ENLIL: A Numerical Code for Solar Wind Disturbances

Dusan Odstrcil

June 11, 2004

Contents

1 Introduction

ENLIL is a three-dimensional magnetohydrodynamic code which calculates the time-dependent behavior of an ideal fluid due to various initial and boundary conditions with or without an external gravitational field taken into account. This ideal magneto-fluid dynamics can simulate the global behavior of a gas or plasma in situations when microscopic processes are negligible. The code enables to solve problems with shocks in either Cartesian, cylindrical, or spherical geometry.

The ENLIL code can be understood from a one-to-one correspondence between the listing of the code and this document. The listing refers to the equations of this document which in turn refers to modules of the code. The OLYMPUS notation is adopted here, e.g. `<2.3>` means subprogram 3 of class 2. There is similar correspondence for variable and array names.

The document describes the physical model, method of solution, initial and boundary, numerical control, structure of the code, input and output, and instructions for the user. Test problems, heliospheric problems, and file structure and conventions are described in separate documents.

2 Mathematical Description

This section specifies the mathematical description of time-dependent magnetohydrodynamic problems.

2.1 Fluid Description

The ideal fluid approximation is used and microscopic processes are neglected.

A fully ionized hydrogen gas with equal electron and proton densities ($n = n_e = n_p$) and equal electron and proton temperatures ($T = T_e = T_p$) is assumed. Properties of this fluid are further characterized by the mean ratio of specific heats γ and the magnetic permeability μ_o [1.2566×10^{-6} H.m^{-1}]. State of the fluid is described by the following macroscopic parameters: mass density ρ [kg.m^{-3}], velocity $\mathbf{V}(V_r, V_\vartheta, V_\varphi)$ [m.s^{-1}], pressure p [N.m^2], temperature T [K], total energy density U [J.m^{-3}] and magnetic field induction $\mathbf{B}(B_r, B_\vartheta, B_\varphi)$ [T]. The external gravitation field is considered and characterized by mass of the Sun M_S [1.991×10^{30} kg] as well as by the gravitational constant G [6.670×10^{-11} N.m^2.kg^{-2}].

2.2 Basic MHD Model

The MHD model describes the fluid dynamics so that all macroscopic parameters are functions of the radial position r [m], meridional angle θ [rad], azimuthal angle ϕ [rad] and time t [s]. This model can be expressed in various mathematical forms, however, the conservative form is preferred for the numerical solution. This means that the equations explicitly represents the conservation of mass, momentum, total energy and induction of magnetic field. Further, it is suitable to express an effect of the magnetic field as effects of the magnetic field pressure and tension of magnetic field lines. Using the vector operator ∇, we have the following system of equations (e.g. [?]):
equation of continuity

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

equation of motion

$$\frac{\partial}{\partial t}(\rho \mathbf{V}) + \nabla \cdot (\rho \mathbf{V}\mathbf{V}) = -\nabla(P) + \nabla \cdot \left(\frac{\mathbf{B}\mathbf{B}}{\mu}\right) + \frac{\rho G M_s}{r^2} \ , \tag{2}$$

equation of total energy

$$\frac{\partial}{\partial t}(U) + \nabla \cdot (U \mathbf{V}) = -\nabla \cdot (P \mathbf{V}) + \nabla \cdot \left(\frac{\mathbf{B}\mathbf{B}}{\mu}\mathbf{V}\right) + \frac{\rho \mathbf{V} G M_s}{r^2} \ , \tag{3}$$

equation of magnetic field

$$\frac{\partial}{\partial t}(\mathbf{B}) = \nabla \times (\mathbf{V} \times \mathbf{B}) \ , \tag{4}$$

where

$$P = p + \frac{\mathbf{B}^2}{2\mu} \tag{5}$$

is the total pressure, a sum of the thermal and magnetic pressure.

This system of equations is closed by the expression for the total energy density, as sum of the thermal, kinetic, and magnetic energy densities

$$U = \frac{p}{\gamma - 1} + \frac{\rho \mathbf{V}^2}{2} + \frac{\mathbf{B}^2}{2\mu} \ , \tag{6}$$

which serves for determination of the thermal pressure. The mean temperature T $[K]$ is determined from the equation of state for two-fluid (protons and electrons) single-temperature plasma

$$p = 2nkT \ , \tag{7}$$

where $k = 1.38044 \times 10^{-23}$ $[\mathrm{J.K}^{-1}]$ is the Boltzmann constant. Finally, the number density n $[\mathrm{m}^{-3}]$ is determined from the relation

$$\rho = mn \ , \tag{8}$$

where $m = 1.6733 \times 10^{-27}$ kg is the sum of the mass of proton and electron.

