Reducing the continuously stratified equations to the shallow water equations

We start with the dry, linearized primitive equations in a continuous atmosphere, and follow a similar procedure as Vallis (section 3.4). We include uniform linear friction α in the momentum and thermodynamic equations. All notations are usual (s is dry static energy divided by c_p).

$$\begin{aligned}
&(\partial_t + \alpha)\mathbf{u} + f\mathbf{k} \times \mathbf{u} + \nabla\phi = 0,\\
&(\partial_t + \alpha)T + \omega\partial_p s = Q_{diab},\\
&\nabla \cdot \mathbf{u} + \partial_p\omega = 0,\\
&\partial_p\phi = -RT/p.
\end{aligned}$$

We seek a vertical structure function for $\mathbf{u}$ and ϕ that would make this set of equations equivalent to a shallow water system. Combining the thermodynamic and hydrostatic equations:

$$(\partial_t + \alpha)\left[-\frac{p}{R\partial_p s}\partial_p\phi\right] + \omega = \frac{Q_{diab}}{\partial_p s}$$

Differentiating with respects to p and using continuity,

$$(\partial_t + \alpha)\,\partial_p\left[-\frac{p}{R\partial_p s}\partial_p\phi\right] - \nabla\cdot\mathbf{u} = \partial_p\left[\frac{Q_{diab}}{\partial_p s}\right]$$

Suppose that one can find a vertical structure function $V_n(p)$ such that $\partial_p\left[-\frac{p}{R\partial_p s}\partial_p V_n\right] + K_n V_n = 0$ for some eigenvalue K_n (with boundary conditions $\partial_p V_n = 0$ at $p = 0$ and $p = p_s$, in order for the vertical velocity to vanish there). Plugging

$$\begin{cases}
\mathbf{u}(x,y,p) &= \mathbf{u}_n(x,y)V_n(p),\\
\phi(x,y,p) &= \phi_n(x,y)V_n(p),\\
Q_{diab}(x,y,p) &= -Q_n(x,y)K_n\partial_p s\int V_n\,dp',
\end{cases}$$

our equation set reduces to

$$\begin{aligned}
&(\partial_t + \alpha)\mathbf{u}_n + f\mathbf{k}\times\mathbf{u}_n + \nabla\phi_n = 0,\\
&(\partial_t + \alpha)\phi_n + \frac{1}{K_n}\nabla\cdot\mathbf{u}_n = Q_n.
\end{aligned}$$

which is close to the Gill problem (we would need to add the beta-plane approximation, and the longwave approximation in the meridional momentum equation, to have the exact Gill problem). Another limitation is that the initial equation set assumes a constant, uniform surface pressure p_s, whereas the stationary wave model lets p_s vary.

A particular choice of stratification

One choice that gives simple vertical structure functions is $\partial_p s = -\gamma p$ where γ is a constant. Then, our eigenvalue problem becomes

$$\frac{1}{R\gamma}\partial_{pp}V_n + K_n V_n = 0$$

and the solutions are $V_n(p) = \cos\left(n\pi\frac{p}{p_s}\right)$, along with $K_n = \dfrac{n^2\pi^2}{R\gamma p_s^2}$.

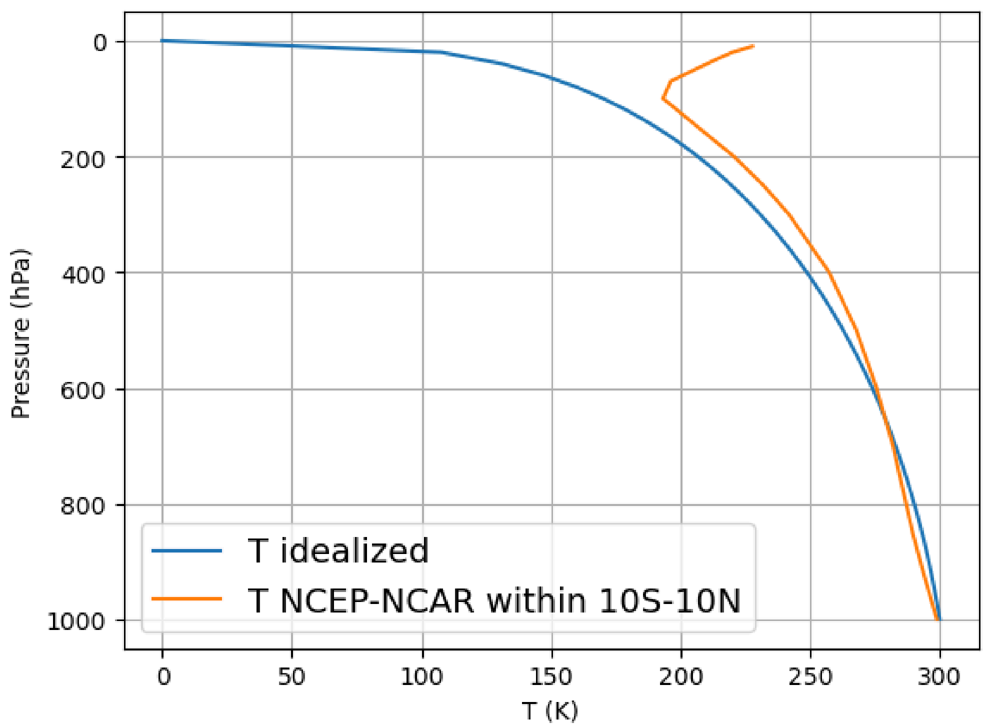
With this choice of stratification, we solve for the vertical temperature profile by solving

