PART II. Incompressible flow

3. The Navier-Stokes equations.

Recall that our goal was to solve the incompressible momentum equation. For our purposes, we can think of momentum as a scalar which is being advected by an incompressible velocity field. The velocity field is divergence-free so that:

$$\nabla \cdot \underset{\sim}{u} = 0 \tag{1}$$

At the moment, we can actually handle the following model for the Navier-Stokes equation:

$$\frac{\partial s}{\partial t} + (\underset{\sim}{u} \cdot \nabla) s = \varepsilon \Delta s \tag{2}$$

where $\underset{\sim}{u} = \underset{\sim}{u}(\underset{\sim}{x})$ and ε is some diffusion coefficient. We know how to handle the advection and diffusion pieces separately; what remains is to explore hybrid approaches which allow us to combine these skills to solve equation (-1) above. The basic difference approximation is:

$$s_{ij}^{n+1} = s_{ij}^{n} - \Delta t (\underset{\sim}{u} \cdot \nabla s)_{ij}^{n+1/2} + \frac{\Delta t}{2} \varepsilon \Delta^h (s^n + s^{n+1})_{ij}$$

or, rewriting:

$$\left(I - \frac{\Delta t}{2} \varepsilon \Delta^h\right) s_{ij}^{n+1} = s_{ij}^{n} - \Delta t (\underset{\sim}{u} \cdot \nabla s)_{ij}^{n+1/2} + \frac{\Delta t}{2} \varepsilon \Delta^h s_{ij}^{n} \tag{3}$$

What are we doing here? We want to use Crank-Nicholson temporal differencing on the diffusion part of the equation since this scheme has desirable stability properties and can avoid gross mismatch of the spatial and temporal errors. We must require the advection scheme to have a compatible time step. We also want the advection part to be explicit to avoid its dependency on the (n+1)st time level. Furthermore, if the diffusion coefficient is zero, we want the hybrid scheme to reduce to second-order upwinding. Using the procedures which we by now have in place, we can do this. We have obtained a well-behaved system.

3.1 Treatment of advection term

In computing the advection terms, what we will actually compute is $\nabla \cdot (\underset{\sim}{u}s)^{n+1/2}$. This is justified because the velocity field is divergence-free. In the discrete formulation, we will assume that the scalar *s* lives at the cell centers. On the other hand, the velocity $\underset{\sim}{u}$ has components *(u, v)* which escort the scalar around the flowfield and will be defined at the edges of the cell as shown in figure +1:

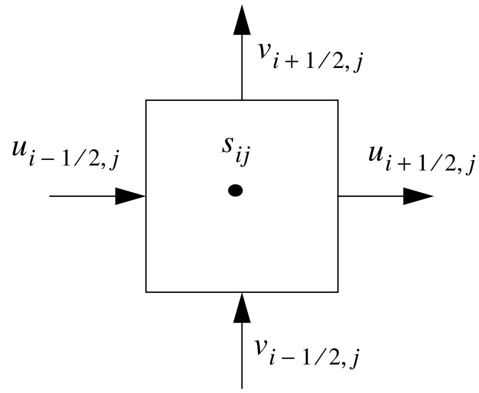

FIGURE 1. Representative cell

The divergence-free condition gives the following first-order approximation:

$$\frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta y} = 0 \tag{4}$$

If the function $\underset{\sim}{u}(\underset{\sim}{x})$ is known analytically, then we can think of the velocity at the right edge as an average integrated value across the cell edge, i.e.,

$$u_{i+1/2,j} = \frac{1}{\Delta y} \int_{(j-\frac{1}{2})\Delta y}^{(j+\frac{1}{2})\Delta y} u((i+1/2)\Delta x, y)\, dy \tag{5}$$

This satisfies the divergence-free condition by the divergence theorem. Note that the r.h.s of equation (4) is one of four terms of the r.hs. of:

$$\int_V (\nabla \cdot \underset{\sim}{u})\, dV = \oint_S (\underset{\sim}{u} \cdot \hat{n})\, dA \tag{6}$$

We can now write a finite-difference approximation for $\nabla \cdot (\underset{\sim}{u}s)^{n+1/2}$. Recalling that we have specified that $\underset{\sim}{u} = \underset{\sim}{u}(\underset{\sim}{x})$ alone:

$$\nabla \cdot (\underset{\sim}{u}s) = \frac{1}{\Delta x}(u_{i+1/2,j}s^{n+1/2}_{i+1/2,j} - u_{i-1/2,j}s^{n+1/2}_{i-1/2,j}) +$$

$$\frac{1}{\Delta y}\left(v_{i,j+1/2}s^{n+1/2}_{i,j+1/2} - v_{i,j-1/2}s^{n+1/2}_{i,j-1/2}\right) \tag{7}$$

Now, recall that the discretized version of s exists at cell centers. To extrapolate to the cell edges, perform a Taylor expansion:

$$s^{n+1/2}_{i+1/2,j} \approx s^n_{ij} + \frac{\Delta x}{2}\frac{\partial s}{\partial x} + \frac{\Delta t}{2}\frac{\partial s}{\partial t} \tag{8}$$

We were able to write the PDE as

$$\frac{\partial s}{\partial t} = \nabla \cdot (\underset{\sim}{u} s) + \varepsilon \Delta s$$

or, for our case of two-dimensional Cartesian coordinates:

$$\frac{\partial s}{\partial t} = \frac{\partial (us)}{\partial x} + \frac{\partial (vs)}{\partial y} + \varepsilon \Delta s \tag{9}$$

Putting equation (9) into equation (8), we obtain:

$$s^{n+1/2}_{i+1/2,j} = s^n_{ij} + \frac{\Delta x}{2}\frac{\partial s}{\partial x} - \frac{u\Delta t}{2}\frac{\partial s}{\partial x} - \frac{\Delta t}{2} s \frac{\partial u}{\partial x} - \frac{\partial (vs)}{\partial y} - \varepsilon \frac{\Delta t}{2}\Delta s$$

or

$$s^{n+1/2}_{i+1/2,j} = s^n_{ij} + \left(1 - \frac{u\Delta t}{\Delta x}\right)\frac{\Delta x}{2}\frac{\partial s}{\partial x} - \frac{\Delta t}{2} s \frac{\partial u}{\partial x} - \frac{\Delta t}{2}\frac{\partial (vs)}{\partial y} - \varepsilon \frac{\Delta t}{2}\Delta s \tag{10}$$

The fully discretized form of this equation is:

$$\begin{aligned} s^{n+1/2}_{i+1/2,j} = {} & s^n_{ij} + \frac{1}{2}\left(1 - \frac{u_{i+1/2,j}\Delta t}{\Delta x}\right)(\Delta_x s)_{ij} - \frac{\Delta t}{2\Delta x} s^n_{ij}\left(u_{i+1/2,j} - u_{i-1/2,j}\right) - \\ & \frac{\Delta t}{2\Delta y}\left(v_{i,j+1/2}s^{up}_{i,j+1/2} - v_{i,j-1/2}s^{up}_{i,j-1/2}\right) - \varepsilon \frac{\Delta t}{2}\left(\Delta^h s^n\right)_{ij} \end{aligned} \tag{11}$$

where the upwind value of s was denoted by s^{up}, i.e.

$$s^{up}_{i,j+1/2} = \begin{cases} s^n_{ij} \ldots if \ldots (v_{i,j+1/2} > 0) \\ s^n_{i,j+1} \ldots if \ldots (v_{i,j+1/2} < 0) \end{cases} \tag{12}$$

There are several comments to be made regarding this approach.

- Should we be disturbed by putting in an explicit estimate of $(\varepsilon\Delta t\Delta s)/2$? The answer is no, since the Crank-Nicholson scheme gives the requisite stability, as is easily verified by computing the symbol of the scheme.