2.3 Tracking Equations

Two additional continuity equations are solved simultaneously,

$$\frac{\partial}{\partial t}\left(\rho_c\right) + \nabla \cdot \left(\rho_c \mathbf{V}\right) = 0 \ ,$$

$$\frac{\partial}{\partial t}\left(\rho_p\right) + \nabla \cdot \left(\rho_p \mathbf{V}\right) = 0 \ ,$$

where ρ_c and ρ_p are quantities used to trace the injected CME material and the magnetic field polarity, respectively. The IMF in interplanetary space is frozen in the solar wind plasma, and the IMF polarity is passively advected by solar wind flow. The tracking of the IMF polarity by marker particles, ρ_p, is thus the same as the tracking of injected mass by marker particles, ρ_c, the only difference being that a different set of markers is traced in each case.

2.4 Specific MHD Model

The time-dependent MHD model written in conservative form and specified for the spherical coordinate system (r, θ, φ) consists of the following equations:
equation of continuity

$$\frac{\partial}{\partial t}\left(\rho\right) + \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 \rho V_r\right) + \frac{1}{r\sin\theta}\frac{\partial}{\partial \theta}\left(\sin\theta \rho V_\vartheta\right) + \frac{1}{r}\frac{\partial}{\partial \varphi}\left(\rho V_\varphi\right) = 0 \ , \tag{9}$$

equation of motion

$$\begin{aligned} \frac{\partial}{\partial t}\left(\rho V_r\right) + \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 \rho {V_r}^2\right) + \frac{1}{r\sin\theta}\frac{\partial}{\partial \theta}\left(\sin\theta \rho V_r V_\vartheta\right) + \frac{1}{r\sin\theta}\frac{\partial}{\partial \varphi}\left(\rho V_r V_\varphi\right) = \frac{\rho {V_\vartheta}^2 + \rho {V_\varphi}^2}{r} \\ -\frac{\partial}{\partial r}(P) - \frac{\rho G M_s}{r^2} + B_r \frac{\partial}{\partial r}\left(\frac{B_r}{\mu}\right) + \frac{1}{r}\frac{\partial}{\partial \theta}\left(\frac{B_r B_\vartheta}{\mu}\right) + B_\varphi \frac{\partial}{\partial \varphi}\left(\frac{B_r}{\mu}\right) - \frac{{B_\vartheta}^2 + {B_\varphi}^2}{\mu r} \end{aligned} \quad , \tag{10}$$

$$\begin{aligned} \frac{\partial}{\partial t}\left(\rho V_\vartheta\right) + \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 \rho V_\vartheta V_r\right) + \frac{1}{r\sin\theta}\frac{\partial}{\partial \theta}\left(\sin\theta \rho {V_\vartheta}^2\right) + \frac{1}{r\sin\theta}\frac{\partial}{\partial \varphi}\left(\rho V_\vartheta V_\varphi\right) = -\frac{\rho V_r V_\vartheta}{r} + \frac{\rho {V_\varphi}^2 \cot\theta}{r} \\ -\frac{1}{r}\frac{\partial}{\partial \theta}(P) + \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 \frac{B_\vartheta B_r}{\mu}\right) + \frac{1}{r\sin\theta}\frac{\partial}{\partial \theta}\left(\frac{\sin\theta {B_\vartheta}^2}{\mu}\right) + \frac{1}{r\sin\theta}\frac{\partial}{\partial \varphi}\left(\frac{B_\vartheta B_\varphi}{\mu}\right) = +\frac{B_r B_\vartheta}{\mu r} - \frac{{B_\varphi}^2 \cot\theta}{\mu r} \end{aligned} \tag{11}$$

