Implementation of CENO3

Norma Meeting

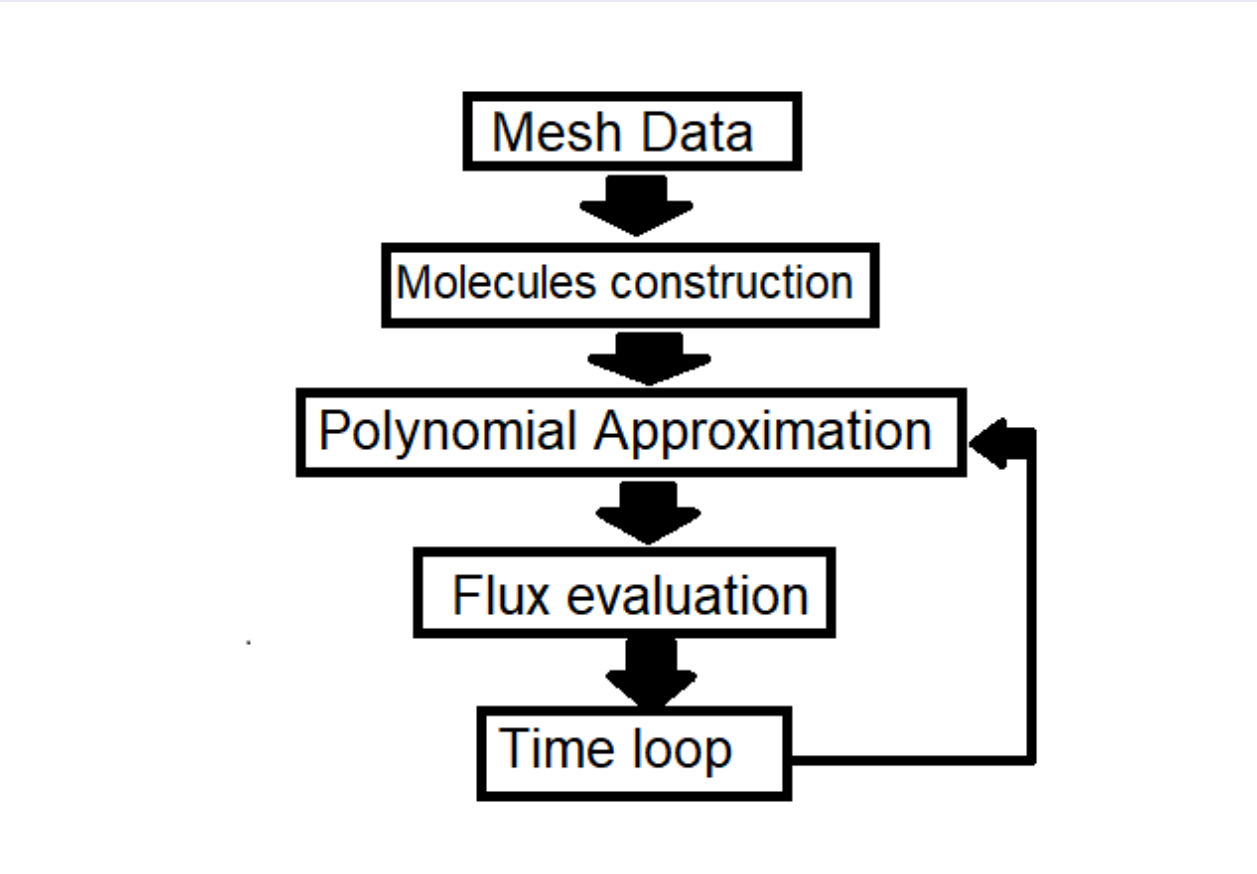
Matthieu Gschwend and Alain Dervieux

09/02/2021

Contents

Algorithm structure

We implement our algorithm though the following steps



Libraries used

-Viennagrid : http://viennagrid.sourceforge.net
-Eigen: https://eigen.tuxfamily.org

Reconstruction analysis

Least square approximation

The CENO3 scheme requires a molecule construction for every cell.
Flux evaluation requires a polynomial reconstruction over every molecule.
For a given cell i we note M the molecule such that $C_i \subset M$.
Then the polynomial aproximation P_i^n is defined as :

$$argmin \sum_{C_{k \neq i} \subset M} \left(\bar{P}_i^{k,n} - \bar{u}^{k,n} \right)^2$$

This problem leads to a linear system of equation :

$$A_i c_i^n = b_i^n$$

Where c_i^n represents the polynomial coefficients.

