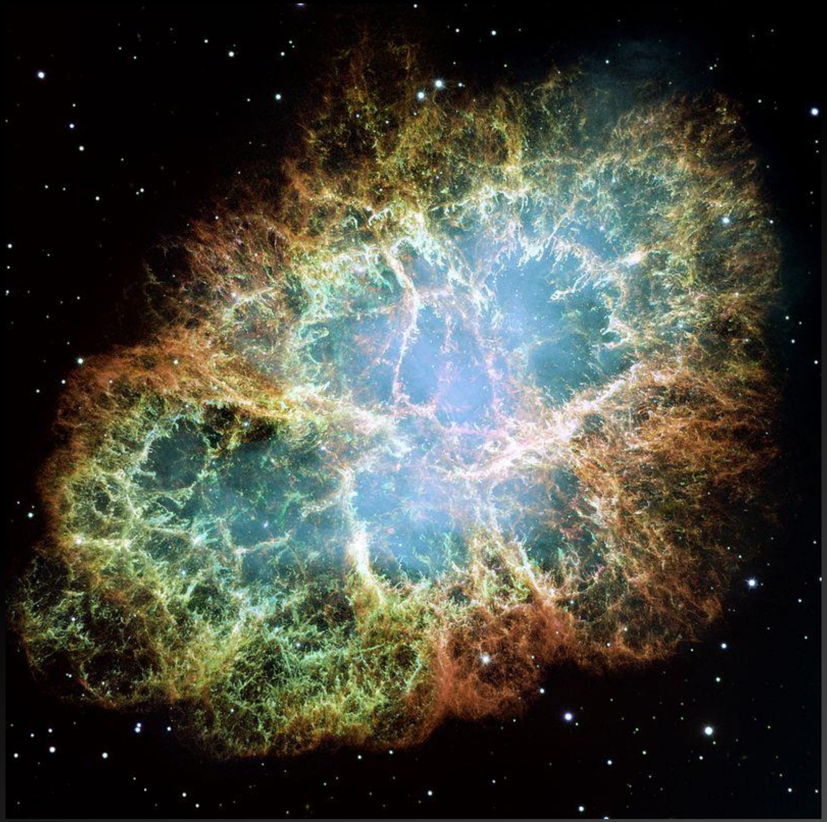
Maxime Rey
Supervisé par Jeremy Blaizot & Joakim Rosdahl

Développement de code hydrodynamique 2D et application à la modélisation d’une supernova



Intérêts

- Simulations numériques

Equations sans solution analytique
Conditions extrêmes
Tester/Valider des théories

- Supernovæ

Milieu Interstellaire (ISM)
Taux de Formation Stellaire (SFR)

Intérêts

- Simulation de supernova

  → solutions analytiques

Comparaison !

- Pédagogique

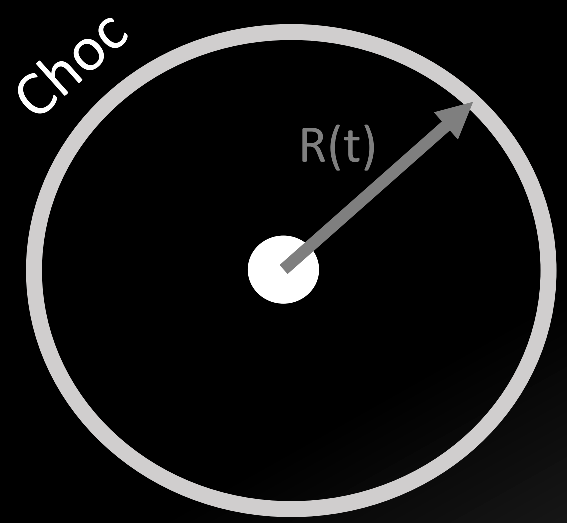
technique très utilisée

La solution de Sedov-Taylor

Hypothèses:

_ Adiabatique
_ Symétrie sphérique
_ Injection d’énergie

$$R(t) = \left(\frac{25E_{SN}}{4\pi\rho_0}\right)^{\frac{1}{5}} t^{\frac{2}{5}}$$



$$\boxed{v(t) = \left(\frac{25E_{SN}}{4\pi\rho_0}\right)^{\frac{1}{5}} t^{\frac{-3}{5}}}$$