$$\frac{\partial}{\partial t}\left(\rho V_\varphi\right)+\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\rho V_\varphi V_r\right)+\frac{1}{r\mathrm{sin}\theta}\frac{\partial}{\partial\theta}\left(\mathrm{sin}\theta\rho V_\varphi V_\vartheta\right)=+\frac{1}{r\mathrm{sin}\theta}\frac{\partial}{\partial\varphi}\left(\rho {V_\varphi}^2\right)=-\frac{\rho V_r V_\varphi}{r}-\frac{\rho V_\vartheta V_\varphi\cot\theta}{r}$$
$$-\frac{1}{r}\frac{\partial}{\partial\varphi}(P)+B_r\frac{\partial}{\partial r}\left(\frac{B_\varphi}{\mu}\right)+\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{B_\varphi B_r}{\mu}\right)+\frac{1}{r}\frac{\partial}{\partial\theta}\left(\frac{B_\varphi B_\vartheta}{\mu}\right)+B_\varphi\frac{1}{r}\frac{\partial}{\partial\varphi}\left(\frac{B_\varphi}{\mu}\right)+\frac{B_r B_\varphi}{\mu r}+\frac{B_\vartheta B_\varphi\cot\theta}{\mu r} \quad (12)$$

equation of total energy density

$$\frac{\partial}{\partial t}(U)+\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2UV_r\right)+\frac{1}{r\mathrm{sin}\theta}\frac{\partial}{\partial\theta}\left(\mathrm{sin}\theta UV_\vartheta\right)+\frac{1}{r\mathrm{sin}\theta}\frac{\partial}{\partial\varphi}\left(UV_\varphi\right)=$$
$$-\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2PV_r\right)-\frac{1}{r\mathrm{sin}\theta}\frac{\partial}{\partial\theta}\left(\mathrm{sin}\theta PV_\vartheta\right)-\frac{1}{r\mathrm{sin}\theta}\frac{\partial}{\partial\varphi}\left(PV_\varphi\right)-\frac{\rho V_r GM_s}{r^2}$$
$$+\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{B_r}{\mu}\left(B_rV_r+B_\vartheta V_\vartheta+B_\varphi V_\varphi\right)\right)+\frac{1}{r\mathrm{sin}\theta}\frac{\partial}{\partial\theta}\left(\mathrm{sin}\theta\frac{B_\vartheta}{\mu}\left(B_rV_r+B_\vartheta V_\vartheta+B_\varphi V_\varphi\right)\right)$$
$$+\frac{1}{r\mathrm{sin}\theta}\frac{\partial}{\partial\varphi}\left(\frac{B_\varphi}{\mu}\left(B_rV_r+B_\vartheta V_\vartheta+B_\varphi V_\varphi\right)\right) \quad , \quad (13)$$

equation of magnetic field induction

$$\frac{\partial}{\partial t}\left(B_r\right)+\frac{1}{r\mathrm{sin}\theta}\frac{\partial}{\partial\theta}\left(\mathrm{sin}\theta B_rV_\vartheta\right)+\frac{1}{r\mathrm{sin}\theta}\frac{\partial}{\partial\varphi}\left(B_rV_\varphi\right)=\frac{1}{r\mathrm{sin}\theta}\frac{\partial}{\partial\theta}\left(\mathrm{sin}\theta B_\vartheta V_r\right)+\frac{1}{r\mathrm{sin}\theta}\frac{\partial}{\partial\varphi}\left(B_\varphi V_r\right) \ , \quad (14)$$

$$\frac{\partial}{\partial t}\left(B_\vartheta\right)+\frac{1}{r}\frac{\partial}{\partial r}\left(rB_\vartheta V_r\right)+\frac{1}{r\mathrm{sin}\theta}\frac{\partial}{\partial\varphi}\left(B_\vartheta V_\varphi\right)=\frac{1}{r}\frac{\partial}{\partial r}\left(rB_rV_\vartheta\right)+\frac{1}{r\mathrm{sin}\theta}\frac{\partial}{\partial\varphi}\left(B_\varphi V_\vartheta\right) \ , \quad (15)$$

$$\frac{\partial}{\partial t}\left(B_\varphi\right)+\frac{1}{r}\frac{\partial}{\partial r}\left(rB_\varphi V_r\right)+\frac{1}{r}\frac{\partial}{\partial\theta}\left(B_\varphi V_\vartheta\right)=\frac{1}{r}\frac{\partial}{\partial r}\left(rB_rV_\varphi\right)+\frac{1}{r}\frac{\partial}{\partial\theta}\left(B_\vartheta V_\varphi\right) \ , \quad (16)$$

where

$$P=p+\frac{{B_r}^2+{B_\vartheta}^2+{B_\varphi}^2}{2\mu} \quad (17)$$

is the total pressure and

$$U=\frac{p}{\gamma-1}+\frac{1}{2}\rho({V_r}^2+{V_\vartheta}^2+{V_\varphi}^2)+\frac{{B_r}^2+{B_\vartheta}^2+{B_\varphi}^2}{2\mu} \quad (18)$$

is the total energy density. Thus we have to solve the system of eight coupled, nonlinear, partial differential equations.