Verification

As a quadratic reconstruction is performed, we focus on the exactness for monomials :

$$x,\ y\ ,z\ ,x^2\ ,y^2,\ z^2,\ xy,\ xz\ ,yz$$

as well as the constant term fixed by $\bar{P}_i^n(0,0,0) = \bar{u}^{i,n}$.
Verification process : for every cell we compare the expected coefficient with the computed one

$$\Delta c_i = \max_{1 \leq l \leq 9} (|c_{i,l}^{expected} - c_{i,l}^{compuded}|)$$

We count the number of failing polynomials. For a given tolerance $\tau > 0$, the reconstruction fail when $\Delta c_i > \tau$.

monomial	x	xy	xz	x^2
N = 18	130	130	130	130
N=33	0	0	0	0

Table : Number of failing polynomials for $\tau = 10^{-6}$

It seems that when a molecule has an insufficient number of cells, some reconstructions can be wrong.
In certain cases some coefficients $c_{i,l}^{computed}$ explode

monomial	x	xy	xz	x^2
Δc_i max for N=18	$>10^6$	$>10^8$	$>10^7$	$>10^8$

Table: Δc_i max

In order to fix the problem we add a stabilization term $\varepsilon>0$ in our linear system, this leads to solve

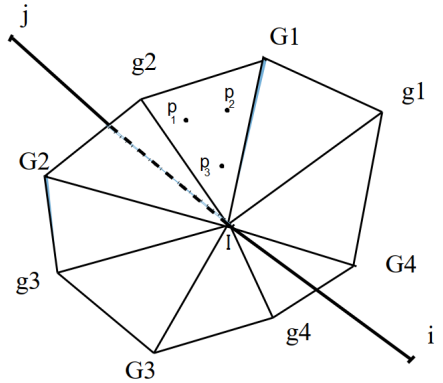
$$(A_i + \varepsilon Id)c_i^n = b_i^n$$

monomial	x	xy	xz	x^2
N=18	70	105	102	78
N=33	0	0	0	0
Δc_i max for N=18	0.1	0.2	0.4	1.1

Table: Number of failing polynomials for $\tau = 10^{-6}$

Flux evaluation

The interface between two median cells $\partial C_i \cap \partial C_j$ can be split in several triangular faces $R_{i,j}$. Flux is computed using polynomials evaluations at three Gauss integration points $p^l_{\{i,j,\}}$.



$$\int_{\partial C_i \cap \partial C_j} F(u(x,y,t)) \cdot \mathbf{n} ds =$$
$$\sum_{1 \leq r \leq R_{i,k}} \sum_{l \in [1,2,3]} \Phi\left(P_i\left(p^l_{\{i,k,r\}}, t\right), P_k\left(p^l_{\{i,k,r\}}, t\right), \mathbf{n}_{\{i,k,r\}}\right)$$

Result

Case : pure gaussian advection

$$\frac{\partial u}{\partial t}(x,y,z,t) + \nabla \cdot (Vu(x,y,z,t)) = 0. \text{ with } V = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

initial and boundaries conditions are given by

$$\begin{cases} u(x,y,z,0) &= e^{-20((x-0.5)^2+(y-0.5)^2+(z-0.5)^2)} \\ u(x,y,z,t) &= 0 \quad \text{for } (x,y,z) \in \partial\Omega \end{cases}$$

Donner cell Riemann solver is used.