Equations

- Conservation de la masse: $$\frac{\partial \rho}{\partial t} + \vec{\nabla}(\rho\vec{u}) = 0$$
- Conservation de l'impulsion : $$\frac{\partial(\rho\vec{u})}{\partial t} + \rho\left(\vec{u}.\vec{\nabla}\right)\vec{u} = -\vec{\nabla}(P) + \cancel{\rho\vec{g}}$$
- Conservation de l'énergie: $$\frac{\partial(\rho E)}{\partial t} + \vec{\nabla}(\rho E\vec{u}) = -\vec{\nabla}(P\vec{u}) + \cancel{Cooling}$$
- Equation d'état: $$P = (\gamma - 1)\rho E_{th} = (\gamma - 1)\rho(E - E_{kin})$$

Passage en 2D

2D → $\vec{u} = (u, v)$

$$
\begin{cases}
\dfrac{\partial \rho}{\partial t} + \dfrac{\partial \rho u}{\partial x} + \dfrac{\partial \rho v}{\partial y} = 0 \\
\dfrac{\partial \rho u}{\partial t} + \dfrac{\partial (\rho u^2 + P)}{\partial x} + \dfrac{\partial \rho v u}{\partial y} = 0 \\
\dfrac{\partial \rho v}{\partial t} + \dfrac{\partial \rho u v}{\partial x} + \dfrac{\partial (\rho v^2 + P)}{\partial y} = 0 \\
\dfrac{\partial \rho E}{\partial t} + \dfrac{\partial u(\rho E + P)}{\partial x} + \dfrac{\partial v(\rho E + P)}{\partial y} = 0
\end{cases}
$$

$$\boxed{\frac{\partial U}{\partial t} + \frac{\partial F_x(U)}{\partial x} + \frac{\partial F_y(U)}{\partial y} = 0}$$

Passage en 2D

$$\frac{\partial U}{\partial t} + \frac{\partial F_x(U)}{\partial x} + \frac{\partial F_y(U)}{\partial y} = 0$$

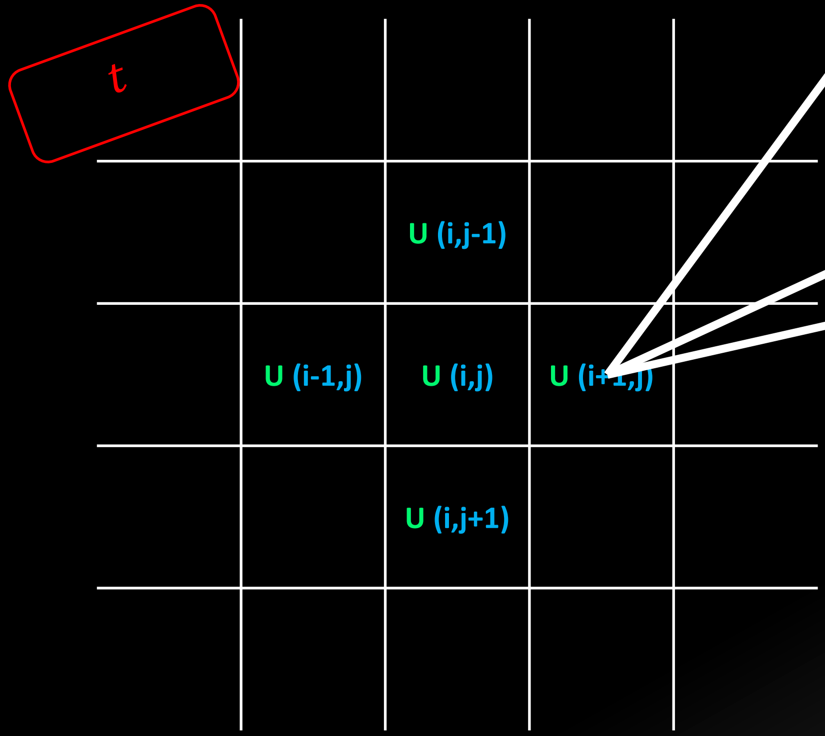
$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}$$

$$F_x(U) = \begin{pmatrix} \rho u \\ \rho u^2 + P \\ \rho u v \\ \rho E + P \end{pmatrix}$$

$$F_y(U) = \begin{pmatrix} \rho v \\ \rho v u \\ \rho v^2 + P \\ \rho E + P \end{pmatrix}$$