$$\partial_p T - \frac{\kappa T}{p} = -\gamma p,$$

where $\kappa = R/c_p$. Imposing $T(p = p_s) = T_s$, this gives

$$T(p) = \left[1 + \frac{\gamma p_s^2}{T_s(2-\kappa)}\right] T_s \left(\frac{p}{p_s}\right)^{\kappa} - \frac{\gamma p^2}{2-\kappa}$$

which looks like (compare to the NCEP-NCAR reanalysis profile):



With this choice of stratification, Q_{diab} is given by $Q_{diab}(x,y,p) = Q_n(x,y)K_n\gamma p\dfrac{p_s}{n\pi}\sin\left(n\pi\dfrac{p}{p_s}\right)$

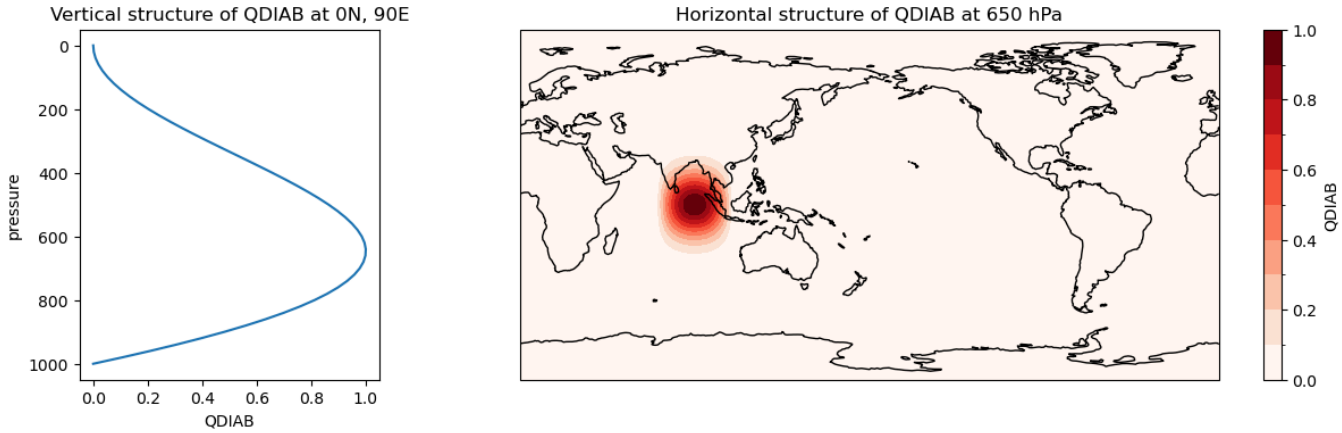
Horizontal structure of the forcing

To stay as close as possible to the Gill model (for which analytical solutions exist - see e.g. Vallis, section 8.5.2), we take the horizontal structure of the forcing to be of the form

$$Q_n(x,y) \propto e^{-y^2/(4L_{eq}^2)} \begin{cases} \cos(\pi x/(2L)), & |x| \leq L \\ 0, & |x| > L \end{cases}$$

where $L_{eq} = \sqrt{c_n/(2\beta)}$ ($c_n = 1/\sqrt{K_n}$ is the gravity wave speed) is the equatorial Rossby radius.

For the first baroclinic mode ($n = 1$), the structure of the forcing looks like:



Numerical solutions

- The shallow-water problem is set up and solved in `shallow_water_ideal_gill.py`.
- The continuously stratified problem is solved using the sigma-level stationary wave model in `run_ideal_gill.py`. Its basic state and forcing are calculated in the first section of `ideal_gill.ipynb`
- The solutions are analyzed in the remaining sections of `ideal_gill.ipynb`