1 | Simulation Set Up

In this section, I briefly describe the simulation set up from the most trivial case all the way up to the true problem. We start with a brief introduction about how the inversion process proceed.

1.1 Inversion Process (Primal Space)

From hydrostatic balance, the surface pressure is propotional to the surface sea height (SSH).

$$\eta = \frac{p\{z=0\}}{g} \tag{1.1}$$

In all the QG models, the pressure is expanded asymptotically in the form of

$$p_{total} \sim p^0 + \epsilon p^1 \tag{1.2}$$

The first order pressure is directly related to the first order potential Φ^0 by the formula

$$p^0 = f\Phi^0$$

As from geostrophic balance we have

$$p_y^0 = -fu^0 = -f\partial_y\Phi^0 \qquad p_x^0 = fv^0 = f\partial_x\Phi^0$$

Now we have

$$\eta^0 = \frac{f\Phi^0}{g}$$

Similarly from the first order correction we have

$$\eta^1 = \frac{p^1}{g}$$

So in general, considering the first order correction to the hydrostatic balance Eq 1.1, we have

$$\eta \sim \eta^0 + \epsilon\eta^1 = \frac{f\Phi^0}{g} + \epsilon\frac{p^1}{g} \tag{1.3}$$

This is the equation 40 in Ryan's note.

Now remember our goal is to use η to invert for Φ^0. What is the relationship between p^1 and Φ^0? The relationship is given by eq **??**. We copy it here

$$\boxed{\nabla^2 p^1 - f\zeta^1 = 2J(\Phi_x^0, \Phi_y^0)}$$

Where J is the Jacobian operator. This is a non-linear relationship, recall that ζ^1 is related to the potentials via Eq **??**. So

$$\nabla^2 p^1 = f\left(\nabla^2\Phi^1 + F_{zy}^1 - G_{zx}^1\right) + 2J(\Phi_x^0, \Phi_y^0) \quad \rightarrow \quad \Phi^{0,s} + \epsilon\mathcal{N}(\Phi^{0,s}) = \eta(x,y)$$

Where $\mathcal{N}$ is a non-linear operator. Here the relationship between Φ^1, F^1 and G^1 are given by Eq **??**, **??** and **??**. Then the whole inversion problem is clear. Given $\eta(x,y)$ we wish to find a $\Phi^{0,s}$ [1]

[1] given $\Phi^{0,s}$ we can reconstruct the 3D Φ already.

1.2 Inversion Process (Spectral Space)

The method introduced in the previous section isn't very efficient. As all the problem is solved mainlyin the physical space, however, the inversion would be much easier if we do it in the spectral space as all the calculation here would be 2D and thereform much easier.

The key equation is still Eq **??**.

$$\nabla^2 p^1 - f\zeta^1 = 2J(\Phi^0_x, \Phi^0_y)$$

We wish to obtain p^1 in the spectral space, then use Eq 1.3 to ge the surface SSH.

$$\eta \sim \frac{f}{g}\Phi^{0,s} + \epsilon\frac{1}{g}p^{1,s}$$

Since $p^{1,s}$ is related to the surface potential $\Phi^{0,s}$ nonlinearly, so we can write [2]

[2] *I use the upper-index s and t interchangeably all for surface data. I should change this in the future.*

$$\eta(x,y)^s = \frac{f}{g}\Phi^{0,s} + \frac{\epsilon}{g}\mathcal{N}(\Phi^{0,s})$$

and for the fourier component.

$$\widehat{\eta^s} = \frac{f}{g}\widehat{\Phi}^{0,s} + \frac{\epsilon}{g}\mathcal{N}(\widehat{\Phi}^{0,s})$$

No we just need a why to get this $\mathcal{N}$ operator.

For the inversion process, we have the true SSH data

$$\eta(x,y)^{true}$$

And its fourier transform

$$\widehat{\eta^{true}}$$

Our goal is to find a $\Phi^{0,s}$ to minimize

$$\sum_{k,l}\left(\hat{\eta} - \eta^{\hat{true}}\right)^2$$

1.2.1 First Order Potential

To get Φ^1, we use Eq **??**.

$$\boxed{\hat{\Phi}^1 = \underbrace{\frac{1}{2\mathrm{Bu}}\widehat{\Phi^0_z\Phi^0_z}}_{\text{Fourier transform of interior part}} - \underbrace{\frac{1}{\mathrm{Bu}}\widehat{\Phi^{0,t}_z\Phi^{0,t}_{zz}}\frac{e^{\mu z}}{\mu} + \frac{C_b}{\mu}e^{\mu z}}_{\text{Fourier transform of surface part}}}$$

and to get G^1, similarly we use Eq **??**.

