

NASA Contractor Report 4725

solwnd: A 3D Compressible MHD Code for Solar Wind Studies. Version 1.0: Cartesian Coordinates

Anil E. Deane

GRANT NAG5-2652
MAY 1996



NASA Contractor Report 4725

**solwnd: A 3D Compressible MHD Code
for Solar Wind Studies
Version 1.0: Cartesian Coordinates**

Anil E. Deane
George Mason University
Fairfax, VA 22030-4444

Prepared for
Goddard Space Flight Center
under Grant NAG5-2652



National Aeronautics
and Space Administration

**Scientific and Technical
Information Branch**

1996

This publication is available from the NASA Center for Aerospace Information,
800 Elkridge Landing Road, Linthicum Heights, MD 21090-2934, (301) 621-0390.

**solwnd: A 3D Compressible MHD Code
for Solar Wind Studies.
Version 1.0: Cartesian Coordinates.**

Anil E. Deane

Institute for Computational Science and Informatics,
George Mason University &
High Performance Computing Branch,
NASA Goddard Space Flight Center
deane@laplace.gsfc.nasa.gov

Abstract

`solwnd_1.0` is a three-dimensional compressible MHD code written in Fortran for studying the solar wind. Time-dependent boundary conditions are available. The computational algorithm is based on Flux Corrected Transport and the code is based on the existing code of Zalesak and Spicer[1]. The flow considered is that of shear flow with incoming flow that perturbs this base flow. Several test cases corresponding to pressure balanced magnetic structures with velocity shear flow and various inflows including Alfvén waves are presented.

Version 1.0 of `solwnd` considers a rectangular Cartesian geometry. Future versions of `solwnd` will consider a spherical geometry. Some discussion of this issue is presented.

1 Introduction

For the purposes of studying 3D, fully compressible MHD flows with the capacity to capture shocks a new series of codes based on Flux Corrected Transport (FCT) termed `solwnd` have been developed. The present version of the code considers a rectangular geometry while future versions will be cast in spherical coordinates and consider a general frustum geometry. Time-dependent boundary conditions are available. These code are targeted towards addressing three effects previously unapproachable in solar wind studies:

1. Time-dependent inflow boundary conditions which can be imposed as well as various combinations of outflow and periodic boundary conditions.
2. The capacity to capture shocks, i.e. oscillation-free discontinuities spread over not more than a few grid points allowing the consideration of severely nonlinear cases and high Mach numbers.
3. The consideration of spherical geometry allowing the study of the fanning out of the flow with radius.

`solwnd_1.0` addresses the first two. The next version, `solwnd_2.0`, will address number 3. The code solves the inviscid compressible MHD equations in conservation form. The FCT procedure is well known [2, 3] and has been employed previously for MHD [4, 1]. A purely hydrodynamic version has been parallelized to run on several machines [5]. The FCT procedure proceeds by first forming a low-order and a high-order flux. Then an *anti-diffusive* flux is formed as their difference. This is subsequently (*flux-limited*) to prevent the formation of spurious extrema. Finally, the solution is updated based on the low-order solution and the anti-diffusive flux.

The magnetic field is calculated from the electric field via Faraday's Law. The cell-averaged magnetic field is cell-centered, while the electric field is edge-based. In this way the divergence condition is met to within numerical accuracy. The effect of computing the magnetic field via Faraday's Law is that \mathbf{B} is computed in planes and variables are packed and unpacked in two dimensional arrays. This reduces storage.

The code advances the primary cell-averaged variables of density, the three components of momenta, the total energy (internal + kinetic + magnetic), and the three components of magnetic field. The equation of state is taken to be ideal gas with a fixed adiabatic index. All other variables (pressure, electric field, low-order solution etc.) are considered temporary.

2 Equations

The basic equations of MHD in standard notation are[6]:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} = 0, \quad (1)$$

$$\rho \left[\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right] = -\nabla p + \frac{1}{c} (\mathbf{J} \times \mathbf{B}), \quad (2)$$

$$\frac{\partial \mathbf{B}}{\partial t} = -c \nabla \times \mathbf{E}, \quad (3)$$

$$\mathbf{J} = \frac{c}{4\pi} \nabla \times \mathbf{B}, \quad (4)$$

$$\mathbf{J} = \sigma (\mathbf{E} + \mathbf{u} \times \mathbf{B}). \quad (5)$$

The induction equation,

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{u} \times \mathbf{B}) + \frac{c^2}{4\pi\sigma} \nabla^2 \mathbf{B}. \quad (6)$$

For infinite conductivity, $\sigma \uparrow$ (5) implies,

$$\mathbf{E} + \mathbf{u} \times \mathbf{B} = 0. \quad (7)$$

The induction equation becomes,

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{u} \times \mathbf{B}). \quad (8)$$

The last term can be written,

$$\nabla \times (\mathbf{u} \times \mathbf{B}) = \mathbf{u}(\nabla \cdot \mathbf{B}) - \mathbf{B}(\nabla \cdot \mathbf{u}) + (\mathbf{B} \cdot \nabla)\mathbf{u} - (\mathbf{u} \cdot \nabla)\mathbf{B}, \quad (9)$$

or

$$= \nabla \cdot (\mathbf{B}\mathbf{u} - \mathbf{u}\mathbf{B}). \quad (10)$$

The dyadic terms are interpreted as

$$\nabla \mathbf{a} \mathbf{b} = (\mathbf{a} \cdot \nabla)\mathbf{b} + (\nabla \cdot \mathbf{a})\mathbf{b}.$$

The Lorentz force term in the momentum equation can be written,

$$\frac{1}{c}(\mathbf{J} \times \mathbf{B}) = -\frac{1}{4\pi}\mathbf{B} \times (\nabla \times \mathbf{B}). \quad (11)$$

The identity,

$$\frac{1}{2}\nabla(\mathbf{B} \cdot \mathbf{B}) = (\mathbf{B} \cdot \nabla)\mathbf{B} + \mathbf{B} \times (\nabla \times \mathbf{B}), \quad (12)$$

gives,

$$\frac{1}{c}(\mathbf{J} \times \mathbf{B}) = -\nabla\left(\frac{B^2}{8\pi}\right) + \frac{1}{4\pi}(\mathbf{B} \cdot \nabla)\mathbf{B}. \quad (13)$$

In addition to the above equations there is the energy equation, one form of which is [7],

$$\rho \frac{De}{Dt} - \frac{p}{\rho} \frac{D\rho}{Dt} = 0. \quad (14)$$

Taking $\mathbf{u} \cdot$ (2) and $\mathbf{B} \cdot$ (8) and adding to the energy equation (14) gives the equation for total energy,

$$\frac{\partial}{\partial t}(\rho e + \frac{1}{2}\rho u^2 + \frac{B^2}{8\pi}) + \nabla \cdot [(\rho e + \frac{1}{2}\rho u^2 + p)\mathbf{u} + \frac{c}{4\pi}\mathbf{E} \times \mathbf{B}] = 0. \quad (15)$$

Thus the equations as solved in the code become,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} = 0, \quad (16)$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot [(p + \frac{B^2}{8\pi})\delta_{ij} + \rho \mathbf{u} \mathbf{u} - \frac{1}{4\pi}\mathbf{B} \mathbf{B}], \quad (17)$$

$$\frac{\partial}{\partial t}(\rho e + \frac{1}{2}\rho u^2 + \frac{B^2}{8\pi}) + \nabla \cdot [(\rho e + \frac{1}{2}\rho u^2 + p)\mathbf{u} + \frac{1}{4\pi}\mathbf{u} \times \mathbf{B} \times \mathbf{B}] = 0, \quad (18)$$

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot [\mathbf{u} \mathbf{B} - \mathbf{B} \mathbf{u}] = 0, \quad (19)$$

with the relations,

$$\nabla \cdot \mathbf{B} = 0, \quad (20)$$

$$p = \rho RT. \quad (21)$$

3 Spherical Coordinates

For solving these equations in a spherical geometry which future versions of `solwnd` will consider, the code must be written with general metric coefficients so that any system of orthogonal coordinates are usable.

