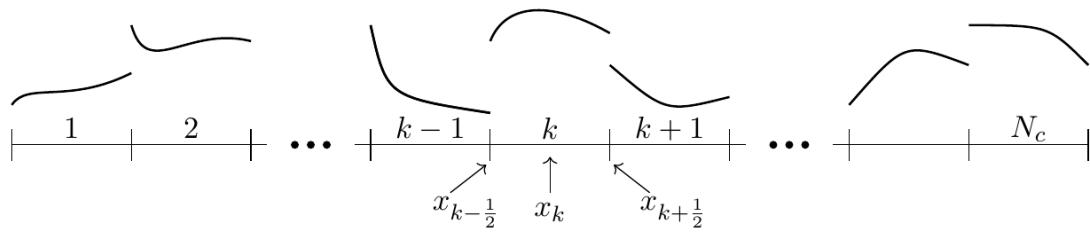
Recall that to get higher-order accuracy in FV via MOL we used higher-order reconstruction + Riemann solver. In DG, the reconstruction is local to each FV cell.

(2)

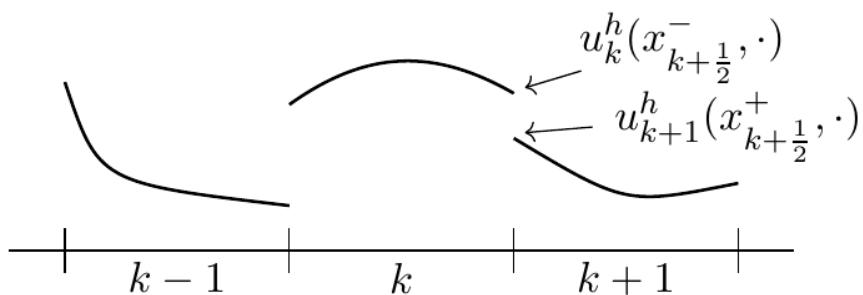
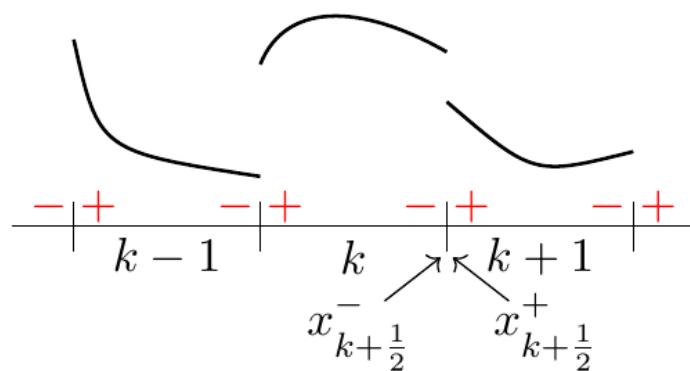


Solution on each cell = polynomial
 of degree N_p (instead of a
 scalar cell average as in FV),
 just like in FEM. But here
 we do not impose any continuity
 between cells in the polynomials
 (we will impose it weakly via
 the r.h.s. of the linear system).
 This is the same as polynomial
 reconstruction as we did in
 FV+MOL (e.g. third order upwind
 biased was quadratic reconstruction)

Element of new "DG" function
space \mathcal{V}^{N_p} :



(image from Sandra May)



At each face, we have a value from the left and a value from the right, just as in FV

(4)

If we use a Riemann solver at face $k+1/2$ we can get a flux - just upwinding for linear advection as we saw in FV. What to do with this flux?

Go back to FEM machinery & write weak form in \mathcal{V}^{N_p} :

Take test function $\psi \in \mathcal{V}^h$ and integrate over cell $I_h = [x_{k-1/2}, x_{k+1/2}]$

$$\int_{x_{k-1/2}}^{x_{k+1/2}} (u_t + f_x) \psi^h dx = 0$$

f_x ^{advection flux}

$$f(u) = c u$$

(5)

Weak form :

$$\int_{I_k} (\partial_t u) \varphi^h dx - \int_{I_k} c u \varphi_x^h dx$$

$$= - \left[f \varphi^h \right]_{x_{k-1/2}}^{x_{k+1/2}}$$

To define φ^h at the face,

$$\begin{cases} \varphi^h(x_{k-1/2}) = \varphi(x_{k-1/2}^+) \\ \varphi^h(x_{k+1/2}) = \varphi(x_{k+1/2}^-) \end{cases}$$

To define flux f at face,
use a Riemann solver / upwind.

In DG parlance this is done
via a flux function H .

(6)

To discretize weak form, set

$$u(x \in I_k, t) = u_k^h(x, t) = \sum_{i=1}^{N_f = N_p + 1} u_i^k(t) \psi_i^h(x)$$