Code en grille: Approche Eulérienne

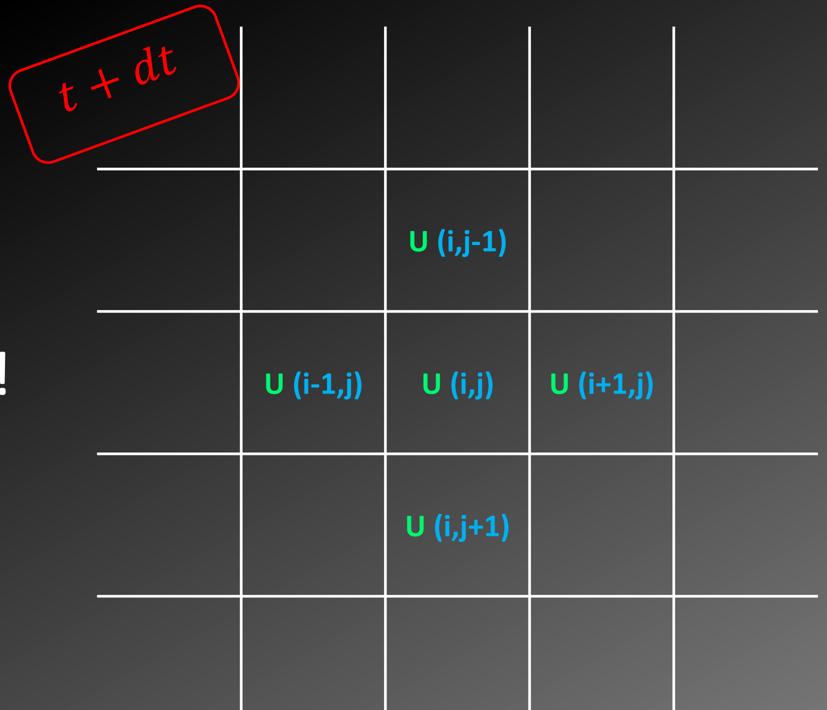
Discrétisation de l'espace…



ρ (i+1,j)
E (i+1,j)
u (i+1,j)
v (i+1,j)

Evolution de
l'état U
des cellule

…et du temps !



Méthode des volumes finis

$$\frac{\partial U}{\partial t} + \frac{\partial F_x(U)}{\partial x} + \frac{\partial F_y(U)}{\partial y} = 0$$

$\iint cellule$

+

Discrétisation spatiale

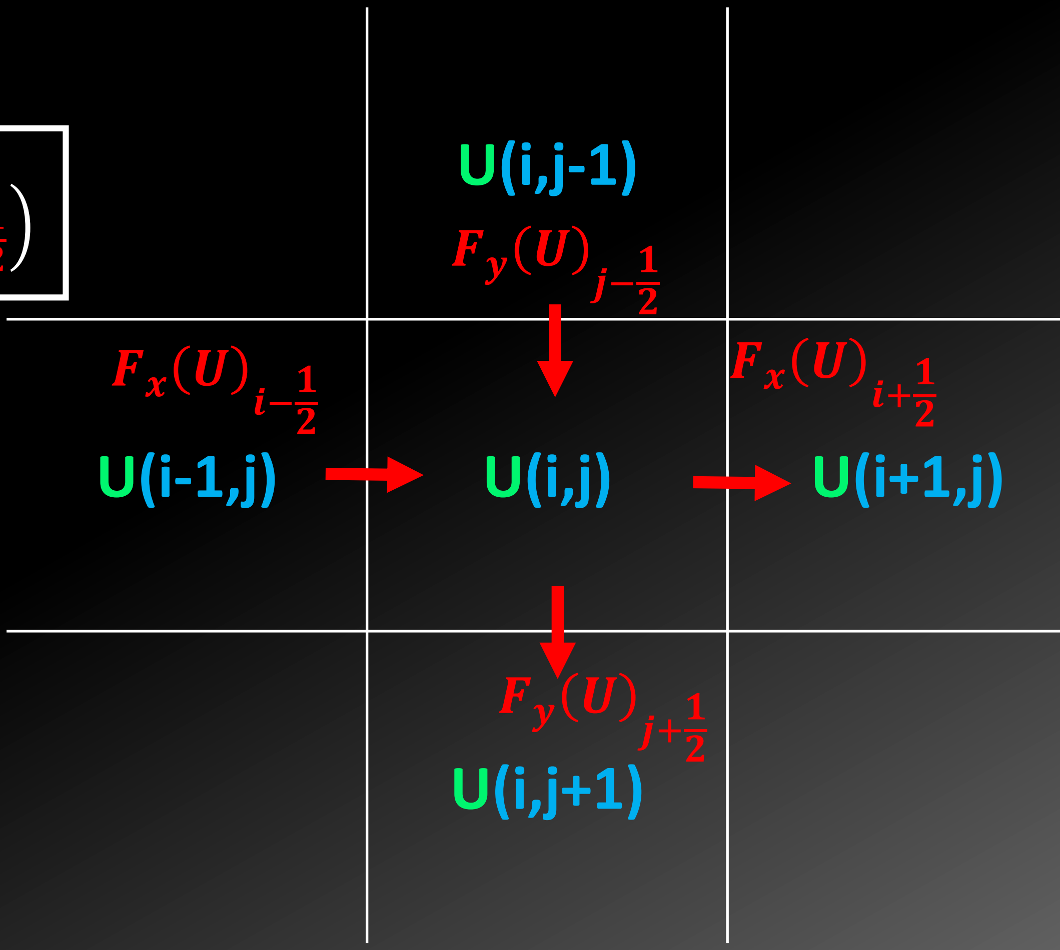
$$\frac{\partial U}{\partial t} = \frac{1}{\Delta x}\left(F_x(U)_{i+\frac{1}{2}} - F_x(U)_{i-\frac{1}{2}} + F_y(U)_{j+\frac{1}{2}} - F_y(U)_{j-\frac{1}{2}}\right)$$

