A Rough Guide to the MAS Code

Pete Riley, Jon Linker, Roberto Lionello, Zoran Mikic, and J. Wijaya

Contents

List of Figures

List of Tables

Preface

This user guide is intended to provide a short and straightforward guide to installing and running Predictive Science's global, 3-D MHD code, MAS. We also provide a brief description of the physics contained in MAS as well as the numerical techniques employed. If you need more information than is contained here, please see the list of references in Appendix A.

0.1 How to use this Guide

You should be able to read this guide from cover to cover, following the installation instructions, basic configuration, running the test examples, and visualizing the output.

0.2 About Us

Zoran Mikic (mikicz@predsci.com) is the principal code developer, assisted by Roberto Lionello (lionel@predsci.com) and Jon Linker (linkerj@predsci.com). Pete Riley (pete@predsci.com) is involved primarily with running (and breaking) the code and managing visualization efforts and web-based tools for accessing the model results. Janvier Wijaya (wijayaj@predsci.com) developed and maintains our visualization applications and is involved in maintaining our modeling website.

0.3 Your Feedback

As a first edition, this guide likely contains a number of errors. We would really welcome your comments and suggestions on how to improve it. Please send them to pete@predsci.com . For your effort, we will send you a FREE electronic copy of the revised edition!

Chapter 1

Introduction

1.1 Acknowledgments

MAS is the collective effort of a number of people. In addition to the co-authors of this guide, who developed and maintain the current version of MAS, we have benefited from the help and guidance of several code developers, including Dalton Schnack at the University of Wisconsin.

1.2 MAS in a Nutshell

The MAS (Magnetohydrodynamics Around a Sphere) code has been developed to study the large-scale structure and dynamics of the solar corona and inner heliosphere. What makes the corona particularly difficult to study is that it exhibits a wide range of plasma parameters. Thus a successful model must calculate efficiently in these different regimes. This is exemplified by the wide disparity in the Alfvén time scale (τ_A), evolution time scale (τ_{evol}), and rotation time scale (τ_R): $\tau_A < \tau_{evol} < \tau_R$).

MAS, in one form or another, has been under development for ~ 17 years or so. It is built on a rich base of experience in computational physics and the modeling of solar coronal and fusion plasmas and has the following features:

- Time-dependent resistive MHD
- Incorporation of observed photospheric magnetic field data
- Evolution of boundary data

- Coronal and heliospheric components
- Non-uniform meshes (structured)
- 3D finite differences in spherical (r,θ,ϕ) coordinates
- Implicit and semi-implicit time differencing
- Comprehensive physics model including the solar wind and energy transport (radiation, parallel thermal conduction, heating, and Alfvn waves)
- Has been used to model CMEs
- Written in FORTRAN 90
- Designed to run on massively parallel computers using MPI: IBM/SP3 + SP4 (xlf); Linux & Beowulf (lf95, pgf90, Intel Fortran); Mac (Absoft and xlf); and SGI/Altix (ifort)
- Mesh decomposition among processors in 3D
- Dynamic allocation allows mesh size and number of processors to be selected at run time
- Restart capability using HDF files (for long runs)
- Many applications and comparisons with observational data (eclipses, IPS, in situ solar wind measurements, coronal holes, pB images, current sheet topology and spacecraft crossings, CMEs)
- A rich set of post-processing tools has been developed

The capabilities of the MAS model at the CCMC have been deliberately curtailed to minimize the chance of code failure. At present, the capabilities of MAS at CCMC are:

- Polytropic MHD model
- Driven by (filtered) Kitt Peak/SOLIS magnetic field maps
- Choice of low ($61 \times 51 \times 32$) and medium ($85 \times 81 \times 64$) resolution meshes
- Choice of base density and temperature in the corona
- Relaxation to steady state

- Increased viscosity and resistivity
- Runs on a single CPU

We are currently compiling and testing the fully parallel version of MAS at the CCMC (as part of our NASA Strategic Capability program and a collaboration with the NSF CISM program). Thus, in the near future, it is possible that both the serial and parallel versions will be available for users to run. In addition, we are continuing to develop MAS in-house. Thus, there are effectively three versions of the code: The old serial version, the new MPI production version, and the latest experimental version. In this document, we will focus on descriptions of the first two models, and, where appropriate, we will differentiate between which features or attributes are applicable to which model. If no distinction is made, then it can be assumed that the discussion refers to both models.

1.3 System Requirements

MAS can be compiled on any Linux/Unix-based machine. It will run on both 32-bit and 64-bit architectures. At an absolute minimum, a single-processor run can be completed (albiet after a very long time) on a system with at least 1 GB RAM and 500 MB of available disk space. However, for anything but test cases, the following requirements represent a reasonable minimum configuration for compiling and running MAS:

- 32-bit or 64-bit Intel, AMD, or IBM processor.
- Linux, Unix, or AIX operating system
- 1 GB RAM per node.
- 10 GB of available disk space.
- HDF and MPI Libraries, F90/95 Compiler

The precise system requirements will depend somewhat on the hardware and software. We currently compile and test MAS on: Mac Pro desktops and Beowulf Clusters containing 64 64-bit AMD Opteron processors (2 processors per node, and sharing 1 GB RAM). Our largest production runs are made on NASA's Columbia Supercomputer and NSF's Texas Ranger. We use, or have used: the Portland Group (PG) F90, Pathscale F95, Absoft, and Intel compilers.

1.4 Planned Improvements

MAS is constantly under development, and these improvements will be delivered to the CCMC as they are validated, both internally and through our participation in NASA's Strategic Capability andNSF's CISM programs. Some of our planned capabilities include the following:

- Add more flexibility to MAS at the CCMC:
  - Fractional Carrington rotations
  - Improved visualization and post-processing (interactive field lines, flying spacecraft trajectories through the model results)
  - Runs with evolution of boundary data
  - User-supplied magnetic field boundary data
  - Generation of higher-level data products
  - Implementation of the model with an improved energy equation, leading to a better solar wind model
- Implementation of the MPI version of MAS
- Coupling of the new MPI version of MAS to the heliospheric model, ENLIL

Additionally, we are currently investigating a number of physics advances, including the following:

- Desire to model fast CMEs ($\geq 1,000$ km/s), which requires stronger magnetic fields 2 kG and smaller length scales 100,000 km in active regions
- Desire to study specific CME events (e.g., May 12, 1997)
- Elements of CME initiation:
  - Initial equilibrium specification (p, ρ, B, solar wind)
  - Energization [e.g., shear flow profile vt(x,t), twist profile ?(x,t), eventually vector magnetograms]
  - Trigger mechanism [e.g., flux cancellation Br(x,t), or continued shear flows]

Chapter 2

Compiling and Running MAS

2.1 Introduction

In this chapter, we describe the basic steps required to compile, run, and restart the MAS code.

2.2 Compiling the Code

The code can be compiled as follows. Go to the source directory, and type:

```
make
```

which uses the file `Makefile` to compile the code. For this to work, it is necessary to have the proper shell environment variables defined to point to the HDF library (version 4) and to the MPI library, and to invoke the proper switches for the compiler. Typical environment variables that control these choices are defined in the CSH as follows:

```
LIBDF=-L/sw/lib -L/usr/local/HDF4.2r1/lib \
      -I/usr/local/HDF4.2r1/include \
      -lmfhdf -ldf -lsz -ljpeg -lz

OPENMPI=-I/usr/local/openmpi/include \
        -L/usr/local/openmpi/lib \
        -Wl,-u,_munmap -Wl,-multiply_defined,suppress \
        -lmpi_f90 -lmpi_f77 -lmpi -lopen-rte -lopen-pal \
```

```
F90=ifort

F90_OPTS=-O3 -assume byterecl -heap-arrays

LD_OPTS=-Wl,-stack_size,0x10000000
```

These particular choices are for the Intel Fortran compiler on Mac OS X with the OpenMPI library.