- What happens if $s_{ij} = s_0$, a constant, in the neighborhood of s_{ij}? In this case, in the computation of $s^{n+1/2}_{i+1/2,j}$, the term yielding $\partial v/\partial y \neq 0$, which looks like trouble. However, the terms

$$\frac{\Delta t}{2}\frac{\partial(vs)}{\partial y} + \frac{\Delta t}{2}s\frac{\partial u}{\partial x}$$

  cancel each other out because of the divergence-free condition. So, this case is not a problem. In the jargon of aerodynamics, it's *freestream-preserving*.
- What about stability? The CFL condition for the advective part is

$$\left|\frac{\Delta t}{\Delta x}u_{i+1/2,j}\right|, \left|\frac{\Delta t}{\Delta x}v_{i,j+1/2}\right| \leq 1 \tag{13}$$

  and recall that $\Delta t = O(\Delta x, \Delta y)$. To get accuracy out of the advection part, we want to limit the CFL numbers between braces above to the range of 0.5 to 1.0. Crank-Nicholson allows us to take a time step that is limited only by the advective CFL condition.

3.2 The incompressible Navier-Stokes equations

We would like to solve for the two-dimensional velocity field $\underset{\sim}{u}$ on the bounded domain Ω with fixed, impermeable solid-wall boundaries. For now. The governing equations are continuity and momentum, which can be written:

$$\nabla \cdot \underset{\sim}{u} = 0 \tag{14}$$

$$\frac{\partial \underset{\sim}{u}}{\partial t} + (\underset{\sim}{u} \cdot \nabla)\underset{\sim}{u} = -\frac{1}{\rho}\nabla p + \nu\Delta\underset{\sim}{u} \tag{15}$$

The boundary conditions are either no-slip:

$$\underset{\sim}{u} = 0 \ldots on \ldots \partial D \tag{16}$$

or, for inviscid walls:

$$\underset{\sim}{u} \cdot \hat{n} = 0 \ldots on \ldots \partial D \tag{17}$$

To simplify notation, let's take our units so that ρ = constant. We realize that p is not a separately specified dependent variable; if we know $\underset{\sim}{u}(\underset{\sim}{x}, t)$, we can compute p: formally, the divergence of the momentum equation yields a Poisson equation for the pressure.Taking this formal observation to seriously as a perscription for solving the problem can lead to some serious conceptual and technical difficulties; to avoid those difficulties, we take a slightly indirect approach.

3.2.1 Definition of divergence, gradient and the inner product

We begin with somedefinitions. For any two-dimensional vector field, the divergence of $\underset{\sim}{u} = (u, v)$ is given by:

$$div(\underset{\sim}{u}) = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \quad (18)$$

For any scalar field in two dimensions, the gradient is given by:

$$grad(\phi) = (\phi_x, \phi_y) = (\frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial y}) \quad (19)$$

We observe, with the aid the divergence theorem and the Leibnitz rule, that:

$$\int_D \phi div(\underset{\sim}{u})\, dV = \int_D div(\phi \underset{\sim}{u})\, dV - \int_D grad(\phi) \cdot \underset{\sim}{u} dV$$

$$\int_D \phi div(\underset{\sim}{u})\, dV = \oint_{\partial D} \phi(\underset{\sim}{u} \cdot \hat{n})\, dA - \int_D grad(\phi) \cdot \underset{\sim}{u} dV \quad (20)$$

The solid-wall boundary condition shows that $\underset{\sim}{u} \cdot \hat{n} = 0$ on ∂D, so that for solid walls:

$$\int_D \phi div(\underset{\sim}{u})\, dV = -\int_D grad(\phi) \cdot \underset{\sim}{u} dV \quad (21)$$

The inner product is defined for two vectors as:

$$\langle \underset{\sim}{v}, \underset{\sim}{w} \rangle_V = \int_D \underset{\sim}{v}(\underset{\sim}{x}) \cdot \underset{\sim}{w}(\underset{\sim}{x})\, dV \quad (22)$$

while for two scalars, the inner product is:

$$\langle \phi, \psi \rangle_S = \int_D \phi(\underset{\sim}{x}) \cdot \psi(\underset{\sim}{x})\, dV \quad (23)$$

We note that, for solid walls:

$$\langle div(\underset{\sim}{u}), \psi \rangle_S = -\langle \underset{\sim}{u}, grad(\phi) \rangle_V \quad (24)$$

3.2.2 Hodge (Helmholtz) decomposition

Given the vector field, $\underset{\sim}{w}$, on a simply connected domain, *D,* which can be described by:

$$\underset{\sim}{w} = \underset{\sim}{w}(\underset{\sim}{x}) \ldots in \ldots D \quad (25)$$

$$\underset{\sim}{w} \cdot \hat{n} = 0 \ldots on \ldots \partial D \quad (26)$$

there is a unique decomposition of $\underset{\sim}{w}$ into a divergence-free part, $\underset{\sim}{w}_d$, and the gradient of some scalar function, ϕ:

$$\underset{\sim}{w} = \underset{\sim}{w}_d + grad(\phi) \tag{27}$$

To get a handle on what this $\underset{\sim}{w}_d$ is, take the divergence of equation (27). We realize that by definition $div(\underset{\sim}{w}_d) = 0$. This means that in order to solve

$$div(\underset{\sim}{w}) = div(grad(\phi)) \tag{28}$$

we can find $\underset{\sim}{w}_d$ by:

$$\underset{\sim}{w}_d = \underset{\sim}{w} - grad(\phi) \tag{29}$$

If we take the divergence of equation (29) above, we see that the divergence of $\underset{\sim}{w}_d$ vanishes:

$$div(\underset{\sim}{w}_d) = div(\underset{\sim}{w}) - div(grad(\phi))$$

The terms on the r.h.s cancel each other out, as shown in equation (-2) above, leaving a big fat goose egg.

We can also see that the components of $\underset{\sim}{w}$ are orthogonal, i.e.:

$$\langle \underset{\sim}{w}_d, grad(\phi) \rangle_V = \int_D \underset{\sim}{w}_d \cdot grad(\phi)\, dV$$

Now, using the no-normal flow boundary condition

$$\begin{aligned} \langle \underset{\sim}{w}_d, grad(\phi) \rangle_V &= -\int_D div \underset{\sim}{w}_d \phi dV \\ &= 0 \end{aligned}$$

The orthogonality comes from the fact that divergence and gradient are adjoints.

3.2.3 The projection operator

One of the difficulties of working with the incompressible Navier-Stokes equation is that we must both satisfy the divergence-free condition imposed by continuity and find the pressure gradient. We can in a sense blend these constraints by using a projection operator which forces the velocity field to be divergence-free; whatever remains from the right-hand side of the momentum equation is absorbed intothe pressure gradient. Another advantage of the projection formulation is that it leads to pure initial value formulation for $\underset{\sim}{u}$.

We can define a projection operator, P, as something which takes a vector field and extracts its divergence-free part. So, we can hit the vector field $\underset{\sim}{w}$ with the projection operator to obtain $\underset{\sim}{w}_d$. If $\underset{\sim}{w}$ is sufficiently smooth, from equations (27) and (29):

$$\underset{\sim}{w}_d = P\underset{\sim}{w} = (I - grad(div grad)^{-1} div)\underset{\sim}{w} \tag{30}$$

P can be extended on $L^2(D)$. It is a way of extracting the divergence-free part, even if we don't have enough derivatives for the individual operations in (27) and (29) to be well-defined.