If $A = \{A_1 \mathbf{x}_1, A_2 \mathbf{x}_2, A_3 \mathbf{x}_3\}$, the divergence operator in curvilinear coordinates is:

$$\nabla \cdot \mathbf{A} = \frac{1}{h_1 h_2 h_3} \left[\frac{\partial}{\partial x_1} (h_2 h_3 A_1) + \frac{\partial}{\partial x_2} (h_1 h_3 A_2) + \frac{\partial}{\partial x_3} (h_1 h_2 A_3) \right], \quad (22)$$

where for spherical coordinates the metric coefficients are

$$h_1 = 1, \quad h_2 = r, \quad h_3 = r \sin \theta. \quad (23)$$

So that the divergence operator is

$$\nabla \cdot \mathbf{A} = \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 A_r) + \frac{1}{r \sin \theta} \frac{\partial}{\partial \theta} (\sin \theta A_\theta) + \frac{1}{r \sin \theta} \frac{\partial}{\partial \phi} A_\phi. \quad (24)$$

The divergence of the dyadic terms is:

$$\begin{aligned} (\nabla \cdot \mathbf{AB})_r &= \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 A_r B_r) + \frac{1}{r \sin \theta} \frac{\partial}{\partial \theta} (\sin \theta A_\theta B_r) + \frac{1}{r \sin \theta} \frac{\partial}{\partial \phi} (A_\phi B_r) \\ &\quad + \frac{A_\theta B_\theta + A_\phi B_\phi}{r}, \end{aligned} \quad (25)$$

$$\begin{aligned} (\nabla \cdot \mathbf{AB})_\theta &= \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 A_r B_\theta) + \frac{1}{r \sin \theta} \frac{\partial}{\partial \theta} (\sin \theta A_\theta B_\theta) + \frac{1}{r \sin \theta} \frac{\partial}{\partial \phi} (A_\phi B_\theta) \\ &\quad + \frac{A_\theta B_r}{r} - \frac{\cot \theta A_\phi B_\phi}{r}, \end{aligned} \quad (26)$$

$$\begin{aligned} (\nabla \cdot \mathbf{AB})_\phi &= \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 A_r B_\phi) + \frac{1}{r \sin \theta} \frac{\partial}{\partial \theta} (\sin \theta A_\theta B_\phi) + \frac{1}{r \sin \theta} \frac{\partial}{\partial \phi} (A_\phi B_\phi) \\ &\quad + \frac{A_\phi B_r}{r} + \frac{\cot \theta A_\phi B_\theta}{r}. \end{aligned} \quad (27)$$

4 Flux Corrected Transport

`solwnd` uses Flux Corrected Transport (FCT) as a shock capturing scheme. In flux conservative form (for 1D)[3] the equation,

$$\frac{dw}{dt} + \frac{df}{dx} = 0,$$

is

$$w_i^{n+1} = w_i^n - \frac{\Delta t}{\Delta x} (F_{i+1/2} - F_{i-1/2}).$$

The w are interpreted as density, momentum, and the energy; i.e. in 1D $w \equiv \{\rho, \rho u, \rho E\}$ and $f \equiv \{\rho u, \rho u^2 + p, \rho u E + pu\}$. The dependence of F on f is determined by the order of the scheme (see next section). The 3D calculation is extended analogously.

The procedure is as follows,

1. Form a *low-order flux*: $F_{i+1/2}^L$.
2. Form a *high-order flux*: $F_{i+1/2}^H$.
3. Form an *anti-diffusive flux*:

$$A_{i+1/2} = F_{i+1/2}^H - F_{i+1/2}^L.$$

4. Form a *low-order solution*:

$$w_i^L = w_i^n - \frac{\Delta t}{\Delta x} (F_{i+1/2}^L - F_{i-1/2}^L).$$

5. Limit the anti-diffusive flux to prevent spurious extrema in w_i^{n+1} :

$$A_{i+1/2}^C = C_{i+1/2} A_{i+1/2} \quad 0 \leq C_{i+1/2} \leq 1.$$

Determination of the appropriate C is the critical and complex step.

6. Update the solution:

$$w_i^{n+1} = w_i^L - \frac{\Delta t}{\Delta x} (A_{i+1/2}^C - A_{i-1/2}^C).$$

5 Formation of Fluxes

In spherical coordinates most of the terms can be written in conservation form. However some terms arise that are nonconservative and must be treated as source terms for the otherwise conservative equations. The latter terms can be treated in a straightforward manner by the inclusion of the appropriate metric coefficients in the formation of the various fluxes required

in the scheme. The source terms are not flux-limited and are added just prior to the solution update. Consider for example the momentum equation, the flux conservative form of which is,

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot \left[\left(p + \frac{B^2}{8\pi} \right) \delta_{ij} + \rho \mathbf{u} \mathbf{u} - \frac{1}{4\pi} \mathbf{B} \mathbf{B} \right].$$

In spherical coordinates the x -component becomes,

$$\begin{aligned} \frac{\partial \rho u_r}{\partial t} + \frac{1}{r^2} \frac{\partial}{\partial r} r^2 \left(p + \rho u_r^2 + \frac{B^2}{8\pi} \right) + \frac{1}{r \sin \theta} \frac{\partial}{\partial \theta} (\sin \theta \rho u_\theta u_r) + \frac{1}{r \sin \theta} \frac{\partial}{\partial \phi} (\rho u_\phi u_r) \\ - \frac{1}{4\pi r^2} \frac{\partial}{\partial r} (r^2 B_r^2) - \frac{1}{4\pi r \sin \theta} \frac{\partial}{\partial \theta} (\sin \theta B_\theta B_r) - \frac{1}{4\pi r \sin \theta} \frac{\partial}{\partial \phi} (B_\phi B_r) \\ + \frac{u_\theta^2 + u_\phi^2}{r} - \frac{B_\theta^2 + B_\phi^2}{4\pi r}. \end{aligned}$$

The last two terms, proportional to $1/r$, are non-conservative and have no analog in Cartesian coordinates and are treated as source terms in spherical coordinates. All the other terms in the above equation are conservative and are treated as in the Cartesian case in the FCT procedure. In 1D for instance,

$$w_i^{n+1} = w_i^n - \frac{\Delta t}{\Delta r} (F_{i+1/2} - F_{i-1/2}),$$

represents the update of a quantity w . The consistent formation of fluxes is the key first step in the procedure. For a fourth-order scheme [3],

$$F_{ijk} = \frac{7}{12} (f_{ijk} + f_{i+1,jk}) - \frac{1}{12} (f_{i-1,jk} + f_{i+2,jk}). \quad (28)$$

6 Calculation of B

The electric field is calculated from the the magnetic field via (7). The magnetic field in turn is updated from the electric field via Faraday's Law.

The cell-averaged magnetic field (e.g. $B_z(i, j, k)$) is located at the face of the cell centered at (i, j, k) (see Figure 1), while the electric field is edge-based. Thus the divergence condition is met to within numerical accuracy. Also, storage is reduced since \mathbf{B} is computed in planes and variables are packed and unpacked in two-dimensional temporary arrays.

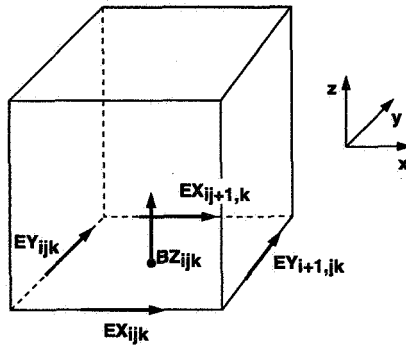


Figure 1: Calculation of \mathbf{B} from \mathbf{E} .

7 Time stepping

The code presently uses 2nd order Runge-Kutta (modified Euler) scheme for time stepping. With this scheme,

$$w^{n+1/2} = w^n - \frac{\Delta t}{2} F(w^n),$$

$$w^{n+1} = w^n - \Delta t F(w^{n+1/2}).$$

The scheme has a weak instability along the imaginary axis, but for the length of integrations typically considered here it does not manifest itself. The advantage of the scheme of course is that only two time levels of storage are required.

8 Geometry

The present code considers a rectangular Cartesian grid. However future versions of the code will consider a general frustum (Fig. 2) geometry. This has the advantage that the expansion of the flow with radial distance from the Sun is naturally incorporated.

