# A Numerical Study of Vertical Propagation of Planetary Waves in the Stratosphere

Candidate Number: 13138

MPhys Project 2002

10300 Words

**Abstract**

A linear numerical model of planetary (Rossby) waves in the stratosphere is used to explore the dependence of the vertical structure of the wave amplitude and phase on the form of the mean zonal flow and other parameters. We consider the effects of tropospheric forcing of these waves from below and find that, for a given wave amplitude, the form of the stratospheric response can vary significantly depending upon conditions higher up. We couple the wave model with a simple nonlinear mean-flow model, with the waves accelerating or decelerating the mean flow; such changes in turn modifying the wave structure. The time-development of the system is followed.

# Contents

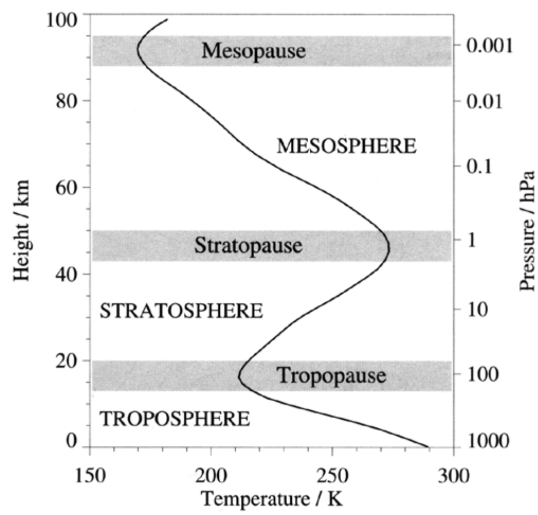


Figure 1: Schematic mid-latitude temperature profile of the lowest 100 km of the atmosphere. From [1].

# 1 Introduction

*Planetary waves*, or *Rossby waves*, are the most important class of atmospheric waves involved in large-scale meteorological processes. They have wavelengths ranging from 10s to 100s of kilometers. They owe their existence to the Coriolis forces produced by the rotation of the Earth and to the planet's spherical geometry, more specifically to the variation of the Coriolis forces with latitude.

In this project, we are concerned with the propagation of planetary waves in the stratosphere, which is the region of the middle atmosphere between about 10 and 60 km altitude. The basic vertical temperature structure of the atmosphere is shown in Figure 1; the stratosphere can be seen to be the region in which the temperature first begins to increase with altitude. It contains around one sixth of the total atmospheric mass and the bulk of the atmosphere's ozone. Processes in the stratosphere have an influence on the development of the main weather-producing systems of the troposphere, through dynamical, chemical and radiative interactions.

The planetary waves we study are forced from below by weather systems in the troposphere. We ignore the precise details of such systems, modelling them only as fluctuations in the geometric height of a surface of constant temperature. We make use of a linear numerical model of planetary waves in a regime of approximations known as *quasi-geostrophic theory*, and explore the dependence of the vertical structure of the wave amplitude and phase on the form of the zonal mean atmospheric flow[1].

A key motivation for this project is to investigate how a quantity known as the *Eliassen-Palm (EP) flux* at the base of the stratosphere varies when the mean flow is modified higher up. The EP flux[2] is a measure of the upward transfer of zonal (eastward) mean momentum by the waves; its value at the bottom of the stratosphere

[1]Formally, this quantity is the zonal-mean zonal flow, the westerly atmospheric flow averaged in that direction, but it is hereafter just referred to as the mean zonal flow.

[2]See Appendix B.

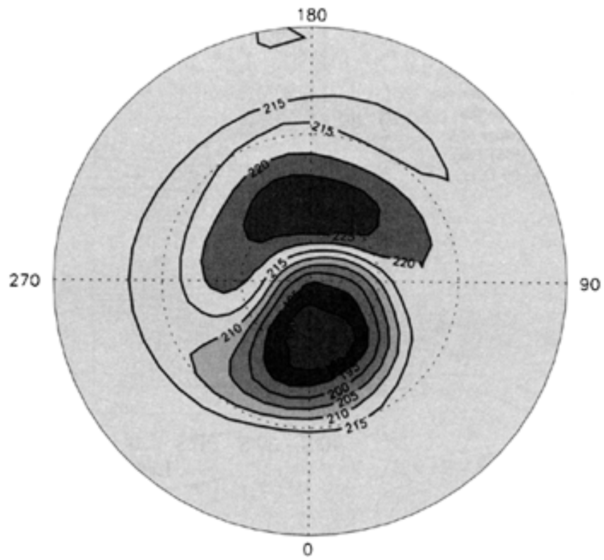


Figure 2: Polar stereographic map of strong Rossby-wave disturbances in the stratosphere, 9 January 1992. From [1].

can thus be regarded as a measure of the 'forcing' of the stratosphere by the troposphere below. It is often assumed that the value of this forcing is prescribed by the tropospheric flow and that the stratosphere merely responds passively. We test the validity of this assumption.

We go on to couple this wave model to a simple mean-flow model. This allows us to examine the effect of the accelerations and decelerations, produced by the planetary waves, on the background mean zonal flow. We are able to follow the time-development of the system.

This report begins with a general outline of the theory of planetary waves. The remainder of the project is split into two major sections. In the first, we describe the development up a numerical computer model which allows us to examine the propagation of planetary waves in the stratosphere. Following this, in the second part, we discuss the theoretical basis of mean-flow interactions and the coupling of our wave model with a simple non-linear mean-flow model.

# 2 The Dynamical Model

## 2.1 General Fluid Dynamics in the Atmosphere

### 2.1.1 The Primitive Equations of Motion

We are concerned with the time-development of wavelike disturbances in the atmosphere, which we model as a fluid-dynamical system. Where appropriate, we define local quantities such as temperature, pressure and velocity and follow the evolution of the fluid, by considering the forces acting upon small parcel of air. We follow what is known as the *Lagrangian* picture of fluid motion.

We may define both a *local* time derivative of a fluid property (i.e. at a fixed point) $\partial/\partial t$ and a time derivative *following the motion* of a fluid parcel $D/Dt$, the *material derivative*. From [1] we find that these are related by,

$$\frac{D}{Dt} \equiv \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla, \tag{1}$$

where $\mathbf{u} = (u, v, w)$ is the fluid velocity.

Applying Newton's Second Law to the small parcel, following the discussion in [1], gives the *Navier-Stokes equation* for fluid flow in an inertial frame. The fores included are the pressure gradient force, gravitation and friction,

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla p - g\mathbf{k} + \mathbf{F}_r, \tag{2}$$

where $p$ is pressure, $\rho$ is density, $g$ is acceleration due to gravity and $\mathbf{F}_r$ designates the frictional force.

It is generally convenient to use a coordinate system fixed with respect to the surface of the Earth. This is not an inertial frame and (2) must be modified with the introduction of terms representing the *Coriolis* and *Centripetal accelerations* that arise from the Earth's rotation. Following [1] we have

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla p - 2\Omega \wedge \mathbf{u} - \mathbf{g}' + \mathbf{F}_r \tag{3}$$

where,

$$\mathbf{g}' = -g\mathbf{k} - \Omega \wedge (\Omega \wedge \mathbf{r})$$

is the *effective gravity* and $\Omega$ is the rotation rate of the Earth.

We may express (3) in spherical polar coordinates giving the eastward, northward and vertical component momentum equations,

$$\frac{Du}{Dt} - \left(2\Omega + \frac{u}{r\cos\phi}\right)(v\sin\phi - w\cos\phi) + \frac{1}{\rho}\frac{\partial p}{\partial x} = F_r^{(x)} \tag{4}$$

$$\frac{Dv}{Dt} + \frac{wv}{r} + \left(2\Omega + \frac{u}{r\cos\phi}\right)u\sin\phi + \frac{1}{\rho}\frac{\partial p}{\partial y} = F_r^{(y)} \tag{5}$$

$$\frac{Dw}{Dt} - \frac{u^2 + v^2}{r} - 2\Omega u\cos\phi + \frac{1}{\rho}\frac{\partial p}{\partial z^*} + g = F_r^{(z)} \tag{6}$$

where $u$, $v$ and $w$ are the eastward ($x$), northward ($y$) and vertical ($z^*$) velocity components, $\phi$ is the latitude and $r$ is the distance from the centre of the earth.

In addition, we will also need to make use of the *continuity equation*, representing the law of conservation of mass,

$$\frac{D\rho}{Dt} + \rho\nabla \cdot \mathbf{u} = 0. \tag{7}$$

Taken together, these are the primitive atmospheric equations of motion.

### 2.1.2 Scale Analysis and the Geostrophic Approximation

In this project we concentrate on slow large-scale (synoptic) mid-latitude disturbances and may apply scaling arguments to (4), (5), (6), to find the relative importance of each term in this regime. From [5], we see that for such phenomena, (6) reduces to hydrostatic equilibrium, and Coriolis forces may be neglected in (4) and (5).

For simplicity, we use the *beta plane approximation* to the spherical geometry of the Earth[3], which enables us to work on a tangent plane, approximating the vertical component of the earth's rotation vector $f = 2\Omega \sin\phi$ by the linear $f = f_0 + \beta y$ with $f_0 = 2\Omega \sin\phi$ and $\beta = 2\Omega \cos\phi / a$ being constants. We also set the radial coordinate $r = a + z^* \simeq a$ where $a$ is the radius of the earth, as we will be considering geometric heights $z^*$ which are small compared to $a$.

The equations are further simplified by *defining* a new vertical 'log-pressure' coordinate,

$$z \equiv -Hln\left(\frac{p}{p_s}\right) \qquad H = \frac{RT_s}{g}, \tag{8}$$

where $T_s \sim 240$ K is a representative stratospheric temperature, $R = 287$ JK$^{-1}$kg$^{-1}$ is the gas constant for dry air, $H \sim 7$ km is a representative stratospheric scale height and $p_s$ is the pressure at $z = 0$[4].

Defining a quantity, known as the *geopotential*, $\Phi = gz$[5], and noting that $p = p_s e^{-z/H}$, we establish the following relationships,

$$-\frac{1}{\rho}\left(\frac{\partial p}{\partial x}\right)_z = -\left(\frac{\partial \Phi}{\partial x}\right)_p, \qquad -\frac{1}{\rho}\left(\frac{\partial p}{\partial y}\right)_z = -\left(\frac{\partial \Phi}{\partial y}\right)_p,$$

allowing us to rewrite the primitive equations as follows,

$$\frac{Du}{Dt} - fv + \frac{\partial \Phi}{\partial x} = 0, \tag{9}$$

$$\frac{Du}{Dt} + fu + \frac{\partial \Phi}{\partial y} = 0, \tag{10}$$

$$\frac{\partial \Phi}{\partial z} = \frac{RT}{H}, \tag{11}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{1}{\rho_0}\frac{\partial}{\partial z}(\rho_0 w) = 0. \tag{12}$$

These equations are still complex and applicable to a large range of atmospheric flows. In this project we wish to concentrate on larger scale slow motions at mid-latitudes. For such flows, approximate *geostrophic balance* holds where the Coriolis terms $-fv$, $fu$ in equations (9) and (10) are roughly balanced by the horizontal gradients of geopotential. The horizontal wind $(u, v, 0)$ therefore satisfies,

$$u \simeq u_g, \qquad v \simeq v_g, \tag{13}$$

where $(u_g, v_g)$ is the *geostrophic wind* defined in terms of the geopotential by,

$$u_g = -\frac{\partial \psi}{\partial y} \qquad v_g = \frac{\partial \psi}{\partial x}, \tag{14}$$

[3] See [1] p110.
[4] See [1] for a discussion of the origin of these values.
[5] Formally equal to the specific work done against gravity

where,

$$\psi = f_0^{-1}(\Phi - \Phi_0). \tag{15}$$

is the geostrophic streamfunction and $\Phi_0(z)$ is a suitable reference geopotential profile.

Following [1], we may take the ratio of the horizontal advection and Coriolis terms in primitive equation 4, which is approximately,

$$\frac{u\partial u/\partial x}{fv} \sim \frac{U^2/L}{fU} = \frac{U}{fL} \equiv \mathrm{Ro},$$

where $L$ and $U$ are characteristic horizontal length and velocity scales. Ro is a dimensionless number, called the *Rossby Number*. If Ro $\ll 1$, the Coriolis term is comparatively large and the geostrophic approximation is usually valid.

It must be noted that geostrophic balance is a *diagnostic* expression giving a relationship between two quantities, independent of time. Equations (14) cannot therefore be used to predict the evolution of the velocity field. In the next section we discuss a better approximation that allows us to follow the time development of the geostrophic flow.

### 2.1.3 Quasi-Geostrophic Theory

In the geostrophic approximation we essentially ignore the $Du/Dt$ and $Dv/Dt$ terms in equations (9) and (10) and replace $f = f_0 + \beta y$ by $f_0$. In the regime where Ro $\ll 1$ this is the leading approximation to the full primitive equations. We now discuss the next-closest, *quasi-geostrophic* approximation. Here we define the so-called *ageostrophic* velocity $(u_a, v_a, w_a)$ where,

$$u = u_g + u_a, \qquad v = v_g + v_a, \qquad w = w_a, \tag{16}$$

and $|u_a| \ll |u_g|$, $|v_a| \ll |v_g|$; the ageostrophic terms are much smaller than the geostrophic wind. It can be shown[6] that, under the quasi-geostrophic approximation, equations (7), (9), (10) become,

$$D_g u_g - f_0 v_a - \beta y v_g = 0, \tag{17}$$

$$D_g v_g + f_0 u_a + \beta y u_g = 0, \tag{18}$$

$$\frac{\partial u_a}{\partial x} + \frac{\partial v_a}{\partial y} + \frac{1}{\rho_0}\frac{\partial}{\partial z}\left(\rho_0 w_a\right) = 0, \tag{19}$$

where,

$$D_g \equiv \frac{\partial}{\partial t} + u_g\frac{\partial}{\partial x} + v_g\frac{\partial}{\partial y}.$$

is the time derivative following the geostrophic wind.

These equations can be combined to give the *quasi-geostrophic potential vorticity equation* (QGPV) following the procedure given in [5]. With no atmospheric friction or heating this may be written as,

$$D_g q_g = 0, \tag{20}$$

[6] See [5] p153-158.

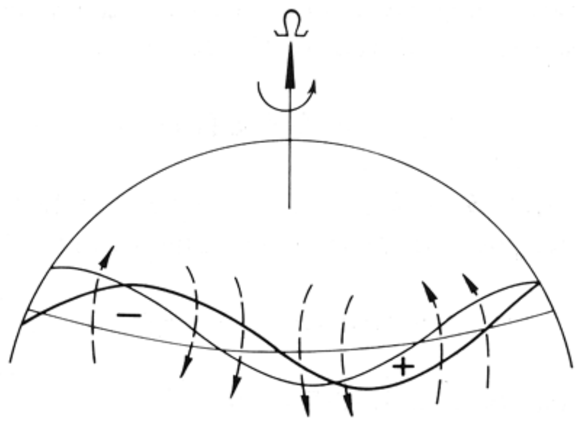


Figure 3: A chain of parcels subjected to a meridional displacement with the dashed arrows representing the induced velocity field. The heavy wavy line shows the original perturbation position and the light line, the westward displacement of the pattern due to advection by the induced velocity. This westward propagating field is a planetary wave. From [5].

where $q_g$ is known as the *quasi-geostrophic potential vorticity*. In full, this is

$$\left(\frac{\partial}{\partial t} + u_g\frac{\partial}{\partial x} + v_g\frac{\partial}{\partial y}\right)\left\{f_0 + \beta y + \frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial y^2} + \frac{1}{\rho_0}\frac{\partial}{\partial z}\left(\rho_0\epsilon\frac{\partial\psi}{\partial z}\right)\right\} = 0, \qquad (21)$$

where,

$$N^2(z) = \frac{R}{H}\left(\frac{dT_0}{dz} + \frac{\kappa T_0}{H}\right), \qquad (22)$$

is a measure of the density stratification of the atmosphere[7]. The quantity $q_g$ is conserved following the geostrophic wind.

In principle, with a knowledge of $q_g$ equation 21 may be solved to give $\psi$ and hence $v_g, u_g$ etc. Planetary waves *are* the wavelike motions described by this equation.

## 2.2 Theoretical Formulation of the Model

### 2.2.1 Overview

We want to examine the form of the geostrophic streamfunction $\psi$ across a range of different mean zonal flows. Given any such a flow profile, specified as a function of log-pressure height, we seek to find $\psi(z)$ by solving the quasi-geostrophic potential vorticity (QGPV) equation (21).

In this section, we apply a series of simplifications to the QGPV equation, reformulating it for use with perturbations on a mean zonal flow. We then consider the form of the boundary conditions and the wavelike solutions we expect to result from sinusoidal wave forcing at the top of the troposphere. The resulting equation is solved

[7] $N^2(z)$ is more formally known as the *buoyancy frequency*. It is related to the frequency of buoyancy oscillations that result when a parcel of air is displaced from its equilibrium position in the atmosphere. For a more detailed discussion see [1].

analytically for the special case of a zonal wind that is constant with height. We go on to make our equations more realistic by including the effects of simple linear dispersion and conclude by finding suitable numerical values for the constants in our equations.

### 2.2.2 Linearisation of the QGPV Equation

The full QGPV equation is non-linear in $\psi$, making it difficult or impossible to solve. However, if we only consider small departures from the mean zonal flow $\bar{u}(y,z)$, we are able to linearise the equation. Corresponding to this flow, we have a basic stream-function $\bar{\psi}$ satisfying $\bar{u} = -\partial\bar{\psi}/\partial y$. We set

$$\psi(x,y,z,t) = \bar{\psi}(y,z) + \psi'(x,y,z,t), \tag{23}$$

where $\psi'$ is a small disturbance.

By substituting (23) into (21) and neglecting terms quadratic in $\psi'$ and its derivatives, we have, following, [2],

$$\left(\frac{\partial}{\partial t} + u_g \frac{\partial}{\partial x}\right) q' + \bar{q}_y \frac{\partial \psi'}{\partial x} = 0, \tag{24}$$

where,

$$\bar{q}_y = \frac{\partial \bar{q}}{\partial y} = \beta - \frac{\partial^2 \bar{u}}{\partial y^2} - \frac{1}{\rho_0}\frac{\partial}{\partial z}\left(\rho_0 \epsilon \frac{\partial \bar{u}}{\partial z}\right),$$

and

$$q' = \frac{\partial^2 \psi'}{\partial x^2} + \frac{\partial^2 \psi'}{\partial y^2} + \frac{1}{\rho_0}\frac{\partial}{\partial z}\left(\rho_0 \epsilon \frac{\partial \psi'}{\partial z}\right).$$

Here $\bar{q}_y$ is the mean northward potential vorticity gradient, or 'effective $\beta$', and $q'$ is the disturbance potential vorticity.