Discrétisation temporelle

$$\boxed{U^{n+1} = U^n - \frac{\Delta t}{\Delta x}\left(F_x(U)_{i+\frac{1}{2}} - F_x(U)_{i-\frac{1}{2}} + F_y(U)_{j+\frac{1}{2}} - F_y(U)_{j-\frac{1}{2}}\right)}$$

Flux entre les cellules

$$\boxed{U^{n+1} = U^n - \frac{\Delta t}{\Delta x}\underbrace{\left(F_x(U)_{i+\frac{1}{2}} - F_x(U)_{i-\frac{1}{2}} + F_y(U)_{j+\frac{1}{2}} - F_y(U)_{j-\frac{1}{2}}\right)}_{?}}$$



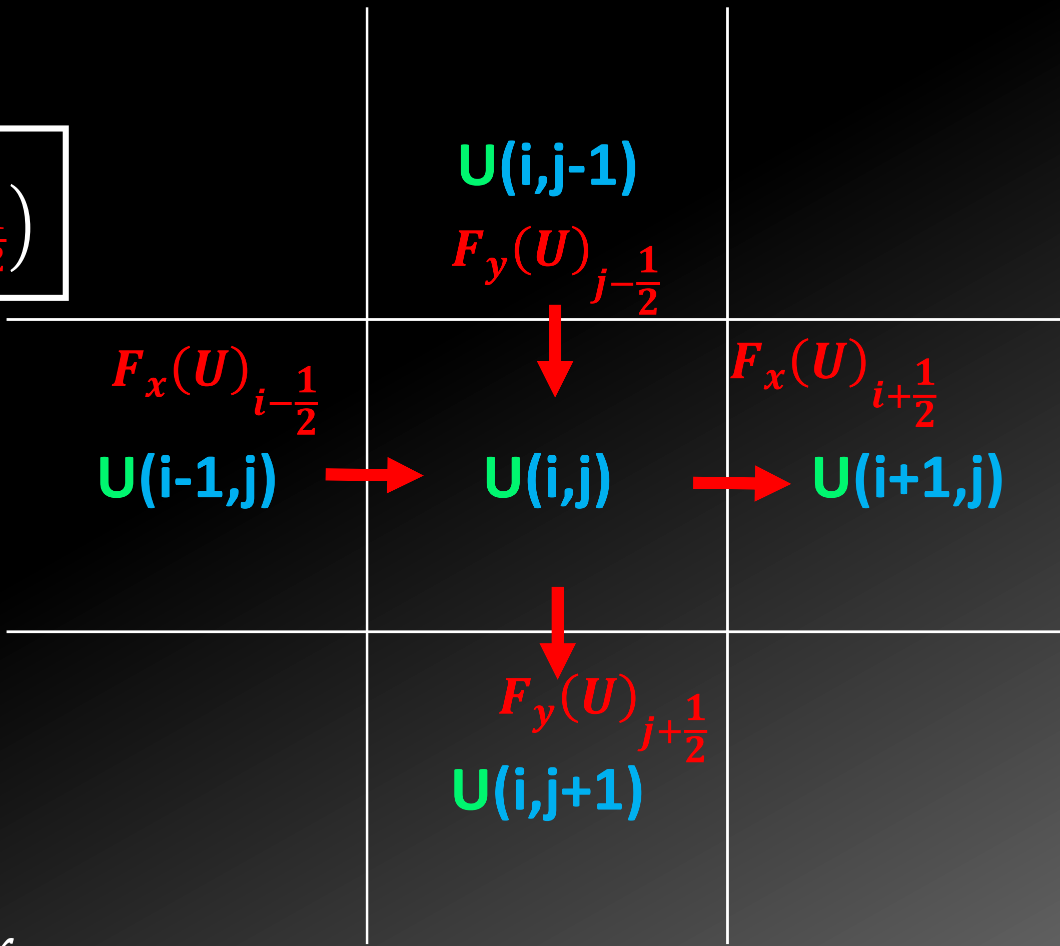
- Méthode “naïve ”