Some properties of the projection operator are that it is symmetric;

$$\langle \underset{\sim}{w}, P\underset{\sim}{w} \rangle = \langle P\underset{\sim}{w}, \underset{\sim}{w} \rangle \tag{31}$$

$$P = P^T \tag{32}$$

and bounded:

$$\| P \| = 1 \tag{33}$$

$$\| P\underset{\sim}{w} \| \leq \| \underset{\sim}{w} \| \tag{34}$$

Also, since the decomposition is unique, P is idempotent.

$$P^2 = P \tag{35}$$

What do projection operators look like in space? If we set the domain to be $\Re^3$ and $\underset{\sim}{w}$ is of compact support, then we can write $\underset{\sim}{w}_d$ as:

$$\underset{\sim}{w}_d = \int_{\Re^3} K(\underset{\sim}{x} - \underset{\sim}{x}')\, \underset{\sim}{w}(\underset{\sim}{x})\, d\underset{\sim}{x} \tag{36}$$

where

$$K(\underset{\sim}{x}) = \delta(\underset{\sim}{x}) I - \frac{1}{4\pi} \left(\frac{I}{|\underset{\sim}{x}|^3} - \frac{3\underset{\sim}{x}\underset{\sim}{x}^T}{|\underset{\sim}{x}|^3} \right) \tag{37}$$

For bounded domains, we want a projection formulation of the Navier-Stokes equation:

$$\frac{\partial \underset{\sim}{u}}{\partial t} + (\underset{\sim}{u} \cdot \nabla)\, \underset{\sim}{u} = -\nabla p + \nu \Delta \underset{\sim}{u} \tag{38}$$

subject to the no-slip boundary condition:

$$\underset{\sim}{u} = 0 \ldots on \ldots \partial D \tag{39}$$

or, alternatively, if $\nu = 0$, we will impose the no-normal flow condition:

$$\underset{\sim}{u} \cdot \hat{n} = 0 \ldots on \ldots \partial D \tag{40}$$

First, apply the projection operator to equation (30):

$$P\left(\frac{\partial \underset{\sim}{u}}{\partial t}\right) + P\left((\underset{\sim}{u} \cdot \nabla)\, \underset{\sim}{u}\right) = P(-\nabla p) + P(\nu \Delta \underset{\sim}{u})$$

The projection of a pure gradient is zero, so that $P(-\nabla p) = 0$. Rewriting,

$$\frac{\partial \underset{\sim}{u}}{\partial t} + P\left(\left(\underset{\sim}{u} \cdot \nabla \right) \underset{\sim}{u} - \nu \Delta \underset{\sim}{u} \right) = 0 \quad (41)$$

with continuity automatically satisfied ($\nabla \cdot \underset{\sim}{u} = 0$). If we compare equation (41) to the original equation (36), we arrive at an expression for the pressure gradient:

$$\nabla p = (I - P) \left(-\left(\underset{\sim}{u} \cdot \nabla \right) \underset{\sim}{u} + \nu \Delta \underset{\sim}{u} \right) \quad (42)$$

We see from equation (38) above that the pressure gradient absorbs any divergence that results from the piece of $\frac{\partial u}{\partial t}$ which is not divergence-free.

We might also be tempted eliminate the viscous terms from (42) using

$$P \nu \Delta \underset{\sim}{u} = \nu \Delta P \underset{\sim}{u}$$

Since we would expect that the divergence and gradient operators commute with Δ. This is valid in infinite space, but is not correct in the presence of no-slip boundaries. In the latter case, the effect of the viscous terms, combined with the boundary conditions, is to cause the velocity field to move off of the subspace divergence-free velocity fields.

3.3 The projection method

Let's examine the discretization of the projection formulation in two dimensions. Assume that we have discretizations for the advective and diffusive parts which are stable, satisfy the boundary conditions, and are denoted by the following:

$$A_{ij} \approx \left(\underset{\sim}{u} \cdot \nabla \right) \underset{\sim}{u} \quad (43)$$

$$\left(\Delta^h \underset{\sim}{u} \right)_{ij} \approx \Delta \underset{\sim}{u} \quad (44)$$

The discrete evolution equation can be written:

$$u_{ij}^{n+1} = u_{ij}^{n} + \Delta t \left(P \left(-A + \nu \Delta^h \underset{\sim}{u} \right) \right)_{ij} \quad (45)$$

where P is a discretization of a continuous projection operator. Typically, we will use a fractional-step method:

$$\underset{\sim}{u}_{ij}^{*} = \underset{\sim}{u}^{n} - \Delta t A_{ij} + \Delta t \nu \left(\Delta^h \underset{\sim}{u} \right)_{ij} \quad (46)$$

$$\underset{\sim}{u}_{ij}^{n+1} = P \underset{\sim}{u}_{ij}^{*}$$

But what exactly is the projection operator? Let's assume we're operating in a discrete domain $\Omega = [1 \ldots N] x [1 \ldots N]$ with either solid walls, for which:

$$\underset{\sim}{u} \cdot \hat{n} = 0 \ldots on \ldots \partial \Omega \quad (47)$$

or doubly periodic boundaries. Our strategy will be to:

(1) Define the discrete divergence D using finite differences; and

(2) Define the discrete gradient $G = -D^T$ with respect to some pair of inner products

We define the inner products for vector fields as:

$$\langle \underset{\sim}{v}, \underset{\sim}{w} \rangle_V = h^2 \sum_{ij} \underset{\sim}{v}_{ij} \cdot \underset{\sim}{w}_{ij} \tag{48}$$

and for scalar fields as:

$$\langle \phi, \psi \rangle_S = h^2 \sum_{ij} \phi_{ij} \cdot \psi_{ij} \tag{49}$$

As a preliminary example, we will define discrete analogs of the divergence, gradient and projection operators based on centered difference approximations in the following sections. Ultimately, though, these will prove unsatisfactory for a variety of reasons,an we will replace them with more appropriate objects.

3.3.1 The discrete divergence

If the components of $\underset{\sim}{u} = (u, v)$, for the cell *(i,j)* which is sufficiently removed from the boundaries, we can define the discrete divergence by:

$$(D\underset{\sim}{u})_{ij} = \frac{u_{i+1/2,j} - u_{i-1/2,j}}{h} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{h} \tag{50}$$

where the cell-edge velocities are assumed to be the average of the two adjacent cell centers:

$$u_{i+1/2,j} = \frac{u_{ij} + u_{i+1,j}}{2} \tag{51}$$

and

$$v_{i,j+1/2} = \frac{v_{ij} + v_{i,j+1}}{2} \tag{52}$$

In the interior, this discretization reduces to centered differences:

$$(D\underset{\sim}{u})_{ij} = \frac{u_{i+1,j} - u_{i-1,j}}{2h} + \frac{v_{i,j+1} - v_{i,j-1}}{2h} \tag{53}$$

For the boundary cells with solid walls, e.g.,

$$(D\underset{\sim}{u})_{1,j} = \frac{u_{2,j} + u_{1,j}}{2h} + \frac{v_{1,j+1/2} - v_{1,j-1/2}}{h} \tag{54}$$

or for periodic boundary conditions:

$$(D\underset{\sim}{u})_{1,j} = \frac{u_{2,j} - u_{N,j}}{2h} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{h} \tag{55}$$

3.3.2 The discrete gradient

Next, we want to find the discrete gradient. For the continuous case, we know that the boundary terms vanish ($\underset{\sim}{u} \cdot \hat{n} = 0$) and that the velocity field is divergence-free ($\nabla \cdot \underset{\sim}{u} = 0$). The relationship between the vector fields, the adjoint condition and the scalar inner product will define the discrete gradient.