3 Method of Solution

Here we present a description of a TVD algorithms in 1D on a stationary uniform grid with the space variable denoted by x.

In the following descriptions the value of the discretized conservative variable U_j^n is defined on the mesh at the discrete time level t^n as a volume average within the j-th mesh cell centered at position x_j. The cell interfaces are at $x_{j-1/2}$ and $x_{j+1/2}$, where $x_{j+1/2}=(x_j+x_{j+1})/2$. We shall also use the notation $\Delta U_{j+1/2}=U_{j+1}-U_j$ for the difference of variables in adjacent cells. The subscript $j+1/2$ will always refer to a quantity centered on the cell interface at $x_{j+1/2}$, while a time centered quantity will be denoted by a superscript $n+1/2$.

A second-order time accuracy, the one-step forward Euler time-discretization shall be replaced by the two-step time-discretization.

3.1 Predictor

The first (half) step, the predictor is used to compute values at $t^{n+1/2}$. Hancock (unpublished; see Yee (1989)) suggested the following predictor scheme

$$U_j^{n+1/2}=U_j^n-\frac{1}{2}\frac{\Delta t}{\Delta x}\left[F\left(U_j^n+\frac{1}{2}\tilde{\Delta}U_j^n\right)-F\left(U_j^n-\frac{1}{2}\tilde{\Delta}U_j^n\right)\right]+\frac{\Delta t}{2}S_j^n \ . \quad (19)$$

where $\tilde{\Delta}U_j^n$ is the limited difference (slope) as described later.

3.2 Corrector

The second-step (full step, corrector)

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x}\left(H_{j+1/2}^{n+1/2} - H_{j-1/2}^{n+1/2}\right) + \Delta t S_j^{n+1/2} \tag{20}$$

where time-centered numerical fluxes are

$$H_{j+1/2}^{n+1/2} = \frac{1}{2}\left[F\left(U_{j+1/2}^R\right) + F\left(U_{j+1/2}^L\right) - C_{j+1/2}^{n+1/2}\left(U_{j+1/2}^R - U_{j+1/2}^L\right)\right] \quad , \tag{21}$$

where the values at the cell interfaces are specified as follows.

Hancock (unpublished; see Yee (1989)) used

$$U_{j+1/2}^L \;=\; U_j^{n+1/2} + \frac{1}{2}\tilde{\Delta}U_j^n \quad , \tag{22}$$

$$U_{j+1/2}^R \;=\; U_{j+1}^{n+1/2} - \frac{1}{2}\tilde{\Delta}U_{j+1}^n \quad . \tag{23}$$

where $\tilde{\Delta}U_j^n$ is the limited difference (slope) as described later.

3.3 Maximum Characteristic Speed

Barmin and Pogorelov (1995) used

$$R_{j+1/2} = C_{j+1/2}^{max} \quad , \tag{24}$$

where $C_{j+1/2}^{max}$ is the spectral radius of the Jacobian matrix $\partial F/\partial U$, i.e., the maximum characteristic speed in the given system. For the MHD system of equations, this speed is

$$C^{max} = |V| + C^f = |V| + \frac{1}{2}\left(\frac{\gamma p + \mathbf{B}^2}{\rho} + \sqrt{\left(\frac{\gamma + \mathbf{B}}{\rho}\right)^2 - 4\frac{\gamma B_x^2}{\rho^2}}\right) \tag{25}$$

where V is the flow velocity, C^F is the fast-mode MHD wave, C^s is the sound velocity, and $C^A =$ is the Alfven velocity.