In our model, we deal with the situation where $\bar{u}$ is independent of $y$ which further reduces $\bar{q}_y$ and $q'$ to

$$\bar{q}_y = \beta - \frac{1}{\rho_0}\frac{\partial}{\partial z}\left(\rho_0 \epsilon \frac{\partial \bar{u}}{\partial z}\right), \tag{25}$$

and

$$q' = \frac{\partial^2 \psi'}{\partial x^2} + \frac{1}{\rho_0}\frac{\partial}{\partial z}\left(\rho_0 \epsilon \frac{\partial \psi'}{\partial z}\right). \tag{26}$$

### 2.2.3 The Vertical Structure Equation and Wavelike Solutions

We suppose that the planetary waves are forced from below in the lower stratosphere by the fluctuations in height of an isothermal temperature surface. This is discussed in more detail in Section 2.2.4. Here we just note that we need to specify the differential of $\psi'$ with respect to $z$ at the lower boundary. We take the form of $\psi$ at the lower boundary is given by,

$$\psi'_0 = \mathrm{Re}\, \hat{\psi}_0\, e^{ik(x-ct)} \sin ly \qquad \text{at} \qquad z = 0, \tag{27}$$

which represents a wavy pattern of zonal wavelength $2\pi k^{-1}$ and meridional wavelength $2\pi l^{-1}$ moving at phase speed c.

We look for solutions to (24) of the form

$$\psi' = \mathrm{Re}\, \hat{\psi}(z) e^{ik(x-ct)} \sin ly. \tag{28}$$

where $\hat{\psi}(z)$ is a function to be found. Substituting (28) into (24) gives

$$\frac{1}{\rho_0}\frac{d}{dz}\left(\rho_0\epsilon\frac{d\hat{\psi}}{dz}\right)+\left[\frac{\bar{q}_y}{\bar{u}-c}-(k^2+l^2)\right]\hat{\psi}=0, \tag{29}$$

where, from before, we know that $\rho_0 \propto e^{-z/H}$ and that $\epsilon(z)=f_0^2/N^2(z)$. If we take $N^2=$ constant, $\epsilon=$ constant, and make the substitution $\hat{\psi}(z)=e^{z/2H}\Psi(z)$ we obtain,

$$\frac{d^2\Psi}{dz^2}+F(z)\Psi=0, \tag{30}$$

where,

$$F(z)=\frac{N^2}{f_0^2}\left[\frac{\bar{q}_y}{\bar{u}-c}-(k^2+l^2)-\frac{\epsilon}{4H^2}\right]. \tag{31}$$

This is the basic equation we investigate in this project. It may be solved analytically in the special case where $u(z)=$ constant. Otherwise, for more general cases, it may be solved numerically.

#### 2.2.4 Boundary Conditions

The differential equation (30) can only be solved over a region of space with well defined boundary conditions. The conditions used in this project are discussed below.

**Lower Boundary Condition** The waves are forced at the lower boundary. The precise nature of the forcing may take a variety of different physical forms; for computational convenience we treat it as a perturbation in temperature.

To express the condition mathematically we must first obtain an expression for the temperature $T$ as a function of $z$ in terms of the streamfunction $\psi$. From the definition of $\psi$ (15) we have,

$$\psi=f_0^{-1}(\Phi-\Phi_0),$$

which we differentiate with respect to $z$,

$$\frac{\partial\psi}{\partial z}=\frac{1}{f_0}\left(\frac{\partial\Phi}{\partial z}-\frac{\partial\Phi_0}{\partial z}\right). \tag{32}$$

From the equation of hydrostatic balance (11) we know that,

$$\frac{\partial\Phi}{\partial z}=\frac{RT}{H},$$

so, with some rearrangement, we may write (32) as,

$$T=T_0+\frac{Hf_0}{R}\frac{\partial\psi}{\partial z}. \tag{33}$$

We recall from equation (23) that,

$$\psi=\bar{\psi}+\psi',$$

so,

$$\frac{\partial\psi}{\partial z}=\frac{\partial\bar{\psi}}{\partial z}+\frac{\partial\psi'}{\partial z}.$$

We may then write (33) as,

$$T = T_0 + \frac{Hf_0}{R}\frac{\partial\bar{\psi}}{\partial z} + T', \tag{34}$$

where,

$$T' = \frac{Hf_0}{R}\frac{\partial\psi'}{\partial z},$$

is the temperature perturbation that we specify at the lower boundary. We know from ref[8] that typical tropospheric forcing causes perturbations in the geometric height ($z^*$) of $100-300$ m so the numerical value of $\partial\psi'/\partial z$ must be set to produce disturbances of this size. We know that $|\psi'|$ equals $|\hat{\psi}|$ and we set $d\hat{\psi}/dz \equiv A$, where $A$ is chosen to give a suitable perturbation size.

**Upper Boundary Condition** The upper boundary condition must be considered carefully. We have to regard our waves as being unbounded at the top of the stratosphere as, despite its computational convenience, any sort of rigid 'lid' to the region would not be physical. In this project we apply what is known as a *radiation condition*: at the top of stratosphere we specify that 'information' is transferred upwards, not downwards. In practical terms, this means controlling the form of $\Psi(z)$, to ensure that it represents an upwardly propagating wave.

**Side Boundary Conditions** We use an *idealised channel model* on the $\beta$-plane, with vertical 'walls' imposed along latitudes corresponding to $y = 0, L$. The northward wind $\psi'_x = v'$ vanishes at the walls because of the $\sin ly$ term of the wavelike solution, provided that $lL$ is an integer multiple of $\pi$.

### 2.2.5 Analytical Solutions

In this section we look briefly at the analytical solution of equation (30) when $\bar{u}(z) =$ constant. This will indicate the form of the solutions for more complicated wind profiles and provide a useful diagnostic test for the computer model.

With $u(z) =$ constant, equation (25) gives $\bar{q}_y = \beta$ and therefore $F(z) = F =$ constant. We look for solutions to (30) of the form,

$$\Psi \sim e^{\lambda z}. \tag{35}$$

On substitution, we then have,

$$(\lambda^2 + F)\Psi = 0,$$

or,

$$\lambda^2 = -F, \tag{36}$$

where,

$$F = \frac{N^2}{f_0^2}\left[\frac{\beta}{\bar{u} - c} - (k^2 + l^2) - \frac{\epsilon}{4H^2}\right].$$

We must consider the solution in the cases where $F < 0$ and $F > 0$:

[8] Q. how do we know this?

***F* < 0** We set $F = -\mu^2$ which from (36) gives $\lambda = \pm\mu$. Recalling that $\hat{\psi} = e^{z/2H}\Psi$, our solutions are $\hat{\psi} \sim e^{(\frac{1}{2H}\pm\mu)z}$. To determine the correct sign in the exponent we consider the wave kinetic energy per unit volume which is given by $\frac{1}{2}\rho_o(\psi_x'^2 + \psi_y'^2)$; here this will be proportional to $e^{\pm 2\mu z}$. The lower sign is the physically acceptable one, with the quantity decreasing exponentially with height. We therefore have,

$$\psi' = \psi_0'(x, y, t)e^{(\frac{1}{2H}-\mu)z}. \tag{37}$$

This represents a *vertically trapped mode*; the amplitude changes exponentially with $z$ and the phase remains constant. It can be seen from (37) that $\psi'$ may grow with height but the exponentially decreasing density $\rho_0 = \rho_s e^{-z/H}$ ensures that the kinetic energy density always falls with height.

***F* > 0** We set $F = m^2$ which gives $m = \pm F^{1/2}$ and $\lambda = im$ from (36). Recalling that $\hat{\psi} = e^{z/2H}\Psi$, we obtain the following from equation (28),

$$\psi' = \mathrm{Re}\,\hat{\psi}_0\, e^{(\frac{z}{2H}+i\{kx-kct+mz\})}\sin ly. \tag{38}$$

This represents *vertical propagation*; the $imz$ term means that the phase changes with height. We may write,

$$F = m^2 > 0,$$

which, using the definition of $F$, becomes,

$$\frac{\beta}{\bar{u}-c} - (k^2 + l^2) - \frac{\epsilon}{(2H)^2} = m^2\epsilon > 0,$$

or

$$\frac{\beta}{\bar{u}-c} = k^2 + l^2 + \epsilon\left\{m^2 + \frac{1}{(2H)^2}\right\} > 0.$$

This shows that for a propagating wave solutions we must have $\bar{u} > c$; the phase speed must be westward with respect to the mean zonal flow for vertically propagating Rossby waves to occur. If $k, l, \epsilon, H$ are fixed with $m^2$ positive we also see that,

$$\frac{\beta}{\bar{u}-c} > k^2 + l^2 + \frac{\epsilon}{(2H)^2},$$

so that,

$$0 < \bar{u} - c < \beta\left\{k^2 + l^2 + \frac{\epsilon}{(2H)^2}\right\} \equiv \bar{u}_c, \qquad \text{say.} \tag{39}$$

This expression is known as the Charney-Drazin criterion[9] and it states that we can only have vertical propagation when $\bar{u} - c$ is not too large. For stationary waves, where $c = 0$, we must have,

$$0 < \bar{u} < \bar{u}_c \tag{40}$$

for vertical propagation; the winds must be westerly and not too strong.

We must now consider the sign of $m$ by applying the radiation condition discussed in Section 2.2.4. Following [2], we find the Rossby wave dispersion relation,

$$\omega = k\bar{u} - \frac{\beta k}{k^2 + l^2 + \epsilon\left(m^2 + \frac{1}{(2H)^2}\right)} \tag{41}$$

[9] See [4].

and use it to find the wave group velocity,

$$c_g^{(z)} = \frac{\partial \omega}{\partial m} = \frac{2\epsilon\beta k m}{\left\{k^2 + l^2\epsilon\left(m^2 + \frac{1}{(2H)^2}\right)\right\}^2}, \tag{42}$$

which takes a positive value if $m > 0$ ($k$ is taken to be positive). Thus the positive $m = +F^{1/2}$ is appropriate. The phase lines $kx - kct + mz = constant$ therefore tilt *westward* with height.

### 2.2.6 Linear Dissipation

All equations discussed up to now have dealt with an atmosphere that is physically unrealistic in the sense that there is no wave dissipation. For the results produced by our model to be meaningful, it is necessary to take this into account. However, the inclusion of a full treatment of the phenomena would entail significant computational difficulties and indeed is not required in a model of this accuracy. A simple but demonstrative form of dissipation can be introduced into equation (24) by including a linear term proportional to $q'$,

$$\left(\frac{\partial}{\partial t} + u_g\frac{\partial}{\partial x}\right)q' + \bar{q}_y\frac{\partial \psi'}{\partial x} = -\gamma(z)q', \tag{43}$$

where $1/\gamma$ is the timescale, which will be taken to be a function of height or be set to a single constant value as appropriate. We are effectively setting the frictional and thermal damping to be equal. With the $e^{ik(z-ct)}$ dependence of our wavelike solutions, equation (43) gives,

$$\begin{aligned} &\left\{ik(\bar{u} - c) - \gamma\right\}q' + \bar{q}_y\frac{\partial \psi'}{\partial x} = 0, \\ \rightarrow \qquad &\left\{ik\left(\bar{u} - c - \frac{\gamma}{ik}\right)\right\}q' + \bar{q}_y\frac{\partial \psi'}{\partial x} = 0. \end{aligned}$$

Hence, equation (29) becomes,

$$\frac{1}{\rho_0}\frac{d}{dz}\left(\rho_0\epsilon\frac{d\hat{\psi}}{dz}\right) + \left[\frac{\bar{q}_y}{\bar{u} - c - \frac{i\gamma}{k}} - (k^2 + l^2)\right]\hat{\psi} = 0, \tag{44}$$

and equation (31),

$$F(z) = \frac{N^2}{f_0^2}\left[\frac{\bar{q}_y}{\bar{u} - c - \frac{i\gamma}{k}} - (k^2 + l^2) - \frac{\epsilon}{4H^2}\right]. \tag{45}$$

A typical value for $1/\gamma$ in the stratosphere would be between 10 and 20 days.

### 2.2.7 Model Parameters

Typical stratospheric values must be chosen for the constants in the QGPV equation.

$\mathbf{c = 0}$ We take the wave phase speed to be zero giving zonally 'stationary waves'.

$\boldsymbol{\phi = 45^{\circ}}$ **N** The approximations discussed above apply most accurately to mid-latitude disturbances. We take $\phi = 45^{\circ}$ N as the latitude.

**$N^2 = 5\times10^{-4}$ s$^{-2}$** This measure of the atmosphere's density stratification, defined by equation (22), is given a constant representative stratospheric value.

**$l = \pi/(10\ 000$ km$)$** The parameter $l$ determines the meridional wavelength ($2\pi l^{-1}$). The value of $l$ is chosen to give a wavelength of 20 000 km.

**$k = s/(a \cos\phi)$** The parameter $k$ determines the zonal wavelength ($2\pi k^{-1}$). Here, $a = 6371$ km is the radius of the earth and $s = 1, 2, 3\ldots$ is the zonal wavenumber, the number of wavelengths circling the Earth. $s = 1$ is used throughout this project.

**$\Omega = 7.292\times10^{-5}$ s$^{-1}$** This is the rotation rate of the Earth.

**$H = 7$ km** This representative stratospheric scale height is found using equation (8) with $T_s = 240$ K.

## 2.3 The Numerical Model

### 2.3.1 Formal Requirements of the Model

Following the theoretical discussions in the previous section, we may now formally specify the numerical computer model.

We seek solutions to the following linear differential equation:

$$\frac{d^2\Psi}{dz^2} + F(z)\Psi = 0, \tag{46}$$

where,

$$F(z) = \frac{N^2}{f_0^2}\left[\frac{\bar{q}_y}{\bar{u} - c - \frac{i\gamma}{k}} - (k^2 + l^2) - \frac{\epsilon}{4H^2}\right], \tag{47}$$

and,

$$\bar{q}_y = \beta - \frac{1}{\rho_0}\frac{\partial}{\partial z}\left(\rho_0 \epsilon \frac{\partial \bar{u}}{\partial z}\right). \tag{48}$$

We also note that,

$$\rho_0 = \rho_s e^{-z/H}, \tag{49}$$

and[10],

$$\Psi = \hat{\psi} e^{-z/2H}. \tag{50}$$

We require a solution between the log-pressure heights of 10 and 70 km. For numerical convenience we set the lower boundary at $z = 0$ km and the upper at $z = 60$ km, choosing a $\rho_s$ such that this is possible. From here onwards the values of z are to be taken as those above 10 km. From Section 2.2.4 we have our boundary conditions: at $z = 0$ km we specify the value of $\mathrm{d}\hat{\psi}/\mathrm{d}z$ and at $z = 60$ km we require that $\Psi \sim e^{i\lambda z}$.

We must be able to systematically specify and modify:

- $\bar{u}(z)$, the mean zonal flow as a function of log-pressure height. We require control over the shape and magnitude of this function.
- $\gamma(z)$, the strength of the linear dissipation.

[10] Recall from Section 2.2.2 that the assumed solution to the linearised QGPV equation was $\psi' = \mathrm{Re}\,\hat{\psi}(z)e^{ik(x-ct)}\sin ly$.

- The various numerical constants in equations (46) to (50)[11].
- Various parameters such as the step size used in the numerical method.

The computer model should provide graphical and numerical output of the following quantities:

- Mod $\Psi(z)$, Arg $\Psi(z)$, Mod $\hat{\phi}(z)$, Arg $\hat{\phi}(z)$
- $\bar{u}(z)$, $F(z)$, $\bar{q}_y(z)$
- Intermediate variables involved in the numerical method as functions of $z$. These are to be used for diagnostic purposes.
- The z-component of the EP Flux[12] $F^{(z)}(z)$, a quantity representing the mean upward transfer of horizontal momentum by the waves.

We also require the ability to systematically vary some features of a given wind profile, at each step recording the value of some quantity, such as $F^{(z)}(z)$ at a given height.

#### 2.3.2 Programming Language - IDL

IDL (Interactive Data Language) is a high level programming language produced by Research Systems Inc. It is particularly useful for programs involving a lot of array and matrix manipulation and for those requiring good quality graphical output. These qualities make it suitable for use throughout this project.

#### 2.3.3 Numerical Methods

We solve equation (46) using a method designed specifically for two point boundary value problems of this type. The method is a form of Gaussian elimination; it is discussed in detail in Appendix A.3.

We divide our $z$-domain into a number of discrete levels $z_0, z_1, z_2, \ldots, z_M$ where $z_0 = 0$ and the remaining levels are equally spaced with a separation $\delta z$. From equations (119) to (121) we obtain a relationship between the values of $\Psi(z)$ at adjacent levels, $z_m$ and $z_{m+1}$,

$$\Psi(z_m) = -\frac{1}{F(z_m)(\delta z)^2 - 2 + \alpha_{m-1}} \cdot \Psi(z_{m+1}) - \frac{\beta_{m-1}}{F(z_m)(\delta z)^2 - 2 + \alpha_{m-1}}, \qquad (51)$$

where the coefficients $\alpha_m$ and $\beta_m$ satisfy,

$$\alpha_m = \frac{-1}{F(z_m)(\delta z)^2 - 2 + \alpha_{m-1}}, \qquad (52)$$

$$\beta_m = \frac{-\beta_{m-1}}{F(z_m)(\delta z)^2 - 2 + \alpha_{m-1}}. \qquad (53)$$

If we know the values of $\alpha_0$ and $\beta_0$ we can find all $\alpha_m$'s and $\beta_m$'s as the first step to determining $\Psi(z)$ . To find $\alpha_0$ and $\beta_0$ we must consider the lower boundary condition.

[11] See Section 2.2.7.
[12] See Appendix B.

Equation (50) gives $\Psi(z) = \hat{\psi(z)} e^{-z/2H}$ which, on differentiation with respect to $z$, becomes,

$$\frac{d\Psi}{dz} = e^{-z/2H} \frac{d\hat{\psi}}{dz} - \frac{1}{2H}\Psi.$$

Following the discussion of Section 2.2.4 we specify the value of $d\hat{\psi}/dz$ at $z = 0$ and call it $A$. At the lower boundary we then have,

$$\left.\frac{d\Psi}{dz}\right|_{z_0} + \frac{1}{2H}\Psi(z_0) = A. \tag{54}$$

Using the method of finite differences we approximate the differential on the LHS,

$$\left.\frac{d\Psi}{dz}\right|_{z_0} \simeq \frac{\Psi(z_1) - \Psi(z_0)}{\delta z},$$

and equation (54) becomes,

$$\Psi(z_0) = \alpha_0 = \frac{1}{1 - \frac{\delta z}{2H}} \cdot \Psi(z_1) - \frac{A\delta z}{1 - \frac{\delta z}{2H}}. \tag{55}$$

We know from equation (117) that $\Psi(z_m) = \alpha_m \Psi(z_{m+1}) + \beta_m$ so,

$$\alpha_0 = \frac{1}{1 - \frac{\delta z}{2H}}, \qquad \beta_0 = \frac{-A\delta z}{1 - \frac{\delta z}{2H}}. \tag{56}$$

With equations (52) and (53) we can proceed to find all $\alpha_m$'s and $\beta_m$'s.