For the continuous case, recall that:

$$\int_D \phi\, div(\underset{\sim}{u})\, dV = -\int_D grad(\phi) \cdot \underset{\sim}{u}\, dV \tag{56}$$

We would also like this to hold for the discrete case so that:

$$\langle D\underset{\sim}{u}, \phi\rangle_S = -\langle \underset{\sim}{u}, G\phi\rangle_V \tag{57}$$

or

$$G = -D^T \tag{58}$$

How do we compute the discrete gradient in the *x*-direction, $(G\phi)^x_{ij}$? We will need to use the adjoint relation. Let $\hat{e}^x_{kl}$ be a vector field defined on Ω by

$$\hat{e}^x_{kl;ij} = \begin{cases} (1,0)\ \ldots if\ldots\ (i,j) = (k,l) \\ (0,0)\ \ldots otherwise \end{cases} \tag{59}$$

The function of $\hat{e}^x_{kl}$ is to pick out the *x* component of the gradient at *(k, l)*. From the adjoint relation, we know that:

$$\begin{aligned} \langle D\hat{e}^x_{kl}, \phi\rangle_S &= -\langle \hat{e}^x_{kl}, G\phi\rangle_V \\ &= -h^2 \sum_{ij} \hat{e}^x_{kl;ij} \cdot (G\phi)_{ij} \\ &= -h^2 (1,0)\,(G\phi)_{kl} \\ &= -h^2 (G\phi)^x_{kl} \end{aligned}$$

so that we can find the gradient by:

$$(G\phi)^x_{ij} = -\frac{1}{h^2}\langle D\hat{e}^x_{ij}, \phi\rangle_S \tag{60}$$

For the *y* direction, we find that:

$$(G\phi)^y_{kl} = -\frac{1}{h^2}\langle D\hat{e}^y_{kl}, \phi\rangle_S \tag{61}$$

where

$$\hat{e}^y_{kl;ij} = \begin{cases} (0,1) \ldots if \ldots (i,j) = (k,l) \\ (0,0) \ldots otherwise \end{cases} \tag{62}$$

3.3.3 The discrete projection operator

In order to define the discrete projection operator, apply the discrete divergence to the definition of the Hodge decomposition of the vector field $\underset{\sim}{w}$:

$$D\underset{\sim}{w} = D(\underset{\sim}{w}_d + G\phi)$$

We know that $D\underset{\sim}{w}_d = 0$ since this component is divergence free. Then we can say that:

$$D\underset{\sim}{w} = DG\phi \tag{63}$$

If we can solve equation (-1) for $G\phi$, then we can solve for $\underset{\sim}{w}_d$ by:

$$\underset{\sim}{w}_d = \underset{\sim}{w} - G\phi \tag{64}$$

We note that:

$$\begin{aligned} \langle \underset{\sim}{w}_d, G\phi\rangle_V &= -\langle D\underset{\sim}{w}_d, \phi\rangle_S \\ &= 0 \end{aligned}$$

since $D\underset{\sim}{w}_d$ is by definition zero. This indicates that $\underset{\sim}{w}_d$ is orthogonal to $G\phi$.

The definition of the discrete projection operator is:

$$P = (I - G(DG)^{-1}D) \tag{65}$$

Just as for the continuous projection operator, the discrete projection is symmetric, bounded, and idempotent.

. Can we be sure that equation (63) is always solvable? We must examine this question through consideration of solvability conditions and the Fredholm alternative. The Fredholm alternative tells us that $L\phi = \rho$ is solvable if the statement that $L^T\pi = 0$ implies that π and ρ are orthogonal, i.e., that ρ is normal to the kernel of L. For our particular case, we have shown that $D = -G^T$. Let's swap the notation of equation (-3) to emphasize this result. If we denote the discretized divergence matrix as the rectangular matrix A, then the discretized gradient matrix is $-A^T$. Then, equation (63) has the form:

$$-AA^T\Phi = Af \tag{66}$$

We also know that in our case, A is symmetric so that $(-AA^T)^T = -AA^T$. The Fredholm alternative says that if $AA^T\Phi = 0$, then $\langle\Phi, Af\rangle = 0$. Let's use as a starting point our statement that $AA^T\Phi = 0$:

$$\begin{aligned} 0 &= \langle AA^T\Phi, \Phi\rangle \\ &= \langle A^T\Phi, A^T\Phi\rangle \\ &= \left\|A^T\Phi\right\|^2 \end{aligned}$$

This means that $A^T\Phi$ is itself zero. (This is equivalent to saying that if $DG\phi = 0$, then $G\phi = 0$.) So, we have shown that

$$\begin{aligned} \langle\Phi, Af\rangle &= \langle A^T\Phi, f\rangle \\ &= \langle 0, f\rangle \\ &= 0 \end{aligned}$$

Thus, $D\underset{\sim}{w} = DG\phi$ is always solvable because of the adjoint condition $D = -G^T$. In particular, iterative methods have a chance of working, even though the operator is not invertible.

3.3.4 Chorin's projection method.

In Chorin's original version of the projection method, we are given $\underset{\sim}{u}$ as initial data. We will solve the discretized equation

$$D\underset{\sim}{u} = DG\phi \tag{67}$$

for ϕ. We know that this equation is solvable because D and G are adjoints. Next, we calculate the divergence-free velocity by:

$$\underset{\sim}{u}_d = \underset{\sim}{u} - G\phi \tag{68}$$

The evolution equation is:

$$u_{ij}^{*} = \underset{\sim}{u}^n - \Delta t A_{ij} + \Delta t \nu (\Delta^h \underset{\sim}{u})_{ij} \tag{69}$$

where $A = A(\underset{\sim}{u})$ and is the discrete analog of $(\underset{\sim}{u} \cdot \nabla)\underset{\sim}{u}$. Then

$$\underset{\sim}{u}^{n+1} = P\underset{\sim}{u}^{*} \tag{70}$$

where P is the discrete projection operator.

Between two time levels, we can define the gradient of pressure as:

$$(Gp)^{n, n+1} \approx \frac{\underset{\sim}{u}^{*} - \underset{\sim}{u}^{n+1}}{\Delta t} = (I - P)\left(\frac{\underset{\sim}{u}^{*} - \underset{\sim}{u}^{n}}{\Delta t}\right) \tag{71}$$

This is an intrinsically first-order method..

3.3.5 The second-order projection method

We would like to increase the accuracy of the projection to $O(\Delta t^2)$. By analogy with the advection/diffusion equation, we can again use some sort of hybrid scheme which handles different pieces of the discretized equation in different ways. We had some success in using Crank-Nicholson previously, so let's examine what good this method would do here. For this case, the evolution equation is:

$$\underset{\sim}{u}^{n+1} = \underset{\sim}{u}^n + \Delta t P\left(-A(\underset{\sim}{u})^{n+1/2} + \frac{\nu}{2}\Delta^h(\underset{\sim}{u}^n + \underset{\sim}{u}^{n+1})\right) \tag{72}$$

The first term in brackets represents the discretization of the $(\underset{\sim}{u} \cdot \nabla)\underset{\sim}{u}$ in the Navier-Stokes equation. Our experience obtained from studying the scalar advection - diffusion equation indicates that we should use an explicit hyperbolic predictor/corrector method on this piece of the equation. The second term in brackets is the diffusive part. We note immediately that there is a spot of trouble here: grouping together terms which are dependent on the new time level leads to having to solve a problem of the form

$$\left(I - \frac{\nu\Delta t}{2}P\Delta^h\right)\underset{\sim}{u}^{n+1} = RHS.$$

The presence of the projection operator here is a serious problem. We could solve for this iteratively, but with every iteration, we would have to do a lot of work in solving linear systems ($D\underset{\sim}{u} = DG\phi$). In addition, applying the projection operator to Δ^h is rather ugly and can lead to conditioning problems. The fix is to assume at the beginning of the time step that we know both $\underset{\sim}{u}^n$ and a time-centered value: $(Gp)^{n-1/2}$. The basic idea is to first compute in step 1:

