Anelastic Approximation without linearization

June 19, 2023, *Revision*

1 The anelastic scheme

We started with the definition of anelastic approximation given by,

$$\boldsymbol{\nabla}.(\rho \boldsymbol{u}) = 0 \tag{1}$$

This is contrary to what most people do when they take an anelastic approximation. Usually all thermodynamic variables are written in terms of a base state and a perturbed state. However we have avoided this linearization because in a nonlinear problem like the Navier Stokes it may be possible that the thermodynamic perturbations become larger than the base state.

Taking the divergence of the Navier Stokes equation we get a Poisson equation for Pressure. The anelastic set of equations may be given by,

$$\rho = \rho(p, s) \tag{2}$$

$$\frac{\partial s}{\partial t} = -(\boldsymbol{u}.\boldsymbol{\nabla})s - \frac{\boldsymbol{R}^s}{\rho T} \tag{3}$$

$$\nabla^2 p = \nabla_i[\rho \boldsymbol{R}^v_i - \nabla_j(\rho u_i u_j)] = g(\rho, \boldsymbol{u}) \tag{4}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} = -(\boldsymbol{u}.\boldsymbol{\nabla})\boldsymbol{u} - \frac{\boldsymbol{\nabla} p}{\rho} + \boldsymbol{R}^v \tag{5}$$

I have not explicitly written down all the terms in Navier-Stokes or entropy equation, but denoted them with $\boldsymbol{R}^v$ or $\boldsymbol{R}^s$. The number of variables are $\boldsymbol{u}, \rho, p, s$; 4 in number. The temperature, T can be obtained from the ideal gas equation $p = \rho R_g T$.

An ideal gas has to satisfy Eq. (2) between p, ρ and s given by

$$\ln \rho = \frac{1}{\gamma} \ln p - (s - s_0)/c_p \tag{6}$$

where s_0 is a constant. For this we use and have extended the eos_idealgas.f90 module to include a case with ivars=ipp_ss. The following routines need to be modified.

```
module procedure eoscalc_pencil   ! explicit f implicit m,n
module procedure eoscalc_point    ! explicit lnrho, ss
module procedure eoscalc_farray   ! explicit lnrho, ss
```

Following this modification, the eos_idealgas module provides ρ as a pencil which is a function of pencils p and s. Now the number of equations is also 4 (Eq. 2, 3, 4, 6).

At any given time, the pressure is calculated such that the divergence of $\rho\boldsymbol{u}$ vanishes. This calculation is based on the values of ρ, $\boldsymbol{u}$, and s at a given time. Since $\boldsymbol{u}$ and s are evolving, p will evolve, and therefore also ρ. It would not be correct to iterate at a given time p and modify only ρ, without at the same time recalculating all the other terms on the right hand sides of the other equations.

So, essentially we do the following steps in the pencil code to incorporate the anelastic approximation.