To finally determine $\Psi(z)$ all we now need is the function's value at the upper boundary, $\psi(z_M)$. From the radiation condition and the discussion of Section 2.2.5 we know that at the boundary, $\Psi(z) = Ce^{i\lambda z}$ where $C$ is a constant. Differentiating and using the method of finite differences we have,

$$\left.\frac{d\Psi}{dz}\right|_{z_M} = i\lambda\Psi(z_M),$$

$$\rightarrow \left\{ \frac{\Psi(z_M) - \Psi(z_{M-2})}{2\delta z} \right\} = i\lambda\Psi(z_{M-1}). \tag{57}$$

Using equation (117), we may write the following relationships,

$$\Psi(z_M) = \alpha_M \Psi(z_{M+1}) + \beta_M,$$

$$\Psi(z_{M-1}) = \alpha_{M-1} + \Psi(z_M) + \beta_{M-1},$$
$$\Psi(z_{M-2}) = \alpha_{M-2}\Psi(z_{M-1}) + \beta_{M-2},$$

which, in conjunction with equation (57), allows us to write,

$$\Psi(z_M) = \frac{\beta_{M-2} + \beta_{M-1}(2i\lambda\delta z + \alpha_{M-2})}{1 - \alpha_{M-1}(2i\lambda\delta z + \alpha_{M-2})}. \tag{58}$$

This expression gives the value of $\Psi$ at the upper boundary in terms of the values of $\alpha$ and $\beta$ which we already know.

We now summarise the procedure for applying this numerical method:

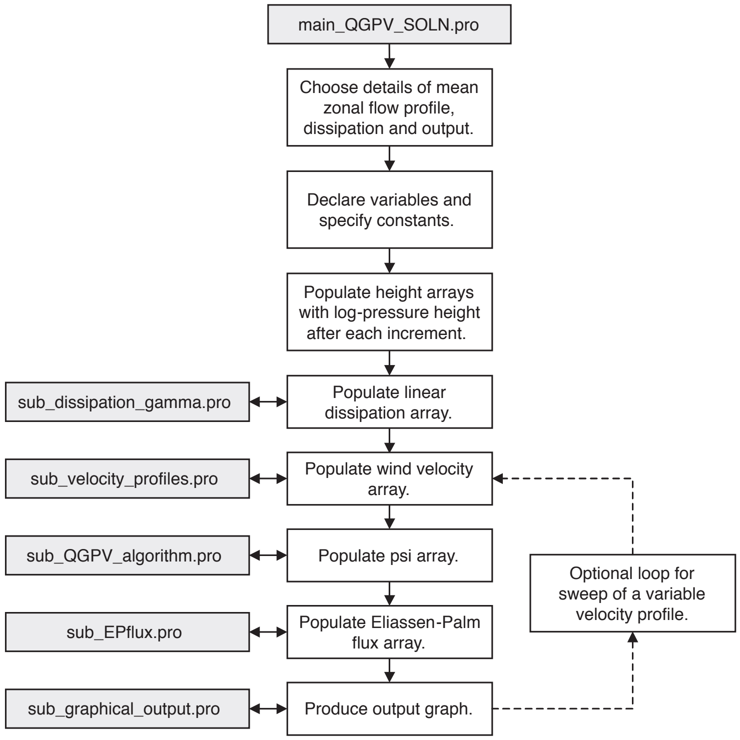


Figure 4: Broad structure of the IDL program used to numerically solve the linearised QGPV equation.

1. Use the lower boundary condition on $\Psi$ to find $\alpha_0$ and $\beta_0$.
2. Use equations (52) and (53) to find the values of $\alpha_m$ and $\beta_m$ over the whole $z$ domain.
3. Use the upper boundary condition on $\Psi$ to find $\Psi$ at this level.
4. Use equation (51) to find the value of $\Psi$ at all lower levels.

### 2.3.4 Program Structure

Figure 2.3.4 shows the broad structure of the IDL program used to perform the numerical method discussed above. The program has a modular design with the main routine calling various procedures and functions, from other files, to accomplish different subtasks. The full program code can be found in Appendix D along with technical details of its operation. In this section we restrict ourselves to a brief overview.

The main program (`main_QGPV_soln.pro`) begins with a set of options where the user can specify the mean zonal wind, $\delta z$, $\gamma(z)$, the required output from the model and other parameters.

The number of steps in the numerical method is calculated first and arrays are declared with the appropriate number of entries. The main body of the program consists of stepping through these arrays calculating a series of quantities at each

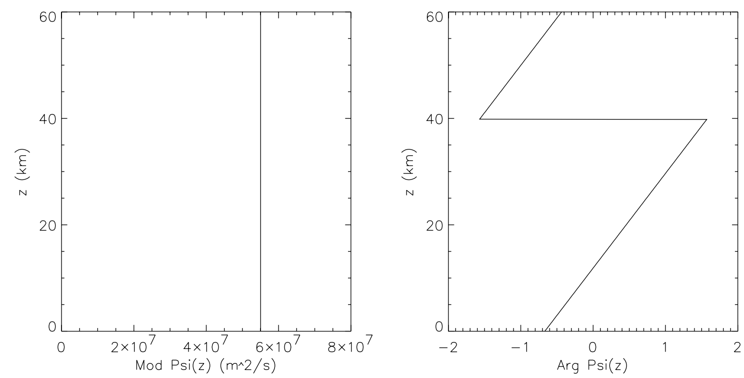


Figure 5: Testing the accuracy of the model. Plots of $|\Psi|(z)$ and Arg $\Psi(z)$ for profile 1, a forcing of A=5000 and, level separation $\delta z = 50$ m and no dissipation.

point. There is therefore a loop through the numerical method but also a larger loop that repeats the numerical method for a varying mean flow profile (most mean flow profiles have a free parameter $zp$ that can be varied; for example, in profile 2, the magnitude of the flow increases linearly with height up to $zp$, at which point it becomes constant). This enables us to systematically vary a mean flow profile and examine, for example, the EP flux at 0 km at each step of the variation.

The functions called by the main program are:

`sub_dissipation_gamma.pro`
: Populates an array with the value of the linear dissipation $\gamma(z)$ at each level.

`sub_velocity_profiles.pro`
: Populates an array with the value of the mean zonal flow $\bar{u}(z)$ at each level. There are six profiles available, all idealised representations of typical natural flows. The profiles are shown in Appendix C.

`sub_QGPV_algorithm.pro`
: Solves the QGPV equation to find $\psi(z)$ at each level.

`sub_EPflux.pro`
: Calculates the Eliassen-Palm flux at each level.

`sub_graphicaloutput.pro`
: Produces graphical output of the requested variable. This can be either in a window or postscript file.

## 2.4 Results

### 2.4.1 Comparison of the Model with Analytical Solutions

In this section we compare the results of our numerical model in the case where $\bar{u}(z) =$ constant (profile 1) to the analytical solution discussed in Section 2.2.5. This provides a test of the accuracy of the model.

| $\delta z$ (m) | $\lvert\Psi\rvert$ Numerical $\times 10^{-7}$ | $\lvert\Psi\rvert$ Analytical $\times 10^{-7}$ | Error (%) | Arg $\Psi(z=0)$ Numerical | Arg $\Psi(z=0)$ Analytical | Error (%) |
|---|---|---|---|---|---|---|
| 25 | 5.502 | 5.500E-7 | 0.036 | -0.667164 | -0.6668803 | 0.043 |
| 50 | 5.504 | | 0.068 | -0.667414 | | 0.080 |
| 100 | 5.508 | | 0.137 | -0.667955 | | 0.161 |
| 500 | 5.538 | | 0.693 | -0.672307 | | 0.814 |
| 1000 | 5.578 | | 1.406 | -0.677826 | | 1.641 |

Table 1: A comparison between the analytical and numerical solutions of the linearised QGPV equation for a constant zonal wind of 50 $\mathrm{ms}^{-1}$. The dependence of the numerical error on the model interval size is shown.

With $\bar{u} = 50.0$ $\mathrm{ms}^{-1}$, a forcing of $A = 5000$ (used throughout this project), a level separation $\delta z = 50.0$ m and no dissipation we have $|\Psi|$ and Arg $\Psi$ as shown in Figure 5.

As expected, we see a wave amplitude that is constant with height and a phase that increases linearly[13], representing undamped vertical wave propagation.

The value of $|\Psi|$ is not completely constant, as it appears in Figure 5; there is a degree of variation in it's value, arising from the numerical method, that decreases as we decrease the step size $\delta z$. With $\delta z = 50$ m this is of order $1 \times 10^{-4}$%.

Using the analytical solution of Section 2.2.5 we are able to calculate the exact values of $|\Psi|$ and Arg $\Psi$, so finding the accuracy of the numerical method as a function of step size. This is shown in Table 1. We see, for example, that with $\delta z = 50$ m we can find $|\Psi|$ accurate to within 0.07%.

#### 2.4.2 Effects of Stratospheric Wind on the Geostrophic Streamfunction

We consider the results of the numerical model for mean zonal flow profile 2 which are shown in Figure 6. We use a level separation of 50 km and no dissipation.

The velocity takes a positive constant value above 30 km and, as with profile 1 discussed above, here we would expect $|\Psi|$ to be constant and Arg $\Psi$ to increase linearly. This is indeed what is seen.

Below 30 km, where the velocity is increasing, we have reflected waves. These are expected to interfere with the upwardly propagating waves from the initial forcing. In Figure 6 we see undulations in $|\Psi|$, characteristic of this interference, and disturbances to Arg $\Psi$ because we have wave components propagating both up and down.

In other profiles, where the mean zonal flow becomes negative we expect, and indeed find, no wave propagation. Consider for example Figure 7. The velocity falls to zero at 30km and propagation stops abruptly; the wave is totally reflected.

#### 2.4.3 Effects of Dissipation

In this section we discuss the effects of wave dissipation. The level of dissipation used is $1/\gamma = 20$ days, which is a value typical in the stratosphere[14].

[13] Phase is only plotted between $-\pi/2$ and $+\pi/2$, with the sharp discontinuity occurring when Arg $\Psi$ passes $+\pi/2$ and $\pi$ is deducted from its value.

[14] See Section 2.2.6.

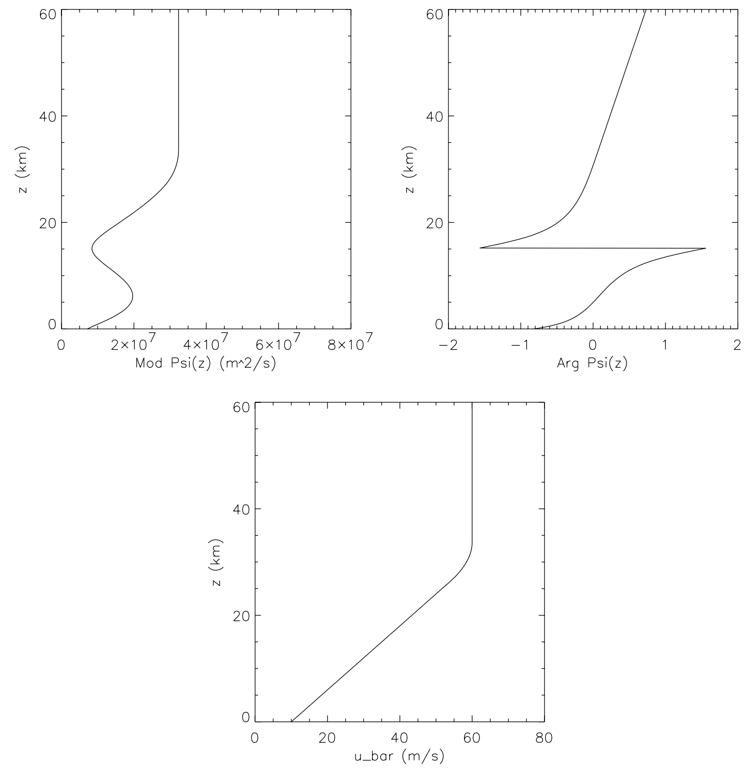


Figure 6: Plot of $|\Psi(z)|$, Arg $\Psi(z)$ and $\bar{u}(z)$ for mean zonal flow profile 2 with no dissipation.

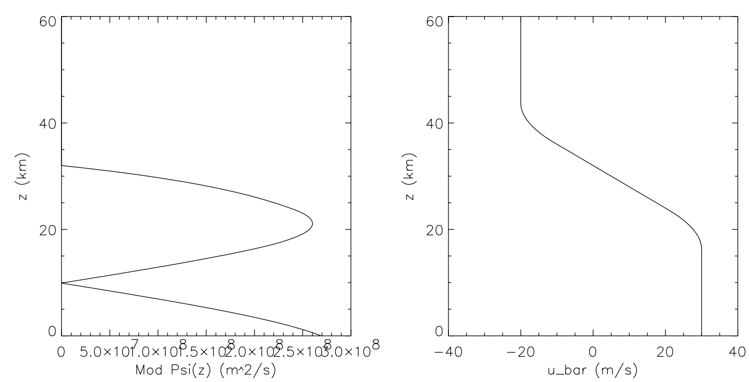


Figure 7: Plot of $|\Psi(z)|$ and $\bar{u}(z)$ for mean zonal flow profile 4 with no dissipation.

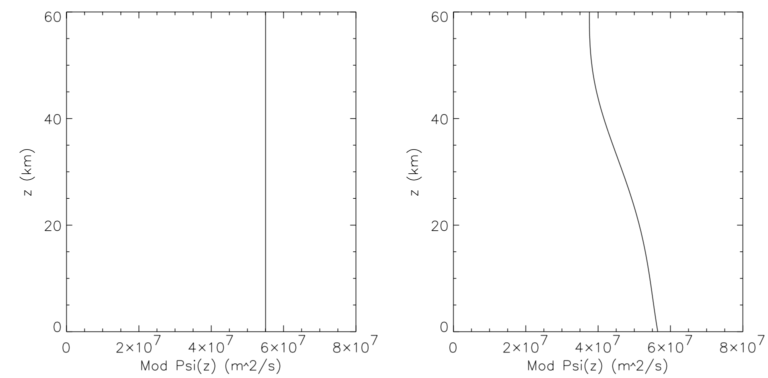


Figure 8: Plots of $|\Psi(z)|$ for mean zonal flow profile 1. The right hand plot includes wave dissipation, the left hand does not.

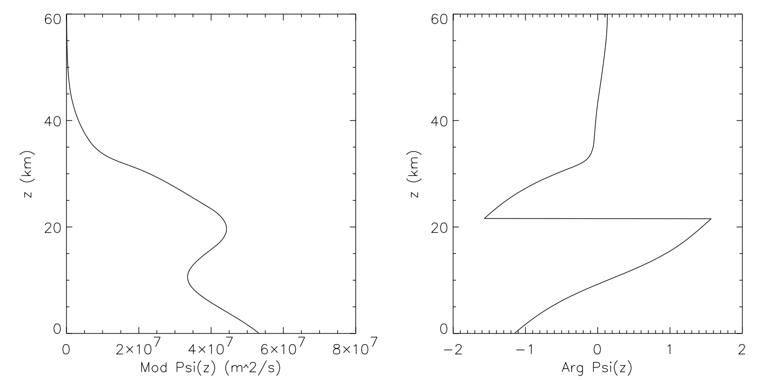


Figure 9: Plots of $|\Psi(z)|$ and Arg $\Psi(z)$ for mean zonal flow profile 4 with dissipation included. Compare with Figure 2.4.2

Figure 8 shows the effect of including wave dissipation when using mean zonal flow profile 1. A non-linear reduction of $|\Psi|$ with height can be seen. There is no change in Arg $\Psi$ with height. The dissipation effects only the wave amplitude.

Shown in Figure 9 are plots of $|\Psi(z)|$ and Arg $\Psi(z)$ for mean zonal flow profile 4, with dissipation included. They should be compared with the plots of Figure 7 where there is no dissipation. At 30 km, where the velocity drops to zero, we see that Arg $\Psi$ tends to a constant value indicating no wave propagation. However, $|\Psi(z)|$ does not fall sharply to zero; the wave is no longer totally excluded from the region above 30 km. In equation (45) dissipation is included via a complex term in $F(z)$; following through the algebra of Section 2.2.5, it can be seen that the imaginary term generates a non-zero component in the $\bar{u}(z) = 0$ region.

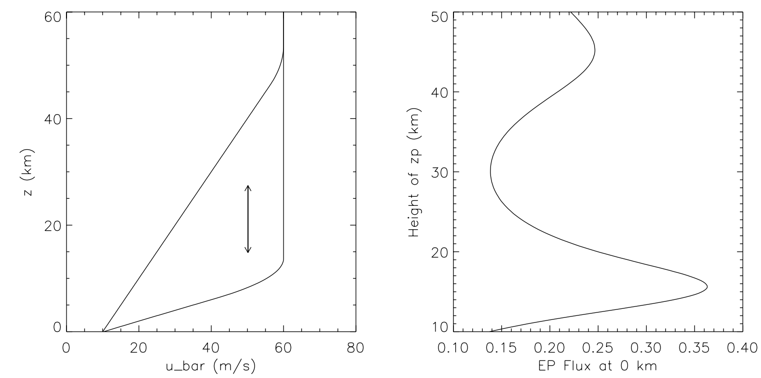


Figure 10: Plot of the EP flux at the base of the stratosphere, with zonal mean flow profile 2, against $zp$, the height at which $\bar{u}$ becomes constant.

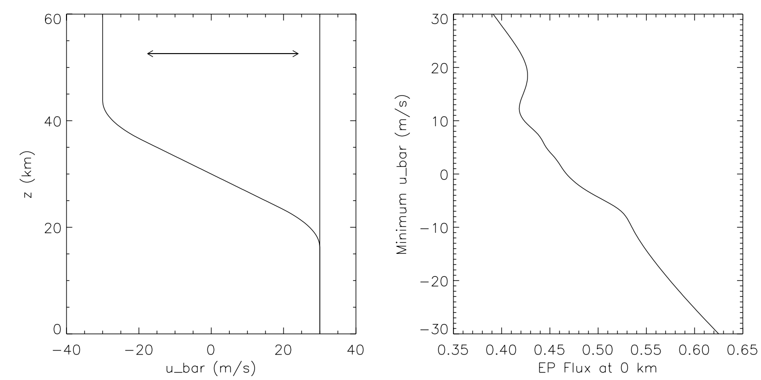


Figure 11: Plot of the EP flux at the base of the stratosphere, with zonal mean flow profile 4, against $zp$, the minimum value of $\bar{u}$.

### 2.4.4 Effects of Stratospheric Wind on the EP-Flux At The Base of the Stratosphere

With reference to the motivation for this project, we are particularly concerned with how the zonal mean flow high up in the stratosphere affects conditions lower down. As discussed in Section 2.3.4, all $\bar{u}$ profiles have a variable parameter which can be systematically swept. We alter this parameter in a range of different profiles and investigate the resulting effects on the EP flux at 0 km.

All the plots discussed below are produced with a forcing of $A = 5000$, increment size $\delta z = 50.0$ m and dissipation based on 20 days.

With profile 2, we vary the altitude at which a linearly increasing $\bar{u}$ takes on a constant value. We expect reflected waves whenever $\bar{u}$ changes and here, with the sharp change at $zp$, we expect a reflected wave with a significant amplitude. The interference between the incident and reflected waves produces the plot shown in Figure 10; the pattern is characteristic of this interference.