Step 1

$$\underset{\sim}{u}^* = \underset{\sim}{u}^n + \Delta t\left[-A(\underset{\sim}{u})^{n+1/2} + \frac{\nu}{2}\Delta^h(\underset{\sim}{u}^n + \underset{\sim}{u}^*) - (Gp)^{n-1/2}\right] \tag{73}$$

followed by step 2:

Step 2

$$\underset{\sim}{u}^{n+1} = P\underset{\sim}{u}^* \tag{74}$$

$$(Gp)^{n+1/2} = (Gp)^{n-1/2} + \frac{\underset{\sim}{u}^* - \underset{\sim}{u}^{n+1}}{\Delta t} \tag{75}$$

$(Gp)^{n+1/2}$ can also be written as:

$$(Gp)^{n+1/2} = -(I-P)\left[-A(\underset{\sim}{u})^{n+1/2} + \frac{\nu}{2}\Delta^h(\underset{\sim}{u}^n + \underset{\sim}{u}^*)\right] \tag{76}$$

To compute $\underset{\sim}{u}^*$ in step 1, we assumed that we knew $(Gp)^{n-1/2}$, and also that $A(\underset{\sim}{u})^{n+1/2}$ was explicit. These two assumptions are easy to satisfy. Also, to solve for the incremental value $\underset{\sim}{u}^*$, we had to solve a system of the form:

$$\left(I - \frac{\nu\Delta t}{2}\Delta^h\right)\underset{\sim}{u}^* = RHS \tag{77}$$

This is not a problem either, and in fact converges faster than the Poisson equation .

What about step 2? For the projection, we needed to solve for the discrete divergence, D, which is given by:

$$(D\underset{\sim}{u})_{ij} = \frac{u_{i+1,j} - u_{i-1,j}}{2h} + \frac{v_{i,j+1} - v_{i,j-1}}{2h} \tag{78}$$

and the discrete gradient $G = -D^T$. Mathematically, $DG\phi$ looks like:

$$DG\phi = \frac{1}{4h^2}\left[\phi_{i+2,j} + \phi_{i,j+2} + \phi_{i-2,j} + \phi_{i,j-2} - 4\phi_{ij}\right] \tag{79}$$

The stencil for $DG\phi$ looks like:

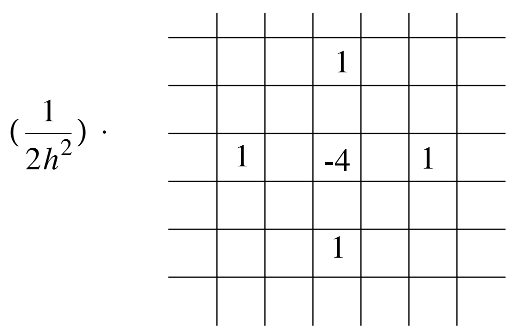

FIGURE 2. Stencil for DGϕ

This looks somewhat like an extended five-point Laplacian, although it will be a little more tricky to implement multigrid and will require additional care at the boundaries. However, it is now a matter of applying known techniques, ultimately boiling down to easy linear algebra.

To verify the second-order accuracy of this method, let's denote the time level by n and the iteration level by l. Recall that we needed a guess for pressure gradient which lags the velocity field by half a time step. We set the first guess for gradient of pressure:

$$(Gp)^{n+1/2,0} = (Gp)^{n-1/2} \tag{80}$$

Then solve for the velocity by the fractional step method:

$$\underset{\sim}{u}^{*l} = \underset{\sim}{u}^n + \Delta t\left[-A(\underset{\sim}{u})^{n+1/2} + \frac{\nu}{2}\Delta^h(\underset{\sim}{u}^n + \underset{\sim}{u}^*)^l - (Gp)^{n-1/2,l}\right] \tag{81}$$

and in the second step:

$$\underset{\sim}{u}^{n+1} = P\underset{\sim}{u}^{*l} \tag{82}$$

$$(Gp)^{n+1/2,l+1} = (Gp)^{n-1/2,l} + \frac{1}{\Delta t}(I-P)(\underset{\sim}{u}^{*l} - \underset{\sim}{u}^{n+1,l+1}) \tag{83}$$

Let's contrast the Bell-Colella-Glaz formulation above with the Crank-Nicholson approach. When starting at the same time, Crank-Nicholson evolves as follows:

$$\underset{\sim}{u}^{CN} = \underset{\sim}{u}^n + \Delta t\left[-A(\underset{\sim}{u})^{n+1/2} + \frac{\nu}{2}\Delta^h(\underset{\sim}{u}^n + \underset{\sim}{u}^{CN}) - (Gp)^{CN}\right] \tag{84}$$

Comparing equations (-1) and (-4):

$$\underset{\sim}{u}^{CN} - \underset{\sim}{u}^* = \frac{\nu}{2}\Delta^h(\underset{\sim}{u}^{CN} - \underset{\sim}{u}^*) - (Gp)^{CN} + (Gp)^{n-1/2}$$

Rewriting

$$\left(I - \frac{\nu\Delta t}{2}\Delta^h\right)(\underset{\sim}{u}^{CN} - \underset{\sim}{u}^*) = -\Delta t\left((Gp)^{CN} - (Gp)^{n-1/2}\right)$$

$$= O(\Delta t^2)$$

since the pressure gradient can only accumulate error of *O*(Δt) in one time step. But what we'd really like to compare is $\underset{\sim}{u}^{CN} - P\underset{\sim}{u}^*$:

$$\underset{\sim}{u}^{CN} - \underset{\sim}{u}^{n+1} = P\left(\frac{\nu\Delta t}{2}\Delta^h(\underset{\sim}{u}^{CN} - \underset{\sim}{u}^*)\right) \tag{85}$$

and we have just shown above that the quantity of $\underset{\sim}{u}^{CN} - \underset{\sim}{u}^* = O(\Delta t^2)$. Numerical simulations also indicate that this scheme is of $O(\Delta t^2)$.

The tricky thing to compute in this technique is the advection term, $A^{n+1/2}$. We will use Godunov methodology, that is, extrapolate for *u* at the cell edge from the adjacent cell centers and find the upwind state. So, for the predictor step, we will extrapolate for velocity at, e.g., the right cell edge:

$$\underset{\sim}{u}^{n+1/2}_{i+1/2,j} = \underset{\sim}{u}^n_{ij} + \frac{\Delta x}{2}\frac{\partial \underset{\sim}{u}}{\partial x} - \frac{\Delta t}{2}u\frac{\partial \underset{\sim}{u}}{\partial x} - \frac{\Delta t}{2}v\frac{\partial \underset{\sim}{u}}{\partial y}$$

$$+\frac{\nu\Delta t}{2}\Delta\underset{\sim}{u}-\frac{\Delta t}{2}\nabla p+O(\Delta x^2,\Delta t^2)$$

or:

$$\underset{\sim}{u}^{n+1/2}_{i+1/2,j} = \underset{\sim}{u}^{n}_{ij}+\frac{1}{2}\left[1-\underset{\sim}{u}^{n}_{ij}\frac{\Delta t}{\Delta x}\right]\frac{\partial\underset{\sim}{u}}{\partial x}\bigg|_{ij}\Delta x-\frac{\Delta t}{2}v_{ij}[u]^{y}_{upwind}$$

$$+\left(\frac{\nu\Delta t}{2}(\Delta^h\underset{\sim}{u}^n)_{ij}-\frac{\Delta t}{2}(Gp)^{n-1/2}\right) \tag{86}$$

where the second term on the r.h.s. incorporates geometric limiting (cf. section xxx.xx.x); the third term uses the jump in the velocity component, *[u]*, in the *y* direction, choosing the upwind state (w.r.t. v_{ij}); and the fifth term approximates the pressure gradient term with accuracy up to $O(\Delta t^2)$.