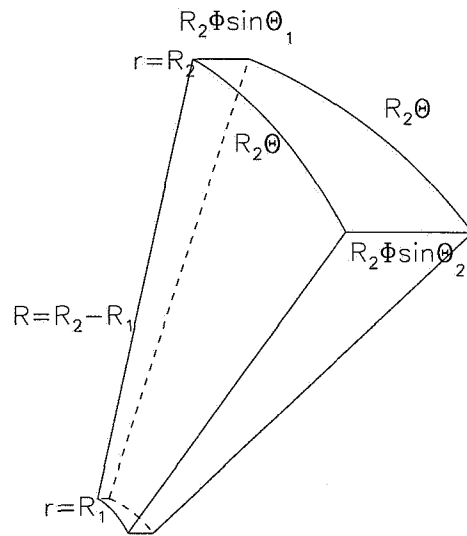


Figure 2: Frustum geometry of solwnd.2.0

9 Initial conditions

To simulate one set of model problems relevant to the solar wind, we consider¹ that the base flow is in the x -direction, with the direction of shear in the z direction and periodic conditions in y and z directions. Then the base flow

¹This scheme was suggested by A. Roberts and M. Goldstein.

is given by²

$$B_{0x}(x, y, z) = B_0 + \delta B_0(y, z), \quad (29)$$

or

$$B_{0x}(x, y, z) = B_0 + \delta B_0 f(y, z), \quad (30)$$

where $f(y, z)$ can be chosen, as e.g.,

$$f(y, z) = \cos(2\pi k_y y) \cos(2\pi k_z z). \quad (31)$$

We introduce the ratio η_B such that,

$$B_0 = \frac{RMS(\delta B_0 f(y, z))}{\eta_B}. \quad (32)$$

The transverse magnetic field amplitude is given by,

$$\delta B_{yz} = \eta_{\perp} B_0. \quad (33)$$

The initial velocity profile is given by,

$$V_x(x, y, z) = \begin{cases} V_0 + \delta V_0 \tanh(z - L_z/4)/z_s & \text{if } z \leq L_z/2 \\ V_0 - \delta V_0 \tanh(z - 3L_z/4)/z_s & \text{otherwise} \end{cases} \quad (34)$$

z_s is the shear layer thickness parameter.

We also introduce η_V such that

$$\delta V_0 = \eta_V V_0, \quad (35)$$

²To indicate what quantities are true constants we show them in roman type. Thus $B_0, \delta B_0 V_0, \delta V_0$, are numerical constants. Other constants, not so indicated, are parameters such as M and M_A . Spatial dependence is indicated on fields as (x, y, z) . Thus B_0 is a constant while $B_0(x, y, z)$ is an initial field. $f(x, y, z)$ or $f(y, z)$ etc. indicate arbitrary functions of the variables, and can indicate different functional dependence from relation to relation.

δV_0 is determined once V_0 is specified.

Initially the transverse velocities are set to zero,

$$V_y(x, y, z, 0) = 0, \quad (36)$$

and

$$V_z(x, y, z, 0) = 0. \quad (37)$$

Now setting,

$$\rho_0(x, y, z) = \frac{1}{4\pi}, \quad (38)$$

gives the Alfven speed,

$$V_A = \frac{B_0}{(4\pi\rho_0)^{1/2}} \quad (\equiv B_0, \text{ by construction}). \quad (39)$$

Since the Alfven Mach number is a parameter V_0 can now be specified,

$$V_0 = M_A V_A. \quad (40)$$

The sound speed is determined from the sonic Mach number, M ,

$$c_s = M V_0. \quad (41)$$

To obtain pressure balance, if

$$p(x, y, z, 0) = \frac{\delta B_0(y, z)^2}{8\pi} + \frac{c_s^2}{\gamma}(\rho_0 + \delta\rho(x, y, z)) + p_{tot} = \frac{B_0^2}{8\pi} + \frac{c_s^2}{\gamma}\rho_0 + p_{tot},$$

canceling common terms,

$$\rho(x, y, z, 0) = \rho_0 + \frac{\gamma}{8\pi c_s^2} (B_0^2 - \delta B_0(y, z)^2). \quad (42)$$

Finally note that since both the sonic and Alfvén Mach numbers are parameters, the plasma β is a derived quantity,

$$\beta = \frac{V_A^2}{c_s^2} = \frac{M^2}{M_A^2}. \quad (43)$$

Table 1 lists the user specified parameters and their Fortran names in the `solwnd` code.

Parameter	symbol in text	Fortran name
Perturbation B_x	δB_0	<code>bx_pert</code>
Initial B ratio, $\text{RMS} \frac{\delta B_0}{B_0}$	η_B	<code>eta_b</code>
Initial V ratio, $\frac{\delta V_0}{V_0}$	η_V	<code>eta_v</code>
Initial transverse B ratio, $\frac{\delta B_{yz}}{B_0}$	η_\perp	<code>eta_perp</code>
sonic Mach number	M	<code>fmach</code>
Alfvén Mach number	M_A	<code>fmacha</code>
B_0 wavenumber in y direction	k_y	<code>xky</code>
B_0 wavenumber in z direction	k_z	<code>xkz</code>

Table 1: Initial condition parameters of `solwnd` code. Also see section 2 (page 30).

10 Time-dependent boundary conditions

The solar wind can be modeled as a spatially evolving shear layer with time-dependent inflow. For the case when the inflow is considered to consist of Alfvén waves, at each time step these conditions are imposed at the $x = 0$ boundary. Namely,

$$B_y(0, y, z, t) = \delta B_{yz} \sin \omega_A t, \quad (44)$$

$$B_z(0, y, z, t) = \delta B_{yz} \cos \omega_A t, \quad (45)$$

$$V_y(0, y, z, t) = -B_z(0, y, z, t), \quad (46)$$

$$V_z(0, y, z, t) = -B_z(0, y, z, t). \quad (47)$$

Here ω_A is the frequency of the incoming Alfvén wave and is an input parameter for the code. The density and pressure are fixed as,

$$\rho(0, y, z, t) = \rho(0, y, z, 0), \quad (48)$$

$$p(0, y, z, t) = p(0, y, z, 0). \quad (49)$$

Since the code evolves density, momenta and energy, the above imposed fields are used to calculate these variables at $x = 0$.

11 Test results

We show three test case results. The first case is a two-dimensional purely hydrodynamic calculation. The run was made by making `nx=324`, `ny=5`, `nz=36`; the update on the y -momentum was skipped; and the boundary condition in the y -direction was such that all planes in y identical. Figure 3 shows the vortex roll-up from spatially evolving double shear layer.

The second test case (Figure 4) considers the triply-periodic geometry of `solwnd_0.2`. This temporally evolving shear layer is what is typically studied with spectral (Fourier) codes.

The third test case is with the version `solwnd_0.9` with initial conditions set forth in the previous sections (version 0.9 contains the same physics and algorithm of version 1.0). The parameters of the case are given in Table 2. The initial velocity profile is shown in Figure 5. Figures 6 and 7 show the fluid and magnetic fields in the $x-z$ plane while Figure 8 show these field in the $y-z$ plane. Subsequent figures 9-14 show cuts of velocity and magnetic field as arrow plots so that a sense of direction can be discerned. A streaming set of shock waves and magnetic structures are seen to be swept through the computational domain. A small subsonic and sub-Alfvénic region is found to form near the inflow, no doubt corrupting the flow somewhat. However the effect of upstream traveling waves appears to be minimal. Figures 15 and 16 show the max/min values over the entire computational domain of the sonic and Alfvénic Mach numbers, density and pressure.

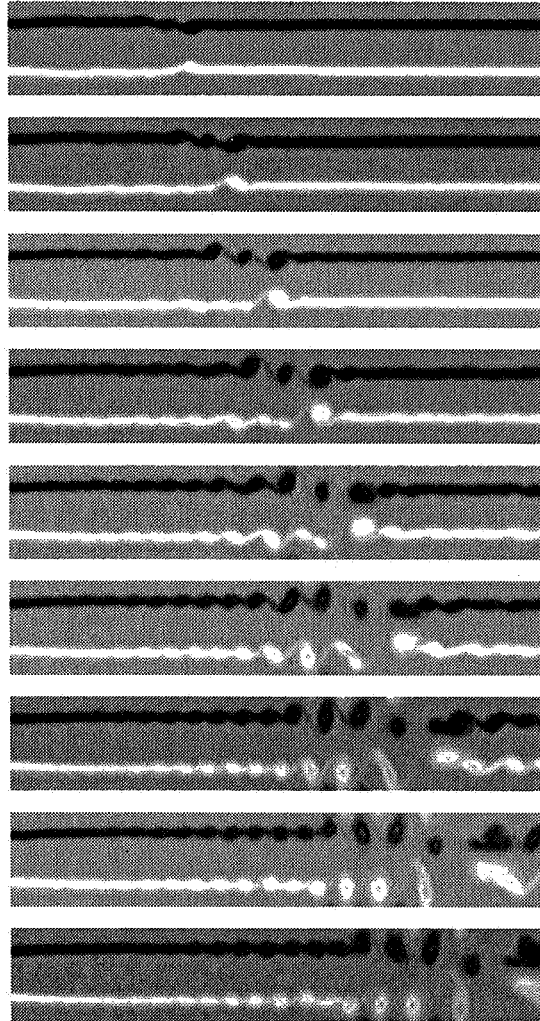


Figure 3: Test case 1. Vortex roll-up. Two-dimensional purely hydrodynamic calculation. The resolution is 320×32 . The vertical scale has been exaggerated.

