Chapter 10

Coordinate systems and gridding techniques

So far we have assumed that we are doing modeling in a cartesian coordinate system with a rectangular mesh. For many applications this is, however, too restrictive. For instance, in astrophysics, if we are interested in modeling the hydrodynamics of a rotating star or a rotating collapsing molecular cloud core, then we know (or assume) beforehand that our solution will be nearly spherical, and that we are modeling deviations from such a spherical distribution. It is then clear that a polar coordinate system (r, θ, ϕ) (often also called a "spherical coordinate system") is better and easier to use. Another example is if we wish to study axisymmetric systems. In this case we can restrict ourselves to two dimensions: (r, θ) in polar coordinates or (r, z) in cylindrical coordinates. Since 2-D calculations are very much faster than 3-D ones, this choice of coordinates saves a lot of computing time for such problems. Therefore non-cartesian coordinates are very often used, in particular in astrophysics.

But sometimes the useage of a different coordinate system does not help. For instance, in astrophysics, if we want to model the collapse of a complex and clumpy molecular cloud, in which a number of clumps contract enormously and eventually become stars. No choice of coordinate system helps here. One must refine the grid around each clump in an adaptive (time-dependent) way. For this type of gridding problem astrophysicists typically use *Adaptive Mesh Refinement (AMR)*.

A more down-to-earth example is the flow of gas or fluid around obstacles, such as the flow of air around a wing or around a racing car. The shape of the obstacle is typically too complex to warp a particular coordinate system around it. On the other hand, if we try to solve this problem on a cartesian grid, then the obstacle surfaces are jagged surfaces (like a slope of a hill created with LEGO blocks). These jagged surfaces tend to produce disturbances that, in high Reynolds number flows, can really cause havoc in the solution. In these cases special purpose *unstructured grids* are used.

In this chapter we will touch upon all three issues (the polar coordinate system, AMR techniques and unstructured grids), but we will be rather brief. We will concern ourselves only with Riemann solvers, but much of the philosophy can be directly transferred to classical hydrodynamics schemes as well.

NOTE: This chapter is written very recently, so there may still be errors in the equations. Beware.

10.1 Hydrodynamics in the polar coordinate system

10.1.1 A scalar conservation law in polar coordinates

There are two popular non-cartesian coordinate systems that are often used: cylindrical coordinates (r, ϕ, z) and polar/spherical coordinates (r, θ, ϕ). In this section we will focus on the polar coordinate system. The equations for cylindrical coordinates can be derived from these by taking $z \simeq (\pi/2 - \theta)\, r$ and taking the limit $z \ll r$ to remove all sine and cosine factors (and once the equations are derived, z can again be large).

The coordinates are: $r > 0$, the radial coordinate, $0 \leq \theta < \pi$ the polar coordinate ($\theta = 0$ is the north pole, $\theta = \pi$ is the south pole and $\theta = \pi/2$ is the equatorial plane), and $0 \leq \phi < 2\pi$ the azimuthal coordinate. Let us define $\vec{u}$ the velocity vector and (u, v, w) its components in r, θ and ϕ directions.

Each grid point (r_i, θ_j, ϕ_k) is at the center of a cell. A cell is given by: $r_{i-1/2} \leq r < r_{i+1/2}$, $\theta_{j-1/2} \leq \theta < \theta_{j+1/2}$, $\phi_{k-1/2} \leq \phi < \phi_{k+1/2}$. The cell walls in r and θ direction are curved while the cell walls in ϕ direction are straight surfaces. This curvature will turn out to produce new terms in the momentum equations, but let us first look at how we can formulate a scalar conservation problem in polar coordinates. The partial differential equation for a scalar conservation law in polar coordinates is:

$$\frac{\partial q}{\partial t} + \nabla \cdot (q\vec{u}) = 0 = \frac{\partial q}{\partial t} + \frac{1}{r^2}\frac{\partial (r^2 q u)}{\partial r} + \frac{1}{r\sin\theta}\frac{\partial(\sin\theta\, q v)}{\partial \theta} + \frac{1}{r\sin\theta}\frac{\partial (q w)}{\partial \phi} = 0 \tag{10.1}$$

As one can see, there are various geometric factors involved.