2.3 Running the Code

The most important parameters for the MAS code are set via the input file. We use a scheme in which a particular run is identified by a "run ID," which we call `<runid>` henceforth. This should be a short string, such as `cme01a`, that is a descriptive of the type of run being done. It is recommended that each run be performed in its own directory. It is strongly recommended that this run directory be named `<runid>`. The convention for the input and output files produced is:

`i<runid>`:	input file
`o<runid>`:	output file (generated)
`h<runid>###`:	time histories (generated)
`v<runid>###`:	additional time histories (generated)

where "`###`" are sequence numbers of the form `001, 002, ...`. The code also writes the fields (at specified time step intervals or time unit intervals) into HDF (version 4) machine-independent binary files. These files are named:

```
{br,bt,bp,vr,vt,vp,jr,jt,jp,rho,p,t}###.hdf
```

These are some of the basic fields of interest; additional quantities may be selected at run time.

The code can be executed as follows (in CSH):

```
nohup mpirun -n 32 mas [-timer mpi] <runid> >& <runid>.log &
```

This launches a parallel job on 32 processors, placing the job in the background, reading the input file `i<runid>`, assuming that the executable is named `mas`. The output from the screen is directed to the log file `<runid>.log`.

Typically, multi-user computer centers use a queueing system to submit jobs. The job submission details are site-depenedent, and are detailed elsewhere.

The code can be ended prematurely (i.e., before it reaches the requested execution time or number of time steps) by creating a file named `STOPRUN` in the run directory. This can be done by executing the UNIX command:

```
touch -a STOPRUN
```

which terminates the code gracefully at the next time step interval.

2.4 Restarting a Run

Typically, due to user time and queue limits at computer centers, a single run of the MAS code for a challenging set of parameters (e.g., high spatial resolution, long time evolution) may require more resources than are available for a single run of the code. In this case, we use the concept of "restarts." Namely, a simulation can be completed in several consecutive runs, with the code being "restarted" from a previous run to continue a simulation. For example, consider a sequence of runs, with run IDs `cme01a`, `cme01b`, and `cme01c`. These are continuations of a single simulation that is incrementally run out to larger end times. For example, run `cme01a` would be the initial run that simulated the corona from $t = 0$ to $t = 15.37$ (say) code time units. In this case, the input file of the first run, `icme01a`, would have the input variable setting

```
rsifile=' '
```

(i.e., the variable `rsifile` is set to a blank string) to indicate that this is the first run of the chain. The continuation run would have the variable `rsifile` set to point to the "restart file" produced by run `cme01a`, which contains the state of all variables and fields at the end of the first run. This restart file is named `rs<runid>.hdf`. Thus, the input file `icme01b` for the second run would contain:

```
rsifile='../cme01a/rscme01a.hdf'
```

which would continue run `cme01a` from $t = 15.37$ to $t = 33.14$ (say).

2.5 Post-Processing the Output Files

The MAS code produces a range of output files, including 3-D HDF files of all magnetofluid parameters (e.g., B_r, B_t, B_p, v_r, v_t, v_p, T, p, and ρ. Other parameters, such as the Alfven speed, V_A, or polarized brightness, pB, to name but a few, can also be computed using a set of command-line tools. Since the principal variables are on staggered grids, it is important to combine them carefully to produce new parameters. The MAS code also produces a sequence of time history files. The parameters may be local variables at a specific location within the simulation region, or volume-integrated averages. Run-related parameters, such as the time step, may be used to investigate the quality of the solutions.

2.6 Visualizing the Results

The HDF produced during the run can be read and visualized using most scientific visualization applications (e.g., IDL, MatLab, Mathematica). At our website (www.predsci.com), we provide several tools for viewing the data. These include: (1) An IDL routine to read the 3-D HDF files; and (2) a Mac/Linux visualization tool (called "View") for opening and displaying 3-D HDF files. The website also contains a set of online IDL-driven tools for viewing runs that are already stored within our database.

Chapter 3

Understanding the Input Parameters

3.1 Overview

There's no easy way to present the data for this chapter: There are literally hundreds of parameters that can be set in MAS. Fortunately, there are perhaps 10-20 that ever really need to be considered for a particular run. So we have broken the parameter list into three main groups: The often, sometimes, and seldom modified parameters. In the sections that follow, we define each parameter, explain the rationale for its inclusion, and provide an example why/how it might be varied. There are some subtle and not-so subtle differences between the parameters in the older serial code and the new parallel code. We begin by summarizing the parameters in the old code and then discuss the parameters in the new code.

3.2 Parameters in the older, serial code

Here are the parameters in the older, serial code.

3.2.1 The Often-Varied Parameters

Name : ntmax
Description: Maximum number of time steps
Value : ≥ 0
Default : 500

Name : tmax
Description: Maximum time of simulation
Value : ≥ 0.
Default : 200 code units (CU), where 1 CU = 24 minutes

Name : dtmax
Description: Maximum time step
Value : ≥ 0.
Default : 1 code unit (CU), where 1 CU = 24 minutes

Name : dtmin
Description: Minimum time step
Value : ≥ 0.
Default : 0.001 code units (CU), where 1 CU = 24 minutes

Name : rl
Description: Distance between inner and outer radial boundary
Value : ≥ 0.
Default : 1 code unit (CU), where 1 CU = 1 $R_\odot$

Name : mmodes
Description: Modes with histories that will be saved
Value : $0 \leq$ mmodes $\leq$ np/2 (np=number of longitudal mesh points)
Default : -1 (do not save histories)

Name : slund
Description: Lundquist number
Value : ≥ 0.
Default : 1000.

Name : visc
Description: Viscosity
Value : ≥ 0.
Default : 0.

Name : rsifile
Description: Restart file
Value : String of characters
Default : rsmas

Name : option
Description: Initial state (see routine initial)
Value : One of these: wave1, wave2, streamer, dipole
Default : NO_DEFAULT

Name : eqtype
Description: Initial plasma equilibrium (see routine initial)
Value : One of these: hydrostatic, zero-beta, parker
Default : NO_DEFAULT

Name : fldtype
Description: Initial magnetic field m (see routine initial)
Value : One of these: dipole, potential
Default : NO_DEFAULT

Name : gamma
Description: Ratio of specific heats
Value : $1. < \text{gamma} \leq 5/3$
Default : 1.05

Name : drratio
Description: The ratio of the mesh spacing at the end of a segment to
: that at the beginning for the radial mesh (see routine genmesh)
Value : Array $> 0.$
Default : 1.

Name : rfrac
Description: The normalized positions of the mesh segment boundaries (as a fraction
: of the size of the domain) for radiial mesh (see routine genmesh)
Value : Array $> 0.$. The last value $\geq 1.$
Default : 1.

Name : dtratio
Description: The ratio of the mesh spacing at the end of a segment to that at the
: beginning for the latitudinal mesh (see routine genmesh)
Value : Array $> 0.$
Default : 1.

Name : tfrac
Description: The normalized positions of the mesh segment boundaries (as a fraction
: of the size of the domain) for latitudinal mesh (see routine genmesh)

Value : Array > 0.. The last value ≥ 1.
Default : 1.