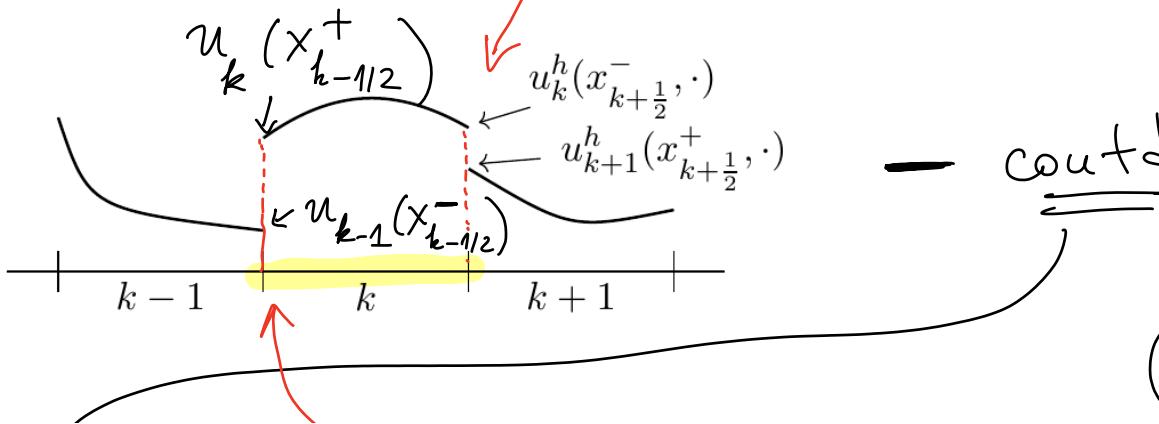
time dependent coefficients / unknowns

polynomial basis functions

$$\begin{bmatrix} f & \psi^h \end{bmatrix}_{x_{k-1/2}}^{x_{k+1/2}} = \begin{bmatrix} H & \psi^h \end{bmatrix}_{\text{notation}} =$$

$$\psi^h(x_{k+1/2}^-) H(u_k^h(x_{k+1/2}^-), u_{k+1}^h(x_{k+1/2}^+))$$

Riemann solver



(7)

$$-\psi^h(x_{k-1/2}^+) H(u_{k-1}^h(x_{k-1/2}^+), u_k^h(x_{k-1/2}^-))$$

For simple advection, the Riemann solver is just simple upwinding

$$H(a, b) = \begin{cases} ca, & c \geq 0 \\ cb, & c < 0 \end{cases}$$

We would get a consistent but not stable scheme if we use a centered flux

$$H(a, b) \cancel{=} \frac{c}{2}(a+b)$$

This is really all the same as FV except we now have a test function as well

⑧

As in FEM, now we simply take ψ^h to be the basis functions, separately in each cell / element, and we get

a linear system similar to FEM (for time-dependent problems).

Discrete weak form: Find

$$u_k^h(x, t) = \sum_{j=1}^{N_f} \hat{u}_j^k(t) \psi_j^k(x)$$

s.t. $\forall \psi_i^k, i = 1, \dots, N_f$, forall cells k

$$\int_{I_k} (\partial_t u_k^h) \psi_i^k dx - \int_{I_k} c u_k^h \frac{d \psi_i^k}{dx} dx$$

$$= - \left[H(u) \psi_i^k \right]_{k-1/2}^{k+1/2}$$

flux function

(9)

In matrix notation:

$$\underbrace{\underline{\underline{M}}^k}_{\text{cell / element mass matrix}} \frac{d\hat{\underline{u}}^h_k(t)}{dt} = \underbrace{\underline{\underline{S}}^k}_{\text{cell stiffness matrix}} (\underline{c}\hat{\underline{u}}^h_k(t)) - \underbrace{\left[\underline{H}(\hat{\underline{u}}^h) \underline{\underline{\Psi}}^k \right]}_{\substack{\text{flux} \\ \text{coupling} \\ \text{between} \\ \text{cells}}}$$

cells/elements are NOT coupled through the mass and stiffness matrices, unlike continuous FEM where these were neighboring elements.

This makes each time step in DG much easier to execute in parallel & more efficient than traditional FEM! And, higher order accuracy is "easy" to achieve unlike in FV.

For completeness, recall

$$\left\{ \begin{array}{l} M_{ij}^k = (\psi_j^k, \psi_i^k)_{I_k} \\ S_{ij}^k = (\psi_j^k, \frac{d\psi_i^k}{dx})_{I_k} \end{array} \right.$$

similar to FEM but now integrals are over each element/cell.

The coupling between cells is only through the face /flux term:

Upwind: $c > 0$

$$[H\psi_i^k]_{k+1/2} = c \underbrace{\left(\sum_{j=1}^{N_f} u_j^k(t) \psi_j^k(x_{k+1/2}^-) \right)}_{\text{only } \psi^k \text{ involved}} \psi_i^k(x_{k+1/2}^-)$$

$$[H\psi_i^k]_{k-1/2} = c \underbrace{\left(\sum_{j=1}^{N_f} u_j^{k-1}(t) \psi_j^{k-1}(x_{k-1/2}^-) \right)}_{\text{couples cell } k \text{ to } k-1} \psi_i^k(x_{k-1/2}^+)$$

(11)

To turn this into an actual scheme we need to:

- a) choose basis functions (polynomials)
- b) compute mass & stiffness local matrices (integrals)
- c) choose an explicit time integrator like RK2 / RK3

While a complete stability theory is lacking, experience says explicit stability condition

$$\Delta t \leq \frac{1}{2N_p + 1} \frac{\Delta x}{|c|}$$

the problem with DG :

requires small time step sizes for higher order, as if we used a subcell grid

Typical convergence result as

$\Delta t \rightarrow 0$:

$$\|u(t) - u_h(t)\|_{L^2} \leq C h^{N_p+1}$$

only L^2
not H^1

same as FEM

Key advantage: Can generalize to unstructured triangular grids in 2d relatively straight forwardly.

Aside:

Define jump across face:

$$[u^h]_{k-1/2} = u(x_{k-1/2}^-) - u(x_{k-1/2}^+)$$

Weak form weakly imposes continuity!

$$\sum_{k=1}^{N_c} \int_{I_k} R^h \varphi^h dx = \sum_{k=1}^{N_c} c [u^h]_{k-1/2} \varphi^h(x_{k-1/2}^+)$$

I_k weak residual "weak" continuity

$$R^h = \partial_t u^h + c u^h_x = \text{residual}$$

(13)