- Start
  1. Initialize $\rho, s, \boldsymbol{u}$ at $t = t0$.
  2. Calculate p_0 at $t = t0$ from equation of state. This is the pressure offset.
  3. Use duu_dt module in hydro to fill up iuu:iuu+2 indices in the df-array except the $\boldsymbol{\nabla} p$ term. This corresponds to $\boldsymbol{R^u}$.
  4. Calculate pressure necessary to make $\boldsymbol{\nabla}.(\rho\boldsymbol{u}) = 0$ at $t0$ from $\nabla^2 p = \boldsymbol{\nabla}.(\rho\boldsymbol{R^u})$ and add to $\boldsymbol{R^u}$.
- Run
  1. $t = t + dt$, time step $\boldsymbol{u}$, s using ρ, p at t.
  2. Use duu_dt module in hydro to fill up iuu:iuu+2 indices in the df-array except the $\boldsymbol{\nabla} p$ term. This corresponds to $\boldsymbol{R^u}$ at t+dt.
  3. Calculate pressure at $t + dt$ from $\nabla^2 p = \boldsymbol{\nabla}.(\rho(t)\boldsymbol{R^u}(t + dt))$ and add back to df array. This can be done in absence of stratification. In presence of gravity, the contribution of pressure to $\boldsymbol{\nabla}.(\rho\boldsymbol{g})$ has to added to $\nabla^2 p$. Adjust boundary conditions for p so that mass flux out of the domain is zero. So at the boundaries we have $\oint \rho\boldsymbol{u}.\boldsymbol{dS} = 0$, which implies,
     $$\oint (\boldsymbol{\nabla} p - \rho\boldsymbol{R^u}).\boldsymbol{dS} = 0$$
     Without loss of generality this means,
     $$\hat{\boldsymbol{n}}.\boldsymbol{\nabla} p = \rho\boldsymbol{R^u}.\hat{\boldsymbol{n}}$$
  4. Calculate ρ at $t + dt$ using $\rho = \rho(s, p)$, where s and p are also at t+dt. The problem lies in step 3 where p,ρ are evaluated at different times. We need to calculate ρ and p simultaneously. One way is iteratively solving step 3 and 4. Another method for isothermal or adiabatic case is to solve the following non-linear Poisson equation. $(\nabla^2 - \boldsymbol{R^u}.\boldsymbol{\nabla} - \boldsymbol{\nabla}.\boldsymbol{R^u})p = 0$
  5. Go to step 1

2 Poisson's solver and boundary conditions

We want to solve

$$\Delta\Psi(x,y,z) = RHS(x,y,z),$$

We Fourier decompose Ψ in the (x,y) plane then

$$\frac{d^2\Psi_k}{dz^2} - k^2\Psi_k = RHS_k \Rightarrow \frac{\Psi_k^{i+1} - 2\Psi_k^i + \Psi_k^{i-1}}{\delta z^2} - k^2\Psi_k^i = RHS_k^i,$$

leading to the following tridiagonal system to solve

$$\frac{1}{\delta z^2} \quad ; \quad \frac{-2}{\delta z^2} - k^2 \quad ; \quad \frac{1}{\delta z^2}$$

Now comes the problem of the boundary conditions. For the self-gravity problem, we assume that the density is zero outside the domain that reduces to $\Delta\Psi = 0$ or

$$\frac{d^2\Psi_k}{dz^2} - k^2\Psi_k = 0 \Rightarrow \Psi \propto e^{+kz} + e^{-kz},$$

and the solution $+kz$ must be eliminated as we have $\Psi \to 0$ as $z \to \infty$. Then $\Psi \propto e^{-kz}$ outside or

$$\Psi e^{kz} = cte \Rightarrow \frac{d\Psi e^{kz}}{dz} = 0 \Rightarrow \frac{d\Psi}{dz} + k\Psi = 0.$$

We therefore have the two following conditions on the boundary points:

$$\begin{cases} \Delta\Psi = 0 \Rightarrow= \dfrac{\Psi_k^{i+1} - 2\Psi_k^i + \Psi_k^{i-1}}{\delta z^2} = 0 \Rightarrow \Psi_k^{i+1} - 2\Psi_k^i + \Psi_k^{i-1} = 0 \\ \dfrac{d\Psi}{dz} + k\Psi = 0 \Rightarrow \dfrac{\Psi_k^{i+1} - \Psi_k^{i-1}}{2\delta z} + k\Psi_k^i = 0 \Rightarrow \Psi_k^{i+1} - \Psi_k^{i-1} + 2\delta z k\Psi_k^i = 0 \end{cases} \tag{7}$$

Elimination of Ψ_k^{i-1} leads to a single equation for Ψ_k^i and Ψ_k^{i+1} only on the first gridpoints of the tridiagonal matrix:

$$2\Psi_k^{i+1} + 2(k\delta z - 1)\Psi_k^i = 0$$

The same demonstration can be applied to the last gridpoint where we want to eliminate Ψ_k^{i+1} in that case.