One can easily see how these factors arise if one derives a discrete version of this conservation equation. Consider a cell (i, j, k) given by $r_{i-1/2} \leq r < r_{i+1/2}$, $\theta_{j-1/2} \leq \theta < \theta_{j+1/2}$ and $\phi_{k-1/2} \leq \phi < \phi_{k+1/2}$. The volume of this cell is:

$$V_{i,j,k} = \frac{1}{3}(r^3_{i+1/2} - r^3_{i-1/2})(\cos\theta_{j-1/2} - \cos\theta_{j+1/2})(\phi_{k+1/2} - \phi_{k-1/2}) \tag{10.2}$$

The $i + 1/2$ surface of the cell in r-direction (i.e. an r =constant surface) is:

$$S_{(r),i+1/2,j,k} = r^2_{i+1/2}(\cos\theta_{j-1/2} - \cos\theta_{j+1/2})(\phi_{k+1/2} - \phi_{k-1/2}) \tag{10.3}$$

and similar for $i - 1/2$. The $j + 1/2$ surface of the cell in θ-direction (i.e. a θ =constant surface) is:

$$S_{(\theta),i,j+1/2,k} = \frac{1}{2}(r^2_{i+1/2} - r^2_{i-1/2}) \sin\theta_{j+1/2}(\phi_{k+1/2} - \phi_{k-1/2}) \tag{10.4}$$

and similar for $j - 1/2$. The $k + 1/2$ surface of the cell in ϕ-direction (i.e. a ϕ =constant surface) is:

$$S_{(\phi),i,j,k+1/2} = \frac{1}{2}(r^2_{i+1/2} - r^2_{i-1/2})(\theta_{j+1/2} - \theta_{j-1/2}) \tag{10.5}$$

and identical for $k - 1/2$.

If we now difference the equations in a conservative form:

$$\begin{aligned}\frac{\partial(q_{i,j,k}V_{i,j,k})}{\partial t} &= F_{i-1/2,j,k}S_{(r),i-1/2,j,k} - F_{i+1/2,j,k}S_{(r),i+1/2,j,k} \\ &+ F_{i,j-1/2,k}S_{(\theta),i,j-1/2,k} - F_{i,j+1/2,k}S_{(\theta),i,j+1/2,k} \\ &+ F_{i,j,k-1/2}S_{(\phi),i,j,k-1/2} - F_{i,j,k+1/2}S_{(\phi),i,j,k+1/2}\end{aligned} \tag{10.6}$$

where F is the flux for q. Eq.(10.6) is the conservatively discretized form of the scalar conservation equation in a general coordinate system. With Eqs.(**??**) inserted into Eq.(10.6), we obtain the conservatively discretized form of the scalar conservation equation in polar coordinates.

For use later in this chapter, let us introduce Δ_r, Δ_θ, and Δ_ϕ, as well as S_r, S_θ and S_ϕ, and finally F_r, F_θ and F_ϕ such that Eq. (10.6) for polar coordinates can be written as:

$$\frac{\partial(q_{i,j,k}V_{i,j,k})}{\partial t} + (\Delta_r F_r S_r)_{i,j,k} + (\Delta_\theta F_\theta S_\theta)_{i,j,k} + (\Delta_\phi F_\phi S_\phi)_{i,j,k} = 0 \tag{10.7}$$

Here the Δ_r symbol denotes that a difference is computed of the argument (here: $F_r S_r$) between its value at the $i+1/2$ interface and the $i-1/2$ interface. Same for θ and ϕ.

$\rightarrow$ **Exercise:** Derive, from this discrete form of the conservation equation, the differential form of the equation, Eq. (10.1).

10.1.2 The Euler equations in polar coordinates

The discretized conservation law in the form Eq.(10.6) (or equivalently Eq.(10.7)) is the form of the equation which we use for our Riemann solvers for hydrodynamics. Because Riemann solvers by definition produce an interface flux ($F_{i+1/2,j,k}$, $F_{i,j+1/2,k}$ or $F_{i,j,k+1/2}$), this form of the conservation equation lends itself perfectly for Riemann solvers. One simply substitutes the scalar $q_{i,j,k}$ with $(\rho e_{\rm tot}, \rho u, \rho v, \rho w, \rho)_{i,j,k}$ and the scalar fluxes $F_{i+1/2,j,k}$, $F_{i,j+1/2,k}$ and $F_{i,j,k+1/2}$ with the fluxes from the Riemann problems in each of these directions. This is indeed the way we can implement Riemann solvers in non-cartesian coordiantes. However, this is unfortunately not yet the full story.