There are a still a couple of problems with this approach which need to be resolved. For one, the use of the term $(Gp)^{n-1/2}$ in the predictor introduces a nonlinear instability, requiring a CFL condition on time step. We need:

$$\frac{|u|\Delta t}{\Delta x},\frac{|v|\Delta t}{\Delta x}\le 0.5 \tag{87}$$

Defining the velocity unambiguously on the cell edges also creates some trouble because there is no reason to expect that $u_{i+1}-u_i = v_{j+1}-v_i$. As an example, advection of a scalar like density is given by:

$$\rho^{n+1}_{ij}=\rho^n_{ij}+\frac{\Delta t}{\Delta x}[u^{n+1/2}_{i-1/2,j}\rho^{n+1/2}_{i-1/2,j}-u^{n+1/2}_{i+1/2,j}\rho^{n+1/2}_{i+1/2,j}]+$$

$$+\frac{\Delta t}{\Delta y}[v^{n+1/2}_{i,j-1/2}\rho^{n+1/2}_{i,j-1/2}-v^{n+1/2}_{i,j+1/2}\rho^{n+1/2}_{i,j+1/2}] \tag{88}$$

But if the density is constant everywhere, i.e., $\rho^n\equiv\rho_0$, then we would like it to remain constant in time so that $\rho^{n+1}\equiv\rho_0$. However, this is not going to happen unless:

$$\frac{u_{i+1/2,j}-u_{i-1/2,j}}{\Delta x}+\frac{v_{i,j+1/2}-v_{i,j-1/2}}{\Delta y} = 0 \tag{89}$$

This version of the BCG technique does not guarantee this condition; thus, continuity may be violated. There are other implications for the momentum equation.

Reference: Bell, Colella and Glaz. *Journal of Computational Physics 85:*257 (1989).

The next task is to find a way to rid ourselves of the CFL condition imposed by the nonlinearity introduced by using $(Gp)^{n-1/2}$ and to fix the velocity flux business. (The footwork is beginning to get fancy.) We learned in the previous section that a basic feature of the BCG approach was to compute advective terms using time-centered cell edge velocities and then using a Riemann solver to find the upwind state. However, we ran into difficulties in maintaining a divergence-free velocity field and in handling the nonlinearities introduced by a lagging pressure gradient. The fix is apply the projection operator to the edge velocities; this will ensure that they will be divergence-free. In keeping with the spirit of terminology proliferation, we will denote this projection of edge velocities as the *mac projection* (mesh-and-cell projection). In addition, we get another piece out of the projection for free, namely the gradient of a scalar. This will be used to solve the pressure gradient problem. The steps are as follows:

(1) Perform the inviscid predictor calculation, ignoring the ∇p term altogether. So, we extrapolate the velocity field to the cell edges and choose the upwind state to find the fractional-step values $\underset{\sim}{u}^*{}_{i+1/2,j}$ and $\underset{\sim}{u}^*{}_{i,j+1/2}$.

(2) Compute the mac divergence of $\underset{\sim}{u}_{ij}$, as defined by edge velocities:

$$(D_{mac}\underset{\sim}{u}) = \frac{u^*{}_{i+1/2,j} - u^*{}_{i-1/2,j}}{\Delta x} + \frac{v^*{}_{i,j+1/2} - v^*{}_{i,j-1/2}}{\Delta y} \tag{90}$$

(3) Solve $\Delta^h \phi = D_{mac}\underset{\sim}{u}$ for ϕ, where Δ^h is the standard five-point Laplacian, subject to appropriate boundary conditions: if solid wall, then $\phi_{0,j} = -\phi_{1,j}$; if periodic, then $(\partial\phi)/(\partial\hat{n}) = 0$.

(4) Then solve for the time-centered velocity components at the cell edges:

$$u^{n+1/2}_{i+1/2,j} = u^*{}_{i+1/2,j} - \frac{\phi_{i+1,j} - \phi_{i,j}}{h} \tag{91}$$

$$v^{n+1/2}_{i,j+1/2} = v^*{}_{i,j+1/2} - \frac{\phi_{i,j+1} - \phi_{i,j}}{h} \tag{92}$$

where h is the mesh spacing. Now u and v satisfy $D_{mac}\underset{\sim}{u}^{n+1/2} = 0$

This takes care of the normal velocities at the cell edges. Now, what about the tangential ones? The simplest approach is, e.g., for the case of u at the top edge:

$$u^{n+1/2}_{i,j+1/2} = u^*{}_{ij} - \frac{1}{4h}\left[\phi_{i+1,j} + \phi_{i+1,j} + \phi_{i-1,j} + \phi_{i-1,j+1}\right] \tag{93}$$

The term in brackets approximates $\left.\frac{\Delta t}{2}\frac{\partial p}{\partial x}\right|_{i,j+1/2}$ and makes the approximation second order.

(5) Now finish off the extrapolation at the cell edges by examining what is coming in from either side of each edge. Coming into the right edge of cell *(i, j)* from the left is:

$$\underset{\sim}{u}^{n+1/2}_{i,j+1/2_L} = \underset{\sim}{u}^n_{ij} + \frac{1}{2}\left[1 - \frac{\Delta t}{\Delta x}\right]\Delta^x \underset{\sim}{u}_{ij} - \frac{v^n_{ij}\Delta t}{2\Delta y}\left[\underset{\sim}{u}\right]^2_{ij} + \frac{\nu\Delta t}{2}\left(\Delta^h \underset{\sim}{u}^n\right)_{ij} \tag{94}$$

where $\Delta^x \underset{\sim}{u}_{ij}$ are van Leer slopes and Δ^h is the five-point Laplacian. The term $[\underset{\sim}{u}]$ denotes the jump in $\underset{\sim}{u}$ and is calculated by:

$$v^n_{ij}\,[\underset{\sim}{u}]^2_{ij} = \begin{cases} v^n_{ij}\,[\underset{\sim}{u}^n_{ij} - \underset{\sim}{u}^n_{i,j-1}]\ \ldots if \ldots\ (v^n_{ij} > 0) \\ v^n_{ij}\,[\underset{\sim}{u}^n_{i,j+1} - \underset{\sim}{u}^n_{ij}]\ \ldots if \ldots\ (v^n_{ij} < 0) \end{cases} \tag{95}$$

We obtain similar values coming from the right:

$$\underset{\sim}{u}^{n+1/2}_{i,j+1/2_R} = \underset{\sim}{u}^n_{i+1,j} + \frac{1}{2}\left[-1 - \frac{\Delta t}{\Delta x}\right]\Delta^x \underset{\sim}{u}_{i+1,j} - \frac{v^n_{ij}\Delta t}{2\Delta y}\,[\underset{\sim}{u}]^2_{i+1,j} + \frac{\nu\Delta t}{2}\,(\Delta^h \underset{\sim}{u}^n)_{i+1,j} \tag{96}$$

(We have now accounted for everything properly in the Taylor expansion except for the $(\Delta t \nabla p)/2$ term. This will be taken care of in the second step.)