Mesh description

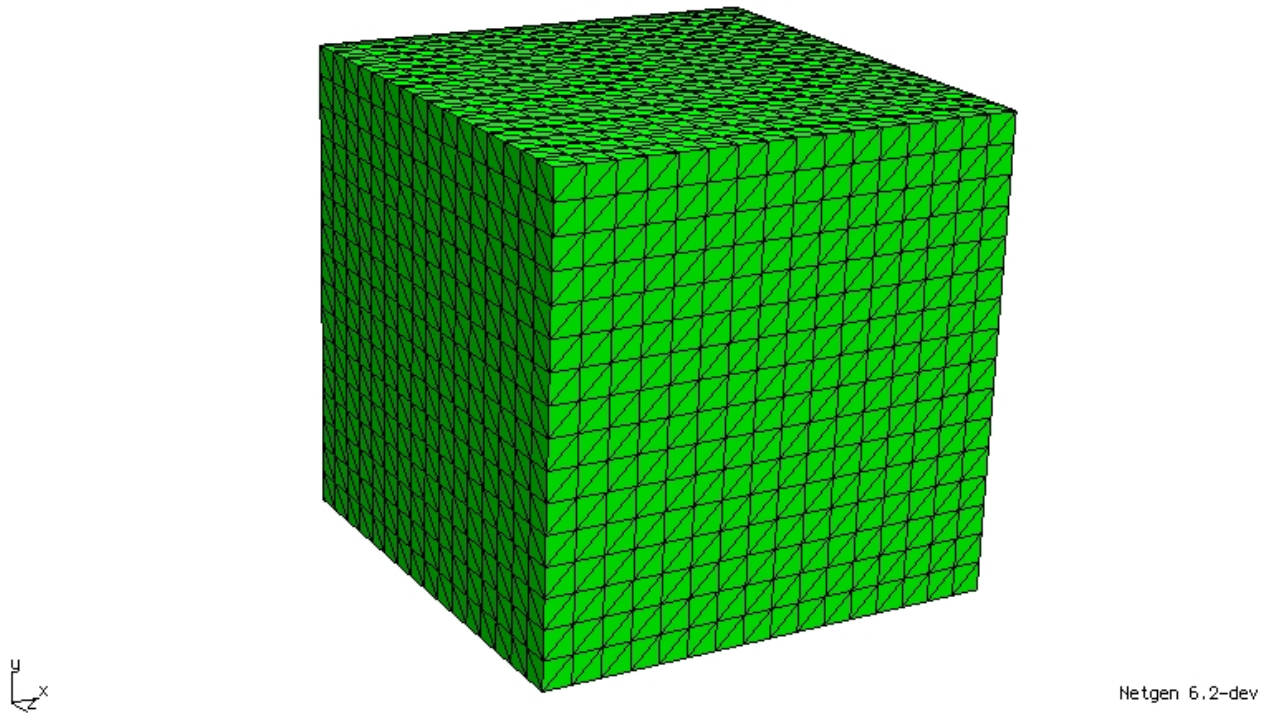
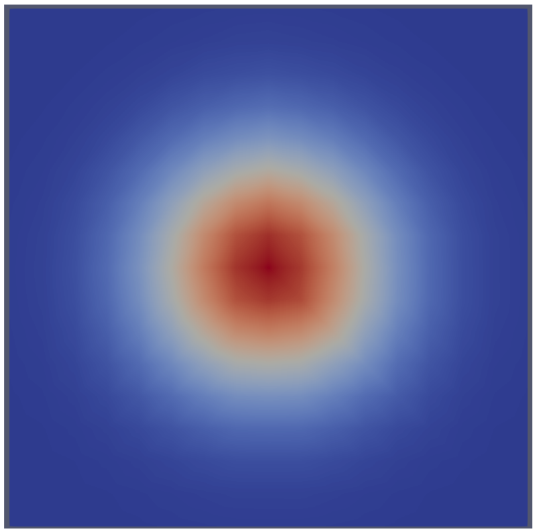


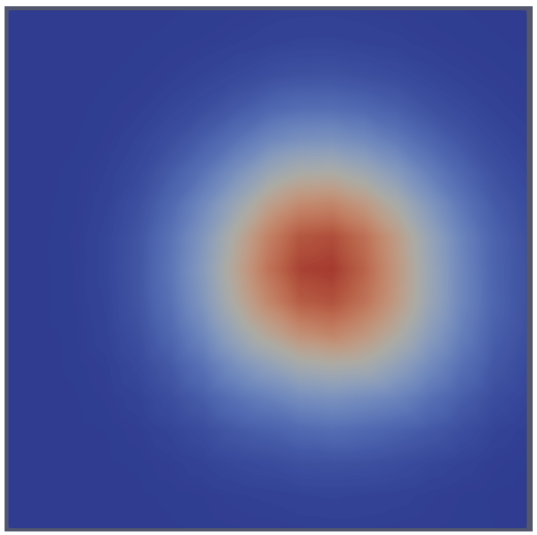
Figure: Unit cube

Structured mesh : 4913 points 24576 elements. Generated with Netgen : https://ngsolve.org

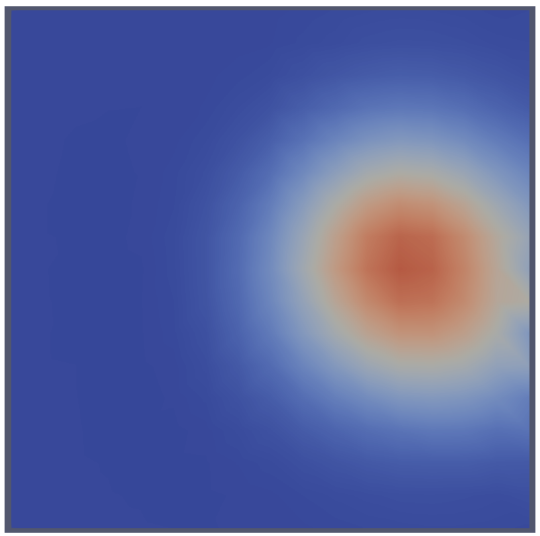
Solution visualization



Initial solution at t = 0



Solution at t = 0.1125



Solution at t = 0.2812

Conclusion

- Currently done.

1. Scalar pure scalar advection case.
2. Paraview visualisation

- Next.

1. Integration in ANANAS at Lemma.
2. Extension to compressible flow (1 month).
3. Beginning of validation on cylinder (beginning of July).