The above derivation only holds for scalar quantities. Indeed, ρ and $\rho e_{\rm tot}$ are scalars, and both obey Eq. (10.1):

$$\begin{aligned}\frac{\partial\rho}{\partial t} &+ \nabla\cdot(\rho\vec{u}) = 0\\ &= \frac{\partial\rho}{\partial t} + \frac{1}{r^2}\frac{\partial(r^2\rho u)}{\partial r} + \frac{1}{r\sin\theta}\frac{\partial(\sin\theta\;\rho v)}{\partial\theta} + \frac{1}{r\sin\theta}\frac{\partial(\rho w)}{\partial\phi} = 0\end{aligned} \tag{10.8}$$

and

$$\begin{aligned}\frac{\partial\rho e_{\rm tot}}{\partial t} &+ \nabla\cdot(\rho h_{\rm tot}\vec{u}) = 0\\ &= \frac{\partial\rho e_{\rm tot}}{\partial t} + \frac{1}{r^2}\frac{\partial(r^2\rho h_{\rm tot}u)}{\partial r} + \frac{1}{r\sin\theta}\frac{\partial(\sin\theta\;\rho h_{\rm tot}v)}{\partial\theta} + \frac{1}{r\sin\theta}\frac{\partial(\rho h_{\rm tot}w)}{\partial\phi} = 0\end{aligned} \tag{10.9}$$

Both these equations can indeed be solved using the discrete form we derived above (see Eq. 10.6). These give for the density equation:

$$\frac{\partial(\rho V)_{i,j,k}}{\partial t} + (\Delta_r F_{\rho r} S_r)_{i,j,k} + (\Delta_\theta F_{\rho\theta} S_\theta)_{i,j,k} + (\Delta_\phi F_{\rho\phi} S_\phi)_{i,j,k} = 0 \tag{10.10}$$

where $F_{\rho r}$, $F_{\rho\theta}$ and $F_{\rho\phi}$ are the Riemann fluxes for the density. For the energy equation:

$$\frac{\partial(\rho e_{\rm tot}V)_{i,j,k}}{\partial t} + (\Delta_r F_{\rho e_{\rm tot}r} S_r)_{i,j,k} + (\Delta_\theta F_{\rho e_{\rm tot}\theta} S_\theta)_{i,j,k} + (\Delta_\phi F_{\rho e_{\rm tot}\phi} S_\phi)_{i,j,k} = 0 \tag{10.11}$$

where $F_{\rho e_{\rm tot}r}$, $F_{\rho e_{\rm tot}\theta}$ and $F_{\rho e_{\rm tot}\phi}$ are the Riemann fluxes for the energy.

However, for the momentum components ρu, ρv and ρw this is not the case. These three form the r, θ and ϕ components of the momentum *vector* $\vec{p}$, which is sensitive to spatial direction. Now here comes the difficulty: Suppose one has a momentum vector $\vec{p}_1 = (1, 0, 0)$ at location $(r = 1, \theta = \pi/2, \phi = 0)$. If we compare this to the momentum vector $\vec{p}_2 = (1, 0, 0)$ at location $(r = 1, \theta = \pi/2, \phi = \pi/2)$, then we see that these two vectors, although they have precisely the same components in polar coordinates, they point in different directions. This is because the vector basis in which the vector components are formulated in the polar coordinates is a *local* vector basis: the first component always points away from the center, the second away from the pole and the third in the azimuthal direction. These directions are different at different positions. Since physical objects, like fluid/gas packages, do not have knowledge of the polar coordinate system and its peculiarities, their momentum vectors behave "naturally" when they are expressed in a *global* vector basis. If they are cast into a local basis that has a different orientation at different positions, then they behave strangely in such a basis. Therefore, in polar coordinates the conservation laws of vector quantities such as the momentum of a fluid parcel yields extra terms in the equations. They look as if an additional force is present, and therefore these terms are also called "pseudo forces". They have the property that they *only* change the direction of the momentum vector as expressed in the local basis of the non-cartesian coordinate system. The absolute value of the momentum vector (and thereby of the velocity vector) is not changed. This is logical, because pseudo forces are not real forces and can therefore not truly accelerate or decellerate a fluid parcel. However, in Section 10.1.5 we will see that if one uses a *rotating* polar coordinate system, then it may seem as if a true acceleration/decelleration is taking place, but again, this turn out to be not a real effect.

It is not trivial to derive the extra terms that arise for the momentum equations. The most robust and reliable way of doing this is to use Riemannian Geometry, and this method also allows one to derive the equations of viscous hydrodynamics (the Navier-Stokes equation), which spawns even more complex additional terms when cast in polar coordinates. We will not go into this mess here, and instead we will simply quote the resulting Euler equations in polar coordinates.