Name : nfrmesh
Description: Number of times to filter the radial mesh (see routine genmesh)
Value : Integer ≥ 0
Default : 0

Name : nftmesh
Description: Number of times to filter the latitudinal mesh (see routine genmesh)
Value : Integer ≥ 0
Default : 0

Name : ihistint
Description: Collect time-histories at intervals of ihistint timesteps
Value : Integer ≥ 0
Default : 1

Name : tpltxint
Description: Plot field diagnostics every tpltxint Alfvèn times
Value : Real $\geq 0.$
Default : 0. code units (CU), where 1 CU = 24 minutes

Name : onedfile
Description: Name of 1D file to load the initial equilibrium
Value : String of characters
Default : NO_DEFAULT

Name : np1d
Description: Number of points in the 1D file to load the initial equilibrium
Value : Integer ≥ 0
Default : 701

Name : plotlist
Description: List of output plot file separated by comma with no space in between
Value : all, vr, vt, vp, br, bt, bp, rho, t, p, jr, jt, jp, ar, at, ap, arr, ari, atr,
: ati, apr, api, vrr, vri, vtr, vti, vpr, vpi, ep, em, ub, rhor0, h1, visc,
: visccell, eta, etacell, et0ef, ep0ef, er0, et0, ep0
Default :

Name : b0

Description: Strength of the initial dipole
Value : $\geq 0.$
Default : 1.

Name : bingauss
Description: Specify magnetic field in Gauss or code units (1 CU = 2.205 G)
Value : Logical
Default : .false. (code units)

Name : rhor0
Description: Base density
Value : $\geq 0.$
Default : 1. code unit (1 CU = 10^8 cm^{-3})

Name : trsdump
Description: Write restart file every trsdump Alfvèn times
Value : If ≤ 0, disable restart file dump at time intervals
Default : 0.

Name : bnfile
Description: File containing the magnetogram data
Value : String ending in .dat
Default : NO_DEFAULT

Name : tcond
Description: Flag to have thermal conduction
Value : 0. or 1.
Default : 0. (no thermal conduction)

Name : radloss
Description: Flag to have radiation losses
Value : 0. or 1.
Default : 0. (no radiation losses)

Name : tbc0
Description: Temperature at the inner radial boundary
Value : $\geq 0.$
Default : 1.80×10^6 K

Name : tbc1
Description: Temperature at the outer radial boundary

Value : If 0. use tbc1, if $> 0.$ use value from pressure and density
Default : 0.

Name : ifaw
Description: Flag to turn on Alfvèn wave pressure
Value : 0 or 1
Default : 0 (no Alfvèn wave pressure)

Name : alpha_nocoll
Description: Flag to turn on collisionless thermal conduction
Value : 0. or 1.
Default : 0. (no collisionless thermal conduction)

3.2.2 The Sometimes-Varied Parameters

Name : ifideal
Description: Flag to have an ideal MHD run
Value : 1 (ideal MHD) or 0 (resistive MHD)
Default : 1

Name : ietatype
Description: Specify kind of resistivity
Value : 1 (uniform) or 2 (custom profile)
Default : 1

Name : nretap
Description: Number of radial points in custom resistivity profile (see routine loadeta)
Value : ≥ 1
Default : 1.

Name : retap
Description: Radial mesh in custom resistivity profile (see routine loadeta)
Value : 1D array of nondecreasing reals
Default : All set to 1.

Name : etapr
Description: Value of custom resistivity in radial profile (see routine loadeta)
Value : 1D array of reals
Default : All set to 1.

Name : ntetap
Description: Number of latitudinal points in custom resistivity profile (see routine loadeta)
Value : ≥ 1
Default : 1.

Name : tetap
Description: Latitudinal mesh in custom resistivity profile (see routine loadeta)
Value : 1D array of nondecreasing reals
Default : All set to 0.

Name : etapt
Description: Value of custom resistivity in latitudinal profile (see routine loadeta)
Value : 1D array of reals
Default : All set to 1.

Name : ivistype
Description: Specify kind of viscosity
Value : 1 (uniform) or 2 (custom profile)
Default : 1

Name : recell
Description: Cell-Reynolds value for viscosity (see routine cellvisc)
Value : Any. If ≤ 0, then no cell viscosity is used.
Default : 0.

Name : recell_facr
Description: Radial factor in cell-Reynolds viscosity (see routine cellvisc)
Value : ≥ 0.
Default : 1.

Name : recell_fact
Description: Latitudinal factor in cell-Reynolds viscosity (see routine cellvisc)
Value : ≥ 0.
Default : 1.

Name : recell_facp
Description: Longitudinal factor in cell-Reynolds viscosity (see routine cellvisc)
Value : ≥ 0.
Default : 1.

Name : recell_viscmax
Description: Maximum viscosity when using cell-Reynolds option (see routine cellvisc)
Value : $\geq 0.$
Default : 10^{20}

Name : recell_unift
Description: Flag to make cell viscosiyy uniform in latitude (see routine cellvisc)
Value : logical
Default : .false.

Name : nrvisp
Description: Number of radial points in custom viscosity profile (see routine loadvisc)
Value : ≥ 1
Default : 1.

Name : rvisp
Description: Radial mesh in custom viscosity profile (see routine loadvisc)
Value : 1D array of nondecreasing reals
Default : All set to 1.

Name : vispr
Description: Value of custom viscosity in radial profile (see routine loadvisc)
Value : 1D array of reals
Default : All set to 1.

Name : ntvisp
Description: Number of latitudinal points in custom viscosity profile (see routine loadvisc)
Value : ≥ 1
Default : 1.

Name : tvisp
Description: Latitudinal mesh in custom viscosity profile (see routine loadvisc)
Value : 1D array of nondecreasing reals
Default : All set to 0.

Name : vispt
Description: Value of custom viscosity in latitudinal profile (see routine loadvisc)
Value : 1D array of reals
Default : All set to 1.

Name : ifprec

Description: Flag to use preconditioning in matrix inversion
Value : 0 (no preconditioning), 1 (diagonal)
Default : 2

Name : epscg
Description: Convergence criterion for matrix inversion except for resistivity
Value : $\geq 0.$
Default : 10^{-8}

Name : ncghist
Description: Write convergence history for matrix inversion every ncghist iterations
Value : Integer ≤ 0
Default : 0

Name : epscga
Description: Convergence criterion for matrix inversion for resistivity
Value : $\geq 0.$
Default : 10^{-8}

Name : ncgmax
Description: Maximum number of iterations in matrix inversion
Value : Integer > 0
Default : 500

Name : ifrho
Description: Flag to advance the density
Value : 1 (advance) or 0 (do not advance)
Default : 1

Name : iftemp
Description: Flag to advance the temperature
Value : 1 (advance) or 0 (do not advance)
Default : 1

Name : dformat
Description: Format of output field files
Value : text or hdf
Default : text

Name : pr0
Description: Base pressure in the initial hydrostatic equilibrium

Value : ≥ 0.
Default : 0.1 code units (1 CU = 0.387 dyne cm^{-2})

Name : dthmax
Description: Angle used to specify shearing profile (see routine setvtrans0)
Value : ≥ 0.
Default : 0.6981

Name : th0
Description: Angle used to specify shearing profile (see routine setvtrans0)
Value : $0 \leq$ th0 $\leq \pi$
Default : $\pi/2$

Name : tnode
Description: Defines temporal nodes for shearing profile (see routine setvtrans0)
Value : Array of non-decreasing reals
Default : -1.