$$F_x(U)_{i+\frac{1}{2}} = \frac{1}{2}\left(F_x(U)_i + F_x(U)_{i+1}\right)$$

Instable !

Flux entre les cellules

$$\boxed{U^{n+1} = U^{n} - \frac{\Delta t}{\Delta x}\underbrace{\left(F_x(U)_{i+\frac{1}{2}} - F_x(U)_{i-\frac{1}{2}} + F_y(U)_{j+\frac{1}{2}} - F_y(U)_{j-\frac{1}{2}}\right)}_{?}}$$

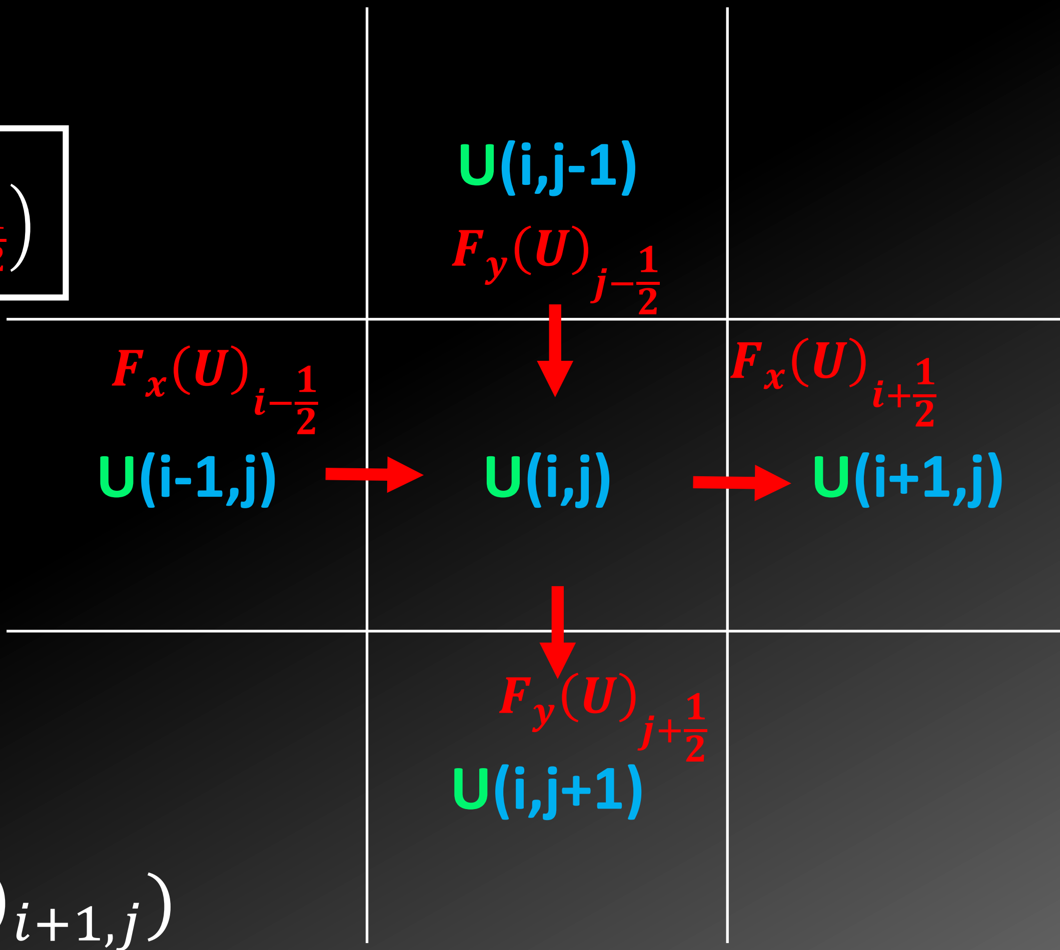


- Méthode de Lax-Friedrichs (1er ordre)

$$F_x(U)_{i+\frac{1}{2}} = \frac{1}{2}\left(F_x(U)_i + F_x(U)_{i+1}\right) - \frac{\Delta x}{2\Delta t}\left((U)_i - (U)_{i+1}\right)$$