The radial momentum equation becomes:

$$\begin{aligned}\frac{\partial \rho u}{\partial t} + \frac{1}{r^2}\frac{\partial (r^2 \rho u^2)}{\partial r} + \frac{\partial P}{\partial r} + \frac{1}{r\sin\theta}\frac{\partial(\sin\theta\rho u v)}{\partial\theta} + \frac{1}{r\sin\theta}\frac{\partial(\rho u w)}{\partial\phi} \\ - \frac{\rho(v^2 + w^2)}{r} = 0\end{aligned} \tag{10.12}$$

In discrete form for the Riemann solver:

$$\begin{aligned}\frac{\partial(\rho u V)}{\partial t} + \Delta_r(F_{rr}S_r) + \Delta_\theta(F_{r\theta}S_\theta) + \Delta_\phi(F_{r\phi}S_\phi) \\ = \left[\frac{2P}{r} + \frac{\rho(v^2 + w^2)}{r}\right] V\end{aligned} \tag{10.13}$$

where V is the volume of the cell, S_r, S_θ and S_ϕ the surfaces of the cell, and $F_{rr} = \rho u^2 + P$, $F_{r\theta} = \rho u v$ and $F_{r\phi} = \rho u w$ the momentum fluxes for this equation. These momentum fluxes (together with the density and energy fluxes) are the ones that we obtain from the Riemann solver of our choice at the interfaces.

The θ momentum equation becomes:

$$\begin{aligned}&\frac{\partial \rho v}{\partial t}+\frac{1}{r^2}\frac{\partial(r^2\rho u v)}{\partial r}+\frac{1}{r\sin\theta}\frac{\partial(\sin\theta\rho v^2)}{\partial\theta}+\frac{1}{r}\frac{\partial P}{\partial\theta}+\frac{1}{r\sin\theta}\frac{\partial(\rho w v)}{\partial\phi}\\&\quad+\rho\frac{vu}{r}-\frac{\cos\theta}{\sin\theta}\rho\frac{w^2}{r}=0\end{aligned}\tag{10.14}$$

In discrete form for the Riemann solver:

$$\begin{aligned}\frac{\partial(\rho v V)}{\partial t}&+\Delta_r(F_{\theta r}S_r)+\Delta_\theta(F_{\theta\theta}S_\theta)+\Delta_\phi(F_{\theta\phi}S_\phi)\\&=-\left[\rho\frac{vu}{r}-\frac{\cos\theta}{\sin\theta}\frac{\rho w^2+P}{r}\right]V\end{aligned}\tag{10.15}$$

where $F_{\theta r}=\rho v u$, $F_{\theta\theta}=\rho v^2+P$ and $F_{\theta\phi}=\rho v w$ are the momentum fluxes for this equation.

The ϕ momentum equation becomes:

$$\begin{aligned}&\frac{\partial \rho w}{\partial t}+\frac{1}{r^2}\frac{\partial(r^2\rho u w)}{\partial r}+\frac{1}{r\sin\theta}\frac{\partial(\sin\theta\rho v w)}{\partial\theta}+\frac{1}{r\sin\theta}\frac{\partial(\rho w^2+P)}{\partial\phi}\\&\quad+\frac{\rho u w}{r}+\frac{\cos\theta}{\sin\theta}\frac{\rho v w}{r}=0\end{aligned}\tag{10.16}$$

In discrete form for the Riemann solver:

$$\begin{aligned}\frac{\partial(\rho w V)}{\partial t}&+\Delta_r(F_{\phi r}S_r)+\Delta_\theta(F_{\phi\theta}S_\theta)+\Delta_\phi(F_{\phi\phi}S_\phi)\\&=-\left[\rho\frac{uw}{r}+\frac{\cos\theta}{\sin\theta}\frac{\rho v w}{r}\right]V\end{aligned}\tag{10.17}$$

where $F_{\phi r}=\rho w u$, $F_{\phi\theta}=\rho w v$ and $F_{\phi\phi}=\rho w^2+P$ are the momentum fluxes for this equation.

The recipe for applying a Riemann solver to polar coordinates is then:

- At the start of the run we pre-compute the cell volumes and cell surfaces, as well as the necessary cosine and sine values. This saves much CPU time, in particular since cosine and sine functions are heavy in terms of CPU time required.
- We perform directional operator splitting (Strang splitting). For each direction we do 1-D independent problems and copy their results back into the 3-D grid.
- For each 1-D problem we compute the left- and right- interface states in precisely the same way as the chapter on Riemann solvers. These left and right states are now used to compute the Riemann interface fluxes. Note that the Roe or HLLC subroutine used here does not know about any geometric issues. It simply produces a flux as if it were a Cartesian Riemann solver.
- These Riemann fluxes are now inserted in the r, θ or ϕ splitted part of Eqs.(**??**). By virtue of operator splitting the other terms in these equations are put to zero.
- The source terms in these equations can be added at the very end of the time step, as another operator splitting step. But sometimes it can be useful instead to add one of them at the end of one directional splitting step, while the other at the end of another directional splitting step. This is something the designer of the algorithm can choose freely. Various choices have various special advantages and disadvantages. It depends on the problem at hand which strategy is the best.