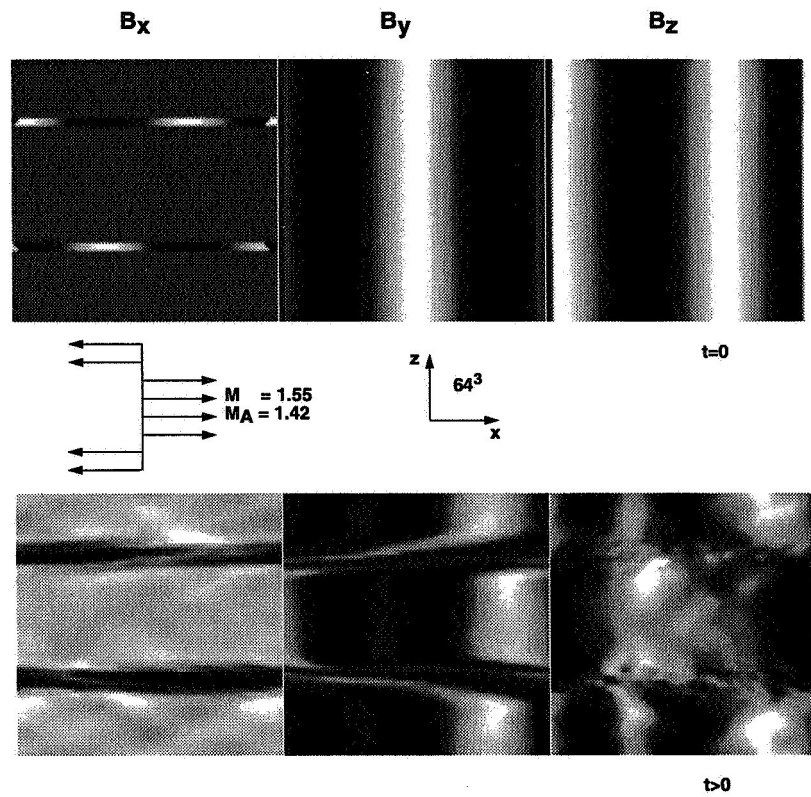


Figure 4: Test case 2. Triply periodic flow with the indicated parameters and configuration. The resolution is 64^3 . The upper panel is at time, $t=0$, while the lower panel is at a later time.

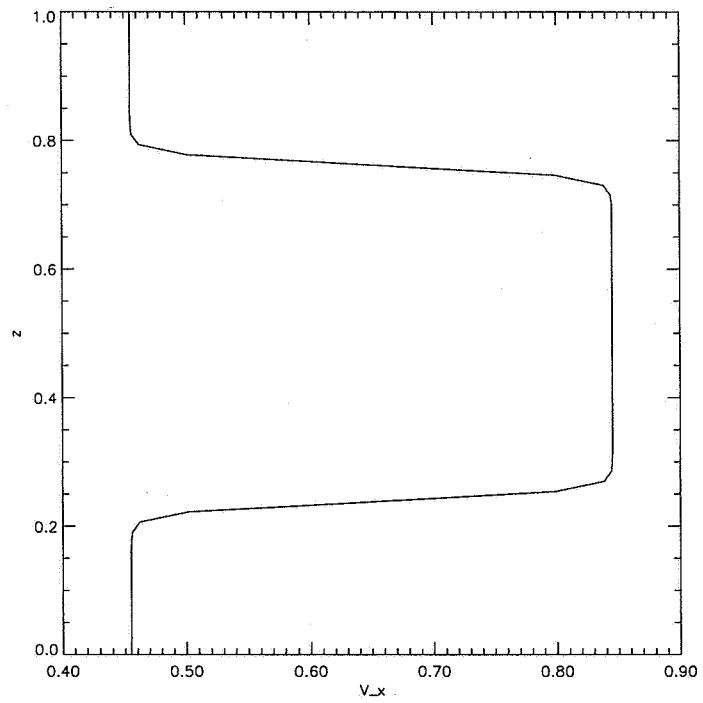


Figure 5: Test case 3. Initial double shear layer velocity profile for `solwnd_0.9` with parameters given in Table 2. The resolution is 256×64^2 .

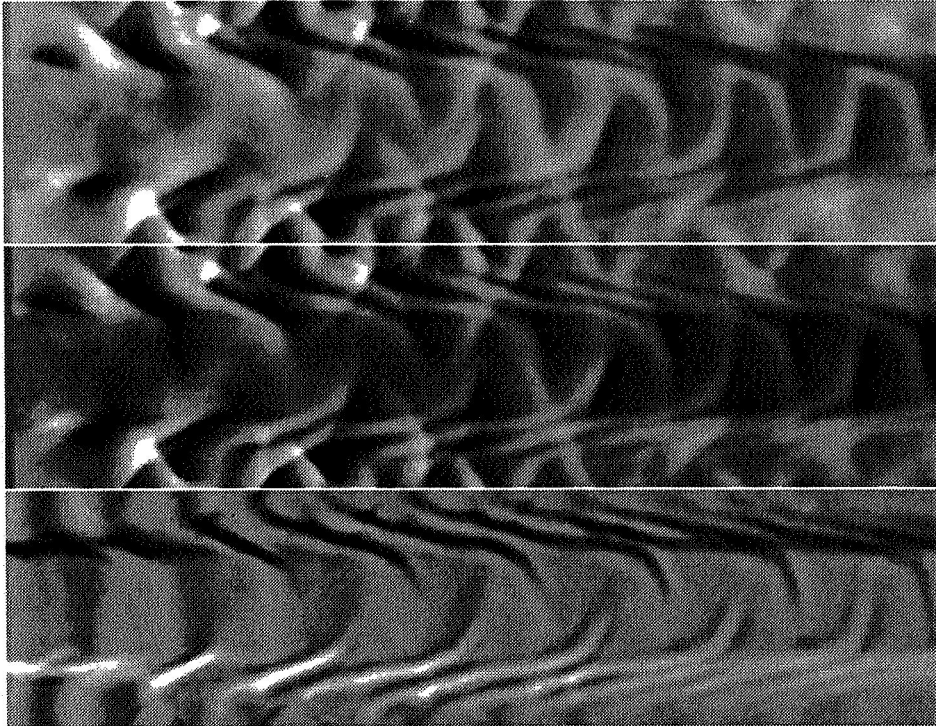


Figure 6: Test case 3. Images of density, pressure and y -vorticity, ω_y , in the $x - z$ plane at time $t = 5.4$ (step 2000) for `solwnd_0.9` with parameters given in table 2. The resolution is 256×64^2 .