$$\boxed{\widehat{G^1} = -\frac{1}{\mathrm{Bu}}\left(\widehat{\Phi^0_x\Phi^0_z} - \widehat{\Phi^{0,t}_x\Phi^{0,t}_z}e^{\mu z}\right)}$$

and to get F^1, use Eq **??** directly.

$$\boxed{\widehat{F^1} = \frac{1}{\mathrm{Bu}}\left(\widehat{\Phi_y^0\Phi_z^0} - \widehat{\Phi_y^{0,t}\Phi_z^{0,t}e^{\mu z}}\right)}$$

1.2.2 Surface Ageostrophic Vorticity

We will invert p^1 in the spectral space following Eq **??**. The philosophy is that fourier transform is a linear operator, so whenever we see a product of two variables in the physical space, we need to complete this operation in the physical space first then do the fourier transform.

From Eq **??** we have

$$\begin{aligned}\widehat{\zeta}^1 &= \widehat{\nabla^2\Phi^1} + \widehat{F^1_{yz}} - \widehat{G^1_{xz}}{}^3 \\ &= -K^2\widehat{\Phi^1} + ik_y\partial_z\widehat{F^1} - ik_x\partial_z\widehat{G^1}\end{aligned}$$

[3] ∇^2 *here is 2D Laplacian operator*

where $K^2 = k_x^2 + k_y^2$. We sepearate the derivation into interior part and surface part.

1. Interior part:
   (a) For Φ^1 its interior part is $\propto \Phi_z^0\Phi_z^0/2$. So applying Laplacian to it we have
   $$\|\nabla\Phi_z^0\|^2 + \Phi_z^0\nabla^2\Phi_z^0$$
   (b) From $F^1_{yz} - G^1_{xz}$, the interior part of F^1 is $\propto -\Phi_y^0\Phi_z^0$ and the interior part of G^1 is $\propto \Phi_x^0\Phi_z^0$.
   $$\nabla\Phi^0\cdot\nabla\Phi_{zz}^0 + \nabla^2\Phi^0\cdot\Phi_{zz}^0 + \underbrace{\|\nabla\Phi_z^0\|^2 + \Phi_z^0\nabla^2\Phi_z^0}_{\text{matches with interior part of }\Phi^1}$$
2. Surface part : The surface part is already an exponential decay in the vertical direction starting from the surface first order potential fourier transform.
   (a) For Φ^1. The fourier transform of the surface part is $\propto -\Phi_z^{0,t}\widehat{\Phi_{zz}^{0,t}e^{\mu z}}/\mu$. So taking the laplacian is equivalent to mulfiply by $-K^2$ in the spectral space. Evaluate at $z=0$, the exponential decay term is 1.
   $$\frac{1}{\mathrm{Bu}}\frac{K^2}{\mu}\widehat{\Phi_z^{0,t}\Phi_{zz}^{0,t}}$$
   (b) For $F^1_{yz} - G^1_{xz}$, the surface part of $\hat{F}_1$ is $\propto -\Phi_y^{0,t}\widehat{\Phi_z^{0,t}e^{\mu z}}$ and $\propto \Phi_x^{0,t}\widehat{\Phi_z^{0,t}e^{\mu z}}$. For $\hat{G}_1$. Then the surface part evaluate at $z=0$ is
   $$-\left(ik_y\mu\widehat{\Phi_y^{0,t}\Phi_z^{0,t}} + ik_x\mu\widehat{\Phi_x^{0,t}\Phi_z^{0,t}}\right)$$

Formula 1.3.1 So to sum up we have

$$\boxed{\widehat{\zeta^{1,s}} = \frac{1}{\mathrm{Bu}}\left[I_1 + I_2 - I_3\right]} \tag{1.4}$$

where the three terms are defined as follows:

$$I_1 = \widehat{\nabla\Phi^{0,t} \cdot \nabla\Phi^{0,t}_{zz}} + \widehat{\nabla^2\Phi^{0,t} \cdot \Phi^{0,t}_{zz}} + 2\|\widehat{\nabla\Phi^{0,t}_z}\|^2 + 2\widehat{\Phi^{0,t}_z \nabla^2\Phi^{0,t}_z} \tag{1.5}$$

$$I_2 = \frac{K^2}{\mu}\widehat{\Phi^{0,t}_z \Phi^{0,t}_{zz}} \tag{1.6}$$

$$I_3 = ik_y\mu\widehat{\Phi^{0,t}_y \Phi^{0,t}_z} + ik_x\mu\widehat{\Phi^{0,t}_x \Phi^{0,t}_z} \tag{1.7}$$

Everything here is evaluated at the surface $z = 0$.