10.1.3 A note of caution for computing the geometric source terms

Let us briefly return to the discrete radial momentum equation Eq.(10.13). The term $2P/r$ appeared because we had to convert the $\partial P/\partial r$ into a $(1/r^2)\partial(Pr^2)/\partial r$ in order to incorporate it into the momentum flux difference. It matters, however, how one computes this term. If done without care, one could easily get a situation in which the case $P(r,\theta,\phi)$ =constant and $u,v,w=0$ is not recognized by the solver as a static solution. One can, however, replace:

$$\frac{2P}{r}V \rightarrow P\Delta_r S_r \equiv (S_{(r),i+1/2,j,k} - S_{(r),i-1/2,j,k})P_{i,j,k} \tag{10.18}$$

If one now computes this geometric pressure source term in this way, then for the case of $P(r,\theta,\phi)$ =constant and $u,v,w=0$ the solution is kept unaltered, as it should be, down to machine precision. In other words, the code now "recognizes" the $P(r,\theta,\phi)$ =constant solution. The reason why this is the case is because the term $2P/r$ arises because of the differences in the surface areas of the cell interfaces at $i-1/2$ and $i+1/2$ which are used in the $\Delta_r(F_{rr}S_r)$ term. So by compensating this with exactly the same but opposite we get an exact cancellation, and thereby a constant pressure will be recognized as such.

The same trick can be applied to the geometric pressure source term of the θ-momentum equation:

$$\frac{\cos\theta}{\sin\theta}\frac{P}{r}V \rightarrow P\Delta_\theta S_\theta \equiv (S_{(\theta),i,j+1/2,k} - S_{(\theta),j-1/2,k})P_{i,j,k} \tag{10.19}$$

For the ϕ-momentum equation no such geometric pressure term arises.

10.1.4 Replacing ϕ-momentum with angular momentum

If one models rotating systems such as accretion disks or rotating stars it can prove to be advantageous in terms of accuracy and reliability to not advect the ϕ-momentum but the *ϕ-angular* momentum instead (see e.g. Kley 1998, A&A 338, L37).

Let us define $l = w\sin\theta r$. We only replace w by l in the advection in r and in θ direction.

The physical form of the ϕ-momentum equation is then:

$$\begin{aligned} \frac{\partial\rho l}{\partial t} + \frac{\partial(\rho u l)}{\partial r} + \frac{1}{r}\frac{\partial(\rho v l)}{\partial\theta} + \frac{\partial(\rho w^2+P)}{\partial\phi} \\ + 2\sin\theta\rho u w + \cos\theta\rho v w = 0 \end{aligned} \tag{10.20}$$

which can be written as:

$$\frac{\partial\rho l}{\partial t} + \frac{1}{r^2}\frac{\partial(r^2\rho u l)}{\partial r} + \frac{1}{r\sin\theta}\frac{\partial(\sin\theta\rho v l)}{\partial\theta} + \frac{\partial(\rho w^2+P)}{\partial\phi} = 0 \tag{10.21}$$

The discrete form for the Riemann solver is:

$$\frac{\partial(\rho l V)}{\partial t} + \Delta_r(r\sin\theta F_{\phi r}S_r) + \Delta_\theta(r\sin\theta F_{\phi\theta}S_\theta) + \Delta_\phi(r\sin\theta F_{\phi\phi}S_\phi) = 0 \tag{10.22}$$

One sees that the source terms on the right-hand side are completely gone. This is good, because source terms are always a menace and if they can be avoided, then that is a positive thing.

For convenience we can now define the fluxes for the ϕ angular momentum:

$$\tilde{F}_{\phi r} = r\sin\theta F_{\phi r} = \rho u l \tag{10.23}$$

$$\tilde{F}_{\phi r} = r\sin\theta F_{\phi\theta} = \rho v l \tag{10.24}$$

$$\tilde{F}_{\phi r} = r\sin\theta F_{\phi\phi} = \rho w l + Pr\sin\theta \tag{10.25}$$

so then we obtain:

$$\frac{\partial(\rho l V)}{\partial t} + \Delta_r(\tilde{F}_{\phi r} S_r) + \Delta_\theta(\tilde{F}_{\phi\theta} S_\theta) + \Delta_\phi(\tilde{F}_{\phi\phi} S_\phi) = 0 \tag{10.26}$$

This means that all we have to do is to construct the normal fluxes F from the Riemann solver, and convert them into $\tilde{F}$ for the angular momentum equation (the other equations remain untouched), and difference them.

10.1.5 Rotating polar coordinate system

For many purposes it may be desirable to let the coordinate system rotate with a certain rate. For instance, if we model the flow of air in the Earth's atmosphere, we want to use a polar coordinate system that rotates with the Earth's rotation rate, so that we can anchor the coordinate system to the surface of the Earth. Another example is the study of accretion disks around young stars and black holes. This will, however, lead to yet again more "pseudo forces": the Coriolis force and the centrifugal force. In fact, they are precisely the same as the pseudo forces we have found in the static spherical coordinates for moving gas. This time, however, they also appear if the gas is not moving with respect to the rotating coordinate system, because having zero velocity in the rotating frame means having non-zero velocity in the lab frame.