Name : vnode
Description: Defines velocity nodes for shearing profile (see routine setvtrans0)
Value : Array of ≥ 0
Default : 0.

Name : ihst
Description: Radial point for collection of history diagnostics
Value : If ≤ 0 or $\geq$ mesh size, use middle point
Default : 0

Name : jhst
Description: Latitudinal point for collection of history diagnostics
Value : If ≤ 0 or $\geq$ mesh size, use middle point
Default : 0

Name : khst
Description: Longitudinal point for collection of history diagnostics
Value : If ≤ 0 or $\geq$ mesh size, use first point
Default : 0

Name : bcr1type
Description: Type of outer radial boundary
Value : open, extrapolation, fixed, 1dchar (1D characteristics)

Default : extrapolation

Name : bcr0type
Description: Type of inner radial boundary
Value : fixed, 1dchar (1D characteristics)
Default : fixed

Name : dipangle
Description: Angle (in degrees) by which the initial dipole is tilted
Value : Any real
Default : 0.

Name : thistint
Description: Collect time-histories every thistint Alfvèn times
Value : ≥ 0
Default : 0.

Name : ipltxint
Description: Plot field diagnostics at intervals of ipltxint time steps
Value : Integer ≥ 0
Default : 0

Name : ubzero
Description: Set flow at inner boundary to zero if it is negative
Value : Logical
Default : .false.

Name : tnode_ch
Description: Defines temporal nodes for coronal heating
Value : Array of non-decreasing reals
Default : -1.

Name : q0_pole_ch
Description: Defines nodes for heat flux at the pole
Value : Array of ≥ 0
Default : 1.

Name : q0_eq_ch
Description: Defines nodes for heat flux at the equator
Value : Array of ≥ 0
Default : 1.

Name : hlen_pole_ch
Description: Defines nodes for length scale of deposition of heat at the pole
Value : Array of ≥ 0
Default : .5 Code Units (1 CU = 6.96×10^{10} cm

Name : hlen_eq_ch
Description: Defines nodes for length scale of deposition of heat at the equator
Value : Array of ≥ 0
Default : .5 Code Units (1 CU = 6.96×10^{10} cm)

Name : q0phys_ch
Description: Defines nodes for heat flux, which is modulated by q0_pole_ch and q0_eq_ch
Value : Array of ≥ 0
Default : 0. erg cm^{-2} s^{-1}

Name : qalfven_ch
Description: Defines nodes for Alfvèn pressure at the inner radial boundary
Value : Array of ≥ 0
Default : 0. dyn cm^{-2}

Name : he_fac
Description: Helium fraction
Value : ≥ 0
Default : 0.

Name : br00
Description: Artificial monopole radial field
Value : ≥ 0
Default : 0. Code Units (1 CU = 2.205 G)

Name : ishearprof
Description: Pick a a type of shear profile (see routine setvtrans0)
Value : 1, 2, or 3
Default : 1

Name : arotate
Description: Parameter used when ishearprof is 2 (see routine setvtrans0)
Value : Real
Default : 0.

Name : brotate
Description: Parameter used when ishearprof is 2 (see routine setvtrans0)
Value : Real
Default : 0.

Name : crotate
Description: Parameter used when ishearprof is 2 (see routine setvtrans0)
Value : Real
Default : 0.

Name : thprof_th
Description: Parameter controlling latitudinal heating profile (see routine load_thprof)
Value : $0. \leq$ thprof_th $\leq \pi$
Default : 0.7

Name : thprof_dth
Description: Parameter controlling latitudinal heating profile (see routine load_thprof)
Value : Real $\geq 0.$
Default : 0.08

Name : emgflux
Description: Flag to emerge new magnetic flux
Value : Logical
Default : .false.

Name : tnode_ef
Description: Defines temporal nodes for emerging flux (see routine eflux)
Value : Array of non-decreasing reals
Default : -1.

Name : brfile_ef
Description: Defines B_r file nodes for emerging flux (see routine eflux)
Value : Array of characters
Default : NO_DEFAULT

Name : ifradbc
Description: Use radiation balance boundary density
Value : 0 (do not use) or 1 (use)
Default : 1

Name : t0_tc

Description: Temperature below which to use modified thermal conduction
Value : ≥ 0.
Default : 250,000 K

Name : dt0_tc
Description: Temperature interval to switch to modified thermal conduction
Value : ≥ 0.
Default : 20,000 K

3.2.3 The Seldom-Varied Parameters

Name : g0
Description: Gravity at the inner radial boundary
Value : ≥ 0.
Default : .823 code units (CU),

Name : ifvdgv
Description: Flag to use advection in the momentum equation
Value : 1 (use) or 0 (do not use)
Default : 1

Name : simult
Description: Semi-implicit multiplier
Value : > 1.
Default : 1.1

Name : dtmult
Description: Totally useless variable
Value : any real
Default : 1.

Name : strmax
Description: Variable to rescale the advective calculated time step
Value : $0. < \text{strmax} < 1$
Default : 0.5

Name : isitype
Description: Switch between fully explicit and semi-implicit algorithm
Value : 0 (fully explicit) or not 0 (semi-implicit)

Default : 1

Name : cfl
Description: In fully explict runs, fraction of the CourantFriedrichsLewy limit
Value : 0. < cfl < 1.
Default : 0.4

Name : omegaeta
Description: Implicit/explicit fraction in the resisitivity advance
Value : 0. < omegaeta < 1.
Default : 0.5

Name : upwinda
Description: Flag to set upwinding in the advection of the vector potential
Value : 0. or 1.
Default : 0. (do not upwind)

Name : upwindv
Description: Flag to set upwinding in the advection of the velocity
Value : 0. or 1.
Default : 0. (do not upwind)

Name : pdma
Description: Partial Donor Method coefficient (see routine uwpdm)
Value : 0. < pdma < 2.
Default : 1.

Name : rmcell
Description: Useless variable: do not attempt to change!
Value : 0.
Default : 0.

Name : rmcell_facr
Description: Totally useless variable
Value : any real
Default : 1.

Name : rmcell_fact
Description: Totally useless variable
Value : any real
Default : 1.

Name : rmcell_facp
Description: Totally useless variable
Value : any real
Default : 1.

Name : rmcell_etamax
Description: Totally useless variable
Value : any real
Default : $1. \times 10^{20}$

Name : rmcell_unift
Description: Totally useless variable
Value : any logical
Default : .false.

Name : ifrsout
Description: Useless variable: do not attempt to change!
Value : 1
Default : 1

Name : icgtype
Description: Totally useless variable
Value : Any integer
Default : Undefined

Name : nwave
Description: Radial mode number in Alfvèn wave test (see routine wave1)
Value : Integer ≤ 1
Default : 1

Name : lwave
Description: Latitudinal mode number in Alfvèn wave test (see routine wave1)
Value : Integer ≤ 1
Default : 1

Name : epswave
Description: Wave amplitude in Alfvèn wave test (see routine wave1)
Value : $> 0.$
Default : 10^{-6}

Name : sigma
Description: Totally useless variable
Value : Any real
Default : 1.

Name : ifpc
Description: Flag to use predictor/corrector scheme
Value : 1 (advance) or 0 (do not advance)
Default : 1

Name : predflow
Description: Flag to add flow part in the predictor step
Value : Logical
Default : .false.