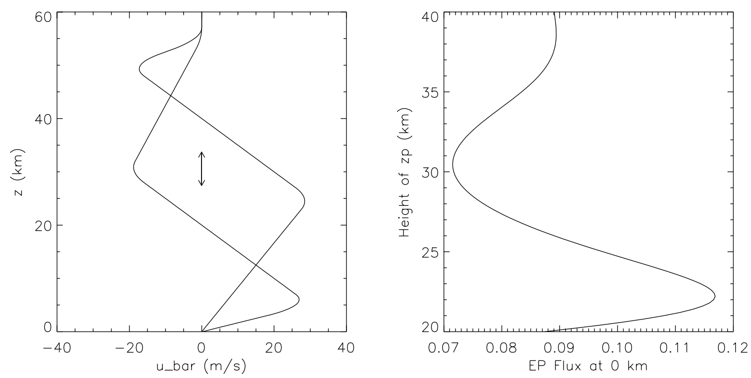


Figure 12: Plot of the EP flux at the base of the stratosphere, with zonal mean flow profile 5, against $zp$, the vertical height of the critical wind $(\bar{u} - c) = 0$ ms$^{-1}$.

We turn now to profile 4 where we have a constant $\bar{u}$ at low $z$, followed by a linear decrease in velocity to a constant minimum value. It is this minimum value that is systematically varied. A detailed interpretation of the results, which are shown in Figure 11, is not clear. However, the broad decrease in the EP flux at 0 km, as the minimum $\bar{u}$ steadily increases, is to due to the decreased magnitude of the wave trapping. Increasing wave trapping, causes an increase in the magnitude of reflected waves; here, a variation in conditions above 20 km is causing a change in EP flux at 0 km of $\sim 50\%$.

With profiles number 5 and 6 we have a steadily increasing $\bar{u}$ at low $z$ followed by a sharp turning point and a decrease through $\bar{u} = 0$ to a negative value; this is followed by a brief increase then a constant value. Profile 5 begins and ends on 0 km while profile 6 begins with a positive value and ends on a constant negative value. In both profiles, $zp$ is the value of $z$ at which $\bar{u} = 0$.

The resulting plots, shown in Figures 12 and 13, have oscillatory behavior similar to that of Figure 10, with presumably the same physical origin. We see a striking difference, however, in the overall magnitudes of 5 and 6 and in the standard deviations. Profile 5 has a mean of 0.09 and a standard deviation of 0.0147 about the mean. Profile 6 has a mean of 0.20 and a standard deviation of 0.0597, four times that of profile 5. The difference arises from the fact that profile 5 has $\bar{u}(z = 0) = 0$: Arg $\Psi$ for both profiles varies little with $z$ so the EP flux is approximately proportional to the square of $|\Psi|$, by equation (125), and $\bar{u}(z = 0) = 0$ causes an initial damping of $|\Psi|$ in profile 5 relative to profile 6.

## 2.5 Conclusions

We have investigated the vertical structure of mid-latitude planetary waves in the stratosphere, forced from below by the troposphere. In our simple model we have found indications that, for a given tropospheric forcing, the size of the stratospheric response can can vary significantly depending upon the form of the zonal mean flow high up in the stratosphere. These results could be investigated further and developed using a more sophisticated wave model.

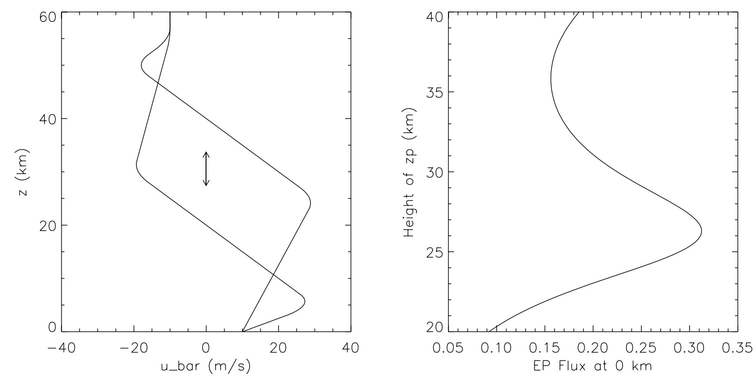


Figure 13: Plot of the EP flux at the base of the stratosphere, with zonal mean flow profile 6, against $zp$, the vertical height of the critical wind $(\bar{u} - c) = 0$ ms$^{-1}$.

# 3 Changes to the Mean Flow

## 3.1 Mean Flow Interactions

### 3.1.1 Introduction

Up to now we have examined the effect of the form of the background mean zonal flow on the properties of a class of planetary waves. These waves have been modelled as a disturbance, or perturbation, on the mean flow. We can extend our discussion by taking into account the accelerations and decelerations of the mean flow produced by the disturbance. The changes caused in this way go on to modify the planetary wave structure and cause further changes to the mean flow. Our system is no longer static in time.

In this part of the project we adapt the routines developed earlier on, coupling them with a series of simple non-linear mean-flow models. This will allow us to follow the time development our system.

We consider three different mathematical approaches to this problem, two relatively rigorous (methods A and B), and one based on a simpler theory (method C). We discuss the effectiveness of these models and the results we are able to obtain using them. The reason for using three models to explore the same physical phenomena highlights some of the limitations of the numerical methods and assumptions we use; within the context of this model, only the simpler theory (method C) produced results of physical interest. We do still present the first two methods in detail as they are instructive and may form the basis for further investigations.

We begin by discussing the theoretical basis of each method in turn, before going on to the numerical models themselves in detail.

### 3.1.2 Changes to the Mean Flow (Method A)

From [2] Chapter 4, we have an expression describing the mean flow interaction described above,

$$\left[\frac{\partial^2}{\partial y^2} + \frac{1}{\rho_0}\frac{\partial}{\partial z}\Big(\rho_0\epsilon\frac{\partial}{\partial z}\Big)\right]\frac{\partial\bar{u}}{\partial t} + \frac{1}{\rho_0}\frac{\partial}{\partial z}\left[\frac{\rho_0\epsilon}{\tau_r}\frac{\partial}{\partial z}(\bar{u}-u_r)\right] = \frac{\partial^2}{\partial y^2}\left[\frac{1}{\rho_0}\nabla\cdot\mathbf{F}\right], \tag{59}$$

where $\tau_r$ is the radiative relaxation time, $u_r(z)$ is the 'radiative equilibrium' wind field and $\mathbf{F}$ is the Eliassen-Palm flux. We make the approximation that $\tau_r \to \infty$[15] and the second term on the left hand side of equation (59), representing thermal relaxation, vanishes. From before, we recall that in our problem, $F^{(x)} = F^{(y)} = 0$, which gives us,

$$\left[\frac{\partial^2}{\partial y^2} + \frac{1}{\rho_0}\frac{\partial}{\partial z}\Big(\rho_0\epsilon\frac{\partial}{\partial z}\Big)\right]\frac{\partial\bar{u}}{\partial t} = \frac{\partial^2}{\partial y^2}\left[\frac{1}{\rho_0}\frac{\partial F^{(z)}}{\partial z}\right]. \tag{60}$$

In principle, this equation can be solved to yield $\partial\bar{u}/\partial t$ at a time $t_1$ allowing us to determine $\bar{u}$ at a later time $t_2 = t_1 + \delta t$.

We must make further approximations to equation (60) to make it suitable for solution by numerical methods. We assume the form of the $y$ dependence of EP flux derivative on the right hand side of equation (60),

$$\frac{1}{\rho_0}\frac{\partial F^{(z)}}{\partial z} = G(z)\sin^2 ly, \tag{61}$$

where $G(z)$ is a function to be determined and $l$ is the meridional wavenumber. We also assume the form of $\frac{\partial\bar{u}}{\partial t}$,

$$\frac{\partial\bar{u}}{\partial t} = A(z,t)\cos 2ly \tag{62}$$

where $A(z,t)$ is a function to be determined.

Substituting equations (61) and (62) into (60), and noting from before that $\epsilon =$ constant and $\rho_0 = \rho_s e^{-z/H}$, we get,

$$\left[\frac{d^2}{dz^2} - \frac{1}{H}\frac{d}{dz} - \frac{4l^2}{\epsilon}\right]A(z,t_2) = \frac{2l^2}{\epsilon}G(z,t_1). \tag{63}$$

If we now take the value of $\bar{u}$ to be that at the centre of our channel, where $y = l/2$, then $\cos 2ly = -1$ and therefore, $A = -\frac{\partial\bar{u}}{\partial t}$. A knowledge of $A$ tells us the rate of change of $\bar{u}(z)$ at that point in time. Thus, given a mean flow profile $\bar{u}(z)$ at a time $t_1$, equation (63) can be solved to find $A(t_1) = -\frac{\partial\bar{u}}{\partial t}\Big|_{t_1}$ and hence $\bar{u}(z)$ at a later time $t_2 = t_1 + \delta t$. By the method of finite differences,

$$\bar{u}(z,t_2) = \bar{u}(z,t_1) + \frac{\partial\bar{u}}{\partial t}\bigg|_{t_1}\partial t$$

$$\rightarrow \bar{u}(z,t_2) = \bar{u}(z,t_1) - A(z,t_1)\cdot\delta t. \tag{64}$$

This model is made more physical by including a term representing the damping of the mean flow accelerations. We modify equation (62),

$$\frac{\partial\bar{u}}{\partial t} = A(z,t)\cos 2ly - \mu(z)\bar{u}(z,t), \tag{65}$$

where $\mu(z)$ is the damping coefficient and is allowed to vary with height. Following through the algebra above with this new expression gives,

$$\left[\frac{d^2}{dz^2} - \frac{1}{H}\frac{d}{dz} - \frac{4l^2}{\epsilon}\right]A(z,t_2) = \frac{2l^2}{\epsilon}\left\{G(z,t_1) + \mu(z)\frac{\epsilon}{2l^2}\left[\frac{d^2}{dz^2} - \frac{1}{H}\frac{d}{dz}\right]\bar{u}(z,t_1)\right\}. \tag{66}$$

[15] Q. What does this correspond to physically?

### 3.1.3 Changes to the Mean Flow with Rayleigh Friction (Method B)

We begin by considering equation (59),

$$\left[\frac{\partial^2}{\partial y^2}+\frac{1}{\rho_0}\frac{\partial}{\partial z}\left(\rho_0\epsilon\frac{\partial}{\partial z}\right)\right]\frac{\partial\bar{u}}{\partial t}+\frac{1}{\rho_0}\frac{\partial}{\partial z}\left[\frac{\rho_0\epsilon}{\tau_r}\frac{\partial}{\partial z}(\bar{u}-u_r)\right]=\frac{\partial^2}{\partial y^2}\left[\frac{1}{\rho_0}\nabla\cdot\mathbf{F}\right].$$

We simplify this in the same way as in the previous section, using the approximation $\tau_r\to\infty$, but in this method we introduce a Rayleigh friction term $-\zeta\bar{u}$ inside the differential on the right hand side, where $\zeta$ is a constant. Rayleigh friction causes damping of the mean flow interaction proportional to the speed of the air flow. We have,

$$\left[\frac{\partial^2}{\partial y^2}+\frac{1}{\rho_0}\frac{\partial}{\partial z}\left(\rho_0\epsilon\frac{\partial}{\partial z}\right)\right]\frac{\partial\bar{u}}{\partial t}=\frac{\partial^2}{\partial y^2}\left[\frac{1}{\rho_0}\frac{\partial F^{(z)}}{\partial z}-\zeta\bar{u}\right]. \tag{67}$$

This equation can in principle be solved to yield $\partial\bar{u}/\partial t$ at a time $t_1$ allowing us to determine $\bar{u}$ at a later time $t_2=t_1+\delta t$.

We must make further approximations to equation (67) to make it suitable for solution by numerical methods. We assume the form of the $y$ dependence of EP flux derivative on the right hand side of (67),

$$\frac{1}{\rho_0}\frac{\partial F^{(z)}}{\partial z}=G(z)\sin^2 ly \tag{68}$$

where $G(z)$ is a function to be determined. We also assume the form of $\bar{u}$,

$$\bar{u}=B(z,t)\cos 2ly \tag{69}$$

where $B(z,t)$ is a function to be determined. Substituting equations (68) and (69) into (67) we get,

$$\left[\frac{1}{\rho_0}\frac{\partial}{\partial z}\left(\rho_0\epsilon\frac{\partial}{\partial z}-4l^2\right)\right]\frac{\partial B(z,t_2)}{\partial t}=2l^2G(z,t_1)+4l^2\zeta B(z,t_1). \tag{70}$$

Using the method of finite differences we can make the following approximation:

$$\frac{\partial B(z,t_2)}{\partial t}=\frac{B(z,t_2)-B(z,t_1)}{\delta t}.$$

If we now take the value of $\bar{u}$ to be that at the centre of our channel, where $y=l/2$, then $\cos 2ly=-1$ and, from equation (69), $B(z,t_1)=-\bar{u}(t_1)$ and $B(z,t_2)=-\bar{u}(t_2)$. With some rearrangement, and noting that $\epsilon=$ constant and $\rho_0=\rho_s e^{-z/H}$, equation (67) becomes,

$$\left[\frac{d^2}{dz^2}-\frac{1}{H}\frac{d}{dz}-\frac{4l^2}{\epsilon}\right]\bar{u}(z,t_2)=P(z,t_1), \tag{71}$$

where,

$$P(z,t_1)=\left[\frac{d^2}{dz^2}-\frac{1}{H}\frac{d}{dz}-\frac{4l^2}{\epsilon}\Big(1-\zeta\delta t\Big)\right]\bar{u}(z,t_1)-\frac{2l^2\delta t}{\epsilon}G(z,t_1), \tag{72}$$

Using this equation, with an initial mean flow profile, we may calculate $\bar{u}$ at later times.

### 3.1.4 Simple Changes to the Mean Flow (Method C)

We begin by considering equation (67),

$$\left[\frac{\partial^2}{\partial y^2} + \frac{1}{\rho_0}\frac{\partial}{\partial z}\left(\rho_0 \epsilon \frac{\partial}{\partial z}\right)\right]\frac{\partial \bar{u}}{\partial t} = \frac{\partial^2}{\partial y^2}\left[\frac{1}{\rho_0}\frac{\partial F^{(z)}}{\partial z} - \zeta \bar{u}\right].$$

and making a further strong simplification; we neglect the term in brackets on the left hand side of the equation[16]. This immediately reduces the equation (67) to,

$$\frac{\partial \bar{u}}{\partial t} = \frac{1}{\rho_0}\frac{\partial F^{(z)}}{\partial z} - \zeta \bar{u}. \tag{73}$$

We assume that we know the form of the mean flow $\bar{u}(z)$ at a time $t_1$ and want to find it's value at a later time $t_2 = t_1 + \delta t$. Using the method of finite differences we can make the following approximation:

$$\frac{\partial \bar{u}(z,t_2)}{\partial t} = \frac{\bar{u}(z,t_2) - \bar{u}(z,t_1)}{\delta t}. \tag{74}$$

Substituting equation (74) into equation (73) gives,

$$\frac{\bar{u}(z,t_2) - \bar{u}(z,t_1)}{\delta t} = \frac{1}{\rho_0}\left.\frac{\partial F^{(z)}}{\partial z}\right|_{t_1} - \zeta \bar{u}(z,t_1),$$

$$\rightarrow \bar{u}(z,t_2) = \bar{u}(z,t_1)(1 - \zeta \delta t) + \frac{1}{\rho_0}\left.\frac{\partial F^{(z)}}{\partial z}\right|_{t_1} \delta t. \tag{75}$$

Using this equation, with an initial mean flow profile, we may calculate $\bar{u}$ at later times.

### 3.1.5 Boundary Conditions

The equations of methods A and B are second order differential equations. Their solutions will therefore contain 2 arbitrary constants that must be found by using two independent boundary conditions[17]. In our problem we can use,

$$\left.\frac{\partial \bar{u}}{\partial t}\right|_{z=0} = 0, \tag{76}$$

$$\left.\frac{\partial}{\partial t}\left(\frac{\partial \bar{u}}{\partial z}\right)\right|_{z=0} = 0. \tag{77}$$

## 3.2 Method A: The Numerical Model

### 3.2.1 Introduction

Following the theoretical discussions of the previous section, we may now formally specify the numerical computer models. We produce a separate IDL program for each of the methods; these programs are similarly structured, using a common set of procedures and numerical methods. In this section we discuss the development of the numerical model used to solve the mean flow interaction equation of method A.

[16] Q. What is the physical justification this?
[17] Q. I'm not sure about the physical meaning of these boundary conditions?

### 3.2.2 Formal Requirements of the Model

We want to solve the following linear differential equation for $A(z, t_2)$:

$$\left[\frac{d^2}{dz^2} - \frac{1}{H}\frac{d}{dz} - \frac{4l^2}{\epsilon}\right]A(z,t_2) = \frac{2l^2}{\epsilon}\left\{G(z,t_1) + \mu(z)\frac{\epsilon}{2l^2}\left[\frac{d^2}{dz^2} - \frac{1}{H}\frac{d}{dz}\right]\bar{u}(z,t_1)\right\}, \quad (78)$$

where,

$$G(z,t_1) = \frac{1}{\rho_0}\frac{\partial F^{(z)}}{\partial z}\bigg|_{t_1}, \quad (79)$$

$$t_2 = t_1 + \delta t,$$

and the boundary conditions are,

$$\frac{\partial \bar{u}}{\partial t}\bigg|_{z=0} = 0, \qquad \frac{\partial}{\partial t}\left(\frac{\partial \bar{u}}{\partial t}\right)\bigg|_{z=0} = 0. \quad (80)$$

We want to find $A(z, t_2)$ for a given mean zonal flow profile $\bar{u}(z, t_1)$. We can then find $\bar{u}(z, t_2)$ using,

$$\bar{u}(z,t_2) = \bar{u}(z,t_1) - A(z,t_1)\delta t. \quad (81)$$

We must therefore set up an iterative process that will allow us to follow the development of $\bar{u}(z)$ over a series of steps in time.

We must be able to systematically specify and modify:

- $\bar{u}(z, t = 0)$, the initial mean zonal flow profile. The profiles used in the first part of this project should be available.
- $\mu(z)$, the strength of the dissipation. This will nominally be taken to be constant with height.
- The numerical method to be used.
- $\delta t$ and the number of iterative steps.

The model should produce plots showing the development of $\bar{u}(z)$ with time.

### 3.2.3 Numerical Methods

Equation (78) is solved using two independent numerical methods, the Method of Variation of Parameters and the Method of Runge-Kutta-Nyström, which are discussed, in detail, in Appendicies A.1 and A.2 respectively. We use two methods, instead of one, as a model diagnostic; the results produced by our model should be independent of the method of solution used.