3.4 Slope Limiters

The slopes of variables ΔU are used to compute values of variables at the cell interfaces. The TVD property is ensured by the procedure that limits these slopes of variables. The following limiters are used:

minmod limiter

$$\tilde{\Delta}U_j = Z_{j-1/2}\max\left[0, \min\left(|\Delta U_{j-1/2}|, Z_{j-1/2}\Delta U_{j+1/2}\right)\right] \quad , \tag{26}$$

Woodward and Colella's limiter

$$\tilde{\Delta}U_j = Z_{j-1/2}\max\left[0, \min\left(|2\Delta U_{j-1/2}|, Z_{j-1/2}2\Delta U_{j+1/2}, Z_{j-1/2}\frac{1}{2}\left(\Delta U_{j+1/2} + \Delta U_{j-1/2}\right)\right)\right] \quad , \tag{27}$$

where the $Z_{j-1/2} = \operatorname{sgn}\left(\Delta U_{j-1/2}\right)$

3.5 Multi-Dimensional Problems

The easiest way to generalize a 1D scheme to multidimensional problems is via a Strang-type [?] operator splitting, e.g. in 2D

$$U^{n+1} = L_x^{\Delta t/2} L_y^{\Delta t} L_x^{\Delta t/2} U^n \tag{28}$$

where L_x and L_y are the appropriate 1D operators for a given time step with spatial derivatives taken in the x and y directions respectively. A somewhat faster and usually satisfactory method is alternating the order of L_x and L_y with every time step, thus

$$U^{n+2} = L_x L_y U^{n+1} = L_x L_y L_y L_x U^n \tag{29}$$

3.6 Div(B) Treatment

$$(B_r)_j^{n+1} - (B_r)_j^n + + \frac{1}{r\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta B_r V_\theta\right) + \frac{1}{r\sin\theta}\frac{\partial}{\partial\varphi}\left(B_r V_\varphi\right) = \frac{1}{r\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta B_\theta V_r\right) + \frac{1}{r\sin\theta}\frac{\partial}{\partial\varphi}\left(B_\varphi V_r\right) \ , \quad (30)$$

$$(B_\theta)_j^{n+1} - (B_\theta)_j^n + + \frac{1}{r}\frac{\partial}{\partial r}\left(rB_\theta V_r\right) + \frac{1}{r\sin\theta}\frac{\partial}{\partial\varphi}\left(B_\theta V_\varphi\right) = \frac{1}{r}\frac{\partial}{\partial r}\left(rB_r V_\theta\right) + \frac{1}{r\sin\theta}\frac{\partial}{\partial\varphi}\left(B_\varphi V_\theta\right) \ , \quad (31)$$

$$(B_\varphi)_j^{n+1} - (B_\varphi)_j^n + + \frac{1}{r}\frac{\partial}{\partial r}\left(rB_\varphi V_r\right) + \frac{1}{r}\frac{\partial}{\partial\theta}\left(B_\varphi V_\theta\right) = \frac{1}{r}\frac{\partial}{\partial r}\left(rB_r V_\varphi\right) + \frac{1}{r}\frac{\partial}{\partial\theta}\left(B_\theta V_\varphi\right) \ , \quad (32)$$

4 Initial and Boundary Conditions

4.1 Initial Conditions

Values of relevant MHD variables at the each computational cell are to be known at the begining of computations. These values define initial conditions for a given run.

If such values are known they are stored at `ini.nc` file. Note that this file contains arrays of values wih shape and dimensions specified also at `grd.nc` file. These parameters has to be identical with the code internal array dimensions.

If such values are unknown, then values at the inner boundary `bnd.bc` has to be specified. Values in the computational region are then extrapolated from bounadary values corresponding to beginning of computations in subroutine `INBND <2.12>`.

$$\begin{aligned} \rho &= \rho^0(1/r)^2 \\ T &= T^0(1/r)^2 \\ V_r &= V_r^0 \\ B_r &= B_r^0(1/r)^2 \\ B_\varphi &= B_\varphi^0(1/r)^2 \qquad . \end{aligned} \quad (33)$$

Note that these values are ad-hoc values that are not consistent with physical conditions. Also div(B). Code robust for most cases

Relaxation. Time. e.e., 5 days from Sun to Earth. It is recommended to `tstart < 0` and at t=0 for sure. (see `namelist NAMRUN` parameters).