1.2.3 Surface Cyclogeostrophic Correction

We also need the cyclogeostrophic correction, evaluate at the surface $z = 0$.

Formula 1.7.1

The result is simple, since derivative and fourier transform commute, we have

$$\boxed{2\widehat{J(\Phi^{0,t}_x, \Phi^{0,t}_y)}} \tag{1.8}$$

1.3 Pre-Defined Velocity Fields

We start with a very simple case to play around with this model.

1.3.1 Toy Model Setup

The toy model start with a pre-defined potential Φ^0. I use a random generator here and make the spectrum follow a power law with slope -3, which is typical for 3D turbulence.

```
rng(42); % Set seed for reproducibility
% We produce a random $\Phi^0$ here.
k_peak = 4;
slope = -3; % This slope matches the energy cascade in 3D turbulance
phase = rand(N, N) * 2 * pi;
amplitude = (K ./ k_peak).^(slope) .* exp(-(K./k_peak).^2);

amplitude(1,1) = 0;
% Compute the hat
phi0_hat = amplitude .* exp(1i * phase);
% Inverse transform to get to the physical space
phi0_surf = real(ifft2(phi0_hat));
% Normalize
phi0_surf = phi0_surf / std(phi0_surf(:));
```

1.3.2 Forward Process to get the true velocity fields

Then some functions are defined for different purposes.

1. This function compute the 3D potential Φ^0 from the surface data. The fourier components has an exponential decay in the vertical direction.

```
    phi0_3d_true = derive_phi0_3d(phi0_surf, K, z, Bu);
```

2. This function compute all the other first order potential F^1, G^1 and Φ^1 using Eq **??**, **??** and **??**. A possion problem is solved in the spectral space.

```
    [F1_true, G1_true, Phi1_true] =
        calculate_higher_order(phi0_3d_true, K, kx, ky, z, Bu, N,
        nz);
```

3. This function computes the first order pressure p^1 using Eq **??**.

```
    p1_true = solve_p1(f, dx, dz, kx, ky, z, Bu, Ro,
        phi0_3d_true, F1_true, G1_true, Phi1_true);
```

4. With all the data above, we can compute the true surface sea height using Eq 1.3.

```
    p1_surf = p1_true(:, :, end);
ssh_true = phi0_surf + Ro * p1_surf;
```

   Here I use end because the z corredinate starts from the bottom to the top.

Now we have the surface sea height. This is where the inversion starts.

1.3.3 Inversion Process solving the Optimization Problem

The inversion process starts with η. We make an initial guess for Φ^0, in my code I use a zero initial Φ^0.

```
phi0_guess_flat = zeros(N, N);
```

Then use the function

```
cost_func = @(phi0_flat) sqg_cost_function(phi0_flat, f, ssh_true, K,
    kx, ky, z, Bu, Ro, N, nz, dx, dz);
```

To compute the cost. In this `sqg_cost_function` basically do the same thing as the forward process, compute the 3D potential Φ^0 first, then solve the Possion equation to get F^1, G^1, Φ^1, then compute p^1 and finally compute the SSH. The difference is that at the final step, we compute the difference between the computed SSH and the true SSH to obtain the cost.

```
    % Cost
    difference = ssh_guess - ssh_obs;
    cost = sum(difference(:).^2);
```

This is the returned value of that function. I then use a built in Matlab toolbox to solve the optimization problem.

```
num_iteration = 20;
options = optimoptions('fminunc', 'Display', 'iter', 'Algorithm',
    'quasi-newton', 'MaxIterations', num_iteration);

% Compute the cost function
cost_func = @(phi0_flat) sqg_cost_function(phi0_flat, f, ssh_true, K,
    kx, ky, z, Bu, Ro, N, nz, dx, dz);

% Run Optimization
tic;
try
    [phi0_opt_flat, fval] = fminunc(cost_func, phi0_guess_flat,
        options);
```

```
    phi0_surf_opt = reshape(phi0_opt_flat, N, N);
    disp('Optimization Complete.');
catch ME
    disp('Optimization failed or interrupted.');
    disp(ME.message);
    phi0_surf_opt = reshape(phi0_guess_flat, N, N); % Fallback
end
toc;
```

Then plot the results. [4]

[4] *The plot part is writen by Gemini*