Name : betapc_p
Description: Fraction of wave term in the predictor step in induction
: and energy equations
Value : $0 \leq$ betapc_p ≤ 1
Default : 0.5

Name : betapc_v
Description: Fraction of wave term in the predictor step in momentum equation
Value : $0 \leq$ betapc_v ≤ 1
Default : 0.5

Name : si_aggressive
Description: Use local semi-implicit coefficient or global
Value : Logical
Default : .false. (global)

Name : fac_cflv
Description: Account for velocity in calculating semi-implicit coefficient
Value : $0 \leq$ fac_cflv ≤ 1
Default : 1.

Name : betapc_si
Description: Use semi-implicit term in the predictor of the momentum equation
Value : 0. (do not use) or 1. (use)
Default : 1.

Name : rhofac1
Description: Totally useless variable
Value : Any real
Default : 0.04

Name : rhofac2
Description: Totally useless variable
Value : Any real
Default : 0.104

Name : ifohmic
Description: Flag to add ohmic heating to the energy equation
Value : If not 0, add ohmic heating
Default : 0

Name : ifhvisc
Description: Flag to add viscous heating to the energy equation
Value : If not 0, add viscous heating
Default : 0

Name : isetrho
Description: Set the density to give a uniform Alfvèn speed
every isetrho timesteps
Value : Integer ≥ 0
Default : 0

Name : omegarho
Description: Tune density when a uniform Alfvèn speed is
requested (see routine setrho)
Value : $0 \leq$ omegarho ≤ 1.
Default : 0.1

Name : upwindar
Description: Flag to upwind only the radial component in the
advection of the vector potential
Value : 0. or 1.
Default : 0. (do not upwind)

Name : upwindat
Description: Flag to upwind only the latitudinal component in the
advection of the vector potential

Value : 0. or 1.
Default : 0. (do not upwind)

Name : upwindap
Description: Flag to upwind only the longitudinal component in the advection of the vector potential
Value : 0. or 1.
Default : 0. (do not upwind)

Name : upwindphi
Description: Flag to upwind the longitudinal component in the momentum equation
Value : 0. or > 0.
Default : 1. (upwind)

Name : ifeqcs
Description: Flag to calculate specially the equatorial current sheet
Value : If not 0, do the special calculation
Default : 0

Name : irsdump
Description: Write restart file every irsdump timesteps
Value : If ≤ 0, disable restart file dump at time step intervals
Default : 0

Name : gbc0err
Description: Error allowed in convergence criterion for characterirstics boundary conditions
Value : > 0.
Default : 10^{-5}

Name : parchar
Description: Solve characteristics parallel to the magnetic field
Value : Logical
Default : .false. (use gas characteristics)

Name : nfiltub
Description: Number of time to filter the boundary velocity
Value : Integer ≥ 0
Default : 0

Name : nfiltne0
Description: Number of time to filter the radiation-balance boundary density

Value : Integer ≥ 0
Default : 0

Name : tne0
Description: Time scale to advance the base density towards the required value when using radiation boundary conditions
Value : ≥ 0.
Default : 1. Code Unit (1 CU = 24 minutes)

Name : rho_aw
Description: Artificial inertia in Alfvèn wave pressure advance
Value : ≥ 0
Default : 1.

Name : tc_r
Description: Radius around which collisional thermal conduction is turned off
Value : Must be withing computational domain
Default : 10. Code Units (1 CU = 6.96×10^{10} cm)

Name : tc_dr
Description: Interval during which collisional thermal conduction is turned off
Value : ≥ 0
Default : 5. Code Units (1 CU = 6.96×10^{10} cm)

Name : tc_fac0
Description: Factor multiplying thermal conduction below tc_r
Value : ≥ 0
Default : 1.

Name : tc_fac1
Description: Factor multiplying thermal conduction below tc_r
Value : ≥ 0
Default : 1. (Must be 0. to turn collisional thermal conduction off)

Name : tc_nocoll_r
Description: Radius around which collisionless thermal conduction is turned on
Value : Must be withing computational domain
Default : 10. Code Units (1 CU = 6.96×10^{10} cm)

Name : tc_nocoll_dr
Description: Interval during which collisional thermal conduction is turned on

Value : ≥ 0
Default : 5. Code Units (1 CU = 6.96×10^{10} cm)

Name : neradmin
Description: Totally useless variable
Value : Any real
Default : -10^{10}

Name : awthprof_th
Description: Parameter controlling Alfvèn wave pressure profile (see routine load_awthprof)
Value : 0. $\leq$ awthprof_th $\leq \pi$
Default : 0.7

Name : bmin
Description: Minimum magnetic field assumed in calculating boundary
velocity in emerging flux runs
Value : Real ≥ 0.
Default : 0.05 Code Units (1 CU = 2.205 G)

Name : awthprof2_th
Description: Parameter controlling Alfvèn wave pressure profile (see routine load_awthprof)
Value : 0. $\leq$ awthprof2_th $\leq \pi$
Default : $\pi - 0.7$

Name : vrbcr0ef
Description: Flag to set the velocity at the lower boundary in emerging flux runs
Value : Characters: exb (calculate from Poynting vector) or not
Default : exb

Name : vtbcr0ef
Description: Flag to set the velocity at the lower boundary in emerging flux runs
Value : Characters: exb (calculate from Poynting vector) or not
Default : zero (set to zero)

Name : vpbcr0ef
Description: Flag to set the velocity at the lower boundary in emerging flux runs
Value : Characters: exb (calculate from Poynting vector) or not
Default : zero (set to zero)

Name : ifbheat
Description: Flag to calculate heating depending on magnetic field

Value : 0 (do not use) or 1 (use)
Default : 0

Name : bheatfile
Description: File to be read in calculating heating depending on magnetic field
Value : Characters
Default : brbyb2.dat

Name : fdphi
Description: Mimic final differences in longitude
Value : Logical
Default : .false.

Name : ifimplrad
Description: Calculate radiation losses semi-implicitly
Value : 0 (do not use) or 1 (use)
Default : 1

Name : ifeta_dyn
Description: Calculate local resistivity depending on the current
Value : 0 (do not use) or 1 (use)
Default : 0

Name : eta_min
Description: Minimum resistivity when ifeta_dyn is used
Value : $\geq 0.$
Default : 0.

Name : eta_max
Description: Maximum resistivity when ifeta_dyn is used
Value : $\geq 0.$
Default : 0.

Name : eta_jmin
Description: Current threshold when ifeta_dyn is used
Value : $\geq 0.$
Default : 30.

Name : eta_dj
Description: Width around current threshold when ifeta_dyn is used
Value : $\geq 0.$

Default : 5.

Name : eta_dj
Description: Proportionality constant for resistivity when ifeta_dyn is used
Value : ≥ 0.
Default : 1.

Name : bsqfloor
Description: Minimum B^2 in calculating resistivity when ifeta_dyn is used
Value : ≥ 0.
Default : 0.

Name : fjfrac
Description: Parameter important when ifeta_dyn is used
Value : ≥ 0.
Default : 0.5

Name : alpha_tc
Description: Exponent used in modified thermal conduction
Value : ≥ 0.
Default : 2.5

Name : nfilt_kappa
Description: Number of times to filter the thermal conduction tensor
Value : Integer ≥ 0
Default : 0

Name : split_visc
Description: Split viscosity advance in momentum equation
Value : Logical
Default : .true.

Name : gamma1
Description: When using variable gamma, this is the inner value
Value : 1. < gamma1 $\leq 5/3$
Default : 0.

Name : gamma2
Description: When using variable gamma, this is the outer value
Value : 1. < gamma2 $\leq 5/3$
Default : 0.

Name : rgamma
Description: When using variable gamma, this is the radius where the change occurs
Value : Within computational domain
Default : 0.