We can derive the equations from the above equations. We replace:

$$\partial_t \quad \rightarrow \quad \bar{\partial}_t - \Omega_0 \partial_\phi \tag{10.27}$$
$$w \quad \rightarrow \quad \bar{w} + \Omega_0 r \sin\theta \tag{10.28}$$

where Ω_0 is the rotation rate of the polar coordinate system around the polar axis and $\bar{w}$ is the ϕ-velocity in the rotating frame. We keep, however, the angular momentum variable l still in the lab frame, because only in the lab frame this quantity makes physical sense.

One can show that this coordinate transformation does not affect the continuity equation. The energy conservation equation is also not affected, as long as one still expresses the kinetic energy in the non-rotating lab frame: $(u^2+v^2+w^2)/2$ and not $(u^2+v^2+\bar{w}^2)/2$. The reason is that energy only has physical meaning in a non-rotating frame. Only the momentum equations are affected.

So let us look at the r-momentum equation. Because one replaces the time derivative with $\bar{\partial}_t - \Omega_0\partial_\phi$, one gains an extra term $-\Omega_0\partial_\phi(\rho u V)$. But one also replaces w with $\bar{w} + \Omega_0 r\sin\theta$ in the term with the Δ_ϕ, which gives another extra term $\Omega_0 r \sin\theta \Delta_\phi(\rho u S_\phi)$. By writing the first extra term into a discrete form, one sees that the two extra terms cancel. One can also replace the Vw^2/r geometric source term with $V(\bar{w}+\Omega_0 r\sin\theta)^2/r$. This gives two extra terms:

$$Vw^2/r = V(\bar{w}+\Omega_0 r\sin\theta)^2/r = V(\bar{w}^2 + 2\bar{w}\Omega_0 r\sin\theta + (\Omega_0 r \sin\theta)^2) \tag{10.29}$$

These are the centrigufal force terms. However, there is no reason why one could not keep the equation hybrid in w and $\bar{w}$, as long as the original w only appears outside of the derivatives or differences. If we use that strategy, then our radial momentum equation in the rotating frame becomes:

$$\bar{\partial}_t(\rho u V) + \Delta_r(F_{rr}S_r) + \Delta_\theta(F_{r\theta}S_\theta) + \Delta_\phi(\bar{F}_{r\phi}S_\phi) = \left[\frac{2P}{r} + \frac{\rho(v^2+w^2)}{r}\right] V \tag{10.30}$$

where we have left the original w in the source term (right hand side) and replaced $F_{r\phi}$ with $\bar{F}_{r\phi} = F_{r\phi} - \Omega_0 r\sin\theta\rho u = \rho u w - \Omega_0 r \sin\theta \rho u = \rho u \bar{w}$, or in other words, $\bar{F}_{r\phi}$ is the $F_{r\phi}$

Riemann flux as measured in the local moving reference frame (i.e. as seen by an observer moving with velocity $\Omega_0 r\sin\theta$. So it follows that the structure of the equation does not change. Just the left-hand-side now refers to fluxes in the moving fame, while the source terms on the right-hand-side still refer to the lab-frame w. One could also write out w^2 in terms of $\bar{w}$ as in Eq. (10.29).

Using the same philosophy we can derive the θ-momentum equation for the case of a rotating coordinate system:

$$\bar{\partial}_t(\rho vV) + \Delta_r(F_{\theta r}S_r) + \Delta_\theta(F_{\theta\theta}S_\theta) + \Delta_\phi(\bar{F}_{\theta\phi}S_\phi) = -\left[\rho\frac{vu}{r} - \frac{\cos\theta}{\sin\theta}\frac{\rho w^2 + P}{r}\right]V \quad (10.31)$$

where $\bar{F}_{\theta\phi} = F_{\theta\phi} - \Omega_0 r\sin\theta\rho v = \rho vw - \Omega_0 r\sin\theta\rho v = \rho v\bar{w}$ is the Riemann flux in the comoving frame.