4.2 Boundary Conditions

There are four external boundaries of the computational region as shown in Fig. 2. The numerical schemes used require to know the values of the main variables at these boundaries, and this knowledge is required at each time step before the solution of the MHD system of equations. In the ENLIL code, it is possible to specify boundary conditions corresponding to a free flow, solid wall, periodic system or constant values. It is also possible to specify prescribed time-dependent values at the left boundary using standard functions.

These different boundary conditions are selected during program compilation. Table 1 gives specification of the left boundary condition by the parameter NBCL. Specifications of the other boundary conditions are similar.

4.3 Standard Conditions

NBCL	ρ_1	description
1	ρ_2	zero-order extrapolation
2	$2\rho_2 - \rho_3$	first-order extrapolation
3	ρ_{n-1}	periodic conditions
4	$\rho_1(t)$	constant or time-dependent values

4.4 Time-Dependent Values

5 Numerical Control

5.1 Timestep Control

The time step must satisfy the condition of stability for explicit difference schemes and the required accuracy of the calculation. The ENLIL code can be run with a constant or a variable value of the time step. The choice between these two cases is provided by the logical variable NLTIMC. Its preset value is .TRUE., i.e. the calculation will be performed with a variable time step which is specified for each step by the code itself. If we set NLTIMC=.FALSE., then the calculation will be performed with a constant time step which may be specified by the user.

The time-step control is performed in subroutine TIMSTP <4.6> and occurs after the required equations ??? have solved for the quantities $Q(= \rho, v, U, B)$ from step n to step $n+1$.

The condition of stability is the Courant-Friedtrichs-Lewy condition ?? which restricts the time-step value to

$$\mathtt{DT} \leq \mathtt{AKCDF} \left(\frac{\Delta x}{|v| + c} \right)$$

where $\mathtt{AKCL}<1$ is a constant, and C is the maximum characteristic speed.

The accuracy of the calculation restricts the time-step value by the rate of change of the main physical variables during the current step

$$\mathtt{DT} < |\mathtt{AKQ}| \left(\frac{Q^n}{Q^{n+1} - Q^n} \right) \mathtt{DTLAST}$$

where AKQ is a constant which determines the permissible relative changes in variable Q (=ρ, v, U, B) and DTLAST is the previous time-step value.

The rate of increase or decrease of DT itself is restricted by the conditions

$$\mathtt{DT} \leq \mathtt{AKDT} * \mathtt{DTLAST}$$

$$\mathtt{DT} \leq \frac{1}{\mathtt{AKDT}} \mathtt{DTLAST}$$

where $\mathtt{AKDT} > 1$ is a constant.

Finally, we demand that DT should lie within the prescribed limits

$$\mathtt{DTMIN} \leq \mathtt{DT} \leq \mathtt{DTMAX}$$

5.2 Examination and Diagnostics

The examination and diagnostics part of the ENLIL code consists of a check of the values of the main variables and of a check of mass and total energy conservation.

5.3 Subroutine EXPERT

Subroutine EXPERT <0.4> is called from many points throughout the code with 3 arguments defining the location from which the call is made. Normally a standard version is loaded, but the user can provide his own version, containing the additions or modifications for the requested locations ??. EXPERT must not call subroutines which themselves call EXPERT in order to avoid recursive calling.

6 Structure of the Code

The structure of ENLIL is based on an updated version of the OLYMPUS system. The ENLIL code comprises of a main program, 88 subprograms, and 8 common blocks. The subprograms are divided into classes and module INDSUB (see listing) lists all subprograms used by the code. SImilarly common blocks are divided into groups and module INDCOM (see listing) lists all common blocks used. The organization of the subprogram structure is shown in Figures 1, 2, 3, and 4.

Table 1: List of Groups

Group	Purpose
Group 0	Administration
Group 9	Boundary Conditions
Group 9	Temporal Evolution and Diagnostics
Group 9	Global Physical Variables
Group 9	Method of Solution
Group 9	Output Control Parameters
Group 9	Physical Parameters
Group 9	Computational Region
Group 9	Total-variation-Diminishing Scheme
Group O	Olympus system

Table 2: List of Common Blocks

Name	Purpose
`comdim`	Array dimensions
`compro`	Program labels
`comadm`	Buffer arrays
`combnd`	Boundary conditions
`comevo`	Values at observing points
`comglo`	Global MHD variables
`commet`	Numerical control
`comout`	Output control
`comphy`	Physical constants
`comphy`	Parameters defining a run
`comreg`	Computational region
`comtva`	Mesh parameters

6.1 Data

6.2 Instructions

7 Run Control

8 Input Parameters

The input parameters required for proper run of the code are in one file which is read by a standard input channel. The input parameters file consist of three parts:

- label
- `namelist` NAMJOB
- `namelist` NAMRUN

Label is a brief characterization of the run on one line, maximum 80 characters.

