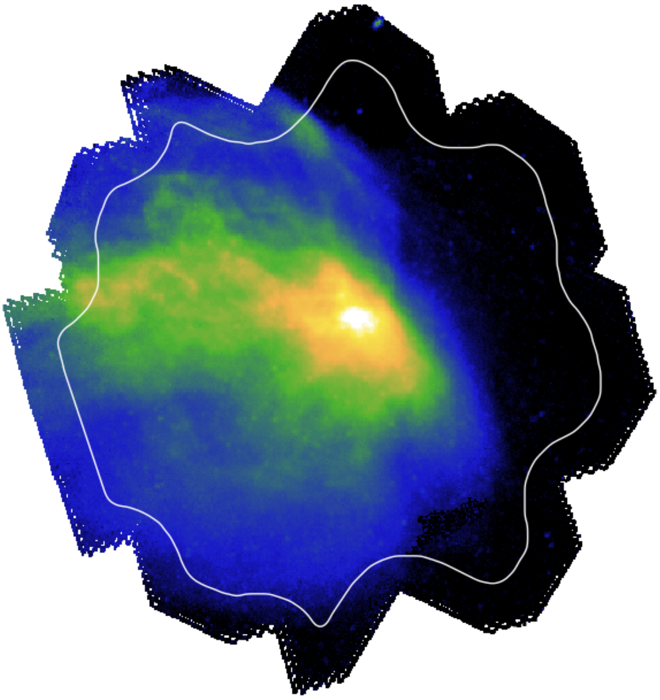
FORTRAN project: Evolution of a prestellar core

Supervision : J. Montillaud
julien.montillaud@univ-fcomte.fr



The dense code G358 as seen by the Herschel Space Observatory at $\lambda = 250\,\mu m$.

This project aims at getting familiar with the language FORTRAN 90 on the example of modelling the evolution of a prestellar core, i.e. the dense condensation of interstellar gas whose gravitational collapse eventually gives birth to a new star or stellar system.

1 Description of the project

1.1 Scientific approach

The general context of this project is the early phases of star formation. Stars form in molecular clouds, i.e. regions of the interstellar medium where the gas is denser ($n_{\rm H} \sim 10^2 - 10^5$ cm^{-3}) and colder (T) than in the rest of the interstellar medium. There, due to gravity and turbulent gas flows, dense gas condensations, dubbed 'dense cores', can form. When dense cores are sufficiently compact, they can become gravitationally unstable and collapse centrally, eventually giving birth to stars. In this case they are called prestellar cores.

In this project, we propose to develop a very simple model of the first steps of the collapse of a prestellar core. To keep this project tractable in a two-week timescale, many assumptions will be adopted. The core is supposed to present a spherical symmetry, so that it is fully characterised by 1-dimensional profiles along the radial direction and only the radial flows are modelled (i.e. all longitudinal flows, including rotation of the core, is supposed to be negligible). The range of evolutionary stages in the model is limited to the initial collapse, so that the law of ideal gas remains applicable even in the innermost layers. The core radial dynamics is assumed to be driven exclusively by gravity and internal thermal pressure, therefore neglecting the effects of turbulence, magnetic fields, external pressure, external radiation fields, etc. It will also be considered to be isothermal, exempting us from energetic balance and radiative transfer calculations.

The fundamental laws of physics included in the model are :

— The continuity equation for the mass volume density ρ and velocity field $\vec{v}$:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \vec{v} = 0, \tag{1}$$

— The Navier-Stokes equation with pressure P and gravitational potential Φ :

$$\rho \left(\frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v} \right) = -\nabla P - \rho \nabla \Phi, \tag{2}$$

— The Poisson equation :

$$\nabla^2 \Phi = 4\pi G \rho, \tag{3}$$

— The law of ideal gas :

$$P = c_{\mathrm{s}}^2 \rho \quad \text{where} \quad c_{\mathrm{s}} = \sqrt{\frac{k_{\mathrm{B}} T}{\mu m_{\mathrm{H}}}}. \tag{4}$$

In the latter equation, c_{s} is the so-called "isothermal sound speed" [1], k_{B} is the Boltzmann constant, T is the temperature m_{H} is the mass of hydrogen nucleus, and μ is the mean particle mass per hydrogen nucleus (to take into account the presence of helium and other chemical elements).

To describe the evolution of the system, we focus on the density $\rho = \rho(r)$ and the velocity $\vec{v} = v(r)\vec{e}_{\mathrm{r}}$ in the frame of the aforementioned assumptions (1D, radial). By injecting eq. 3 and 4 into eq. 2, and expressing the analytical operators in spherical coordinates, we obtain the two following equations :

$$\frac{\partial \rho}{\partial t} = -\frac{1}{r^2}\frac{\partial (r^2 \rho v)}{\partial r}, \tag{5}$$

$$\frac{\partial v}{\partial t} = -v\frac{\partial v}{\partial r} - \frac{c_{\mathrm{s}}^2}{\rho}\frac{\partial \rho}{\partial r} - \int 4\pi G \rho dr. \tag{6}$$

In this project, we aim to integrate numerically these two equations.

