Wigglewave equations

C.Boocock

1 Overview

This document will explain the calculations made in *Wigglewave*. *Wigglewave* uses an RK4 finite difference method to solve the linearised governing equations for a torsion Alfvén wave propagating in a plasma with negligible plasma beta and in a force-free axisymmetric magnetic field with no azimuthal component embedded in a high density divergent tube structure. The solutions calculated are the perturbations to the velocity, v and to the magnetic field, b. All variables are calculated over a uniform grid in radius r and height z.

2 Governing Equations

These linearised governing equations to be solved are as follows,

$$\begin{aligned}
\rho\frac{\partial v}{\partial t} =& \quad \frac{1}{r\mu_0}\left(\mathbf{B_0}\cdot\nabla(rb)\right)+\frac{1}{r}\frac{\partial}{\partial r}\left(\rho\nu r\frac{\partial v}{\partial r}\right)+\frac{\partial}{\partial z}\left(\rho\nu\frac{\partial v}{\partial z}\right) \\
\frac{\partial b}{\partial t} =& \quad r\mathbf{B_0}\cdot\nabla\left(\frac{v}{r}\right)
\end{aligned}$$

Where ρ is the density, μ_0 is the vacuum permeability, ν is the kinematic viscosity and $B_0\cdot\nabla = B_r\partial_r + B_z\partial_z$. To solve these equations in *Wigglewave* we expand the terms involving the del operator and move all the terms to the RHS except for the time derivatives. This gives us,

$$\begin{aligned}
\frac{\partial v}{\partial t} =& \ f_v(t,v,b)=\frac{1}{\mu_0\rho}\left(B_r\frac{\partial b}{\partial r}+B_z\frac{\partial b}{\partial z}+\frac{B_r b}{r}\right)+\frac{1}{r\rho}\frac{\partial}{\partial r}\left(\rho\nu r\frac{\partial v}{\partial r}\right)+\frac{1}{\rho}\frac{\partial}{\partial z}\left(\rho\nu\frac{\partial v}{\partial z}\right) \\
\frac{\partial b}{\partial t} =& \ f_b(t,v,b)=B_r\frac{\partial v}{\partial r}+B_z\frac{\partial v}{\partial z}-\frac{vB_r}{r}
\end{aligned}$$

3 Code Structure

3.1 Constants

This is where fundamental constants π and μ_0 are defined as well as all the parameters, these include

Computational parameters:

- B_0 the characteristic magnetic field strength
- ρ_0 the characteristic density
- dr and dz the number of grid cells in radial and vertical directions
- r_{min} and r_{max} the range of the radial direction
- z_{min} and z_{max} the range of the vertical direction
- t_{end} and $t_{interval}$ the simulation run time and the interval between outputs
- t_0 the ramp-up time
- *save_dir* the location to save outputs to

Simulation parameters:

- H the magnetic scale height
- ν the kinematic viscosity
- T the wave period of the Alfv/'en waves
- α the ratio between magnetic and density scale heights $\alpha = H/H_\rho$
- ζ the density contrast between the centre of the density tube and the background
- u_0 the Alfvén wave amplitude
- r_0 the width of the density tube at the lower boundary
- ω the Alfvén wave frequency, currently defined based on T

Damped boundary parameters:

- *topdamp* logical for top boundary damping
- *outdamp* logical for outer boundary damping

and restart parameters:

- *restart* logical for whether to load from a restart file (same as output files)
- *v_in* and *b_in* file names for restart files
- *restart_time* and *last_output*, simulation time and output index for restart files

3.2 Functions

The functions used in *Wigglewave* are as follows:

3.2.1 funcv

This function calculates $f_v(t, v, b)$ at each grid point,

$$f_v(t, v, b) = \frac{1}{\mu_0 \rho} \left(B_r \frac{\partial b}{\partial r} + B_z \frac{\partial b}{\partial z} + \frac{B_r b}{r} \right) + \frac{1}{r\rho} \frac{\partial}{\partial r} \left(\rho \nu r \frac{\partial v}{\partial r} \right) + \frac{1}{\rho} \frac{\partial}{\partial z} \left(\rho \nu \frac{\partial v}{\partial z} \right)$$

3.2.2 funcb

This function calculates $f_b(t, v, b)$ at each grid point,

$$\frac{\partial b}{\partial t} = f_b(t, v, b) = B_r \frac{\partial v}{\partial r} + B_z \frac{\partial v}{\partial z} - \frac{v B_r}{r}$$

All derivatives in f_v and f_b are calculated using the fourth order central finite difference scheme,

$$\left. \frac{\partial f}{\partial x} \right|_i = \frac{-f(i+2) + 8f(i+1) - 8f(i-1) + f(i-2)}{12h}$$

where f is the function being differentiated with respect to the direction x, i is the index in the $x-$direction and h is the grid spacing in the $x-$direction.