Finally, for the angular momentum equation we get:

$$\bar{\partial}_t(\rho lV) + \Delta_r(\tilde{F}_{\phi r}S_r) + \Delta_\theta(\tilde{F}_{\phi\theta}S_\theta) + \Delta_\phi(\bar{\tilde{F}}_{\phi\phi}S_\phi) = 0 \quad (10.32)$$

where $\bar{\tilde{F}}_{\phi\phi} = \tilde{F}_{\phi\phi} - \Omega_0 r\sin\theta\rho l = \rho lw + Pr\sin\theta - \Omega_0 r\sin\theta\rho l = \rho l\bar{w} + Pr\sin\theta$. In contrast to $\bar{F}_{r\phi}$ and $\bar{F}_{\theta\phi}$, the $\bar{\tilde{F}}_{\phi\phi}$ is not just a multiplication of the comoving flux. Once we obtain the comoving Riemann flux $\bar{F}_{\phi\phi} = \rho\bar{w}^2 + P$ from the Riemann solver, here is how to obtain $\bar{\tilde{F}}_{\phi\phi}$:

$$\bar{\tilde{F}}_{\phi\phi} = r\sin\theta(\tilde{F}_{\phi\phi} + \rho\bar{w}r\sin\theta) \quad (10.33)$$

So we obtain the comoving Riemann flux $\tilde{F}_{\phi\phi}$, modify it with the above equation to obtain $\bar{\tilde{F}}_{\phi\phi}$ and insert this into the difference equation.

10.1.6 The FARGO-trick for modeling nearly Keplerian disks

Using a rotating polar coordinate system is very useful for instance for the study of an accretion disk around a star. There is, however, a slight problem. An accretion disk has a rotation velocity according to Kepler's law $\Omega_K(r) = \sqrt{GM/r^3}$, meaning it is *differentially rotating*. There is no single global Ω_0 that describes the main rotation of the fluid so that one can concentrate on perturbations on this disk. Instead, if one fixes the rotation rate of the grid Ω_0 to the Kepler frequency $\Omega_K(r_0)$ at radius r_0, then the accretion disk will have a comoving frame ϕ-velocity $\bar{w}$ that is about zero at r_0, but it will be strongly negative for $r \gg r_0$ and strongly positive for $r \ll r_0$. Since the highest characteristic velocities are the ones that limit the global time step, one may be forced to take very small time steps if the outer radius of the grid is much larger than r_0 or the inner radius much smaller than r_0. This may put a severe limitation on the spatial dynamic range.

The FARGO code of Frederic Masset solves this problem in an elegant way. The trick is to make a scheme that, for this special case, can advect over more than 1 cell per time step. This may sound peculiar, because if it were so easy, why did we do all the effort to design the advection algorithms in the previous chapters? The reason we had to do it there was that for normal hydrodynamics various characteristics move with different speeds and the same characteristics move with different speeds at different locations. Therefore it was not possible to find a single time step by which one could simply remap the quantity q_i to q_{i+3} for example.

Here we can partly use this remapping technique, while doing the rest using the normal advection schemes. Suppose that within a single time step the gas should move over a distance of 3.423 ϕ-grid cells in ϕ-direction. We can now perform the remapping trick: we first remap all quantities over precisely 3 grid points. What is left is an advection over 0.423 grid points, which is then done using our favorite advection scheme or a Riemann solver. The distance in number of grid cells to move in the remapping step will depend on radius r. The larger abs$(r - r_0)$, the larger the remapping step. The number of steps to take equals the distance of advection of the gas in the present time step, rounded to the nearest integer value. The CFL time step is now determined by requiring that, after the remapping step, none of the characteristics moves farther than 1 cell size in either direction.

10.2 Adaptive Mesh Refinement

Using different coordinate systems to fit best to the problem at hand is a useful technique for problems with a reasonably simple geometry. However, once the geometry becomes more complex, such as the geometry of a Giant Molecular Cloud with multiple dense cores, then polar coordinates do not help. In fact, they only make it more complex. One is then best advised to use 3-D Cartesian coordinates again. The problem is that these coordinates are not adapted to the problem at hand. The dense clumps may require a much finer mesh than the rest of the problem. Taking a fine mesh throughout the volume may be too computationally demanding.

One way to solve this problem is to use the technique of *mesh refinement*. This works by replacing a part of the course grid by a finer grid, possibly even recursively. One thus gets a patch-work hierarchical grid: Coarser grids higher up the hierarchy and finer grids down in the hierarchy. If these patches obey certain rules, for instance that each patch of refinement must either be fully inside a patch of lower resolution or fully outside, then the data management of these patches can be kept reasonably in check. A hierarchical grid is then built up as a tree. The base grid stands at the stem of the tree, and each patch of finer resolution mesh is a branch, which can then branch off itself into even finer patches.

There are several libraries that take care of such hierarchical grid refinement. For instance the CHOMBO library (written mostly in C++ but with some F77 parts in it) allows a very large freedom to choose the size and position of the patches. Another library, PARAMESH, is written in F90/F95 and has a somewhat more rigid structuring of the patches, with the advantage of efficiency. Here the highest level of refinement is a patch of $N_x \times N_y \times N_z$ grid cells. One can combine these into a larger block of $2 \times 2 \times 2$ of these grid patches, and this continues all the way up to the main (lowest resolution) grid.