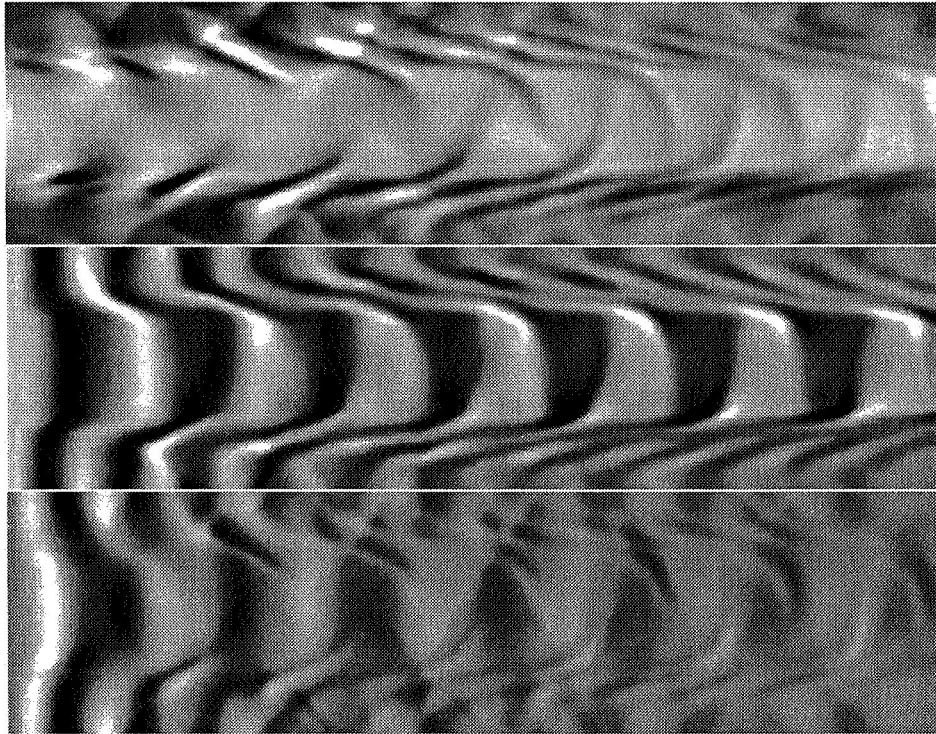


Figure 7: Test case 3. Images of magnetic fields, B_x , B_y and B_z in the $x - z$ plane at time $t = 5.4$ (step 2000) for solwnd_0.9 with parameters given in Table 2. The resolution is 256×64^2 .



Figure 8: Test case 3. Images of x -vorticity, density and B_x , in the $y - z$ plane at time $t = 5.4$ (step 2000) for solwnd_0.9 with parameters given in Table 2. The resolution is 256×64^2 .

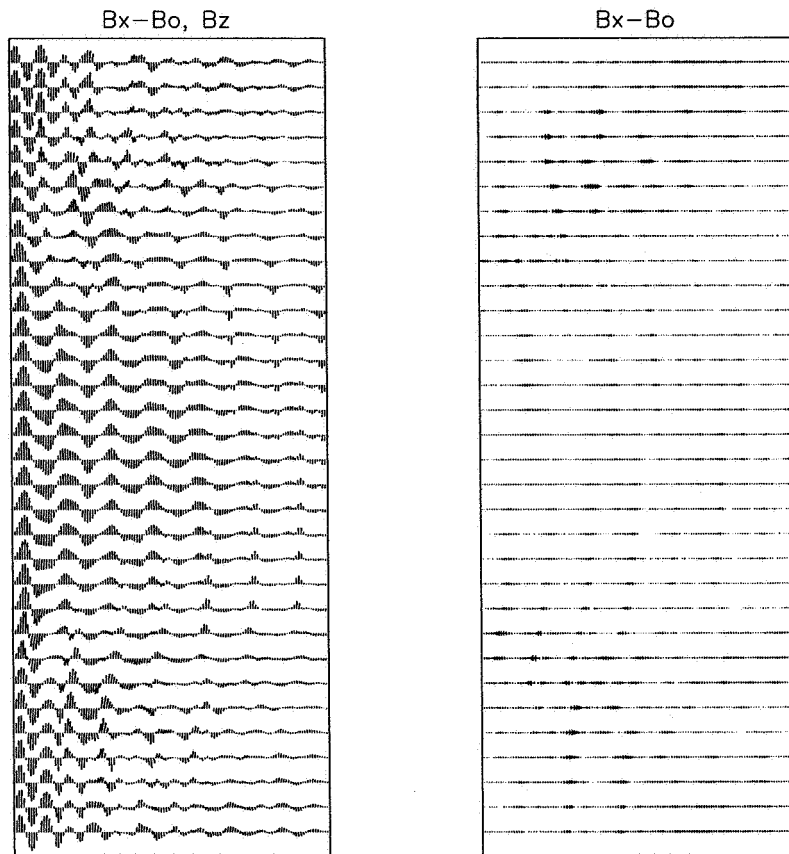


Figure 9: Test case 3. Magnetic field vectors in the $x - z$ plane. Left panel has components $(B_x - B_0, B_z)$. Right panel has components $(B_x - B_0, 0)$.

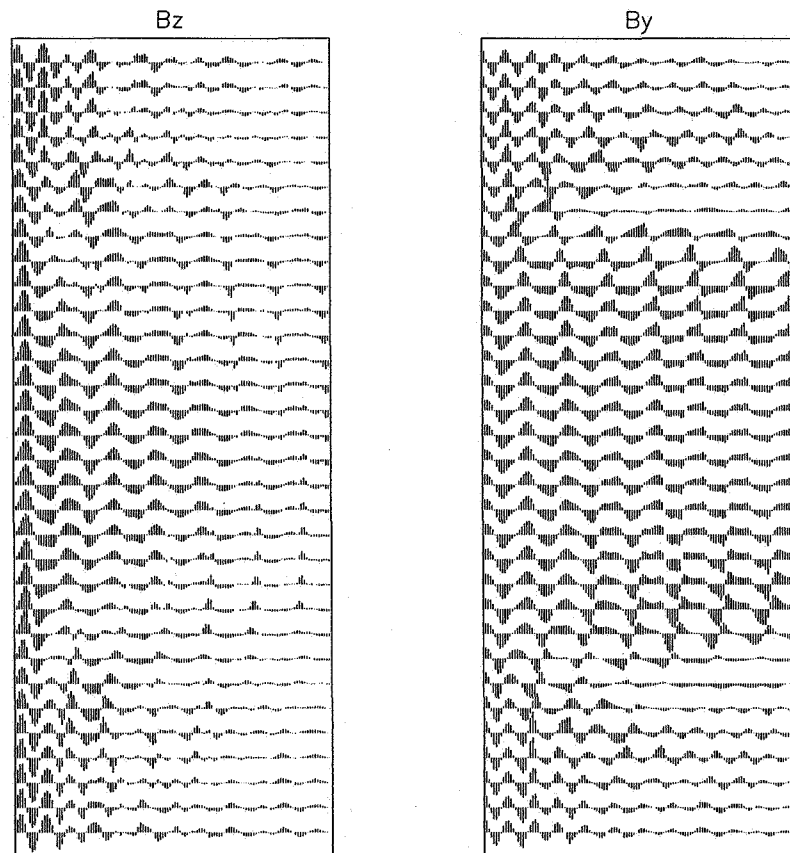


Figure 10: Test case 3. Magnetic field vectors in the $x - z$ plane. Left panel has components $(0, B_z)$. Right panel has components $(0, B_y)$.

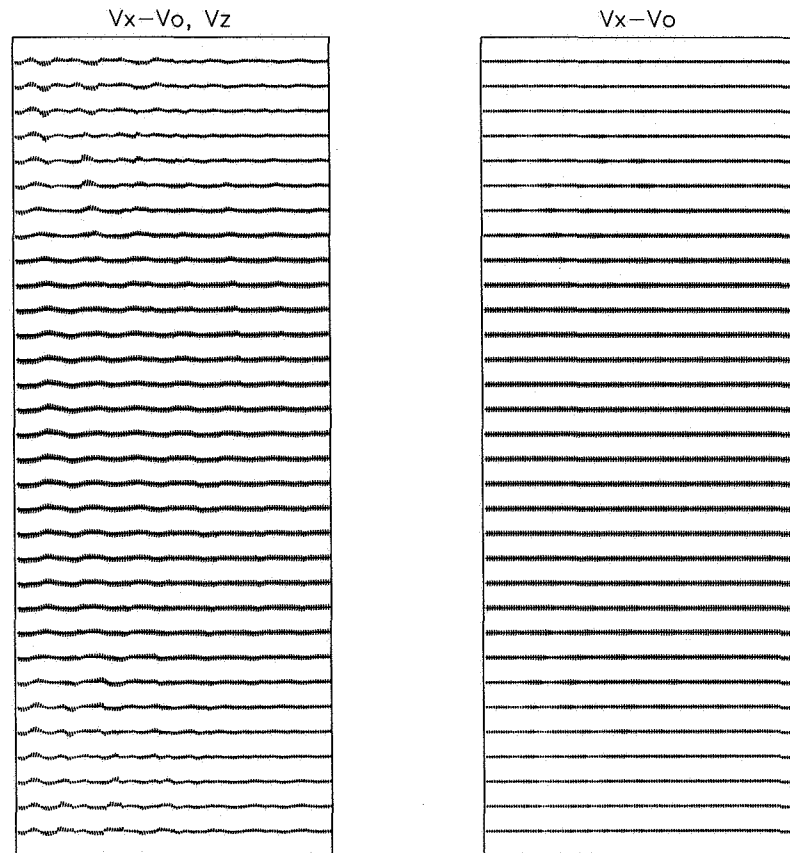


Figure 11: Test case 3. Velocity field vectors in the $x - z$ plane. Left panel has components $(V_x - V_0, V_z)$. Right panel has components $(V_x - V_0, 0)$.