(6) Now, finally, choose the upwind state. Recall that we have gotten to this point by extrapolating to the cell edges from the right and left:

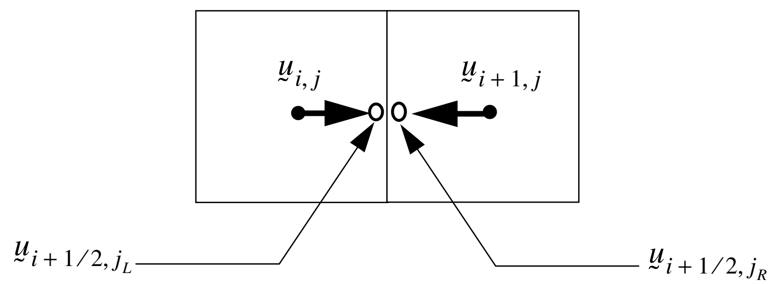

FIGURE 3. Extrapolation of cell-edge velocities from left and right

So, there are three possibilities:

$$\underset{\sim}{u}_{i+1/2,j} = \begin{cases} \underset{\sim}{u}_{i+1/2,j_L} \ldots if \ldots\ (\underset{\sim}{u}_{i+1/2,j_L}, \underset{\sim}{u}_{i+1/2,j_R} > 0) \\ \left\{ \begin{matrix} u_{i+1/2,j} = 0 \\ v_{i+1/2,j} = v_{avg} \end{matrix} \right\} \ldots if \ldots\ (\underset{\sim}{u}_{i+1/2,j_L} \cdot \underset{\sim}{u}_{i+1/2,j_R} < 0) \\ \underset{\sim}{u}_{i+1/2,j_R} \ldots if \ldots\ (\underset{\sim}{u}_{i+1/2,j_L}, \underset{\sim}{u}_{i+1/2,j_R} < 0) \end{cases} \tag{97}$$

where

$$v_{avg} = \frac{1}{2}\,[v_{i+1/2,j_L} + v_{i+1/2,j_R}] \tag{98}$$

3.3.6 Second-order cell-centered projection methods

Here is a nasty piece of business which we can avoid no longer: what happens with cell-centered projections? Given an arbitrary discrete vector field, $\underset{\sim}{u}_{ij}$, defined at cell centers, we would like to hit it with a projection to find $\underset{\sim}{u}_{ij} = \underset{\sim}{u}_{d_{ij}} + (G\phi)_{ij}$. We will examine two cases: that of doubly periodic boundary conditions; and of solid wall boundaries (which means homogeneous Dirichlet b.c. on $\underset{\sim}{u}$ and homogeneous Neumann b.c. on ϕ). Assume that the number of mesh points in each direction, nx and ny, are both even. Assume further that the mesh spacing, h, is uniform in both directions. Recall that we obtain a Poisson-like equation which must be used to solve for ϕ:

$$DG\phi = D\underset{\sim}{u} \tag{99}$$

We define the gradient of ϕ in the x and y directions by:

$$(G\phi)^x_{ij} = \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h} \tag{100}$$

$$(G\phi)^y_{ij} = \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2h} \tag{101}$$

The discrete div grad ϕ can be written:

$$DG\phi = \frac{1}{4h^2}[\phi_{i+2,j} + \phi_{i,j+2} + \phi_{i-2,j} + \phi_{i,j-2} - 4\phi_{ij}] \tag{102}$$

For the case of doubly periodic boundary conditions, this is easy to extend periodically as required. However, the result is a set of four completely decoupled stencils:

	-	+	-	+	-	+	
o	x	o	x	o	x	o	x
+	-	+	-	+	-	+	-
o	x	o	x	o	x	o	x
+	-	+	-	+	-	+	-
o	x	o	x	o	x	o	x
+	-	+	-	+	-	+	-
	x	o	x	o	x	o	

FIGURE 4. Completely decoupled stencils of DGϕ with periodic boundary conditions

The cells marked with x communicate only with other cells marked x at this grid level; likewise for cells marked with o, + and -. Well, what about multigrid which is supposed to eliminate short-wavelength components in the residual? Can we use this technique to our advantage? The answer is no, not right away. Let's focus on a grouping of four fine-mesh cells which is assumed to be repeated throughout the domain, shown in figure +1:

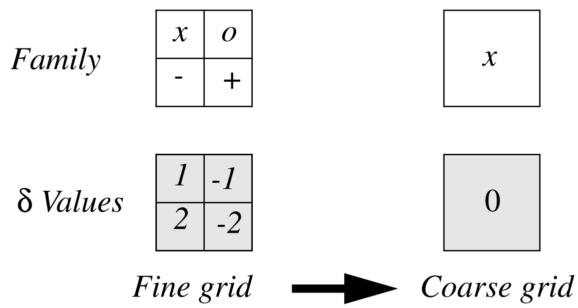

FIGURE 5. Detail of a decoupling example which spells trouble for multigrid

We would like to relax on the fine grid to eliminate short-wavelength components in the quantities R and δ. However, note that for this example, $(DG\delta)_{ij} = 0$. Also, the average of the fine-grid cells onto the coarse grid would be zero. Multigrid would not help us smooth out the error in this case, so multigrid gets a big gongggggggg!

The second boundary condition under consideration, i.e., that of solid walls, is equivalent to extending the solution in the following way:

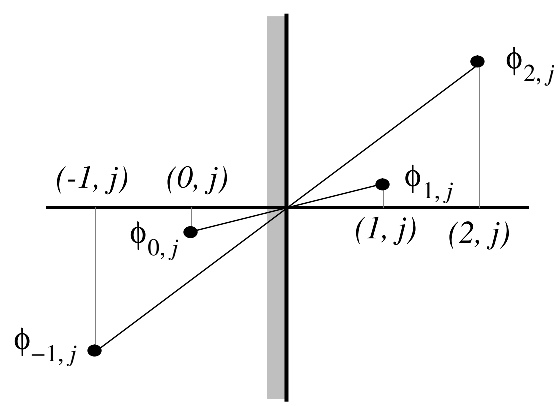

FIGURE 6. Phantom nodes and relation to interior points for solid-wall boundary conditions

Linear interpolation is performed between nodes in the interior of the problem domain and phantom nodes which are inside the boundary to create a value of zero at the wall (i.e., $\phi_{-1,j} = -\phi_{2,j}$ and $\phi_{0,j} = -\phi_{1,j}$). The good news is that there are no more decoupled meshes since everything couples in a global sense at the boundaries, as shown below in figure +1. For example, we need to compute $DG\phi$ at the upper left corner of the domain (the dark circle of type x). Information from the interior of the domain comes only from the x family itself; however, the phantom nodes within the boundary have also taken marching orders from the *"o"* and "-" families and so can communicate their evil intent to family x.

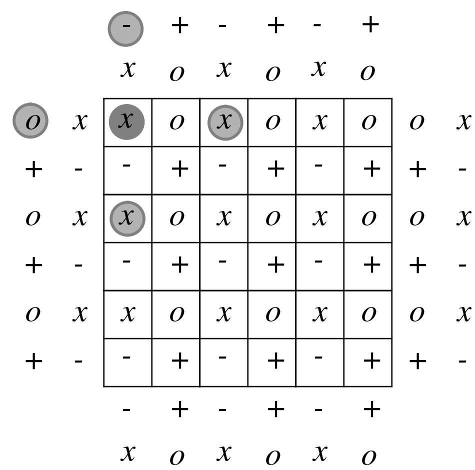

FIGURE 7. Four families of cells communicate at the borders with solid-wall boundary conditions

However, as can be seen from the above, relaxation is a local process. The updating of the solution of ϕ in the interior of the domain must wait for numerical "diffusion" to take place from the information at the boundaries. This also merits a big gong for multigrid.

We might next ask if there is some way around this coupling/decoupling problem. We will try to fix this deficiency by using local decoupling. Two approaches follow: First, we can try to mimic doubly periodic b.c. by using nonstandard averaging and interpolation operators which respect the coupling/decoupling. For example, in going from the fine grid to the coarse grid, we might simply use adjacent cells of the same family to compute the coarse-grid residual. In this case, the coarse-grid residual is given by:

$$R_o^c = \frac{1}{4}\sum_{i=1}^{4} R_{o_i}^f \qquad (103)$$

where f and c denote fine- and coarse-grid values, respectively. The fine-grid values used to interpolate for the coarse-grid value in the dark circle are shown with arrows and fingers in figure +1, below. The quantity $DG\phi$ calculated at the solid-wall boundaries still communicates with the other families and provides the appropriate coupling. Since we're at the coarser grid level, the information “diffusion” from the boundary cells will infiltrate into the interior more readily than for the fine grid.