3.3 Subroutines

The subroutine used in *Wigglewave* are as follows:

3.3.1 edgebc

This subroutine applies boundary conditions to the radial boundaries of the domain. The inner radial boundary at $r = 0$ uses radial anti-symmetry whilst the outer radial boundary uses reflective BC.

3.3.2 drivebc_v

This subroutine applies velocity BC for wave driving at the lower boundary. The wave driving is applied only to the ghost cells at the lower boundary and only to those within the tube radius $r < r_0$. The velocity driving BC take the form,

$$v = \begin{cases} u_0 \left(\frac{r}{r_0}\right)\left(1-\left(\frac{r}{r_0}\right)^2\right) \times \sin\left(\omega\left(\frac{z}{V_A}-t\right)\right) \times \left(1-\exp\left(-\left(\frac{t}{t_0}\right)^3\right)\right), & \text{if } r \leq r_0 \\ 0, & \text{if } r > r_0 \end{cases}$$

where the first, second and third terms control the amplitude, waveform and ramp-up time respectively.

3.3.3 drivebc_b

This subroutine applies magnetic BC for wave driving at the lower boundary. The wave driving is applied only to the ghost cells at the lower boundary and only to those within the tube radius $r < r_0$. The magnetic driving BC take the form,

$$b = \begin{cases} -u_0\sqrt{\mu_0\rho} \left(\frac{r}{r_0}\right)\left(1-\left(\frac{r}{r_0}\right)^2\right) \times \sin\left(\omega\left(\frac{z}{V_A}-t\right)\right) \times \left(1-\exp\left(-\left(\frac{t}{t_0}\right)^3\right)\right), & \text{if } r \leq r_0 \\ 0, & \text{if } r > r_0 \end{cases}$$

where the first, second and third terms control the amplitude, waveform and ramp-up time respectively.

3.3.4 damptop

This subroutine controls exponential damping adjacent to the upper boundary. It is applied if the parameter *topdamp* is set to true. The damping is applied to all grid cells above a height of $d = 4/5$ of the domain height. Above this height the variable is divided by a at each timestep,

$$f_n = \frac{f_{n-1}}{a} \quad \Rightarrow \quad f_n = \frac{f_0}{a^n}$$

where a is,

$$a = \quad 1 + \frac{z-d}{z_{max}-d}\, dt \quad = \quad 1 + q_z\, dt$$

3.3.5 dampout

This subroutine controls exponential damping adjacent to the outer radial boundary. It is applied if the parameter *outdamp* is set to true. The damping is applied to all grid cells beyond a radius of $d = 4/5$ of the domain radius. Beyond this radius the variable f is divided by a at each timestep,

$$f_n = \frac{f_{n-1}}{a} \quad \Rightarrow \quad f_n = \frac{f_0}{a^n}$$

where a is,

$$a = \quad 1 + \frac{r-d}{r_{max}-d}\,dt \quad = \quad 1 + q_r\,dt$$

NB To see that these subroutines are equivalent to exponential damping in time we consider the limit of infinitesimal dt and infinite time steps $n = t/dt$,

$$\lim_{n\to\infty} f_n = \frac{f_0}{\lim_{n\to\infty}(1+q\,dt)^n} = \frac{f_0}{\lim_{n\to\infty}(1+q\frac{t}{n})^n} = \frac{f_0}{\exp{(qt)}} = f_0 e^{-qt}$$

3.3.6 envelope

This subroutine is used to calculate the envelopes of v and b at each timestep this subroutine is applied to every grid point,

$$v_{env} = max\,(v_{env}, |v|)$$
$$b_{env} = max\,(b_{env}, |b|)$$

3.4 Main body

3.4.1 Setup

The code begins by calculating the grid spacings in r and z directions as well as the values of r and z at each grid point and then calculated the equilibrium quantities.