Name : drgamma
Description: Radial distance over which we have a variable gamma
Value : ≥ 0.
Default : 0.

3.3 The New, Parallel Code

Here is a description of the variables in the new code.

3.3.1 The Often-Varied Parameters

Name : ntmax
Description: Maximum number of time steps
Value : ≥ 0
Default : 500

Name : tmax
Description: Maximum time of simulation
Value : ≥ 0.
Default : 100 code units (CU), where 1 CU = 24 minutes

Name : dtmax
Description: Maximum time step
Value : ≥ 0.
Default : 1 code unit (CU), where 1 CU = 24 minutes

Name : dtmin
Description: Minimum time step
Value : ≥ 0.
Default : 0.001 code units (CU), where 1 CU = 24 minutes

Name : r0

Description: Position of the inner radial boundary
Value : $\geq 0.$
Default : 1 code unit (CU), where 1 CU = 1 $R_\odot$

Name : r1
Description: Position of the outer radial boundary
Value : $\geq 0.$
Default : 30 code unit (CU), where 1 CU = 1 $R_\odot$

Name : slund
Description: Lundquist number
Value : $\geq 0.$
Default : 10^5

Name : visc
Description: Viscosity
Value : $\geq 0.$
Default : 10^{-5}

Name : rsifile
Description: Restart file
Value : String of characters
Default :

Name : initial_field
Description: Initial state for magnetic field (see routine initialize_magnetic_field)
Value : One of these: ALFVEN_WAVE1, ALFVEN_WAVE2, POTENTIAL_FIELD, DIPOLE
Default : DIPOLE

Name : initial_plasma
Description: Initial plasma equilibrium
Value : One of these: HYDROSTATIC, ZERO-BETA, 1DFILE, STREAMER
Default : ZERO-BETA

Name : gamma
Description: Ratio of specific heats
Value : 1. < gamma $\leq 5/3$
Default : 5/3

Name : drratio
Description: The ratio of the mesh spacing at the end of a segment to

: that at the beginning for the radial mesh (see routine genmesh)
Value : Array $> 0.$
Default : 1.

Name : rfrac
Description: The normalized positions of the mesh segment boundaries (as a fraction
: of the size of the domain) for radial mesh (see routine genmesh)
Value : Array of increasing reals. First value is 0., last value 1.
Default : 0.

Name : dtratio
Description: The ratio of the mesh spacing at the end of a segment to that at the
: beginning for the latitudinal mesh (see routine genmesh)
Value : Array $> 0.$
Default : 1.

Name : tfrac
Description: The normalized positions of the mesh segment boundaries (as a fraction
: of the size of the domain) for latitudinal mesh (see routine genmesh)
Value : Array of increasing reals. First value is 0., last value 1.
Default : 1.

Name : dpratio
Description: The ratio of the mesh spacing at the end of a segment to that at the
: beginning for the longitudinal mesh (see routine genmesh)
Value : Array $> 0.$
Default : 1.

Name : pfrac
Description: The normalized positions of the mesh segment boundaries (as a fraction
: of the size of the domain) for latitudinal mesh (see routine genmesh)
Value : Array of increasing reals. Must not contain both 0. and 1.
Default : 1.
Name : nfrmesh
Description: Number of times to filter the radial mesh (see routine genmesh)
Value : Integer ≥ 0
Default : 0

Name : nftmesh
Description: Number of times to filter the latitudinal mesh (see routine genmesh)
Value : Integer ≥ 0

Default : 0

Name : nfpmesh
Description: Number of times to filter the longitudinal mesh (see routine genmesh)
Value : Integer ≥ 0
Default : 0

Name : phishift
Description: Amount by which to shift the longitudinal mesh (see routine gen mesh)
Value : Real ≥ 0
Default : 0.

Name : ihistint
Description: Collect time-histories at intervals of ihistint timesteps
Value : Integer ≥ 0
Default : 1

Name : tpltxint
Description: Plot field diagnostics every tpltxint Alfvèn times
Value : Real ≥ 0.
Default : 0. code units (CU), where 1 CU = 24 minutes

Name : onedfile
Description: Name of 1D file to load the initial equilibrium
Value : String of characters
Default :

Name : plotlist
Description: List of output plot file separated by comma with no space in between
Value : all, vr, vt, vp, br, bt, bp, rho, t, p, pres, jr, jt, jp, ar, at, ap,
: sifac, heat, ep, em, vr_old, vt_old, vp_old, eta, visc
Default :

Name : b0_dipole
Description: Strength of the initial dipole
Value : ≥ 0.
Default : 1. Code Units (1 CU = 2.205 G)

Name : b_in_gauss
Description: Specify magnetic field in Gauss or Code Units (1 CU = 2.205 G)
Value : Logical

Default : .true. (use Gauss)

Name : b0
Description: Specify maximum field magnitude if b_in_gauss is .false.
or leave it unchanged if 0.
Value : Real ≥ 0
Default : 0. Code Units (1 CU = 2.205 G)

Name : rho0
Description: Base density
Value : ≥ 0.
Default : 1. code unit (1 CU = 10^8 cm^{-3})

Name : trsdump
Description: Write restart file every trsdump Alfvèn times
Value : If ≤ 0, disable restart file dump at time intervals
Default : 0.

Name : bnfile
Description: File containing the magnetogram data
Value : String ending in .hdf
Default :

Name : tcond
Description: Flag to have thermal conduction
Value : 0. or 1.
Default : 0. (no thermal conduction)

Name : radloss
Description: Flag to have radiation losses
Value : 0. or 1.
Default : 0. (no radiation losses)

Name : tbc0
Description: Temperature at the inner radial boundary
Value : ≥ 0.
Default : $20,000$ K

Name : tbc1
Description: Temperature at the outer radial boundary
Value : If 0. use tbc1, if > 0. use value from pressure and density

Default : 0.

Name : advance_pw
Description: Flag to turn on Alfvèn wave pressure
Value : Logical
Default : .false. (no Alfvèn wave pressure)

Name : alpha_nocoll
Description: Flag to turn on collisionless thermal conduction
Value : 0. or 1.
Default : 0. (no collisionless thermal conduction)

Name : ifheat
Description: Flag to turn on coronal heating (see routine heating)
Value : Integer ≥ 0, corresponding to different models
Default : 0 (no coronal heating)

3.3.2 The Sometimes-Varied Parameters

Name : ifideal
Description: Flag to have an ideal MHD run
Value : 1 (ideal MHD) or 0 (resistive MHD)
Default : 1

Name : eta_profile
Description: Specify kind of resistivity
Value : UNIFORM, RADIALLY_DEPENDENT
Default : UNIFORM

Name : eta_fac_0
Description: Scale radially dependent resistivity below r_eta_crit
Value : Real > 0
Default : 1.

Name : eta_fac_1
Description: Scale radially dependent resistivity above r_eta_crit
Value : Real > 0
Default : 1.

Name : r_eta_crit
Description: Critical radius where radially dependent resistivity changes
Value : Real > 0
Default : 3. Code Units (1 CU $= 6.96 \times 10^{10}$ cm

Name : dr_eta_crit
Description: Width along which radially dependent resistivity changes
Value : Real > 0
Default : 1.

Name : ivistype
Description: Specify kind of viscosity
Value : 1 (uniform) or 2 (custom profile)
Default : 1

Name : recell
Description: Cell-Reynolds value for viscosity (see routine cellvisc)
Value : Any. If ≤ 0, then no cell viscosity is used.
Default : 0.