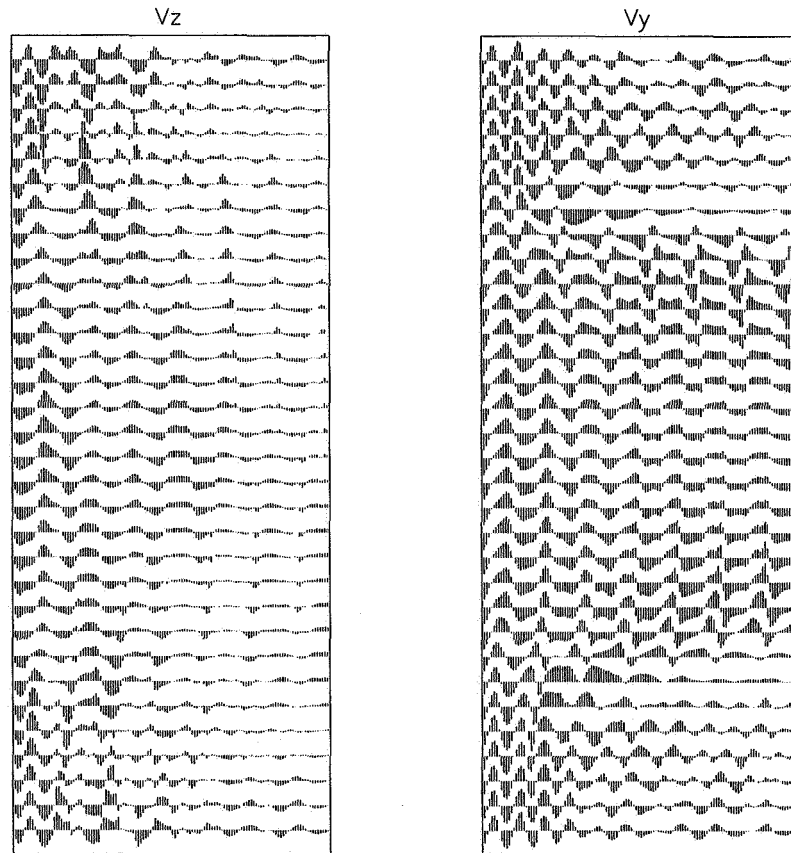


Figure 12: Test case 3. Velocity field vectors in the $x - z$ plane. Left panel has components $(0, V_z)$. Right panel has components $(0, V_y)$.

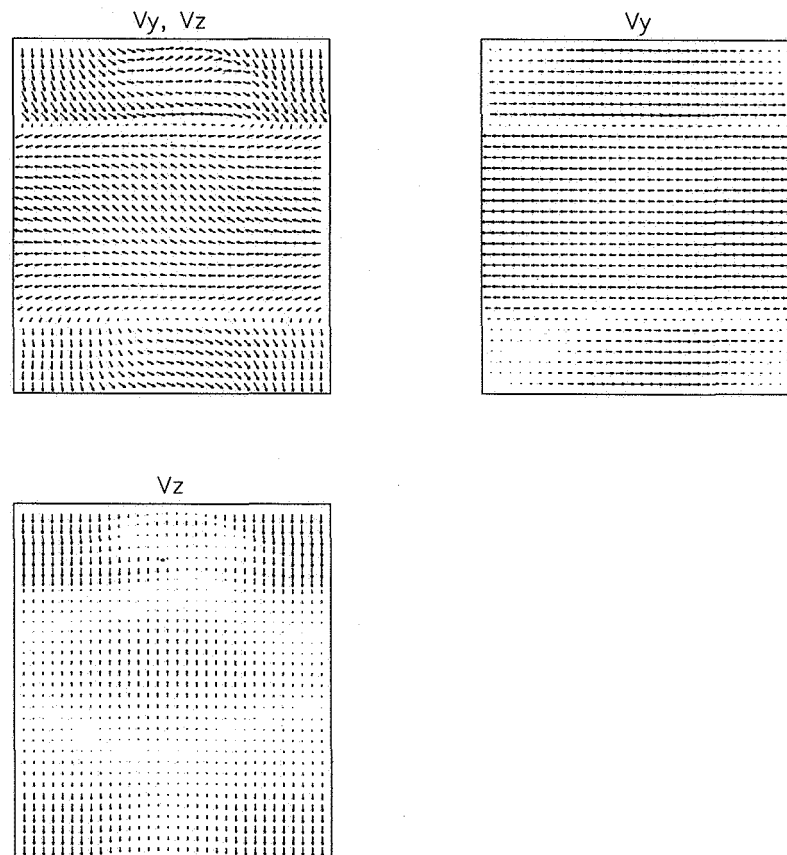


Figure 13: Test case 3. Velocity field vectors in the $y - z$ plane. Top left panel has components (V_y, V_z) . Top right panel has components $(V_y, 0)$. Bottom panel has components $(0, V_z)$.

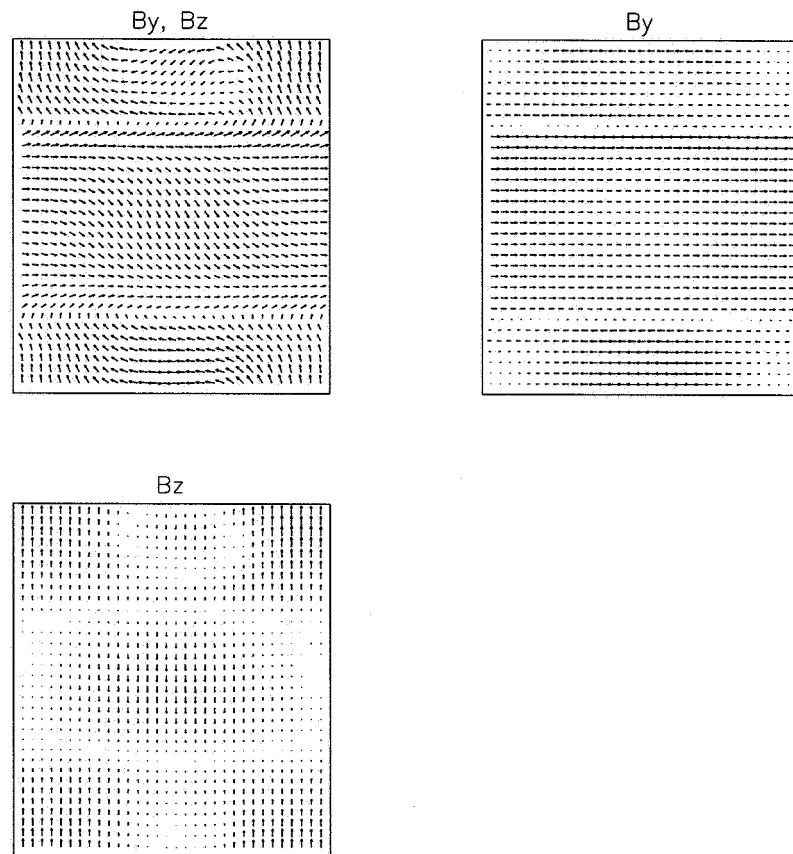


Figure 14: Test case 3. Magnetic field vectors in the $y - z$ plane. Top left panel has components (B_y, B_z) . Top right panel has components $(B_y, 0)$. Bottom panel has components $(0, B_z)$.