The `namelist` NAMJOB contains parameters describing the input and out file names as given in Table tab-namjob. The `namelist` NAMRUN contains physical, numerical, and output control paramters as given in Table tab-namrun.

These parameters control the caluclations and are subjected to user specification. Module docnam (see listing) lists them together with their default values which are assigned to some parameters in subroutine `RUNPRE <1.03>` and with bounds on input values which are checked in subroutine `RUNAUX <1.05>`.

Table 3: List of Classes

Name	Purpose
Class 0	Main Control
Class 1	Prologue
Class 2	Numerical Grid
Class 3	Initial Conditions
Class 4	Boundary Conditions
Class 5	Output
Class 6	Computational Control
Class 7	Magnetohydrodynamic Flow
Class 8	Auxiliary Procedures
Class 9	Utilities

8.1 Input Parameters

The input parameters file `in` file uses the following naming convention:

<*run*>`.in`,

that is equal to (see Figure fig-names):

<*case*>.<*PEs*>-<*model*>.<*parameters*>`.in`,

This means that the <*parameters*> name is to be specified by the user as a combination of 1-8 alpha-numeric characters. Note that the `in` file is placed in the <*run*> directory since it defines a particular computational run for the given <*case*> and <*PEs*>-<*model*>.

Input parameters file contains specification of file names, start and stop times of the computation, frequency of various output files, as well as some physical and numerical parameters of the <*run*>. It s a text (ASCII) file and it has the following structure:
title line
`&namjob`
namelist commands
`/`
`&namrun`
namelist commands
`/`
as detailed below.

8.2 Namelist NAMJOB

The user has to specify values of the following parameters:

`ldir`	Full name of directory (1-80 chars)
`lproj`	Project name (1-8 chars)
`lcode`	Name of <*code*> (1-8 chars)
`lgrd`	Name of <*grid*> (1-12 chars)
`lbgrd`	Name of <*boundary grid*> (1-12 chars)
`lres`	Name of restart values (1-8 chars)
`lini`	Name of *initial values* (1-8 chars)
`lbnd`	Name of *boundary values* (1-8 chars)
`lrun`	Name of *parameters* (1-8 chars)

8.3 Namelist NAMRUN

The user has to specify values of the following parameters:

Table 4: List of Subprograms

Name	No.	Purpose
MASTER	0.1	Main control
JOBSET	0.2	Preset the JOB parameters
JOBCLR	0.3	Clear variables and arrays
JOBPAR	0.4	Read in the JOB parameters
JOBAUX	0.5	Set the JOB auxiliary values
JOBCHE	0.6	Check the JOB parameters
JOBEND	0.7	Terminate the JOB
PARAMS	0.8	Write the code parameters
EXPERT	0.9	Write the code parameters
DIAGNO(2)	0.10	Write the code parameters
VTSETUP	0.11	Setup the VAMPIR trace instrumentation
RUNCLR	1.1	Clear variables and arrays
RUNPRE	1.2	Set default variables
RUNPAR	1.3	Define data specific to run
RUNCHE	1.4	Check input data
RUNAUX	1.5	Set auxiliary values
INPOS	1.6	Find mesh positions of the observing points
INEVO	1.7	Find mesh positions of the observing points
GRID	2.1	Set grid coordinates
GRID1	2.2	Set X1-grid variables and coefficients
GRID2	2.3	Set X2-grid variables and coefficients
GRID3	2.4	Set X3-grid variables and coefficients
RUNINI	3.1	Start the calculation
INDT	3.2	Set the initial time step
BNDINI	3.3	Set the initial time step
RESINI	3.3	Resume at time level
RESRES	3.3	Resume at time level
RESBND	3.3	Resume at time level
RESPAS	3.3	Resume at time level
RESGRD	3.3	Resume at time level
RUNOUT	5.1	Control the output
OUTDIA	5.2	Output of diagnostics
OUTFIN	5.3	Output at time levels
OUTTIM	5.4	Output of diagnostics
OUTEVO	5.5	Output of diagnostics
OUTEVL	5.6	Output of diagnostics
OUTEVG	5.7	Output of diagnostics
OUTVAL(1)	5.8	Output of diagnostics
OUTX1S	5.9	Output of diagnostics
OUTX2S	5.10	Output of diagnostics
OUTX3S	5.11	Output of diagnostics

Table 5: List of Subprograms (cnt.)