Name : recell_facr
Description: Radial factor in cell-Reynolds viscosity (see routine cellvisc)
Value : ≥ 0.
Default : 1.

Name : recell_fact
Description: Latitudinal factor in cell-Reynolds viscosity (see routine cellvisc)
Value : ≥ 0.
Default : 1.

Name : recell_facp
Description: Longitudinal factor in cell-Reynolds viscosity (see routine cellvisc)
Value : ≥ 0.
Default : 1.

Name : recell_viscmax
Description: Maximum viscosity when using cell-Reynolds option (see routine cellvisc)
Value : ≥ 0.
Default : 10^{20}

Name : nrvisp

Description: Number of radial points in custom viscosity profile (see routine loadvisc)
Value : ≥ 1
Default : 1.

Name : ifprec
Description: Flag to use preconditioning in matrix inversion
Value : 0 (no preconditioning), 1 (diagonal)
Default : 1

Name : epscg
Description: Convergence criterion for matrix inversion except for resistivity
Value : $\geq 0.$
Default : 10^{-9}

Name : ncghist
Description: Write convergence history for matrix inversion every ncghist iterations
Value : Integer ≤ 0
Default : 0

Name : epscga
Description: Convergence criterion for matrix inversion for resistivity
Value : $\geq 0.$
Default : 10^{-9}

Name : ncgmax
Description: Maximum number of iterations in matrix inversion
Value : Integer > 0
Default : 500

Name : advance_a
Description: Flag to advance the vector potential
Value : Logical
Default : .true.

Name : advance_v
Description: Flag to advance the velocity l
Value : Logical
Default : .true.

Name : advance_rho
Description: Flag to advance the density

Value : Logical
Default : .true.

Name : advance_t
Description: Flag to advance the temperature
Value : Logical
Default : .true.

Name : tprofile
Description: Temporal profiles to ramp up and down variables
Value : Structure
Default :

Name : char_bc
Description: Use characteristics boundary conditions
Value : Logical
Default : .true.

Name : dipangle
Description: Angle (in degrees) by which the initial dipole is tilted
Value : Any real
Default : 0.

Name : thistint
Description: Collect time-histories every thistint Alfvèn times
Value : ≥ 0
Default : 0.

Name : ipltxint
Description: Plot field diagnostics at intervals of ipltxint time steps
Value : Integer ≥ 0
Default : 0

Name : ubzero
Description: Set flow at inner boundary to zero if it is negative
Value : Logical
Default : .false.

Name : heat0
Description: Coronal heating parameter (see routine heating)
Value : Real > 0.

Default : 0.012 Code Units (1CU = 2.6806×10^{-4} erg cm^{-3} s^{-1})

Name : heatscale
Description: Length scale of deposition of heat
Value : Real $> 0.$
Default : 0.4 Code Units (1 CU = 6.96×10^{10} cm)

Name : heat1
Description: Coronal heating parameter (see routine heating)
Value : Real $> 0.$
Default : 0.012 Code Units (1CU = 2.6806×10^{-4} erg cm^{-3} s^{-1})

Name : hlen_eq_ch
Description: Defines nodes for length scale of deposition of heat at the equator
Value : Array of ≥ 0
Default : .5 Code Units (1 CU = 6.96×10^{10} cm)

Name : pw0
Description: Alfvèn pressure at the inner radial boundary
Value : Real ≥ 0
Default : 0. Code Units (1 CU = 0.387 dyn cm^{-2})

Name : he_fac
Description: Helium fraction
Value : ≥ 0
Default : 0.

Name : br00
Description: Artificial monopole radial field
Value : ≥ 0
Default : 0. Code Units (1 CU = 2.205 G)

Name : shear
Description: Set the shear profile (see routine initialize_shear)
Value : Structure
Default : No shear

Name : arotate
Description: Parameter used when shear is DIFFERENTIAL_SHEAR
Value : Real
Default : 0.

Name : brotate
Description: Parameter used when shear is DIFFERENTIAL_SHEAR
Value : Real
Default : 0.

Name : crotate
Description: Parameter used when shear is DIFFERENTIAL_SHEAR
Value : Real
Default : 0.

Name : v_drive
Description: Magnitude of the shear velocity
Value : Real
Default : 0.

Name : flow
Description: Set the surface flow profile (see routine initialize_flow)
Value : Structure
Default : No flow

Name : v_flow_norm
Description: Magnitude of the normal flow
Value : Real
Default : 0.

Name : v_flow_trans
Description: Magnitude of the transverse flow
Value : Real
Default : 0.

Name : thprof_th
Description: Parameter controlling latitudinal heating profile (see routine load_thprof)
Value : $0. \leq$ thprof_th $\leq \pi$
Default : 0.7

Name : thprof_dth
Description: Parameter controlling latitudinal heating profile (see routine load_thprof)
Value : Real $\geq 0.$
Default : 0.08

Name : emerging_flux
Description: Flag to emerge new magnetic flux (see routine initialize_emerging_flux)
Value : Integer 1, 2, 3, 4
Default : 0 (no emerging flux)

Name : eflux_e0f
Description: Emerging flux field amplitude
Value : Real $\geq 0.$
Default : 0.

Name : eflux_psi_file
Description: Specify file from which to read the emerging flux field
Value : Array of characters
Default :

Name : t0_tc
Description: Temperature below which to use modified thermal conduction
Value : $\geq 0.$
Default : 250,000 K

Name : dt0_tc
Description: Temperature interval to switch to modified thermal conduction
Value : $\geq 0.$
Default : 20,000 K

Name : diag
Description: Input structure for specifying diagnostic points
Value : See module diagnostics
Default : No diagnostics

Name : potential_field_bc
Description: Specify boundary condition for initial potential field
Value : SOURCE_SURFACE or CLOSED_WALL
Default : CLOSED_WALL

Name : bfac
Description: Proportionality constant for heating proportional to B
Value : Real $\geq 0.$
Default : Undefined

Name : bexp

Description: Magnetic field exponent for heating
Value : Real *geq*0.
Default : Undefined

Name : heatscale1
Description: Auxiliary length scale of deposition of heating (see routine heating)
Value : Real > 0.
Default : 1. Code Units (1 CU = 6.96×10^{10} cm)

Name : heat_b0
Description: Magnetic field proportionality constant for heating (see routine heating)
Value : Real *geq*0.
Default : 0.

Name : heat_db
Description: Constant for heating proportional to B (see routine heating)
Value : Real *geq*0.
Default : 1.