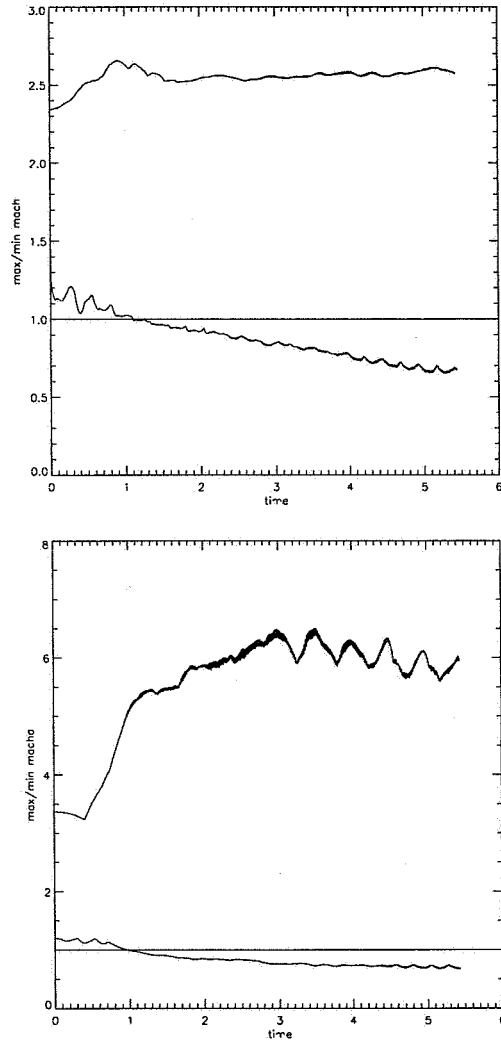


Figure 15: Test case 3. Top: Maximum and minimum sonic Mach number versus time, Bottom: Maximum and minimum Alfvénic Mach number versus time, for solwnd_0.9 with parameters given in table 2. The resolution is 256×64^2 .

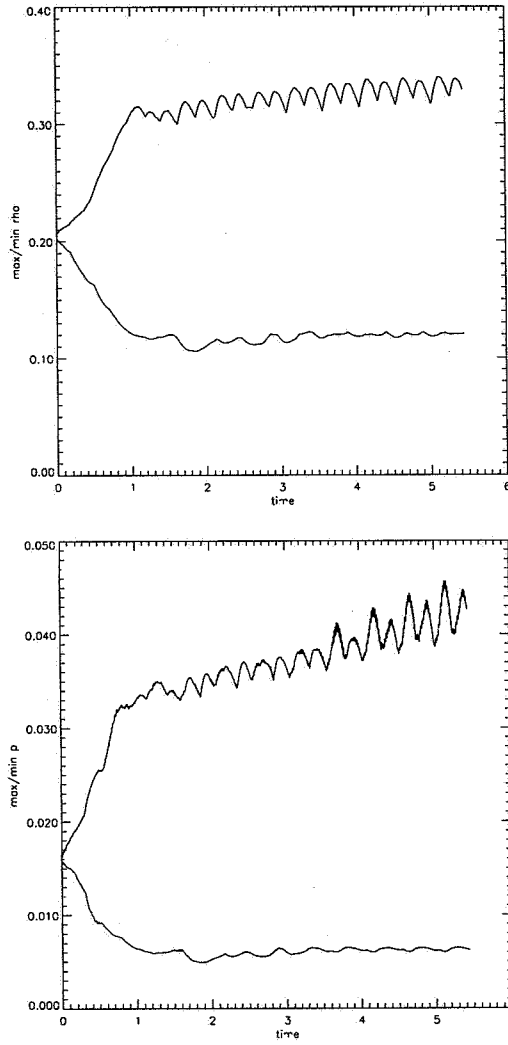


Figure 16: Test case 3. Top: Maximum and minimum density versus time, Bottom: Maximum and minimum pressure versus time, for `solwnd_0.9` with parameters given in table 2. The resolution is 256×64^2 .

Parameter	value
δB_{x0}	0.1
η_B	0.1
η_{\perp}	0.3
η_V	0.3
M	1.8
M_A	1.3
k_x	0
k_y	1
k_z	1
γ	1.67
L_x	4
L_y	1
L_z	1
z_s	1
ω_A	13
λ	10
B_0	0.5
V_0	0.65
δV_{x0}	0.195
ρ_0	0.08
V_A	0.5
c_s	0.36
β	1.9

Table 2: Initial condition parameters of `solwnd` code for test case 3. Derived quantities are shown below the line.

12 Version history

This section records the history of the various versions of `solwnd`.

0.0 Original FCT MHD code of Zalesak and Spicer. Non-working.

0.1 (5-95) Included boundary conditions on cell-averaged \mathbf{B} . Allows code to work.

0.2 (5-95) Triply periodic code. Initial conditions introduced: 2 shear layers in x - direction with tanh profile. 64^3 grid test cases computed

with supersonic and super-Alfvenic initial conditions.

- 0.3 (6-95) Inflow-outflow in x -direction and outflow in y, z -directions code produced. Because the code computes current from B -field, and then updates current to update B field, the $n_x - 3$ point does not get a definition for B . Hence B made continuous at $n_x - 3$. Similarly the end planes in the $x - y$ and $x - z$ planes need to be made continuous.
- 0.4 (6-95) Inflow-outflow in x -direction and periodic in y, z -directions code.
- 0.5 (7-95) Time-dependent boundary conditions added. Initial conditions corresponding to a pressure balanced double shear layer with sinusoidal transverse B and incoming Alfven waves added. 64^3 and 96^3 computations done. First results of spatially evolving cross-helicity shown. Presented in Goddard report ????
- 0.6 (1-96) Modified initial conditions to form a self-consistent model described in section above.
- 0.7 (1-96) Bugs fixed from previous version.
- 0.9 (2-96) Completed version used for test case 3. Modified which data fields are dumped. Added max/min trace values. Added point trace values.
- 0.9.1 (2-96) Restructured code. Put movie calls into routine `do_movie`. Added restart capabilities, single `SIZE` parameter file. Added smart data directory. Released as `solwnd_1.0`.

13 User-friendly features

`solwnd` has some user-friendly features making it a convenient code to work with.

1. **problem size file.** The file `SIZE.sw` sets the problem size and only needs to be altered for different problem sizes.

```
parameter (nx=132, ny=36, nz=36)
c      set nxmax to be the largest of (nx,ny,nz)
```

c set nymax to be the 2nd largest of (nx,ny,nz)
 parameter (nxmax=132, nymax=36)

As indicated, nxmax and nymax need to be appropriately set.

2. **input file.** An input file is provided for the parameters.

```
file solwnd_1.0.in

/disk6/data/solwnd/test/
sw_AAAA                               : header (7 chars)
0 : if restart
/disk6/data/solwnd/test/sw_1005.dat
1000                               nsteps : final iteration step
200                                idump  : number of iterations between dump
1                                   nstart : iteration starting counter
0.25e+0                            crn    : courant number
1.00e-1    bx_pert : delta B_Ox, initial delta B in x direction
0.10e+0                            eta_b  : delta B rms/B0
0.30e+0                            eta_perp: delta B_perp/B0
0.30e+0                            eta_v  : delta V_x rms
1.80e+0                            fmach  : Sonic Mach number
1.30e+0                            fmacha : Alfven Mach number
0.00e+0                            xkx    : B0 wavenumber in x direction
1.00e+0                            xky    : B0 wavenumber in y direction
1.00e+0                            xkz    : B0 wavenumber in z direction
1.67e+0                            gamma  : ratio of specific heats
1.00e+0    fk      : shear layer thickness factor (1-2 is good)
1.30e+1                            omega  : frequency of incoming waves
1.00e+1                            lambda : time constant for inflow ramp
4.00e+0                            xlen   : length of system in units of length
1.00e+0                            ylen   : length of system in units of length
1.00e+0                            zlen   :
t                                   lsynch : fct synchronization
t                                   lhydro : switch to turn on coupled hydro
t                                   lfield : switch to turn on magnetic field solver
```

Line 1 is the directory in which all data, trace and movie files will be placed. The directory must be terminated with a “/”.

Line 2 is the case name that will be prefixed to all the data, trace and movie files. A useful convention is sw_ tagging the AAAA which provide ample opportunity to vary parameters and successive runs can be labeled AAAB, AAAC, AAAD etc. Data files (field files) are created with

the header and with 1000 (e.g. `sw_AAAA1000.dat`, `sw_AAAA1001.dat`, `sw_AAAA1002.dat` etc.) added to the dump number. Thus an `ls` or other unix sort lists them in order.