Flux entre les cellules

$$\boxed{U^{n+1} = U^n - \frac{\Delta t}{\Delta x}\underbrace{\left(F_x(U)_{i+\frac{1}{2}} - F_x(U)_{i-\frac{1}{2}} + F_y(U)_{j+\frac{1}{2}} - F_y(U)_{j-\frac{1}{2}}\right)}_{?}}$$



- Méthode de Rusanov (Lax-Friedrichs locale)

$$C_{max} = \max\left((|u| + C_s)_{i,j} \,;\, (|u| + C_s)_{i+1,j}\right)$$

$$F_x(U)_{i+\frac{1}{2}} = \frac{1}{2}\left(F_x(U)_i + F_x(U)_{i+1}\right) - \frac{C_{max}}{2}\left((U)_i - (U)_{i+1}\right)$$

Initialisations

Paramètres physiques:
- Taille de la simulation: 30×30 pc²
- Energie de la supernova: 10^{30} erg. cm^{-1}
- Indice adiabatique: $\gamma = \frac{5}{3}$

$\mu = 1$
$T = 10^2 K$
$k_B = 10^{-16}$erg.K^{-1}
$m_H = 1{,}67 \times 10^{-24}$ g

Initialisations:
- Densité initiale: $1{,}67 \times 10^{-24}$g.cm^{-3} → $1\ atoms.$cm^{-3}
- Energie spécifique initiale: 6×10^{9}erg.g^{-1}

$$P = (\gamma - 1)\rho E_{th} = (\gamma - 1)\rho(E - \cancel{E_{kin}})$$

$$\rho E = \frac{P}{\gamma - 1} = \frac{\rho k_B T}{\mu m_H}\frac{1}{\gamma - 1} = \rho \times 6 \times 10^{9}\text{erg.cm}^{-3}$$

Initialisations

- Taille de la simulation:

  30 × 30 pc² ↔ 100x100 pixels²

  $$\Rightarrow \boxed{\Delta x = 10^{18}\text{cm}}$$

- Pas de temps:

  Courant condition

  $$\rightarrow \boxed{\Delta \mathrm{t} = min\left(\frac{Courant\ factor}{3}\frac{dx}{max(C_{max})}\right)}$$

  Initialisation:

  _ $\|\vec{\mathrm{u}}\| = 0$

  _ $min(dt_{local}) = dt_{SN}$

  _ $P = (\gamma - 1)\rho E_{th} = (\gamma - 1)\rho_i E_{SN}$

$$\Rightarrow \boxed{\Delta \mathrm{t} \sim 1{,}45 \times 10^8 \text{ s} \sim 4{,}6\ yr}$$

Entrées physiques:

- Energie de la supernova: 10^{30} erg.cm⁻¹
- Indice adiabatique: $\gamma = \frac{5}{3}$
- Densité initiale: $\rho = 1{,}67 \times 10^{-24}$ g.cm⁻³

Le code (Fortran)

Paramètres d'initialisation
Initialisation des tableaux
Ajout de la supernova
Calcul de dt
Boucle sur le temps
- Boucle sur la grille
  - Calcul des flux aux interfaces
- Fin
- Boucle sur la grille
  - Update de l'état des cellules
- Fin
- Itération du temps
- Boucle sur la grille
  - Calcul du dt minimal
- Fin
- Sauvegarde

Fin

Taille de la simulation: 30×10^{20} cm²
Energy de la supernova: 10^{30} erg.cm^{-1}
Indice adiabatique: $\frac{5}{3}$
Densité initiale: 1.67×10^{-24} g.cm^{-3}

+ Condition de courant: $dt < \frac{0,8}{3} \times \frac{dx}{v + C_S}$

+ Condition limite d'écoulement (''outflow'')