1.2 Numerical approach

To integrate eq. 5 and 6 numerically, we propose to use a simple scheme based on finite elements. The profile is sampled by a 1D spatial grid. The density ρ_i and velocity v_i are known at time t_i at every position of the spatial grid ; we determine their values at time $t_{i+1} = t_i + dt$ as follows :

$$\rho_{i+1} = \rho_i + \left(\frac{\partial \rho}{\partial t}\right)_i dt + o(dt) = \rho_i - \frac{1}{r^2}\left(\frac{\partial (r^2 \rho v)}{\partial r}\right)_i dt + o(dt), \tag{7}$$

$$v_{i+1} = v_i + \left(\frac{\partial v}{\partial t}\right)_i dt + o(dt) = v_i + \left[-v_i\left(\frac{\partial v}{\partial r}\right)_i - \frac{c_{\mathrm{s}}^2}{\rho_i}\left(\frac{\partial \rho}{\partial r}\right)_i - \int 4\pi G \rho_i dr\right] dt + o(dt), \tag{8}$$

where $o(dt)$ are vanishing terms of order greater than 1 that will be neglected in the calculations.

The integral term in eq. 8 can be computed numerically by summing the profile ρ_i across the spatial grid, from 0 to x. The three gradients can be computed numerically by a finite element approach :

$$\frac{\partial F}{\partial r} \approx \frac{F(r+dr) - F(r)}{dr}. \tag{9}$$

In this formulation, the gradient is computed to the right of the current point. Using $F(r) - F(r-dr)$ instead of $F(r+dr) - F(r)$ would correspond to compute it to the left of the current point. Both cases bias the calculation and tend to destabilise the integration scheme. Here, we propose to use the mean of these two values to obtain a centred, more stable gradient. More elaborated techniques exist to improve the accuracy and stability of the calculation, and might be proposed by the students.

1. Note that isothermal sound speed is physically meaningless, because the acoustic waves are too fast for their propagation to be isothermal. It differs from the real sound speed by the adiabatic factor γ, where translates the fact that the fast evolution of acoustic waves is rather adiabatic than isothermal.

2 Expected deliverables

2.1 General shape of the program

It is asked to develop a FORTRAN 90 module that contains a set of `FUNCTION`s and `SUBROUTINE`s that an end user could call to study the evolution of dense cores. This module must contain at least :

— a `SUBROUTINE` (hereafter the *main subroutine*) that takes the current state of the system (at $t = t_i$) and returns the next state of the system (at $t = t_{i+1} = t_i + dt$) ;

— a `SUBROUTINE` that takes a field in input and returns its gradient ;

— a `SUBROUTINE` that computes the gravitational forces $\frac{\partial \Phi}{\partial r}$ from the density profile.

It may contain other routines. The user would be responsible for the development of his/her own main program in FORTRAN 90, where the module would be called. The user should be able to compute the evolution of a core by setting the desired initial conditions (density and velocity profiles) and calling the main subroutine of the module at each step of the integration. The other routines of the module will be primarily used by the main subroutine, but it may be useful to the user or the developer to call them directly from the main program to monitor the evolution of some quantities of interest (e.g. the gravitational forces or some gradient) or to debug the program.

2.2 Version 1

The program The first version of the program must contain the implementation of the equations 7 and 8 with the integration of $\frac{\partial \Phi}{\partial r}$ and a routine to compute gradients by centred finite differences. In addition, the program should enable the user to easily compute :

— the total mass of the system ;

— the conversion between mass density ρ and particle density n_{H} ;

— the mass accretion rate $\dot{M}$ at a chosen radius ;

Search what is the "Courant-Friedrichs-Lewy" (CFL) condition. Include a function that enables the user to easily check whether the CFL condition is fulfilled.

Tests

— A set of at least three (3) tests must be proposed to validate the results of the program, based on simple physical situations where the behaviour can be foreseen, possibly separating gravity and the coupling between velocity and pressure.

— The (or some) numerical limits of the program must be characterized.

Scientific analysis Study the system stability as a function of its initial density and of its temperature.

2.3 Version 2

The program The second version of the program can contain :

— a routine to compute a column density profile of the core ;

— a routine to compute the rotational line profile of the CO molecule, assuming that its abundance is constant in the cloud, and that its intrinsic line-width is solely due to thermal broadening ;

Scientific analysis

• Create mock 2D maps of the core column density seen by a radiotelescope with angular resolution θ and a Gaussian beam :

$$f(\alpha) \propto \exp(-\alpha^2/(2\theta^2))$$

where $f(\alpha)$ is the telescope response as a function of the angular distance α between the pointed direction and another direction.

• Study the effect of gravitational infall on the CO line profile.

2.4 Version 3

A third version including a simplified radiative transfer calculation to go beyond the isothermal hypothesis can be developed. Guidelines for this option can be provide on demand.

3 Complementary information

3.1 Provided material

Along with this guide, a `tar` archive is provided which contains files where basic FORTRAN 90 lines of code are included, to help the students to get familiar with this language. The files are the following :

— `main_test.f90` contains the main program, in principle developed by the end user, but also by the developer to test the routine during development (not provided to the end user). It contains the `USE ***` command to include modules from an external file.

— `Constants.f90` contains a module where useful numerical values are defined as public variables. Note as well the definition of `dp`, which makes it easy to change the precision of the floats throughout the code.

— `Prestel.f90` contains a module with one example of a `SUBROUTINE` and one example of a `FUNCTION`.

— `Makefile` is not a FORTRAN file, it contains the compilation instructions to easily compile all the files used in the FORTRAN code. It can be executed by typing `make` in a terminal opened in the same directory. The compilation options can be changed using the `FFLAGS` variable at the beginning of the script.

3.2 Orders of magnitudes

A typical prestellar core gathers a mass of ~ 1 solar mass ($M_\odot$) within a width of ~ 0.1 pc (parsec), with a temperature of ~ 10 K and collapses in a free-fall time, which can be computed using the formula :

$$t_{\rm ff} = \sqrt{\frac{3\pi}{32 G \rho_0}} \tag{10}$$

where ρ_0 is the mean density of the cloud.