**Method of Variation of Parameters** The homogeneous version of equation (78) has solutions,

$$R_1(z) = e^{\alpha_1 z}, \qquad R_2(z) = e^{\alpha_2 z},$$

where,

$$\alpha_{1,2} = \frac{1}{2H} \pm \sqrt{\frac{1}{4H^2} + \frac{4l^2}{\epsilon}}.$$

Substituting these into equation (107) we have, by the Method of Variation of Parameters,

$$A(z) = CR_1(z) + DR_2(z) + \frac{2l^2}{\epsilon(\alpha_1 - \alpha_2)} \int_0^z \left(e^{\alpha_1(z-z')} - e^{\alpha_2(z-z')}\right) M(z')dz',$$

where,

$$M(z') = G(z') + \mu(z')\frac{\epsilon}{2l^2}\left[\frac{d^2}{dz'^2} - \frac{1}{H}\frac{d}{dz'}\right]\bar{u}(z').$$

$C$ and $D$ are constants determined by the boundary conditions. If $\frac{\partial \bar{u}}{\partial t}\big|_{z=0} = 0$ then by equation (62),

$$A(z = 0) = 0, \qquad \text{and,} \qquad \left.\frac{\partial \mathrm{A}}{\partial \mathrm{z}}\right|_{\mathrm{z}=0} = 0.$$

We thus see that,

$$C + D = 0,$$

and, using equation (108),

$$C\alpha_1 + D\alpha_2 = 0,$$

hence,

$$C = D = 0.$$

$A(z, t_2)$ is thus given by the following integral,

$$A(z, t_2) = c_1 \int_0^z \left(c_2 e^{-\alpha_1 z'} - c_3 e^{-\alpha_2 z'}\right) M(z', t_1)dz', \tag{82}$$

where,

$$c_1 = \frac{2l^2}{\epsilon(\alpha_1 - \alpha_2)}, \qquad c_2 = e^{\alpha_1 z}, \qquad c_3 = e^{\alpha_2 z}, \tag{83}$$

and,

$$M(z', t_1) = G(z', t_1) + \mu(z')\frac{\epsilon}{2l^2}\left[\frac{d^2}{dz'^2} - \frac{1}{H}\frac{d}{dz'}\right]\bar{u}(z', t_1). \tag{84}$$

This integral can be evaluated using `INT_TABULATED`, a built-in IDL integration routine.

**Method of Runge-Kutta-Nyström** We rearrange equation (78) into a form suitable for the application of this numerical method,

$$A''(z, A, A') = a_1 A'(z) + a_2 M(z) + a_3 A(z), \tag{85}$$

where,

$$a_1 = \frac{1}{H}, \qquad a_2 = \frac{2l^2}{\epsilon}, \qquad a_3 = \frac{4l^2}{\epsilon}, \tag{86}$$

and,

$$M(z) = G(z) + \mu(z)\frac{\epsilon}{2l^2}\left[\frac{d^2}{dz'^2} - \frac{1}{H}\frac{d}{dz}\right]\bar{u}(z). \tag{87}$$

This is the same form as equation (110) and the numerical method can be followed through as detailed in Appendix A.2.

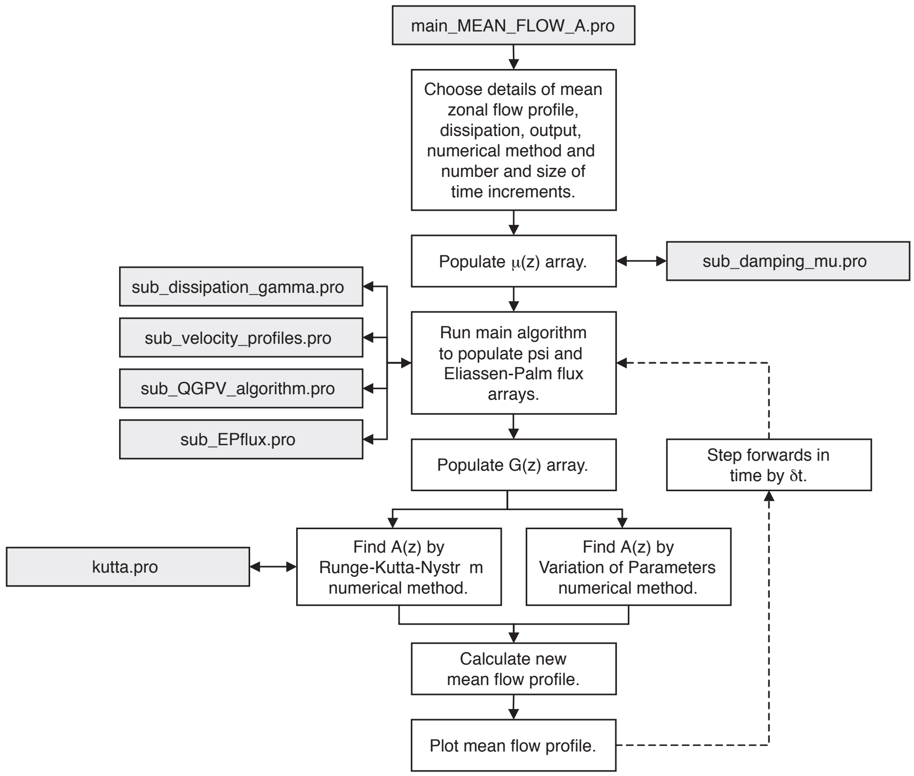


Figure 14: Broad structure of the IDL program for method A.

Note that in the above, we are dealing with a progression through time in steps of $\delta t$. The quantity $M$ is a forcing evaluated at a given time, and $A$ is the quantity derived from it at a time $\delta t$ later.

#### 3.2.4 Program Structure

Figure 14 shows the broad structure of the IDL program used to perform the numerical method described above. The program is an extension of the one discussed in Section 2.3.4. The main difference between the two is the loop to step forwards in time and the routines for carrying out the numerical methods. The only additional functions called are:

`sub_damping_mu.pro`
: Populates an array with the value of the damping $\mu(z)$ at each level.

`sub_kutta.pro`
: Calculates the function $A''(z, A, A')$ which is used repeatedly in the Runge-Kutta-Nyström numerical method.

## 3.3 Method B: The Numerical Model

### 3.3.1 Introduction

The development of this numerical model proceeds in analogy with that of method A; the differential equation we want to solve has the same mathematical form.

### 3.3.2 Formal Requirements of the Model

We want to solve the following linear differential equation for $\bar{u}(z,t_2)$:

$$\left[\frac{d^2}{dz^2} - \frac{1}{H}\frac{d}{dz} - \frac{4l^2}{\epsilon}\right]\bar{u}(z,t_2) = P(z,t_1), \tag{88}$$

where,

$$P(z,t_1) = \left[\frac{d^2}{dz^2} - \frac{1}{H}\frac{d}{dz} - \frac{4l^2}{\epsilon}\Big(1-\zeta\delta t\Big)\right]\bar{u}(z,t_1) - \frac{2l^2\delta t}{\epsilon}G(z,t_1), \tag{89}$$

$$G(z,t_1) = \frac{1}{\rho_0}\frac{\partial F^{(z)}}{\partial z}\bigg|_{t_1}, \tag{90}$$

$$t_2 = t_1 + \delta t, \tag{91}$$

and the boundary conditions are,

$$\frac{\partial \bar{u}}{\partial t}\bigg|_{z=0} = 0, \qquad \frac{\partial}{\partial t}\left(\frac{\partial \bar{u}}{\partial t}\right)\bigg|_{z=0} = 0. \tag{92}$$

We must set up an iterative process that will allow us to follow the development of $\bar{u}(z)$ over a series of steps in time. We must be able to systematically specify and modify:

- $\bar{u}(z,t=0)$, the initial mean zonal flow profile.
- $\zeta$, the magnitude of the Rayleigh friction.
- The numerical method to be used.
- $\delta t$ and the number of iterative steps.

The model should produce plots showing the development of $\bar{u}(z)$ with time.

### 3.3.3 Numerical Methods

As in method A, we make use of the two independent numerical methods, the Method of Variation of Parameters and the Method of Runge-Kutta-Nyström.

**Method of Variation of Parameters** The application of this numerical method here is analogous to its application in method A. $\bar{u}(z,t_2)$ is thus given by the following integral,

$$\bar{u}(z,t_2) = c_1\int_0^z \Big(c_2 e^{-\alpha_1 z'} - c_3 e^{-\alpha_2 z'}\Big)P(z',t_1)dz', \tag{93}$$

where,

$$c_1 = \frac{1}{\epsilon(\alpha_1 - \alpha_2)}, \qquad c_2 = e^{\alpha_1 z}, \qquad c_3 = e^{\alpha_2 z}, \tag{94}$$

and,

$$P(z', t_1) = \left[\frac{d^2}{dz'^2} - \frac{1}{H}\frac{d}{dz'} - \frac{4l^2}{\epsilon}\Big(1 - \zeta\delta t\Big)\right]\bar{u}(z', t_1) - \frac{2l^2\delta t}{\epsilon}G(z', t_1). \tag{95}$$

**Method of Runge-Kutta-Nyström** Again, in analogy with method A, we rearrange equation (88) into a suitable form,

$$\bar{u}''(z, \bar{u}, \bar{u}') = a_1\bar{u}'(z) + a_2 P(z) + a_3\bar{u}(z), \tag{96}$$

where,

$$a_1 = \frac{1}{H}, \qquad a_2 = \frac{4l^2}{\epsilon}, \qquad a_3 = 1, \tag{97}$$

and,

$$P(z) = \left[\frac{d^2}{dz'^2} - \frac{1}{H}\frac{d}{dz} - \frac{4l^2}{\epsilon}\Big(1 - \zeta\delta t\Big)\right]\bar{u}(z) - \frac{2l^2\delta t}{\epsilon}G(z). \tag{98}$$

### 3.3.4 Program Structure

Figure 15 shows the broad structure of the IDL program used to perform the numerical method described above. The program is similar to that used for method A, with only the following new function being called in place of `sub_damping_mu.pro`:

`sub_damping_zeta.pro`
: Populates an array with the value of the damping $\zeta(z)$ at each level.

Note that the equations in the IDL program are written in terms of $B(z,t) = -\bar{u}(z,t)$ not $\bar{u}(z,t)$.

## 3.4 Method C: The Numerical Model

### 3.4.1 Introduction

This method is relatively distinct from the previous two, and simpler from a computational point of view.

### 3.4.2 Formal Requirements of the Model

The forms of $\bar{u}(z)$ at times $t_1$ and $t_2 = t_1 + \delta t$ are related by the expression,

$$\bar{u}(z, t_2) = \bar{u}(z, t_1)(1 - \zeta\delta t) + \frac{1}{\rho_0}\frac{\partial F^{(z)}}{\partial z}\bigg|_{t_1} \cdot \delta t \tag{99}$$

We must set up an iterative process that will allow us to follow the development of $\bar{u}(z)$ over a series of steps in time. We must be able to systematically specify and modify:

- $\bar{u}(z, t = 0)$, the initial mean zonal flow profile.
- $\zeta$, the magnitude of the dissipation.
- $\delta t$ and the number of iterative steps.

The model should produce plots showing the development of $\bar{u}(z)$ with time.

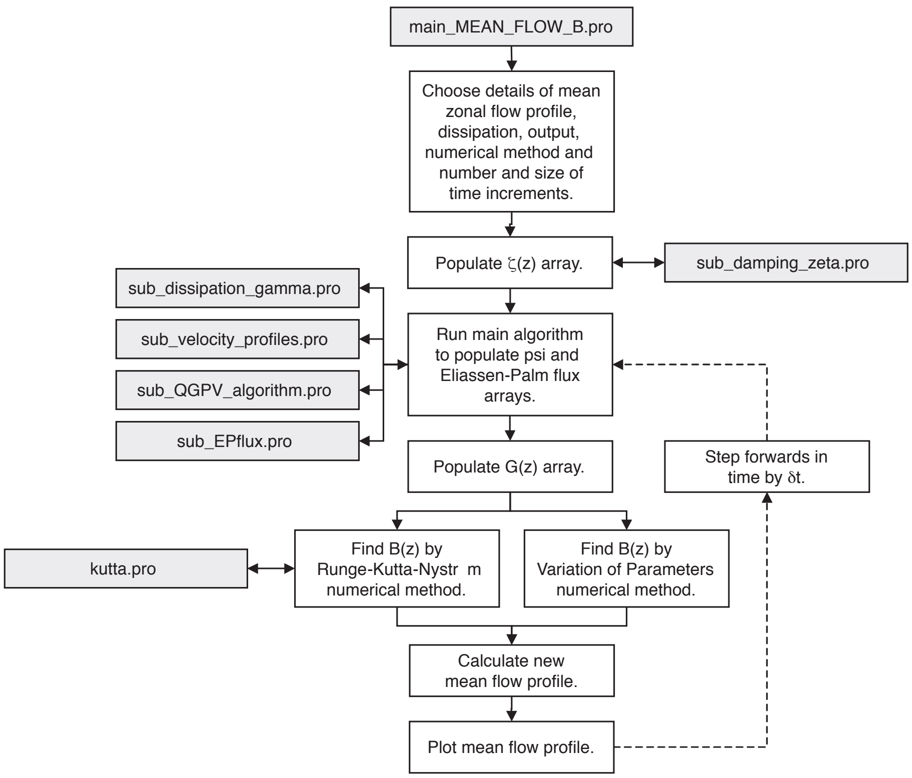


Figure 15: Broad structure of the IDL program for method B.

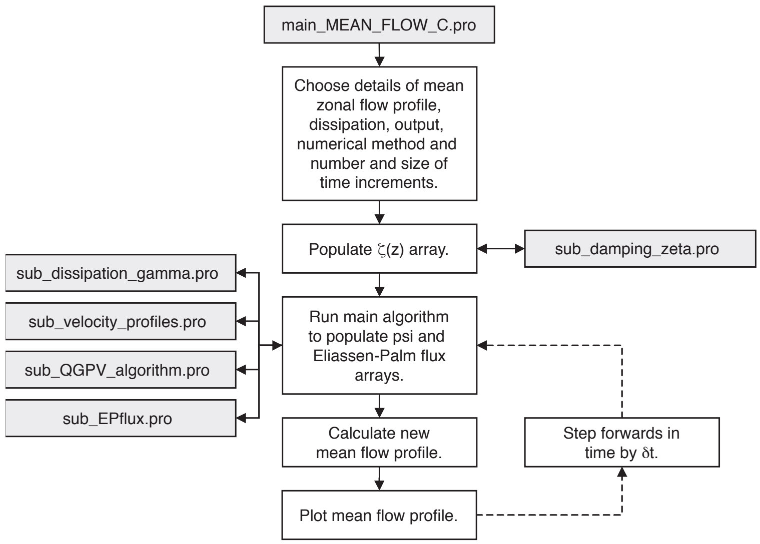


Figure 16: Broad structure of the IDL program for method C.

### 3.4.3 Numerical Methods

No complicated numerical method is required here. At each step of the iterative process we must just evaluate equation (99).

### 3.4.4 Program Structure

The program structure shown in Figure 16 is a simplified version of the programs used for methods A and B.

## 3.5 Results

### 3.5.1 Method A

The accuracy of the computer model is confirmed by the fact that we obtain identical results using both the numerical methods. These results however are highly unphysical. The damping $\mu(z)$ in the equation is swamped by huge accelerations at high altitudes and its value makes little difference to the form of the plots. A typical plot is shown in Figure 17; it has been produced by cycling through 60 steps of $\delta t = 2000$ s with $\delta z = 100$ m and $1/\mu = 20$ days, $\bar{u}(z)$ being plotted every 8000 s.

The model produced the same qualitative results when applied to all profiles and was free of numerical instabilities. The original mathematics on which this model is based does not adequately handle the damping of accelerations of the mean zonal flow.

### 3.5.2 Method B

As with Method A, we obtain consistent but unphysical results. With this model we again observed huge changes in the magnitude of $\bar{u}$ that swamped any details of the original profile. The mathematical formulation of the model is assumed to be inadequate and would require further investigation in any extension of this work.

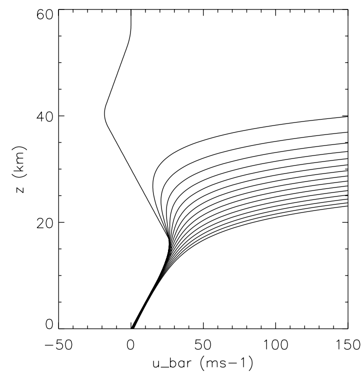


Figure 17: The results produced by method A for mean zonal flow profiles 5 over a period of 1.4 days.

### 3.5.3 Method C

Figure 3.5.3 shows the results of the mean flow interactions on mean zonal flow profiles 1 to 6. The plots for profiles 1 to 3 are produced by cycling through 60 steps of $\delta t = 2000$ s with $\delta z = 100$ m and $1/\zeta = 20$ days. $\bar{u}(z)$ is plotted every 8000 s. The plots for profiles 4 to 6 are produced over 150 steps of $\delta t = 500$ s.

We discuss each profile individually:

1. $\bar{u}(z)$ is a constant giving an EP flux constant with height and an EP flux derivative of zero. Consequently, the mean zonal flow tends to zero sharply.

2. We have two conflicting effects in evidence here; there is a tendency towards large accelerations at high altitudes superimposed on a general decelerating effect.

3. We see a similar pattern to profile 2.

4. This plot shows a general deceleration that is particularly pronounced in the vicinity of the critical wind ($\bar{u} = 0$). Alternatively we may treat critical wind point as descending in altitude.

5. Here, we see only a general deceleration of the mean zonal flow.

6. Like profile 4 we have a general deceleration most strongly felt at the critical wind. We can offer no explanation for discrepancy between profiles 5 and 6 at the critical wind point.

Method C is very crude and we have seen that the effects of wave damping are not modelled perfectly; this is particularly evident in profiles 2 and 3 where there are unphysically large accelerations at high altitudes. However, we are able to note strong decelerations in the vicinity of the critical wind, that may merit further investigation, although their absence in profile 5 remains a puzzle.

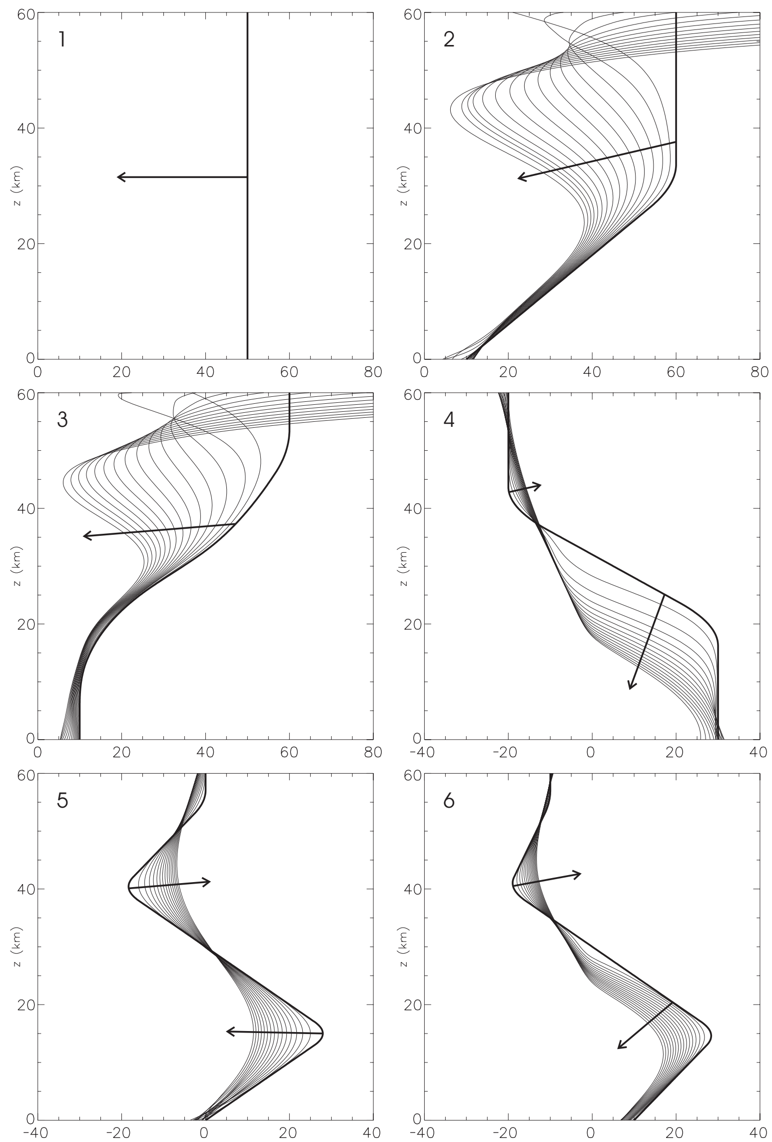


Figure 18: The results produced by method C for mean zonal flow profiles 1 to 6 over a period of 1.4 days for 1 to 3 and 0.9 days for 4 to 6.