Aperçu

```fortran
do while(t<tmax)
    if(istep==1000)EXIT
    do j = 0,ny
      do i = 0,nx
          flux(1,i,j,1) = (subflux1(i+1,j) + subflux1(i,j))/2 - maxspeed(i,j,1)*(state(i+1,j,1)-state(i,j,1))
          flux(2,i,j,1) = (subflux2(i,j+1) + subflux2(i,j))/2 - maxspeed(i,j,2)*(state(i,j+1,1)-state(i,j,1))

          flux(1,i,j,2) = (subflux3(i+1,j) + subflux3(i,j))/2 - maxspeed(i,j,1)*(state(i+1,j,2)-state(i,j,2))
          flux(2,i,j,2) = (subflux4_5(i,j+1) + subflux4_5(i,j))/2 - maxspeed(i,j,2)*(state(i,j+1,2)-state(i,j,2))

          flux(1,i,j,3) = (subflux4_5(i+1,j) + subflux4_5(i,j))/2 - maxspeed(i,j,1)*(state(i+1,j,3)-state(i,j,3))
          flux(2,i,j,3) = (subflux6(i,j+1) + subflux6(i,j))/2 - maxspeed(i,j,2)*(state(i,j+1,3)-state(i,j,3))

          flux(1,i,j,4) = (subflux7(i+1,j) + subflux7(i,j))/2 - maxspeed(i,j,1)*(state(i+1,j,4)-state(i,j,4))
          flux(2,i,j,4) = (subflux8(i,j+1) + subflux8(i,j))/2 - maxspeed(i,j,2)*(state(i,j+1,4)-state(i,j,4))
       end do
    end do
```

Aperçu

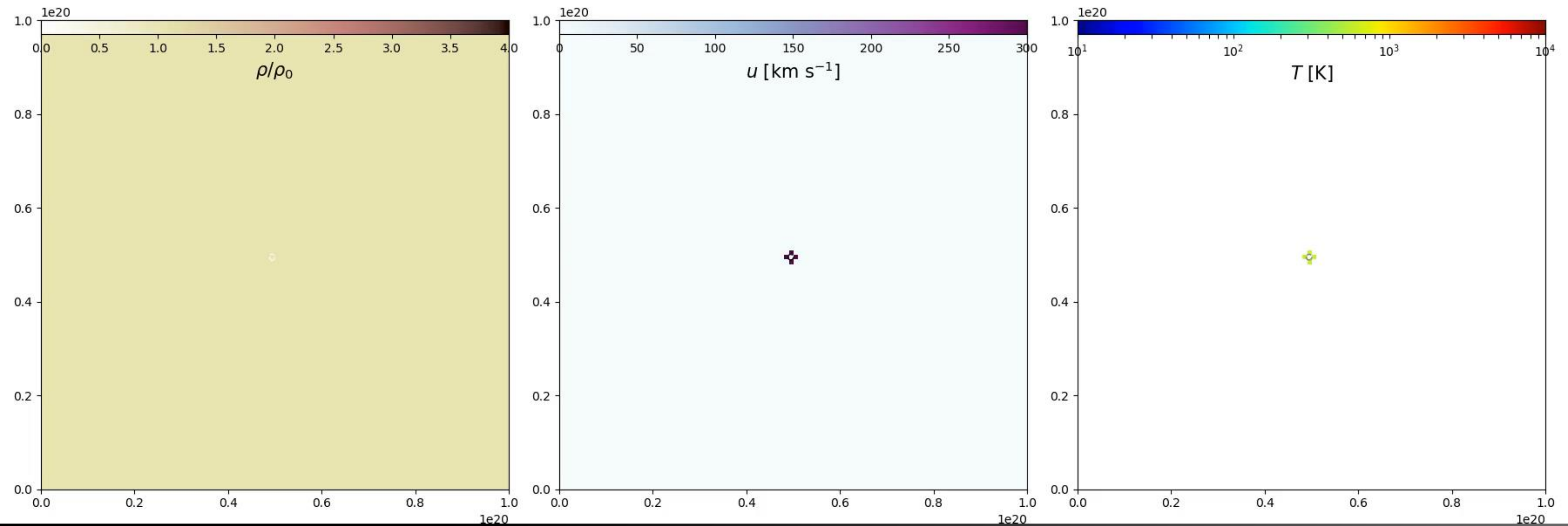
10 fonctions pour
- subflux
- pression
- maxspeed

```
function pression(k,l)
    real(kind=8) :: pression
    integer(kind=4) :: k,l
    real(kind=8) :: rho,u,v,E
    rho=state(k,l,1)
    u=state(k,l,2)/rho
    v=state(k,l,3)/rho
    E=state(k,l,4)/rho
    pression = (E - 0.5d0*(u**2 + v**2))*(gamma-1)*rho
    return
end function
```