Line 3 is a flag; if set to 1 indicates that the file listed in the next line must be read to initialize the data. The code begins the run with the step and time of the restart file.

Line 4 is the restart file name. Note that it can be in a different directory from the present run. Point trace values are either created in a `.trc1` or `.trc2` files or if they exist in the data directory they are appended to these files.

The rest of the lines are either self explanatory or discussed in the text previously.

3. **data directory.** As indicated above the data directory is written in with all data, trace and movie files.
4. **restart.** As indicated above the code can restart from a previous run.
5. **movie files.** The routine `do_movie` in the code creates subfields which are then passed to the routine `dump_movie`. New fields that are needed can be added to the ones already present or the present ones eliminated by editing `do_movie`. The program `dump_movie.c` is a code written by A. Malagoli (U. Chicago) for viewing the movie files. This program is invoked by, for instance,

```
Movie -z 2 2 -p 5 movie_file_name
```

The `-p` option indicates the color map. The `-z` option indicates a zoom of 2 in both directions.
6. **time step CPU counters** . If `-SGI` is chosen in the compile line then a CPU time step counter is used.
7. **data, max/min and point trace values.** Full field data of density, pressure, 3 components of momentum, total energy and 3 components of magnetic fields are saved every `ndump` steps in `.dat` files. Every time step, time, max/min values of sonic and Alfvenic Mach number, density and pressure are saved in `.trc1` files. Point values of each of the primary variables of density, momenta, energy and magnetic fields are put in an unformatted file `.trc2`. The sample points are at the

stations,

$$x = L_x/4, L_x/2, 3L_x/4, L_x,$$

located at $y = L_y/2$. For each of these stations 5 points in the vertical (z) direction at,

$$z = L_z/4 - 2\epsilon, L_z/4 - \epsilon, L_z/4, L_z/4 + \epsilon, L_z/4 + 2\epsilon,$$

are sampled. ϵ is presently chosen as 4 grid points.

14 Supplementary programs

Several IDL (Interactive Data Language, Research Systems Inc.) routines have been written to analyze and display results from `solwnd` simulations.

1. `solwnd.pro` reads the full field data. Depending on `mflag` it displays information about the max and min of sonic and Alfvénic Mach numbers. It is invoked by,

```
solwnd,ifile,mflag,time,nx,ny,nz,rho,p,vx,vy,vz,e,bx,by,bz,mach
```

`ifile` is the file name (including path) of the `solwnd` field data files. `e` is the total energy, internal + kinetic + magnetic. `mach` is the three-dimensional sonic mach number field. Rest of the variables have their usual meaning. All the fields are of size `(nx,ny,nz)`.

2. `solwnd_trc.pro` reads the file `case.trc1` and displays the max/min of sonic and Alfvénic Mach numbers, density and pressure. It is invoked by,

```
solwnd_trc,ifile,n,a
```

`ifile` is the file name (including path) of the `solwnd case.trc1` file. `n` is number of time step data records (`nsteps`). `a` is of dimension `(9,nsteps)`.

`a(0,*)` is t

`a(1,*)` is $\max M(x, y, z)$

`a(2,*)` is $\min M(x, y, z)$

`a(3,*)` is $\max M_A(x, y, z)$

a(4,*) is $\min M_A(x, y, z)$

a(5,*) is $\max \rho(x, y, z)$

a(6,*) is $\min \rho(x, y, z)$

a(7,*) is $\max p(x, y, z)$

a(8,*) is $\min p(x, y, z)$

Upon invocation, `solwnd_trc.pro` reads the trace file and then prompts the user for which variable is to be plotted in a composite max/min plot (where the plots for the Mach numbers also indicate the unity value) as follows,

```
iplot => 0 skip plot
iplot => 1 mach vs i
iplot => 2 mach vs t
iplot => 3 macha vs t
iplot => 4 rho vs t
iplot => 5 p vs t
```

3. `solwnd_trc2.pro` reads the file `case.trc2` (point data file) and plots the data vs. step or if "time" is available (from previous run of `solwnd_trc.pro` for instance) then plots the data vs. time. User is queried for what station (x location) and what height (z location) is plot of $\rho, p, V_x, V_y, V_z, B_x, B_y$, or B_z vs. time is desired. The routine cycles through choices until an exit is indicated with a negative value.
4. `solwnd_arrow.pro` reads the same data files as `solwnd.pro` but produces various arrow plots. It is invoked as,

```
solwnd_arrow,file,jcut,iplot,ires
```

`jcut` is the y -plane in which the $x-z$ view is desired. `ires` is 1, 2, 4 etc. indicating either full, half, or quarter resolution etc. The data is sub-sampled for drawing the arrow plots. For $n_x > 32$ the arrow plots are too crowded to see them as arrows, although interesting effects can still be produced. `iplot` indicates one of the following,

```
iplot => 1 ( $B_x - B_0, B_z$ ); ( $B_x, 0$ ) in  $x - z$  plane (2 plots/page)
```

```
iplot => 2 ( $0, B_z$ ); ( $0, B_y$ ) in  $x - z$  plane (2 plots/page)
```


iplot => 3 ($V_x - V_0, V_z$); ($V_x, 0$) in $x - z$ plane (2 plots/page)
iplot => 4 ($0, V_z$); ($0, V_y$) in $x - z$ plane (2 plots/page)
iplot => 5 (V_y, V_z); ($V_y, 0$); ($0, V_z$) in $x - z$ plane (3 plots/page)
iplot => 6 (B_y, B_z); ($B_y, 0$); ($0, B_z$) in $x - z$ plane (3 plots/page)

Acknowledgements

Thanks are due to S. Zalesak and D. Spicer (GSFC/934) for the use of their original code. Thanks are due to A. Roberts and M. Goldstein (GSFC/692) for suggesting the problem and the inflow boundary conditions, and for reading and commenting on the manuscript. This work is supported by the NASA Goddard HPCC-ESS Project through grant NAG5-2652 to George Mason University.

References

- [1] Zalesak, S. and Spicer, D., unpublished.
- [2] Boris, J. P. and Book, D. L., *J. Comput. Phys.* **11**, 38 (1973).
- [3] Zalesak, S., *J. Comput. Phys.* **31**, 335 (1979).
- [4] DeVore, C. R., *J. Comput. Phys.* **92**, 142 (1991).
- [5] Deane, A., Zalesak, S., and Spicer, D., 3 D compressible hydrodynamics using Flux Corrected Transport on message passing parallel computers, in *High Performance Computing, 1995*, edited by Tentner, A., pages 128–133, 1995.
- [6] Jackson, J., *Classical Electrodynamics*, Wiley, 1975.
- [7] Priest, E. R., *Solar Magnetohydrodynamics*, D. Reidel, 1982.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE May 1996	3. REPORT TYPE AND DATES COVERED Contractor Report	
4. TITLE AND SUBTITLE Solwnd: A 3D Compressible MHD Code for Solar Wind Studies. Version 1.0: Cartesian Coordinates			5. FUNDING NUMBERS Code 934 NAG5-2652	
6. AUTHOR(S) Anil E. Deane				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) George Mason University Fairfax, VA 22030-4444			8. PERFORMING ORGANIZATION REPORT NUMBER 96B00077	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Goddard Space Flight Center National Aeronautics and Space Administration Washington, DC 20546-0001			10. SPONSORING/MONITORING AGENCY REPORT NUMBER CR-4725	
11. SUPPLEMENTARY NOTES Anil E. Deane: George Mason University, Fairfax, Virginia				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified-Unlimited Subject Category: 75 This report is available from the NASA Center for Aerospace Information, 800 Elkridge Landing Road, Linthicum Heights, MD 21090; (301) 621-0390			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) solwnd 1.0 is a three-dimensional compressible MHD code written in Fortran for studying the solar wind. Time-dependent boundary conditions are available. The computational algorithm is based on Flux Corrected Transport and the code is based on the existing code of Zalesak and Spicer. The flow considered is that of shear flow with incoming flow that perturbs this base flow. Several test cases corresponding to pressure balanced magnetic structures with velocity shear flow and various inflows including Alfvén waves are presented. Version 1.0 of solwnd considers a rectangular Cartesian geometry. Future versions of solwnd will consider a spherical geometry. Some discussions of this issue is presented.				
14. SUBJECT TERMS magnetohydrodynamics, solar wind, compressible flow			15. NUMBER OF PAGES 34	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT Unlimited	