The value of ψ at the boundary of the flux tube, ψ_b, is calculated,

$$\psi_b = \frac{r_0^2}{2H}$$

The curvilinear field orientated coordinates ϕ and ψ are calculated,

$$\phi = \quad -H\exp(-z/H)J_0(r/H)$$
$$\psi = \quad r\exp(-z/H)J_1(r/H)$$

The magnetic field components B_r and B_z are calculated,

$$B_r = \quad B_0\exp(-z/H)J_1(r/H)$$
$$B_z = \quad B_0\exp(-z/H)J_0(r/H)$$

The density profile and Alfvén speed are calculated,

$$\hat{\rho}(\psi) = \begin{cases} \frac{\rho_0}{\zeta}(1 + (\zeta - 1)(1 - (\psi/\psi_b))^2), & \text{if } \psi \leq \psi_b \\ 0, & \text{if } \psi > \psi_b \end{cases}$$

$$\rho(\psi, z) = \quad \hat{\rho}(\psi) \exp(-\alpha z/H)$$

$$V_A = \quad \frac{\sqrt{B_r^2 + B_z^2}}{\sqrt{\mu_0 \rho}}$$

All the equilibrium quantities are then saved to files and the timestep is calculated,

$$dt = \quad \frac{1}{10} min\left(\frac{min(dr, dz)}{max(V_A)}, \frac{min(dr, dz)^2}{\nu}\right)$$

The time t is then either initialised at zero or set according to the restart settings. Similarly the variables v and b are then initialised as zero or loaded from a restart file.

3.4.2 Main Loop

The main loop uses an RK4 method to update v and b at each time step. The calculation used is equivalent to the following,

$$\begin{aligned}
k_1 &= f_v(t_n, v_n, b_n)\ dt & m_1 &= f_b(t_n, v_n, b_n)\ dt \\
k_2 &= f_v(t_n + dt/2, v_n + k_1/2, b + m_1/2)\ dt & m_2 &= f_b(t_n + dt/2, v_n + k_1/2, b + m_1/2)\ dt \\
k_3 &= f_v(t_n + dt/2, v_n + k_2/2, b + m_2/2)\ dt & m_3 &= f_b(t_n + dt/2, v_n + k_2/2, b + m_2/2)\ dt \\
k_4 &= f_v(t_n + dt, v_n + k_3, b + m_3)\ dt & m_4 &= f_b(t_n + dt, v_n + k_3, b + m_3)\ dt \\
\\
k &= (k_1 + 2k_2 + 2k_3 + k_4)/6 & m &= (m_1 + 2m_2 + 2m_3 + m_4)/6 \\
\\
t_{n+1} &= t_n + dt \\
v_{n+1} &= v_n + k \\
b_{n+1} &= b_n + m
\end{aligned}$$

However these calculations are not made directly as we need to apply BC each time spatial derivatives are taken, which means each time the functions f_v and f_b are applied, and the BC need to be applied to the arguments of these functions. The BC are already applied to v and b, the remaining arguments to which these functions are applied are,

$$
\begin{aligned}
fk_1 &= v_n + f_v(t_n, v_n, b_n)\ dt/2 &&= v_n + k_1/2 \\
fm_1 &= b_n + f_b(t_n, v_n, b_n)\ dt/2 &&= b_n + m_1/2 \\
\\
fk_2 &= v_n + f_v(t_n + dt/2, fk_1, fm_1)\ dt/2 &&= v_n + k_2/2 \\
fm_2 &= b_n + f_b(t_n + dt/2, fk_1, fm_1)\ dt/2 &&= b_n + m_2/2 \\
\\
fk_3 &= v_n + f_v(t_n + dt/2, fk_2, fm_2)\ dt &&= v_n + k_3 \\
fm_3 &= b_n + f_b(t_n + dt/2, fk_2, fm_2)\ dt &&= b_n + m_3
\end{aligned}
$$

It is therefore these arguments that are calculated directly. After each pair of arguments is calculated the boundary conditions are applied including the driving BC. The values of k_4 and m_4 are calculated directly using fk_3 and fm_3. Then v_{n+1} and b_{n+1} are calculated as,

$$
\begin{aligned}
v_{n+1} &= v_n + (2(fk_1 - v_n) + 4(fk_2 - v_n) + 4(fk_3 - v_n) + k_4)/\,6 \\
b_{n+1} &= b_n + (2(fm_1 - b_n) + 4(fm_2 - b_n) + 4(fm_3 - b_n) + m_4)/\,6
\end{aligned}
$$

After this the BC are applied to the v_{n+1} and b_{n+1}.

The envelope variables v_{env} and b_{env} are then calculated simply as the maximum value of v and b over the lifetime of the simulation using the *envelope* function.

Finally the simulation time is incremented by dt and the timestep is incremented by one.

3.4.3 Output Block

At each output time the code will output files for v, b, v_{env} and b_{env} as well as printing some key diagnostics. The output times are separated by $t_{interval}$.