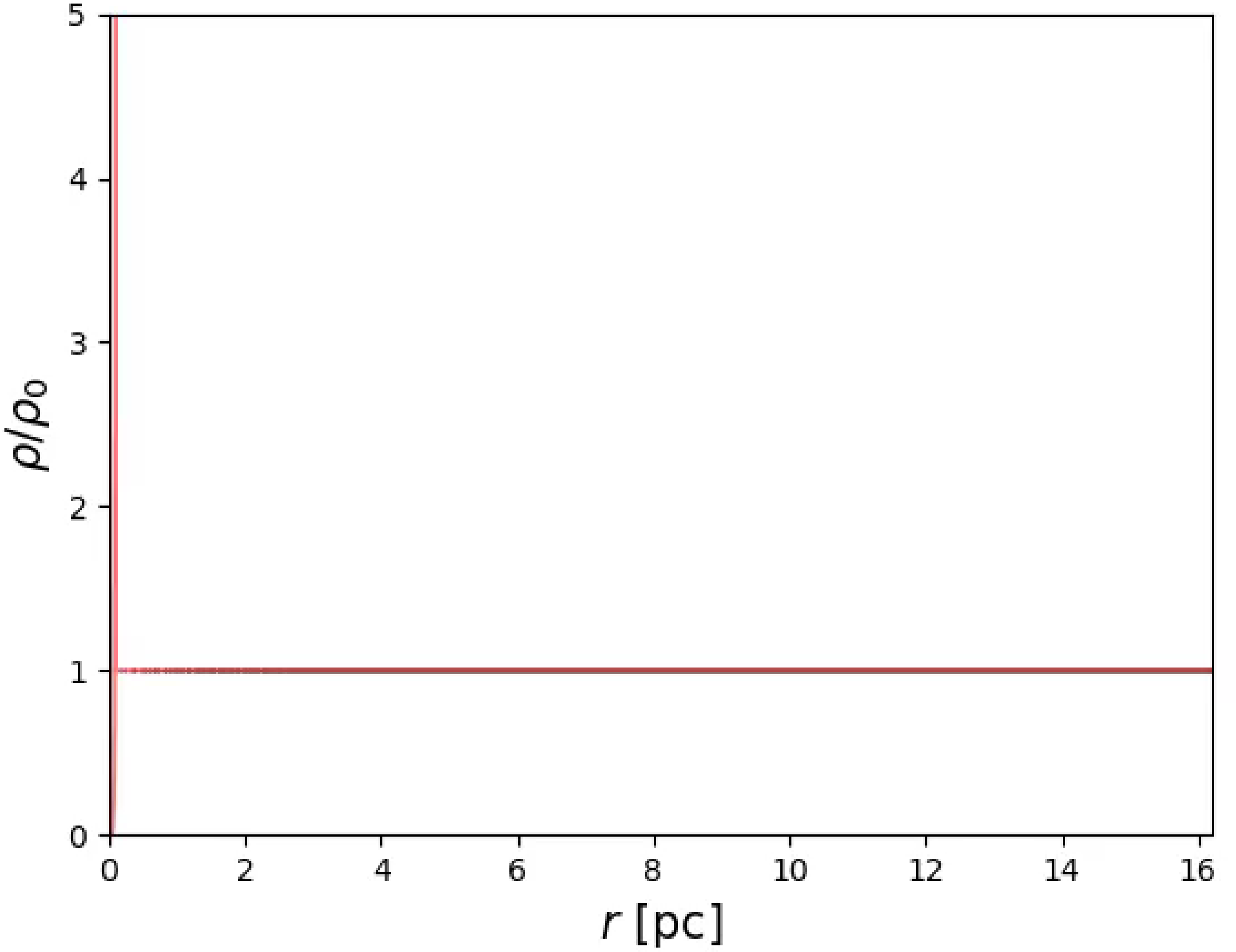
1 subroutine pour sauvegarder

+ Codes pour
- créer des images
- faire une vidéo

Méthode de Rusanov



Comparaison avec la théorie



Travail accompli

Premier code en Fortran 90
Code débuggé (avec aide) → Fonctionne

Pseudo code
Code structuré: fonctions
Plusieurs méthodes numériques

Dérivation équations de volume fini

Perspectives

Ajouter du refroidissement, de la gravité, 3D

Merci de votre attention !