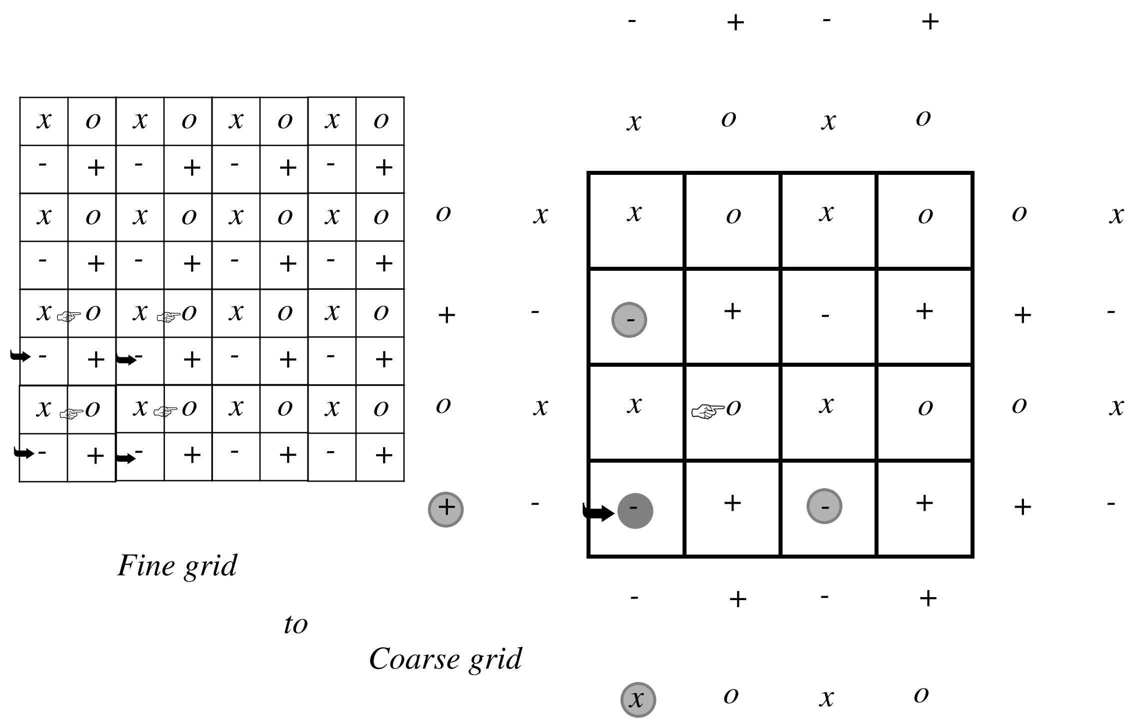

FIGURE 8. Local decoupling approach applied to multigrid

A second approach to eliminate decoupling is by finding an appropriate discrete divergence and gradient such that $DG\phi$ only has constant solutions for doubly periodic boundary conditions. For this case, the discrete divergence is defined by:

$$(D\underset{\sim}{u})_{ij} = \frac{u_{i+1/2,j} - u_{i-1/2,j}}{h} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{h} \tag{104}$$

where the cell edge velocities may be defined by:

$$u_{i+1/2,j} = \frac{1}{2}(u_{ij} + u_{i+1,j}) - \alpha(D^x_+ \ D^x_- \ u)_{ij} \tag{105}$$

$$v_{i,j+1/2} = \frac{1}{2}(v_{ij} + v_{i,j+1}) - \alpha(D^y_+ \ D^y_- \ v)_{ij} \tag{106}$$

and

$$(D^x_+ \ D^x_- \ u)_{ij} = u_{i+1,j} - 2u_{ij} + u_{i-1,j} \tag{107}$$

Now why do we need this extra notation? The second term on the r.h.s. of equations (-2) and (-3) above comes from carrying higher-order terms in the Taylor expansion performed in the discretization. The gradient in the x direction is:

$$\frac{\partial u}{\partial x} \approx (D_o u)_{ij} - \alpha h D_- \ (D^x{}_+ \ D^x_- \ u) \qquad (108)$$

Thus, the second term on the r.h.s. of equation (-1) above is an approximation to $\alpha h^2 \frac{\partial^3 u}{\partial x^3}$

Next, the discrete gradient is related to the divergence by the adjoint condition so that $G = -D^T$. To define G:

$$(G\phi)^x_{ij} = \frac{\phi_{i+1/2,j} - \phi_{i-1/2,j}}{h} \qquad (109)$$

where

$$\phi_{i+1/2,j} = \frac{1}{2}(\phi_{ij} - \phi_{i+1,j}) - \alpha (D^x{}_+ \ D^x_- \ u)_{i+1,j} \qquad (110)$$

Next we need to solve the equation:

$$DG\phi = D\underset{\sim}{w} \qquad (111)$$

and

$$\underset{\sim}{w}_d = \underset{\sim}{w} - G\phi \qquad (112)$$

When $DG\phi$ is zero, ϕ is a constant. This can be shown by using a discrete Fourier transform to diagonalize DG.

We expect that the short-wavelength modes would be damped by point relaxation using Gauss-Seidel with red/black ordering. For this case, in the first sweep the even cells are updated:

$$\phi^{l+1/2}_{ij} = \begin{cases} \phi^l_{ij} + \lambda(\rho - DG\phi^l)_{ij} \ldots if \ldots (i+j) = even \\ \phi^l_{ij} \ldots if \ldots (i+j) = odd \end{cases} \qquad (113)$$

Then perform a second sweep to get from iteration level $l+1/2$ to $l+1$ by reversing even and odd in equation (-1) above to update the odd cells (see the discussion in section xxx.x.x. supra). If we choose the "time" step parameter λ so as to eliminate the diagonal terms in the matrix $DG\phi$, we don't kill the high wavenumber modes, but we do dampen them. For $\lambda = -1/c$, we can write:

$$(DG\phi)_{ij} = c\phi_{ij} + \sum_s \phi_s \qquad (114)$$

This is not the same as the $2\Delta x$ mode which we discussed in section xxx.xx.x previously.

Now, let's discuss the solid-wall boundary. The discrete divergence is defined as before for the doubly periodic case, but the discrete gradient loses accuracy near the boundaries. This manifests itself as a failure in multigrid. So, now comes the moment when we will relax the condition $G = -D^T$ at the boundary, but we must somehow reconcile with the solvability condition.

Let's backtrack for a moment. Recall that to find the divergence-free velocity field by hitting it with a projection, we wanted to solve $\underset{\sim}{w}_d = P\underset{\sim}{w}$. The first step was to solve $DG\phi = D\underset{\sim}{w}$ to obtain ϕ so that we could compute $\underset{\sim}{w}_d = \underset{\sim}{w} - G\phi$. Another way to write the projection operator is:

$$P\underset{\sim}{w} = (I - G(DG)^{-1}D)\underset{\sim}{w} \tag{115}$$

Recall that $P^2 = P$, but the conjugate transpose $P^* \neq P$. The Fredholm alternative tells us that $DG\phi = D\underset{\sim}{w}$ is solvable if, given that, $(DG)^T\phi = 0$ then $\langle D\underset{\sim}{w}, \phi\rangle = 0$. We can reason this out in the following way: if $(DG)^T\phi = 0$, then $DG\phi = 0$. This means that ϕ is a constant, say C. It also means that $\langle D\underset{\sim}{w}, \phi\rangle = 0$ if it is written in conservation form.

Reference: Strikwerda (1984).