We will not go into detail on the precise structure of such AMR libraries in this text. We will just briefly discuss how these things work.

In Fig.10.1 one can see how the two above described AMR techniques work geometrically.

To construct such grids requires rather complex data management. In a hierarchical structure, where levels of gridding are strictly based on sub-division of higher-level grids the data structure is typically a tree-like structure. And typically each level of refinement is an integer multiple of lower resolution grids, such that a cell at higher resolution will at most have one neighbor of equal or lower refinement level at each of its cell walls.

Communication between cells in an AMR tree is complex. The easiest way to do this is to pad each grid block with ghost cells (1 row for first order schemes, 2 rows for second order schemes etc). At the beginning of each time step one can then fill the ghost cells and all the complexity of the AMR data management is done in this ghost-cell filling. Once this is done,

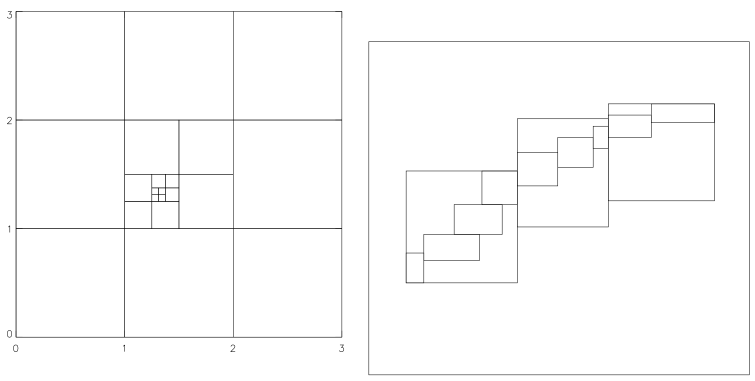

Figure 10.1. Illustration of AMR gridding. Left: A hierarchical cell-division type of AMR (e.g. Paramesh library). Right: AMR based on patches (e.g. Chombo library).

each block of cells can then be handled independently for one time step. This works very well (and is implemented in PARAMESH), but if the highest-refinement level consists of blocks of 1x1x1, then there is a huge overhead of ghost cellss: for a second order scheme in 2-D each cell is then padded with 24 ghost cells (and in 3-D it's even worse: 124). The easiest way out is so make the highest-refinement levels to consist of NxN cells, so that the ghost-cell overhead compared to the true cells is not so large anymore. If one wants to save memory and overhead, then a direct management of communication between true cells is desirable. This is, however, rather challenging from a programming perspective.

In any case, it is strongly advised to use existing libraries for the data-management of AMR, instead of writing such libraries oneself, because of the complexity of the matter.

10.3 Unstructured grids, or partly structured grids

For many purposes the boundary conditions of the problem at hand are very complex. For instance: flow in a complex pipe structure, or in a reactor of a factory, or the flow of gas over a car or airplane etc. For such problems it is important to allow very flexible gridding. There are many ways of doing this. The most flexible is the gridding using triangles in 2-D or tetraeders in 3-D (Fig. 10.2-left). The creation of good grids using this triangulation is non-trivial, and this is an entire field of engineering by itself. One of the main disadvantages of such completely flexible gridding is that it has difficulty making use of the highly parallel computer architectures now available, in particular architectures with extremely parallel graphics card accelerators. Such accelerators need simply structured grids to acquire substantial speeds-ups. For this reason the complex geometries of the grid are sometimes generated out of patches of regular but curved) coordinate systems patched together at their interfaces (see Fig. 10.2-right).

We will not go into details here.

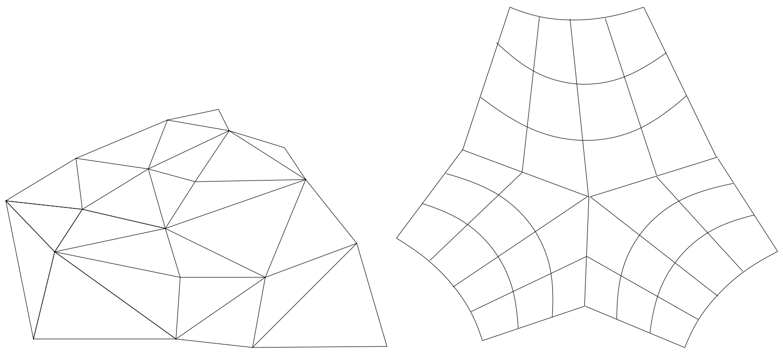

Figure 10.2. Illustration of unstructured gridding. Left: with triangles (or in 3-D: tetraeders). Right: with piecewise structured curved patches.