### 3.6 Conclusions

Both methods A and B have been shown to provide unphysical results. The *tendency* for large accelerations at high altitudes, as demonstrated by these models, *is* physical (it results from the exponentially decreasing air density); but the damping introduced to limit such accelerations is ineffective. From the results of these methods we may only conclude that their mathematical modelling needs to be reviewed.

Method C produced more promising results. The large accelerations were limited and interesting behavior was noted in the vicinity of the critical wind ($\bar{u} = 0$).

# A Numerical Methods

## A.1 Method of Variation of Parameters

This method is suitable for solving differential equations of the form

$$\frac{d^2y}{dx^2} + f(x)\frac{dy}{dx} + g(x)y = S(x). \tag{100}$$

We suppose that we have two independent solutions of this equation when $S = 0$, denoted by $y_1$ and $y_2$. The complementary function, the most general solution with $S = 0$, is then given by $y = Ay_1 + By_2$ where $A$ and $B$ are constants. In the method of variation of parameters we make $A$ and $B$ variable functions of $x$, choosing them so that (100) is satisfied for general $S(x)$. We take,

$$y = P(x)y_1 + Q(x)y_2; \tag{101}$$

then

$$y' = P'y_1 + Q'y_2 + Py_1' + Qy_2'. \tag{102}$$

We are entitled to assume one relation between the introduced functions $P$ and $Q$; we take

$$P'y_1 + Q'y_2 = 0. \tag{103}$$

then

$$y'' = Py_1'' + Qy_2'' + P'y_1' + Q'y_2', \tag{104}$$

which, substituted into (100), gives

$$\{Py_1'' + f(x)Py_1' + g(x)Py_1\} + \{Qy_2'' + f(x)Qy_2' + g(x)Qy_2\} + P'y_1' + Q'y_2' = S. \tag{105}$$

The terms in the brackets must cancel as $y_1$ and $y_2$ satisfy the equation with $S = 0$; hence we have two equations to determine $P'$ and $Q'$ in terms of $S$. Then

$$P' = \frac{Sy_2}{y_1'y_2 - y_2'y_1}, \qquad Q' = -\frac{Sy_1}{y_1'y_2 - y_2'y_1}, \tag{106}$$

which are definite if $y_1$ and $y_2$ are linearly independent. Hence

$$y = Cy_1 + Dy_2 + \int_a^x \frac{y_1(x)y_2(\xi) - y_2(x)y_1(\xi)}{y_1'(\xi)y_2(\xi) - y_2'(\xi)y_1(\xi)} S(\xi)d\xi \tag{107}$$

where $C$ and $D$ are constants that may be obtained from the initial conditions and $a$ may be taken arbitrarily.

It can be shown that (107) satisfies (100), noting that by differentiation we have

$$y' = Cy_1' + Dy_2' + \int_a^x \frac{y_1'(x)y_2(\xi) - y_2'(x)y_1(\xi)}{y_1'(\xi)y_2(\xi) - y_2'(\xi)y_1(\xi)} S(\xi)d\xi \tag{108}$$

(differentiation of the limit being equal to zero as the integrand vanishes there) and

$$y'' = Cy_1'' + Dy_2'' + \int_a^x \frac{y_1''(x)y_2(\xi) - y_2''(x)y_1(\xi)}{y_1'(\xi)y_2(\xi) - y_2'(\xi)y_1(\xi)} S(\xi)d\xi. \tag{109}$$

For further details see [8].

## A.2 Method of Runge-Kutta-Nyström

This method is suitable for solving differential equation of the following form:

$$\frac{d^2y}{dx^2} = f\Big(x, y, \frac{dy}{dx}\Big),$$

which is written more concisely as,

$$y'' = f\big(x, y, y'\big). \tag{110}$$

We divide our $x$-domain into a number of discrete levels $x_0, x_1, x_2, \ldots, x_M$ separated by a distance $h$. We therefore have,

$$x_{n+1} = x_0 + (n+1)h \qquad n = 0, 1, \ldots, N-1$$

At each $x$ level we evaluate the the following quantities:

$$k_1 = \frac{1}{2}hf\Big(x_n, y_n, y_n'\Big),$$

$$k_2 = \frac{1}{2}hf\Big(x_n + \frac{1}{2}h, y_n + K, y_n' + k_1\Big),$$

where,

$$K = \frac{1}{2}h\Big(y_n' + \frac{1}{2}k_1\Big),$$

and,

$$k_3 = \frac{1}{2}hf\Big(x_n + \frac{1}{2}h, y_n + K, y_n' + k_2\Big),$$

$$k_4 = \frac{1}{2}hf\Big(x_n + h, y_n + L, y_n' + 2k_3\Big),$$

where,

$$L = h\Big(y_n' + k_3\Big).$$

The values of $x$, $y$ and $y'$ at the next level are then given by,

$$x_{n+1} = x_n + h, \tag{111}$$

$$y_{n+1} = y_n + h\Big(y_n'\big(k_1 + k_2 + k_3\big)\Big), \tag{112}$$

$$y_{n+1}' = y_n' + \frac{1}{3}\Big(k_1 + 2k_2 + 2k_3 + k_4\Big). \tag{113}$$

In this way be may find $y(x)$. For further details see [9] or [13].

### A.3 Lindzen's Method For Two Point Boundary Values

This method is suitable for solving differential equations of the following form:

$$\frac{d^2y}{dx^2} + F(x)\frac{dy}{dx} = 0 \tag{114}$$

where the boundary conditions are specified at two end points, between which we want to find $y(x)$. This method is a version of Gaussian elimination.

The $x$-domain is divided into a number of discrete levels $x_0, x_1, x_2, \ldots, x_M$ where $x_0 = 0$ and the remaining levels are equally spaced with a separation $\delta x$. We make the following approximation at $x = x_m$ using finite elements

$$\frac{d^2y(x_m)}{dx^2} \approx \frac{y(x_{m+1}) - 2y(x_m) + y(x_{m-1})}{(\delta x)^2}. \tag{115}$$

Substituting (115) into (114) gives

$$y(x_{m+1)} + \{F(x_m)(\delta z)^2 - 2\} \cdot y(x_m) + y(x_{m-1}) = 0 \tag{116}$$

We then let

$$y(x_m) = \alpha_m \cdot y(x_{m+1}) + \beta_m \tag{117}$$

where $\alpha_m$ and $\beta_m$ are new variables. Similarly,

$$y(x_{m-1}) = \alpha_{m-1} \cdot y(x_m) + \beta_{m-1}. \tag{118}$$

Substituting (118) into (116) gives

$$y(x_m) = -\frac{1}{F(x_m)(\delta x)^2 - 2 + \alpha_{m-1}} \cdot y(x_{m+1}) - \frac{\beta_{m-1}}{F(x_m)(\delta z)^2 - 2 + \alpha_{m-1}} \tag{119}$$

which comparing with (117) gives,

$$\alpha_m = \frac{-1}{F(x_m)(\delta x)^2 - 2 + \alpha_{m-1}}, \tag{120}$$

$$\beta_m = \frac{-\beta_{m-1}}{F(x_m)(\delta x)^2 - 2 + \alpha_{m-1}}. \tag{121}$$

Hence, when $\alpha_0$ and $\beta_0$ are known, we can solve for all $\alpha_m$'s and $\beta_m$'s and find $y(x)$ as required.

$\alpha_0$ and $\beta_0$ can be found from a condition on the differential of $y$ at $x_0$ using finite elements to give an expression of the form of (118). Finally, to solve for $y$ we a value for $y_M$ giving the upper boundary condition.

This method is discussed in [12].

# B The Eliassen-Palm (EP) Flux

In this project we make extensive use of the *Eliassen-Palm (EP) Flux*, the $z$-component of which represents the mean upward transfer of horizontal momentum by the waves. It is a useful measure of the mean effect of the waves on the background flow.

Under the approximations used in this project, the z-component of the Eliassen-Palm flux is given by

$$F^{(z)} = \rho_0 \epsilon(z) \overline{\frac{\partial \psi'}{\partial u} \cdot \frac{\partial \psi'}{\partial z}}, \tag{122}$$

where $\rho_0 = \rho_s e^{-z/H}$ and $\epsilon(z) = f_0^2 / N^2(z)$.

From equation (28) we have

$$\psi' = Re \left\{ e^{z/2H} \Psi(z) e^{ik(x-ct)} \right\} \sin ly$$

which we substitute into (122) recalling the standard result that if $a' = \mathrm{Re}\, \hat{\mathrm{a}} \mathrm{e}^{\mathrm{ikx}}$ and $b' = Re\, \hat{b} e^{ikx}$ then $\overline{a'b'} = \frac{1}{2} Re\, (\hat{a}\hat{b}^*)$,

$$F^{(z)} = \frac{1}{2} \rho_s k \cdot \epsilon(z) \sin^2 ly \cdot \mathrm{Re} \left\{ i\varPsi \frac{d\varPsi^*}{dz} \right\}. \tag{123}$$

It is often preferable to work with the integral of this quantity in the y direction across the channel,

$$\langle F^{(z)} \rangle = \int_0^L F^{(z)} \mathrm{dy}, \tag{124}$$

where L is the width of the channel. To evaluate this integral we need to express $\Psi$ in $re^{i\theta}$ form,

$$\Psi(z) = A(z) e^{i\alpha(z)}.$$

Equation (124) therefore becomes,

$$\langle F^{(z)} \rangle = \frac{1}{4} \rho_s k L \cdot \epsilon z A^2(z) \frac{d\alpha}{dz}. \tag{125}$$

Note that when $\Psi(z)$ exhibits pure exponential behavior, $\alpha = 0$ and $\langle F^{(z)} \rangle = 0$.

For more details see [5].

# C Mean Zonal Flow Profiles

The six standard mean zonal flow profiles used in this project are shown in Figure C.

# D Model Code

## D.1 main_MEAN_FLOW_A.pro

```
; MEAN FLOW Method A
; Changes to the Mean Flow

; ***************************************
; MAIN PARAMTERS

;   1. constant
;   2. linear to zp then constant
;   3. 10 then gradual increase to 60
;   4. 30 then linear to -20
;   5. jagged left then back right (0 at top, bottom)
;   6. jagged left then back right (non-zero ends)
```

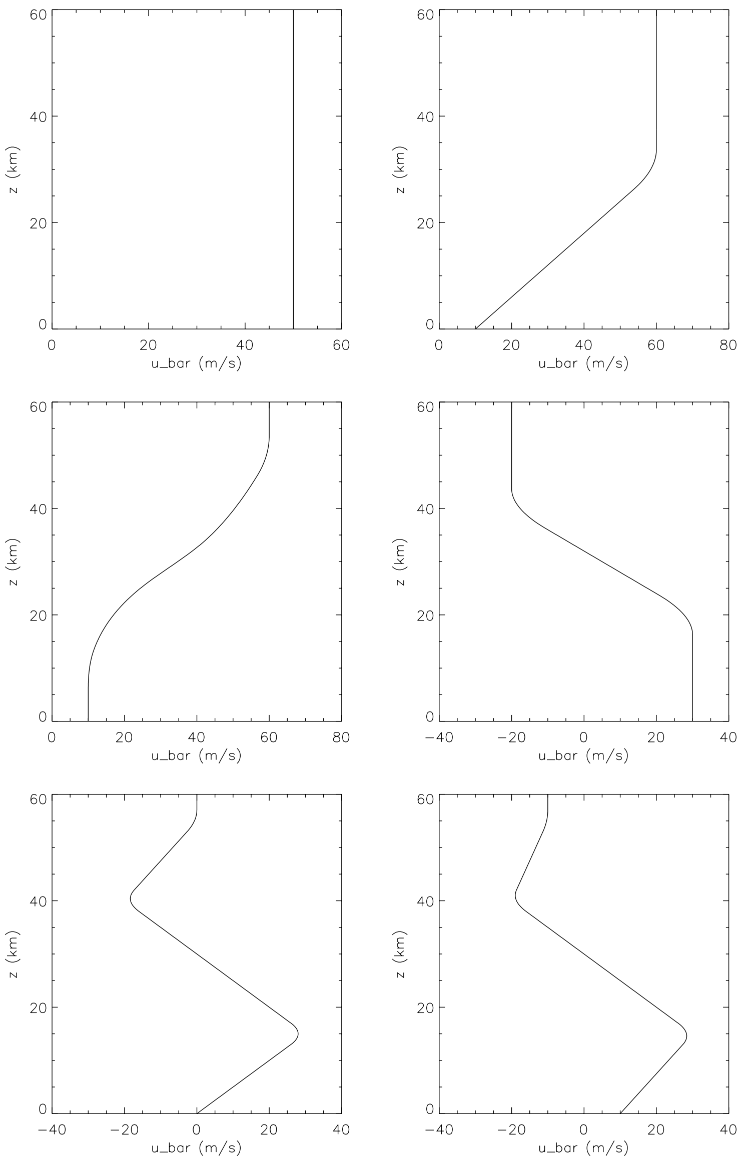


Figure 19: A key to the standard mean zonal flow profiles 1 to 6, numbered from the top left, across and then down.

```
profile_choice = 5
delta_z = 250.0
delta_t = 2000.0
iterations = 60.0
plot_interval = 4.0

;   1. postscript
;   2. window

display = 1

;   1. no profile smoothing
;   2. profile smoothing

smoothing = 2

;   1. no dissipation
;   2. simple linear dissipation
;   3. variable magnitude simple linear dissipation

dissipation = 2

;   1. no damping
;   2. damping (mu) variable
;   3. damping (mu) constant

damping = 1

;   1. variation of parameters
;   2. Runge-Kutta-Nystrom

integration_method = 2

; ****************************************
; VARIABLES and PARAMETERS

z_max = 60000.0
delta_z = delta_z

no = long(z_max / delta_z)

zm = dblarr(no+1)
zkm = dblarr(no+1)

mod_psi = dblarr(no+1)
arg_psi = dblarr(no+1)
q_bar_y = dblarr(no+1)

EPflux_deriv = dblarr(no+1)
G = dblarr(no+1)
U = dblarr(no+1)
A_z = dblarr(no+1)
ddt_u_bar = dblarr(no+1)
ddz_u_bar = dblarr(no+1)
```

```
d2dz2_u_bar = dblarr(no+1)

; ****************************************
; CONSTANTS

lattitude = !pi/4.0
earth_radius = 6.371E6
N_2 = 5.0E-4
l = !pi/(10.0E6)
s = 1.0
k = s / (earth_radius * cos(lattitude))
omega = 7.292E-5
f_0 = 2.0 * omega * sin(lattitude)
H = 7000.0 ; m
beta = 2.0 * omega * cos(lattitude) / earth_radius
c = 0.0 ; stationary waves
epsilon = (f_0^2) / N_2
A = 5000.0 ; c.f. temperature amplitude
i = sqrt(complex(-1.0))

; ****************************************
; CODE

; populate heights

FOR j = 0.0, no DO zm(j) = j * delta_z
zkm = zm / 1000.0

gam = GAM_DISSIPATION(no, zm, zkm, dissipation)  ; DISSIPATION
mu = MU_DAMPING(no, zm, zkm, damping, delta_t)   ; DAMPING

CASE profile_choice OF
   1:
   2: zp = 30.0
   3:
   4: zp = -20.0
   5: zp = 30.0
   6: zp = 30.0
ENDCASE

; ** find initial velocity profile **

u_bar = velocity_profile(profile_choice, no, delta_z, zm, zkm, $
   no_loops, loop_index, dissipation, smoothing, zp)

; plot it

IF display EQ 1 THEN BEGIN
   set_plot, 'ps'
   device, /portrait
   device, filename = $
      'C:\My Documents\Academic Work\Masters Project\Graphs\mf.ps'
   device, xsize = 6, ysize = 6
   device, /times
```

```
   device, font_size = 7
END
IF display EQ 2 THEN set_plot, 'win'

plot, u_bar, zkm, ytitle='z (km)', $
   yrange=[0.0, 60.0], xrange=[-50.0, 150.0], xtitle='u_bar (ms-1)'

; mean flow interaction loop

FOR it = 1.0, iterations DO BEGIN

   u_bar = ts_smooth(u_bar, no/20)
   ddz_u_bar = DERIV(zm, u_bar)
   d2dz2_u_bar = DERIV(zm, ddz_u_bar)
   ddz_u_bar = ts_smooth(ddz_u_bar, no/20)
   ddz_u_bar = ts_smooth(ddz_u_bar, no/20)
   d2dz2_u_bar = ts_smooth(d2dz2_u_bar, no/20)
   d2dz2_u_bar = ts_smooth(d2dz2_u_bar, no/20)

   ; ** find EP flux **

   q_bar_y = beta +  (epsilon/H) * ddz_u_bar - epsilon * d2dz2_u_bar
   psi = MAIN_ALGORITHM (A, H, beta, epsilon, gam, f_0, N_2, k, $
      l, c, zm, delta_z, no, u_bar, dissipation, q_bar_y)

   mod_psi = sqrt(float(psi)^2 + imaginary(psi)^2)
   arg_psi = atan(imaginary(psi) / float(psi))

   EPflux = ELIASSEN_PALM_FLUX(arg_psi, mod_psi, zm, no, k, epsilon)

   ; ** differentiate EP flux and find G **

   EPflux_deriv = DERIV(zm, EPflux)
   G = EPflux_deriv * exp(zm/H)
   G = ts_smooth(G, no/20)
   G = ts_smooth(G, no/20)

   ; ** solve eqn for A(z) *****

   CASE integration_method OF

      1:  BEGIN ; varaition of parameters
             a_1 = (1.0/(2.0*H)) + SQRT((1.0/(4.0*H^2.0)) + $
                 ((4.0*l^2.0)/epsilon))
             a_2 = (1.0/(2.0*H)) - SQRT((1.0/(4.0*H^2.0)) + $
                 ((4.0*l^2.0)/epsilon))
             c_1 = (2.0*l^2.0) / (epsilon*(a_1-a_2))

             FOR limit = 1, no DO BEGIN
                ; must first find array to be integrated along z
                integrand = dblarr(limit+1)
                integral_range = dblarr(limit+1)
                FOR j = 0, limit DO BEGIN ; population of the integrand
                   c_2 = exp(a_1 * zm(limit))
```

```
               c_3 = exp(a_2 * zm(limit))
               integrand(j) = c_1 * ( c_2 * exp(-a_1*zm(j)) - $
                  c_3 * exp(-a_2*zm(j)) ) * $
                  (G(j) + mu(j) * (epsilon/(2.0*l^2)) * $
                  (d2dz2_u_bar(j)-(ddz_u_bar(j)/H)))
               integral_range(j) = zm(j)
            ENDFOR
            A_z(limit) = INT_TABULATED(integral_range, integrand, /DOUBLE)
         ENDFOR
      END

   2:  BEGIN ; Runge-Kutta-Nystrom
         A_dz = fltarr(1)
         k_1 = fltarr(1)
         k_2 = fltarr(1)
         k_3 = fltarr(1)
         k_4 = fltarr(1)
         bigK = fltarr(1)
         bigL = fltarr(1)

         U = G + mu * (epsilon/(2.0*l^2)) * (d2dz2_u_bar-(ddz_u_bar/H))

         FOR j = 0, (no-1) DO BEGIN

            k_1 = (delta_z/2.0) * $
               A_dz2(H, l, epsilon, A_dz, A_z(j), U(j))
            bigK = (delta_z/2.0) * (A_dz + 0.5 * k_1)
            k_2 = (delta_z/2.0) * $
            A_dz2(H, l, epsilon, A_dz + k_1, $
            A_z(j) + bigK, (U(j)+U(j+1))/2.0)
            k_3 = (delta_z/2.0) * $
            A_dz2(H, l, epsilon, A_dz + k_2, $
            A_z(j) + bigK, (U(j)+U(j+1))/2.0)
            bigL = delta_z * (A_dz + k_3)
            k_4 = (delta_z/2.0) * $
            A_dz2(H, l, epsilon, A_dz + 2 * k_3, $
            A_z(j) + bigL, U(j+1))

            A_z(j+1) = A_z(j) + delta_z * $
               ( A_dz + (1.0/3.0) * (k_1 + k_2 + k_3) )
            A_dz = A_dz + (1.0/3.0) * (k_1 + 2 * k_2 + 2 * k_3 + k_4)

         ENDFOR
   END

ENDCASE

u_bar = u_bar - A_z * delta_t

IF LONG(it/plot_interval) EQ (it/plot_interval) THEN $
   oplot, u_bar, zkm ; ** plot results **

ENDFOR
```

```
IF display EQ 1 THEN device, /close

END
```