Name	No.	Purpose
RUNCAL	4.1	Step on the calculation
TIMSTP	4.2	Timestep control
PASVAL	4.3	Save evolution in time
RUNTST	4.4	Test for completion of run
SIGVAL	4.5	Save evolution in time
SELVAL	4.6	Save evolution in time
GEOVAL	4.7	Save evolution in time
BND1	5.1	X1-boundary conditions
BND2	5.2	X2-boundary conditions
BND3	5.3	X3-boundary conditions
BNDRDR	5.4	Read in boundary values
BNDMAP	5.5	Map boundary positions
BNDIN0	5.6	Interpolate boundary values
BNDIN1	5.7	Interpolate boundary values
BNDIN2	5.8	Interpolate boundary values
BNDMPI	5.9	Interpolate boundary values
MHC	6.1	Interpolate boundary values
MHD1	6.2	Interpolate boundary values
MHD2	6.3	Interpolate boundary values
MHD3	6.4	Interpolate boundary values
MHS	6.5	Interpolate boundary values
TOTH1	6.6	Interpolate boundary values
TOTH2	6.7	Interpolate boundary values
POWELL	6.8	Interpolate boundary values

tstart Start computations at this time
tstop Stop computations at this time
trfrom RES output from this time
trto RES output to this time
trstep RES output with this step in time
ttfrom TIM output from this time
ttto TIM output to this time
ttstep TIM output with this step in time
tsfrom X*S output(s) from this time
tsto X*S output(s) to this time
tsstep X*S output(s) with this step in time
tefrom EV* output(s) from this time
teto EV* output(s) to this time
x1hel X1-position(s) of the observing point(s) in heliosphere
x2hel X2-position(s) of the observing point(s) in heliosphere
x3hel X3-position(s) of the observing point(s) in heliosphere
x1sel X1-position(s) of the observing point(s) on Sun-Earth line
gamma Ratio of specific heats (1.0<gamma≤2.0)
vrot Rotation speed of the inner boundary (0.,25.,27)
akcfl Max. value of the CFL stability number (0.1≤akcfl≤0.9)
dtzero Initial value of the timestep
dtmin Minimum value of the timestep
dtmax Maximum value of the timestep
nltimc = .true. if timestep control

8.4 Output Log

The output of examination and diagnostics provided for the running control of the calculation during the execution and is produced by the subroutine `OUTLOG`. The output is preset to a terminal display. The frequency of this output is given by the value `NLOG`.

The main output of results served for displaying results of the calculation. The output consists of five parts:

- copy of the input data
- initial conditions
- periodic output
- final output
- terminal output

8.5 Output Log

Output log is an ASCII file with the following naming convention:

<*run*>.out

(see Figure fig-names for specification of <*run*>). It contains:

- date and time
- information on options selected by the user during compilation of the <*model*>;
- information on input <*parameters*> provided by the user (a copy of `in` file);
- record of the computation progress (step number, physical time, time step, stability condition, as well as which data files are read in and written out).

9 Instructions for the User

9.1 Subroutine EXPERT

Subroutine EXPERT ¡0.04¿ is called from many points throughout the code, as can be senn from the listing, with 3 arguments defining the location from which the call is made. Normally a standard version is loaded, but the user can provide his/her own version, containing the additions or modifications for the requested locations Note that EXPERT must not call subroutines which themselves call EXPERT in order to avoid recursive calling.

10 Examination and diagnostics

The examination and diagnostics part f the ENLIL code consists of a check of the values of the main variables and of a check of total energy conservation. Subroutine `ENERGY <x.xx>` first calculates the distribution of total energy ehich consists of thermal energy, kinetic energy, and magnetic energy.

The sum of these energies.

11 References