## D.2 main_MEAN_FLOW_B.pro

```
; MEAN FLOW Method B
; Changes to the Mean Flow with Rayleigh Friction

; ****************************************
; MAIN PARAMTERS

;   1. constant
;   2. linear to zp then constant
;   3. 10 then gradual increase to 60
;   4. 30 then linear to -20
;   5. jagged left then back right (0 at top, bottom)
;   6. jagged left then back right (non-zero ends)

profile_choice = 6
delta_z = 250.0
delta_t = 500.0
iterations = 20.0

;   1. postscript
;   2. window

display = 2

;   1. no profile smoothing
;   2. profile smoothing

smoothing = 2

;   1. no dissipation
;   2. simple linear dissipation
;   3. variable magnitude simple linear dissipation

dissipation = 2

;   1. no damping
;   2. damping (zeta) constant

damping = 2

;   1. Runge-Kutta-Nystrom
;   2. variation of parameters

integration_method = 1

; ****************************************
; VARIABLES and PARAMETERS

z_max = 60000.0
```

```
delta_z = delta_z

no = long(z_max / delta_z)

zm = dblarr(no+1)
zkm = dblarr(no+1)

mod_psi = dblarr(no+1)
arg_psi = dblarr(no+1)
q_bar_y = dblarr(no+1)

EPflux_deriv = dblarr(no+1)
G = dblarr(no+1)
P = dblarr(no+1)
B_z = dblarr(no+1)
zeta = dblarr(no+1)
ddt_u_bar = dblarr(no+1)
ddz_u_bar = dblarr(no+1)
d2dz2_u_bar = dblarr(no+1)

; ***************************************
; CONSTANTS

lattitude = !pi/4.0
earth_radius = 6.371E6
N_2 = 5.0E-4
l = !pi/(10.0E6)
s = 1.0
k = s / (earth_radius * cos(lattitude))
omega = 7.292E-5
f_0 = 2.0 * omega * sin(lattitude)
H = 7000.0 ; m
beta = 2.0 * omega * cos(lattitude) / earth_radius
c = 0.0 ; stationary waves
epsilon= (f_0^2) / N_2
A = 5000.0 ; c.f. temperature amplitude
i = sqrt(complex(-1.0))

; ***************************************
; CODE

; populate heights

FOR j = 0.0, no DO zm(j) = j * delta_z
zkm = zm / 1000.0

IF dissipation EQ 1.0 THEN gam = 0.0 ELSE $
   gam = GAM_DISSIPATION(no, zm, zkm, dissipation) ; DISSIPATION
zeta = ZETA_DAMPING(no, zm, zkm, damping, delta_t) ; DAMPING

CASE profile_choice OF
   1:
   2: zp = 30.0
   3:
```

```
   4: zp = -20.0
   5: zp = 30.0
   6: zp = 30.0
ENDCASE

; ** find initial velocity profile **

u_bar = velocity_profile(profile_choice, no, delta_z, zm, zkm, $
   no_loops, loop_index, dissipation, smoothing, zp)

; plot it

IF display EQ 1 THEN BEGIN
   set_plot, 'ps'
   device, /portrait
   device, filename = $
      'C:\My Documents\Academic Work\Masters Project\mean flow.ps'
   device, xsize = 9, ysize = 9;standard is 10 by 10
   device, /times
   device, font_size = 7
END
IF display EQ 2 THEN set_plot, 'win'

plot, u_bar, zkm, ytitle='z (km)', $
   yrange=[0.0, 60.0], xrange=[-50.0, 50.0], xtitle='u_bar (ms-1)'

; mean flow interaction loop

FOR it = 1.0, iterations DO BEGIN

   u_bar = ts_smooth(u_bar, no/20)
   ddz_u_bar = DERIV(zm, u_bar)
   ddz_u_bar = ts_smooth(ddz_u_bar, no/20)
   ddz_u_bar = ts_smooth(ddz_u_bar, no/20)
   d2dz2_u_bar = DERIV(zm, ddz_u_bar)
   d2dz2_u_bar = ts_smooth(d2dz2_u_bar, no/20)
   d2dz2_u_bar = ts_smooth(d2dz2_u_bar, no/20)

   ; ** find EP flux **

   q_bar_y = beta +  (epsilon/H) * ddz_u_bar - epsilon * d2dz2_u_bar
   psi = MAIN_ALGORITHM (A, H, beta, epsilon, gam, f_0, N_2, $
      k, l, c, zm, delta_z, no, u_bar, dissipation, q_bar_y)

   mod_psi = sqrt(float(psi)^2 + imaginary(psi)^2)
   arg_psi = atan(imaginary(psi) / float(psi))

   EPflux = ELIASSEN_PALM_FLUX(arg_psi, mod_psi, zm, no, k, epsilon)

   ; ** differentiate EP flux and find G **

   EPflux_deriv = DERIV(zm, EPflux)
   G = EPflux_deriv * exp(zm/H)
   G = ts_smooth(G, no/20)
```

```
G = ts_smooth(G, no/20)
; P = (d2dz2_u_bar - (1.0/H) * ddz_u_bar - $
;  ((4.0*l^2.0)/epsilon) * (1.0 - zeta * delta_t) ) $
;  * u_bar - ((2.0*l^2.0)/epsilon) * delta_t * G

P=((2.0*l^2.0)/epsilon) * delta_t * G

; ** solve eqn for B(z) *****
; 1. variation of parameters
; 2. Runge-Kutta-Nystrom

CASE integration_method OF

   1: BEGIN
      B_dz = fltarr(1)
      k_1 = fltarr(1)
      k_2 = fltarr(1)
      k_3 = fltarr(1)
      k_4 = fltarr(1)
      bigK = fltarr(1)
      bigL = fltarr(1)

      FOR j = 0, (no-1) DO BEGIN

         k_1 = (delta_z/2.0) * $
         B_dz2(H, l, epsilon, B_dz, B_z(j), P(j))
            bigK = (delta_z/2.0) * (B_dz + 0.5 * k_1)
         k_2 = (delta_z/2.0) * $
         B_dz2(H, l, epsilon, B_dz + k_1, B_z(j) + bigK, $
            (P(j)+P(j+1))/2.0)
         k_3 = (delta_z/2.0) * $
         B_dz2(H, l, epsilon, B_dz + k_2, B_z(j) + bigK, $
            (P(j)+P(j+1))/2.0)
         bigL = delta_z * (B_dz + k_3)
         k_4 = (delta_z/2.0) * $
         B_dz2(H, l, epsilon, B_dz + 2 * k_3, $
         B_z(j) + bigL, P(j+1))

         B_z(j+1) = B_z(j) + delta_z * $
            ( B_dz + (1.0/3.0) * (k_1 + k_2 + k_3) )
         B_dz = B_dz + (1.0/3.0) * (k_1 + 2 * k_2 + 2 * k_3 + k_4)

      ENDFOR
      END

   2: BEGIN
         FOR limit = 1, no DO BEGIN
            ; must first find array to be integrated along z
            integrand = dblarr(limit+1)
            integral_range = dblarr(limit+1)
            FOR j = 0, limit DO BEGIN ; population of the integrand
               a_1 = (1.0/(2.0*H)) + $
                  SQRT((1.0/(4.0*H^2.0))+((4.0*l^2.0)/(epsilon)))
               a_2 = (1.0/(2.0*H)) - $
```

```
                    SQRT((1.0/(4.0*H^2.0))+((4.0*l^2.0)/(epsilon)))
                 c_1 = 1.0 / (a_1-a_2)
                 c_2 = exp(a_1 * zm(limit))
                 c_3 = exp(a_2 * zm(limit))
                 integrand(j) = c_1 * ( c_2 * exp(-a_1*zm(j)) $
                    - c_3 * exp(-a_2*zm(j)) ) * P(j)
                 integral_range(j) = zm(j)
              ENDFOR
              B_z(limit) = INT_TABULATED(integral_range, integrand, /DOUBLE)
           ENDFOR
        END

   ENDCASE

   ; ** find du/dt **

   u_bar = - B_z ;

   oplot, u_bar, zkm ; ** plot results **

ENDFOR

IF display EQ 1 THEN device, /close

END
```

## D.3 main_MEAN_FLOW_C.pro

```
; MEAN FLOW Method C
; Simple Method

; ***************************************
; MAIN PARAMTERS

; 1. constant
; 2. linear to zp then constant
; 3. 10 then gradual increase to 60
; 4. 30 then linear to -20
; 5. jagged left then back right (0 at top, bottom)
; 6. jagged left then back right (non-zero ends)

profile_choice = 6
delta_z = 100.0
delta_t = 1000.0
iterations = 10.0
plot_interval = 1.0

; 1. postscript
; 2. window

display = 2

; 1. no profile smoothing
; 2. profile smoothing
```

```
smoothing = 2

; 1. no dissipation
; 2. simple linear dissipation
; 3. variable magnitude simple linear dissipation

dissipation = 2

; 1. no damping
; 2. damping (zeta) constant

damping = 2

; ****************************************
; VARIABLES and PARAMETERS

z_max = 60000.0
delta_z = delta_z

no = long(z_max / delta_z)

zm = dblarr(no+1)
zkm = dblarr(no+1)

mod_psi = dblarr(no+1)
arg_psi = dblarr(no+1)
q_bar_y = dblarr(no+1)

EPflux_deriv = dblarr(no+1)
zeta = dblarr(no+1)
ddt_u_bar = dblarr(no+1)
ddz_u_bar = dblarr(no+1)
d2dz2_u_bar = dblarr(no+1)

; ****************************************
; CONSTANTS

lattitude = !pi/4.0
earth_radius = 6.371E6
N_2 = 5.0E-4
l = !pi/(10.0E6)
s = 1.0
k = s / (earth_radius * cos(lattitude))
omega = 7.292E-5
f_0 = 2.0 * omega * sin(lattitude)
H = 7000.0 ; m
beta = 2.0 * omega * cos(lattitude) / earth_radius
c = 0.0 ; stationary waves
epsilon = (f_0^2) / N_2
A = 5000.0 ; c.f. temperature amplitude
i = sqrt(complex(-1.0))

; ****************************************
```

```
; CODE

; populate heights

FOR j = 0.0, no DO zm(j) = j * delta_z
zkm = zm / 1000.0

IF dissipation EQ 1.0 THEN gam = 0.0 ELSE $
    gam = GAM_DISSIPATION(no, zm, zkm, dissipation) ; DISSIPATION
zeta = ZETA_DAMPING(no, zm, zkm, damping, delta_t) ; DAMPING

CASE profile_choice OF
   1:
   2: zp = 30.0
   3:
   4: zp = -20.0
   5: zp = 30.0
   6: zp = 30.0
ENDCASE

; ** find initial velocity profile **

u_bar = velocity_profile(profile_choice, no, delta_z, zm, zkm, $
   no_loops, loop_index, dissipation, smoothing, zp)

; plot it

IF display EQ 1 THEN BEGIN
   set_plot, 'ps'
   device, /portrait
   device, filename = $
      'C:\My Documents\Academic Work\Masters Project\Graphs\mf.ps'
   device, xsize = 15, ysize = 15;standard is 10 by 10
   device, /times
   device, font_size = 7
END
IF display EQ 2 THEN set_plot, 'win'

plot, u_bar, zkm, ytitle='z (km)', $
   yrange=[0.0, 60.0], xrange=[-40.0, 40.0], xtitle='u_bar (ms-1)'

; mean flow interaction loop

FOR it = 1.0, iterations DO BEGIN

   u_bar = ts_smooth(u_bar, no/20)
   ddz_u_bar = DERIV(zm, u_bar)
   ddz_u_bar = ts_smooth(ddz_u_bar, no/20)
   ddz_u_bar = ts_smooth(ddz_u_bar, no/20)
   d2dz2_u_bar = DERIV(zm, ddz_u_bar)
   d2dz2_u_bar = ts_smooth(d2dz2_u_bar, no/20)
   d2dz2_u_bar = ts_smooth(d2dz2_u_bar, no/20)

   ; ** find EP flux **
```

```
   q_bar_y = (beta +  (epsilon/H) * ddz_u_bar - epsilon * $
      d2dz2_u_bar)
   psi = MAIN_ALGORITHM (A, H, beta, epsilon, gam, f_0, N_2, k, l, c, zm, $
   delta_z, no, u_bar, dissipation, q_bar_y)

   mod_psi = sqrt(float(psi)^2 + imaginary(psi)^2)
   arg_psi = atan(imaginary(psi) / float(psi))

   EPflux = ELIASSEN_PALM_FLUX(arg_psi, mod_psi, zm, no, k, epsilon)

   ; ** differentiate EP flux and find G **

   EPflux_deriv = DERIV(zm, EPflux)

   u_bar = u_bar * (1 - zeta * delta_t) + delta_t * exp(zm/H) * EPflux_deriv

   IF LONG(it/plot_interval) EQ (it/plot_interval) THEN $
   oplot, u_bar, zkm ; ** plot results **

ENDFOR

IF display EQ 1 THEN device, /close

END
```

## D.4 main_QGPV_SOLN.pro

```
; solves quasi-geostrophic potential vorticity equation for
; specified mean zonal wind velocity profile

; ***************************************
; MAIN PARAMTERS

; 1. constant
; 2. linear to zp then constant
; 3. 10 then gradual increase to 60
; 4. 30 then linear to -20
; 5. jagged left then back right (0 at top, bottom)
; 6. jagged left then back right (non-zero ends)

; 1. mod Psi(z)
; 2. arg Psi(z)
; 3. u_bar(z)
; 4. EP Flux(z)
; 5. q_bar_y(z)
; 6. EP_Flux(zp) at 0km
; 7. 3D - adjust manually
; 8. nothing

profile_choice = 5
output_choice = 5
delta_z = 100.0
no_loops = 50.0
```

```
; 1. postscript
; 2. 3D postscript
; 4. window
; 5. nothing

display = 1

; 1. no profile smoothing
; 2. profile smoothing

smoothing = 2

; 1. no dissipation
; 2. simple linear dissipation
; 3. variable magnitude simple linear dissipation

dissipation = 1

; ****************************************
; VARIABLES and PARAMETERS

z_max = 60000.0
delta_z = delta_z

no = long(z_max / delta_z)

zm = dblarr(no+1)
zkm = dblarr(no+1)

zp = 0.0

mod_psi = dblarr(no+1)
arg_psi = dblarr(no+1)
q_bar_y = dblarr(no+1)

loop_value = dblarr(no_loops+1, no+1)
zp_store = dblarr(no_loops+1)

ddz_u_bar = dblarr(no+1)
d2dz2_u_bar = dblarr(no+1)

; ****************************************
; CONSTANTS

lattitude = !pi/4.0
earth_radius = 6.371E6
N_2 = 5.0E-4
l = !pi/(10.0E6)
s = 1.0
k = s / (earth_radius * cos(lattitude))
omega = 7.292E-5
f_0 = 2.0 * omega * sin(lattitude)
H = 7000.0 ; m
```

```
beta = 2.0 * omega * cos(lattitude) / earth_radius
c = 0.0 ; stationary waves
epsilon= (f_0^2) / N_2
A = 5000.0 ; c.f. temperature amplitude
i = sqrt(complex(-1.0))

; ***************************************
; CODE

FOR j = 0.0, no DO zm(j) = j * delta_z ; populates heights
zkm = zm / 1000.0

gam = GAM_DISSIPATION(no, zm, zkm, dissipation) ; dissipation

FOR loop_index = 0.0, no_loops DO BEGIN

   CASE profile_choice OF
      1:
      2: IF no_loops EQ 0.0 THEN zp = 30.0 ELSE $
         zp = 10.0 + 10.0 * (loop_index/(no_loops/4.0))
      3:
      4: IF no_loops EQ 0.0 THEN zp = -20.0 ELSE $
         zp = 30.0 - 10.0 * (loop_index/(no_loops/6.0))
      5: IF no_loops EQ 0.0 THEN zp = 30.0 ELSE $
         zp = 20.0 + 10.0 * (loop_index/(no_loops/2.0))
      6: IF no_loops EQ 0.0 THEN zp = 30.0 ELSE $
         zp = 20.0 + 10.0 * (loop_index/(no_loops/2.0))
   ENDCASE

   print, zp

   u_bar = velocity_profile(profile_choice, no, delta_z, zm, zkm, $
   no_loops, loop_index, dissipation, smoothing, zp)

   ddz_u_bar = DERIV(zm, u_bar)
   d2dz2_u_bar = DERIV(zm, ddz_u_bar)

   q_bar_y = beta +  (epsilon/H) * ddz_u_bar - epsilon * d2dz2_u_bar
   psi = MAIN_ALGORITHM (A, H, beta, epsilon, gam, f_0, N_2, k, l, $
   c, zm, delta_z, no, u_bar, dissipation, q_bar_y)

   mod_psi = sqrt(float(psi)^2 + imaginary(psi)^2)
   arg_psi = atan(imaginary(psi) / float(psi))

   EPflux = ELIASSEN_PALM_FLUX(arg_psi, mod_psi, zm, no, k, epsilon)

   FOR m = 0, no DO loop_value(loop_index, m) = EPflux(m)
   ; FOR m = 0, no DO loop_value(loop_index, m) = arg_psi(m)

   zp_store(loop_index) = zp
   print, zp

   ; call procedure for graphical output
```

```
   output, display, output_choice, profile_choice, mod_psi, $
      arg_psi, u_bar, EPflux, q_bar_y, loop_value, no_loops, $
      zp_store, delta_z, zkm, zm

ENDFOR

END
```

## D.5 sub_EPflux.pro

```
FUNCTION ELIASSEN_PALM_FLUX, arg_psi, mod_psi, zm, no, k, epsilon

; calculates z component of EP flux minus the multiplicative
; factor rho_s

; ****************************************
; INPUT
; arg_psi - argument of capital psi
; mod_psi - modulus of capital psi
; zm - height array in meters
; no - array size
; k - constant
; epsilon - constant

; ****************************************
; VARIABLES

arg_psi_e = dblarr(no+1)
EPflux = dblarr(no+1)
arg_psi_e = arg_psi

; ****************************************
; CODE

FOR m = 1, no DO BEGIN ; smooth jumps out of arg_psi
   IF (arg_psi_e(m) - arg_psi_e(m-1)) GT (!pi/2) THEN $
   FOR n = m, no DO arg_psi_e(n) = arg_psi_e(n) - !pi
   IF (arg_psi_e(m) - arg_psi_e(m-1)) LT (-!pi/2) THEN $
   FOR n = m, no DO arg_psi_e(n) = arg_psi_e(n) + !pi
ENDFOR

EPflux = 0.5 * k * epsilon * (mod_psi)^2 * DERIV(zm,arg_psi_e)
; note factor of rho-s is neglected here

; ****************************************
; RETURN VALUE

RETURN, EPflux

END
```

## D.6 sub_QGPV_algorithm.pro

```
FUNCTION MAIN_ALGORITHM, A, H, beta, epsilon, gam, f_0, N_2, k, l, $
```

```
   c, zm, delta_z, no, u_bar, dissipation, q_bar_y

; numerically solves the QGPV equation using Lindzen's bouncing ball method
;    giving the solution in the psi array (psi is capital psi)

; ***************************************
; INPUT
; misc constants
; delta_z - height increment
; no - array size
; u_bar - wind velocity profile
; dissipation - should linear dissipation be included?
; q_bar_y - q_bar_y array

; ***************************************
; CONSTANTS and VARIABLES

i = sqrt(complex(-1.0))
psi = dcomplexarr(no+1)
alpha_m = dcomplexarr(no+1)
beta_m = dcomplexarr(no+1)
F = dcomplexarr(no+1)

; ***************************************
; CODE

; calculate alpha_m(0), beta_m(0) ; lower boundary

alpha_m(0) = 1.0 / (1 - (delta_z/(2.0*H)))
beta_m(0) = - (A * delta_z) / (1 - (delta_z/(2.0*H)))

FOR m = 1, no DO BEGIN ; find values for alpha_m,beta_m

   ; evaluate F at height zm for different sorts of dispersion - see key
   IF dissipation EQ 1.0 THEN $
      F(m) = ( N_2/(f_0^2) ) * $
      ( q_bar_y(m)/(u_bar(m) - c) - (k^2 + l^2) - (epsilon/(4*H^2)) ) ELSE $
      F(m) = ( N_2/(f_0^2) ) * $
      ( q_bar_y(m)/(u_bar(m) - c - $
      ((i * gam(m))/k)) - (k^2 + l^2) - (epsilon/(4*H^2)) )

  ; populating alpha and beta
  alpha_m(m) = -1.0 / ( F(m) * (delta_z)^2 - 2.0 + alpha_m(m-1) )
  beta_m(m) = -beta_m(m-1) / ( F(m) * (delta_z)^2 - 2.0 + alpha_m(m-1) )

ENDFOR

; upper boundary condition

lambda = complex(SQRT(abs(F(no))))
IF u_bar(no) LT 0.0 THEN psi(no) = 0.00000000001 ELSE psi(no) = (beta_m(no-2) + $
   beta_m(no-1)*(2*i*lambda*delta_z + alpha_m(no-2))) / (1 - alpha_m(no-1) * $
   (2*i*lambda*delta_z + alpha_m(no-2)))
```

```
; work back from upper boundary condition to find values for psi & its mod and arg

FOR m = no-1, 0, -1 DO psi(m) = alpha_m(m) * psi(m+1) + beta_m(m)

; psi = psi * exp(zm / (2*H))

; ****************************************
; RETURN VALUE

RETURN, psi

END
```

## D.7 sub_damping_mu.pro

```
FUNCTION MU_DAMPING, no, zm, zkm, damping, delta_t

; populutes mu array for damping on main_MEAN_FLOW.pro

; ****************************************
; INPUT
; no - array size
; zm - height array in meters
; zkm - height array in kilometers
; damping - choice of damping form
; delta_t

; ****************************************
; VARAIBLES and CONSTANTS

mu = dblarr(no+1)
mu_max = 0.00000001
power = 2.0

; ****************************************
; CODE

IF damping EQ 2 THEN BEGIN ; mu damping 2

FOR m = 0, no DO BEGIN
   IF zkm(m) LE 40.0 THEN mu(m) = mu_max * (zkm(m)/40.0) ^ (1.0/power)
   IF zkm(m) GT 40.0 THEN mu(m) = mu_max
ENDFOR

END

IF damping EQ 3 THEN FOR m = 0, no DO mu(m) = 1.0/(86400.0*20.0)

; ****************************************
; RETURN VALUE

RETURN, DOUBLE(mu)

END
```

## D.8 sub_damping_zeta.pro

```
FUNCTION ZETA_DAMPING, no, zm, zkm, damping, delta_t

; populutes zeta array for damping on main_MEAN_FLOW.pro

; ****************************************
; INPUT
; no - array size
; zm - height array in meters
; zkm - height array in kilometers
; damping - choice of damping form
; delta_t

; ****************************************
; VARAIBLES and CONSTANTS

zeta = dblarr(no+1)

days = 20.0

CASE damping OF

   1: FOR i = 0, no DO zeta(i) = 0.0
   2: FOR i = 0, no DO zeta(i) = 1.0/(86400.0*days)

ENDCASE

RETURN, DOUBLE(zeta)

END
```

## D.9 sub_dissipation_gamma.pro

```
FUNCTION GAM_DISSIPATION, no, zm, zkm, dissipation

; populates the gamma array with values for linear dissipation

; ****************************************
; INPUT
; no - array size
; zm - height array in meters
; zkm - height array in kilometers
; dissipation - choice of damping form

; ****************************************
; VARAIBLES and CONSTANTS

gam = dblarr(no+1)
days = dblarr(no+1)

days_max = 20.0

; ****************************************
```

```
; CODE

IF dissipation EQ 3.0 THEN FOR m = 0, no DO days(m) = $
   SQRT( ((days_max^2-400.0)/60.0)*zkm(m) + 400.0 ) ELSE $
   FOR m = 0, no DO days(m) = 20.0
gam = 1.0/(86400.0 * days)

IF dissipation EQ 1.0 THEN FOR m = 0, no DO gam(m) = 0.0

; ****************************************
; RETURN VALUE

RETURN, gam

END
```

## D.10 sub_graphical_output.pro

```
PRO output, display, output_choice, profile_choice, mod_psi, $
   arg_psi, u_bar,EPflux, q_bar_y, loop_value, no_loops, $
   zp_store, delta_z, zkm, zm

; graphically displays psi and serived functions for ouput
;    as postscript or window

; ****************************************
; INPUT
; display - output method
; output_choice - output field
; profile_choice - wind profile in use
; mod_psi - modulus capital psi
; arg_psi - argument capital psi
; u_bar - wind velocity array
; EPflux - EP flux array
; q_bar_y - q_bar_y array
; loop_value - array storing data for 3D graphs (LOOP)
; no_loops - c.f. dimensions of loop_value (LOOP)
; zp_store - array of variable parameter in wind profile (LOOP)
; delta_z - height increment
; zkm - height array in kilometers
; zm - height array in metres

; ****************************************
; CODE

CASE display OF ; set output method attributes

1: BEGIN ; general postscript plot
      set_plot, 'ps'
      device, /portrait
      device, filename= $
         'C:\My Documents\Academic Work\Masters Project\Graphs\graph.ps'
      device, xsize = 6, ysize = 6 ; standard is 10 by 10
      device, /times
```

```
      device, font_size = 7
   END
2: BEGIN
      set_plot, 'ps'; 3D general ps
      device, xsize = 15, ysize =15
      device, filename= $
         'C:\My Documents\Academic Work\Masters Project\3D plot.ps'
   END
3: set_plot, 'win' ; window plot
4: ; nothing

ENDCASE

CASE output_choice OF

   1: plot, mod_psi, zkm, xtitle = 'Mod Psi(z) (m^2/s)', $
         xrange = [0.0, 8.0E7], ytitle = 'z (km)'
         ;ytitle = 'z (km)'
   2: plot, arg_psi, zkm, xtitle = 'Arg Psi(z)', $
         ytitle = 'z (km)'
   3: plot, u_bar, zkm, xrange = [-40.0,40.0], ytitle='z (km)', $
         yrange=[0.0, 60.0],xtitle='u_bar (m/s)'
   4: plot, EPflux, zkm, ytitle='z (km)', $
         xtitle='Eliassen-Palm Flux',title='delta_z=' + $
         string(delta_z/1000.0) + ' km  ' + string(profile_choice)
   5: plot, q_bar_y, zkm, xrange = [-1.0E-11, 5.0E-11], xtitle = 'q_bar_y', $
         ytitle = 'z (km)', title='delta_z=' + string(delta_z/1000.0) $
         + ' km  ' + string(profile_choice)
   6: plot, extrac(loop_value, 0, 0, no_loops+1,1), zp_store, $
         xtitle = 'EP Flux at 0 km', $
         ytitle = 'Minimum u_bar (m/s)',yrange=[zp_store(0), $
         zp_store(n_elements(zp_store)-1)]
         ; , yrange=[-30.0,30.0], xrange=[0.35,0.65], $
         ; ystyle=1,xstyle=1 ;, $
         ; yrange=[zp_store(0), zp_store(n_elements(zp_store)-1)]
   7: shade_surf, loop_value, AX=45, AZ = 225, charsize = 0.001, zaxis = -1
   8:

ENDCASE

IF (display EQ 1) OR (display EQ 2) THEN device, /close

END
```

## D.11 sub_kutta.pro

```
FUNCTION A_dz2, H, l, epsilon, A, B, C

; required for Runge-Kutta method of solution in main_MEAN_FLOW.pro

; ****************************************
; INPUT
; H - constant
; l - constant
```

```
; epsilon - constant
; A - constant
; B - constant
; C - constant

; ***************************************
; VARIABLES

func = dblarr(1)
a_1 = dblarr(1)
a_2 = dblarr(1)
a_3 = dblarr(1)

; ***************************************
; CODE

a_1 = (1.0/H)
a_2 = ((4.0*l^2)/epsilon)
a_3 = ((2.0*l^2)/epsilon)

func = a_1*A + a_2*B + a_3*C

; ***************************************
; RETURN VALUE

RETURN, func

END

FUNCTION B_dz2, H, l, epsilon, A, B, C

; required for Runge-Kutta method of solution in main_MEAN_FLOW.pro

; ***************************************
; INPUT
; H - constant
; l - constant
; epsilon - constant
; A - constant
; B - constant
; C - constant

; ***************************************
; VARIABLES

func = dblarr(1)
a_1 = dblarr(1)
a_2 = dblarr(1)
a_3 = dblarr(1)

; ***************************************
; CODE

a_1 = (1.0/H)
```

```
a_2 = ((4.0*l^2)/epsilon)
a_3 = -1

func = a_1*A + a_2*B + a_3*C

; ***************************************
; RETURN VALUE

RETURN, func

END
```

## D.12 sub_velocity_profiles.pro

```
FUNCTION velocity_profile, profile_choice, no, delta_z, zm, zkm, $
   no_loops, loop_index, dissipation, smoothing, zp

; populates the u_bar array with a velocity profile
;    (u_bar as a function of z)

; ***************************************
; INPUT
; profile_choice
; no - array size
; delta_z - height increment
; zm - height array in metres
; zkm - height array in kilometers
; no_loops - no loops to move zp through range (LOOP)
; loop_index - index number of current loop (LOOP)
; dissipation - should linear dissipation be included?
; smoothing - should profiles be smoothed?
; zp - variable paramter in velocity profile

; ***************************************
; CODE

u_bar = dcomplexarr(no+1)

FOR j = 0.0, no DO BEGIN

CASE profile_choice OF

1: u_bar(j) = 50.0
2: BEGIN
      IF zkm(j) LT zp THEN u_bar(j) = $
         50.0 * (zkm(j)/zp) + 10.0 ELSE $
      IF zkm(j) GE zp THEN u_bar(j) = 60.0
   END
3: BEGIN
      IF zkm(j) LT 10.0 THEN u_bar(j) = 10.0 ELSE $
      IF (zkm(j) LT 30.0) AND (zkm(j) GE 10.0) THEN u_bar(j) = $
         (1.0/1040.0)*zkm(j)^3.0 + (235.0/26.0) ELSE $
      IF (zkm(j) GE 30.0) AND (zkm(j) LT 50.0) THEN u_bar(j) = $
         ((34625.0/4.0)*(zkm(j)-(6938.0/277.0)))^(1.0/3.0) ELSE $
```

```
      IF zkm(j) GE 50.0 THEN u_bar(j) = 60.0
   END
4: BEGIN
      IF zkm(j) LT 20.0 THEN u_bar(j) = 30.0 ELSE $
      IF (zkm(j) LT 40.0) AND (zkm(j) GE 20.0) $
         THEN u_bar(j) = (zkm(j)-40.0)*((zp-30.0)/20.0) + zp ELSE $
      IF zkm(j) GE 40.0 THEN u_bar(j) = zp
   END
5: BEGIN
      IF zkm(j) LT (zp-15.0) THEN u_bar(j) = $
         (zkm(j)*30.0)/(zp-15.0) ELSE $
      IF (zkm(j) LT (zp+10.0)) AND (zkm(j) GE (zp-15.0)) $
         THEN u_bar(j) = 2.0*(zp-zkm(j)) ELSE $
      IF (zkm(j) LT 55.0) AND (zkm(j) GE (zp+10.0)) $
         THEN u_bar(j) = ((20.0*(zkm(j)-55.0))/(45.0-zp)) ELSE $
      IF zkm(j) GE 55.0 THEN u_bar(j) = 0.0
   END
6: BEGIN
      IF zkm(j) LT (zp-15.0) THEN u_bar(j) = $
         (zkm(j)*20.0)/(zp-15.0) + 10.0 ELSE $
      IF (zkm(j) LT (zp+10.0)) AND (zkm(j) GE (zp-15.0)) $
         THEN u_bar(j) = 2.0*(zp-zkm(j)) ELSE $
      IF (zkm(j) LT 55.0) AND (zkm(j) GE (zp+10.0)) $
         THEN u_bar(j) = ((10.0*(zkm(j)-55.0))/(45.0-zp)) - 10.0 ELSE $
      IF zkm(j) GE 55.0 THEN u_bar(j) = -10.0
   END

ENDCASE

If ((u_bar(j) EQ 0.0) AND (dissipation EQ 1)) THEN u_bar(j) = 0.001
; get around divide by zero

ENDFOR

IF smoothing EQ 2.0 THEN BEGIN ; profile smoothing

   CASE profile_choice OF

      1:
      2: BEGIN
           u_bar = ts_smooth(u_bar,no/8)
           FOR j = ((no+1) - (no/15)), (no+1) DO u_bar(j-1) = 60.0
         END
      3: BEGIN
           u_bar = ts_smooth(u_bar,no/8)
           FOR j = 0, (no/15) DO u_bar(j) = 10.0
           FOR j = ((no+1) - (no/15)), (no+1) DO u_bar(j-1) = 60.0
         END
      4: BEGIN
           IF zp EQ 30.0 then goto, miss ; if we have straight line
           u_bar = ts_smooth(u_bar,no/8)
           FOR j = 0, (no/15) DO u_bar(j) = 30.0
           FOR j = ((no+1) - (no/15)), (no+1) DO u_bar(j-1) = zp
           miss:
```

```
          END
       5: BEGIN
             u_bar = ts_smooth(u_bar,no/15)
             FOR j = ((no+1) - (no/30)), (no+1) DO u_bar(j-1) = -0.0
          END
       6: BEGIN
             u_bar = ts_smooth(u_bar,no/15)
             FOR j = ((no+1) - (no/30)), (no+1) DO u_bar(j-1) = -10.0
          END

    ENDCASE

 ENDIF

 ; ****************************************
 ; RETURN VALUE

 RETURN, u_bar

 END
```

# E Technical Details of Important IDL Routines

For full detail see [15].

## E.1 Calculus

### E.1.1 INT_TABULATED

The `INT_TABULATED` function integrates a tabulated set of data $(x_i, f_i)$ on the closed interval [MIN(x) , MAX(x )], using a five-point Newton-Cotes integration formula. As used in this project, it returns a double precision value.

### E.1.2 DERIV

The `DERIV` function performs numerical differentiation using 3-point, Lagrangian interpolation and returns the single precision derivative.

## E.2 Function Smoothing

### E.2.1 TS_SMOOTH

The `TS_SMOOTH` function is used to compute a central average of an n-element series. Autoregressive forecasting and backcasting are used to extrapolate the series and compute a moving average for each point. The result is an n-element vector of the same data type as the input vector, double precision in our case.

# References

[1] Andrews, D.G., *An Introduction to Atmospheric Physics* (2000) (Cambridge University Press)

[2] Andrews, D.G. *Dynamics of the Middle Atmosphere* (Lecture Notes)

[3] Andrews, D.G. Physics World (November 1991) pp. 41-46 *A Stratospheric Transport System*

[4] VCharney J. G., P.G. Drazin, Journal of Geophysical Research **66** 1 pp. 83-109 (1961) *Propogation of Planetary-Scale Disturbanes from the Lower into the Upper Atmosphere*

[5] Holton, J.R., *An Introduction to Dynamic Meteorology* (1992) (Academic Press)

[6] Holton, J.R., Mass, C., Journal of Atmospheric Sciences **33** 11 pp. 2281-2225 (1976) *Stratospheric Vacillation Cycles*

[7] Houghton, J.T. *The Physics of Atmospheres* (1986) (Cambridge University Press)

[8] Jeffreys, H., Jeffreys, B., *Methods of Mathemetical Physics* (1999) (Cambridge University Press)

[9] Kreyzig, E., *Advanced Engineering Mathematics* (1993) (Wiley)

[10] Lindzen, R.S. Mathematical Problems in the Physical Sciences **14** pp. 293-362 (1971) *Atmospheric Tides*

[11] Lindzen, R.S. and Kuo, H.L., Monthly Weather Review **97** 10 pp. 293-362 (1969) *A Reliable Method for the Numerical Integration of a Large Class of Ordinary and Partial Differential Equations*

[12] Lindzen, R.S., Schoeberl, M.R., Journal of Atmospheric Sciences **39** 5 pp. 1171-1174 (1982) *A Note on the Limits of Rossby Wave Amplitudes*

[13] Press, W.H., Teukolsky, S.A., Vetterlin, W.T., Flannery, B.P., *Numerical Recipies in C* (Cambridge University Press) (1992)

[14] Salby, M.L., *Fundamentals of Atmospheric Physics* (1995) (Academic Press)

[15] Research Systems Inc. *Online Documentation*