Name : r_ss
Description: Source surface radius
Value : Real. If negative then use r1
Default : -1. Code Units (1 CU = 6.96×10^{10} cm)

Name : heat2
Description: Coronal heating parameter (see routine heating)
Value : Real > 0.
Default : 0. Code Units (1CU = 2.6806×10^{-4} erg cm^{-3} s^{-1})

Name : heatscale2
Description: Auxiliary length scale of deposition of heating (see routine heating)
Value : Real > 0.
Default : 1. Code Units (1 CU = 6.96×10^{10} cm)

Name : iftfloor
Description: Flag to set a minumum temperature
Value : Logical
Default : .false. (no floor)

Name : tfloor
Description: Minumum temperature in the calculation

Value : Real ≥ 0.
Default : 20,000 K

Name : heatbr
Description: Proportionality constant for heating where B_r is small (see routine heating)
Value : Real ≥ 0.
Default : 0. Code Units (1CU = 2.6806×10^{-4} erg cm^{-3} s^{-1})

Name : brmax
Description: Parameter for heating where (see routine heating)
Value : Real ≥ 0.
Default : 0. Code Units (1 CU = 2.205 G)

Name : heatrf
Description: Radius to scale down heating hencefore (see routine heating)
Value : Real *geq*0.
Default : Undefined

Name : heatdrf
Description: Interval over which to scale down heating (see routine heating)
Value : Real *geq*0.
Default : Undefined

Name : ifheat_aux
Description: Flag to use a 3D file for heating (see routine heating)
Value : Logical
Default : .false. (no mask)

Name : heat_aux_file
Description: 3D file to be read for heating (see routine heating)
Value : Characters
Default :

3.3.3 The Seldom-Varied Parameters

Name : g0
Description: Gravity at the inner radial boundary
Value : ≥ 0.
Default : .823 code units (CU)

Name : iffvdgv
Description: Flag to use advection in the momentum equation
Value : 1 (use) or 0 (do not use)
Default : 1

Name : simult
Description: Semi-implicit multiplier
Value : $> 1.$
Default : 1.1

Name : isitype
Description: Switch between fully explicit and semi-implicit algorithm
Value : 0 (fully explicit) or not 0 (semi-implicit)
Default : 1

Name : cfl
Description: In fully explict runs, fraction of the CourantFriedrichsLewy limit
Value : $0. < \text{cfl} < 1.$
Default : 0.4

Name : upwind_a
Description: Flag to set upwinding in the advection of the vector potential
Value : 0. or 1.
Default : 0. (do not upwind)

Name : upwind_v
Description: Flag to set upwinding in the advection of the velocity
Value : 0. or 1.
Default : 1. (upwind)

Name : wave_mode_r
Description: Radial mode number in Alfvèn wave test (see routine init_alfven_wave1)
Value : Integer ≤ 1
Default : 1

Name : wave_mode_t
Description: Latitudinal mode number in Alfvèn wave test (see routine init_alfven_wave1)
Value : Integer ≤ 1
Default : 1

Name : wave_amplitude
Description: Wave amplitude in Alfvèn wave test (see routine init_alfven_wave1)
Value : $> 0.$
Default : 10^{-6}

Name : pred_v
Description: Flag to use predictor in the momentum equation
Value : Logical
Default : .true.

Name : pred_a
Description: Flag to use predictor in the induction equation
Value : Logical
Default : .true.

Name : pred_t
Description: Flag to use predictor in the energy equation
Value : Logical
Default : .true.

Name : pred_rho
Description: Flag to use predictor in the density equation
Value : Logical
Default : .true.

Name : betapc_v_flow
Description: Multiplies advection term in the predictor for velocity advance
Value : $0 \leq$ betapc_v_flow ≤ 1
Default : 0.

Name : betapc_v_wave
Description: Multiplies wave term in the predictor for velocity advance
Value : $0 \leq$ betapc_v_wave ≤ 1
Default : 0.5

Name : betapc_t_flow
Description: Multiplies advection term in the predictor for energy advance
Value : $0 \leq$ betapc_t_flow ≤ 1
Default : 0.

Name : betapc_rho_wave

Description: Multiplies wave term in the predictor for density advance
Value : $0 \leq$ betapc_rho_wave ≤ 1
Default : 0.5

Name : betapc_rho_flow
Description: Multiplies advection term in the predictor for density advance
Value : $0 \leq$ betapc_rho_flow ≤ 1
Default : 0.

Name : betapc_t_wave
Description: Multiplies wave term in the predictor for energy advance
Value : $0 \leq$ betapc_t_wave ≤ 1
Default : 0.5

Name : betapc_a
Description: Multiplies term in the predictor for induction equation
Value : $0 \leq$ betapc_a ≤ 1
Default : 0.5

Name : si_aggressive
Description: Use local semi-implicit coefficient or global
Value : Logical
Default : .true. (Use local)

Name : fac_cflv
Description: Account for velocity in calculating semi-implicit coefficient
Value : $0 \leq$ fac_cflv ≤ 1
Default : 1.

Name : si_local_kv
Description: Set the flow modification to the semi-implicit term
Value : Logical
Default : .true. (Use local value)

Name : betapc_si
Description: Use semi-implicit term in the predictor of the momentum equation
Value : 0. (do not use) or 1. (use)
Default : 1.

Name : ohmic_heating
Description: Flag to add ohmic heating to the energy equation

Value : Logical
Default : .false. (do not add)

Name : upwind_ar
Description: Flag to upwind only the radial component in the advection of the vector potential
Value : 0. or 1.
Default : 0. (do not upwind)

Name : upwind_at
Description: Flag to upwind only the latitudinal component in the advection of the vector potential
Value : 0. or 1.
Default : 0. (do not upwind)

Name : upwind_ap
Description: Flag to upwind only the longitudinal component in the advection of the vector potential
Value : 0. or 1.
Default : 0. (do not upwind)

Name : upwind_a_r0
Description: Totally useless variable
Value : Any real
Default : 0.

Name : upwind_t
Description: Upwind the temperature advection
Value : 0. or 1.
Default : 1. (upwind)

Name : upwind_rho
Description: Upwind the density advection
Value : 0. or 1.
Default : 1. (upwind)

Name : irsdump
Description: Write restart file every irsdump timesteps
Value : If ≤ 0, disable restart file dump at time step intervals
Default : 0

Name : nfiltub
Description: Number of time to filter the boundary velocity
Value : Integer ≥ 0
Default : 2

Name : rho_aw
Description: Artificial inertia in Alfvèn wave pressure advance
Value : ≥ 0
Default : 1.

Name : tc_r
Description: Radius around which collisional thermal conduction is turned off
Value : Must be withing computational domain
Default : 10. Code Units (1 CU = 6.96×10^{10} cm)

Name : tc_dr
Description: Interval during which collisional thermal conduction is turned off
Value : ≥ 0
Default : 5. Code Units (1 CU = 6.96×10^{10} cm)

Name : tc_fac0
Description: Factor multiplying thermal conduction below tc_r
Value : ≥ 0
Default : 1.

Name : tc_fac1
Description: Factor multiplying thermal conduction below tc_r
Value : ≥ 0
Default : 0.

Name : tc_nocoll_r
Description: Radius around which collisionless thermal conduction is turned on
Value : Must be withing computational domain
Default : 10. Code Units (1 CU = 6.96×10^{10} cm)

Name : tc_nocoll_dr
Description: Interval during which collisional thermal conduction is turned on
Value : ≥ 0
Default : 5. Code Units (1 CU = 6.96×10^{10} cm)

Name : awthprof_th

Description: Parameter controlling Alfvèn wave pressure profile (see routine load_awthprof)
Value : $0. \leq$ awthprof_th $\leq \pi$
Default : 0.7

Name : awthprof_dth
Description: Parameter controlling Alfvèn wave pressure profile (see routine load_awthprof)
Value : Real $\geq 0.$
Default : 0.08

Name : bmin
Description: Minimum magnetic field assumed in calculating boundary velocity in emerging flux runs
Value : Real $\geq 0.$
Default : 0.05 Code Units (1 CU = 2.205 G)

Name : awthprof2_th
Description: Parameter controlling Alfvèn wave pressure profile (see routine load_awthprof)
Value : $0. \leq$ awthprof2_th $\leq \pi$
Default : $\pi - 0.7$

Name : ifimplrad
Description: Calculate radiation losses semi-implicitly
Value : 0 (do not use) or 1 (use)
Default : 1

Name : dynamic_eta
Description: Calculate local resistivity depending on the current
Value : Logical
Default : .false. (do not use)

Name : eta_min
Description: Minimum resistivity when ifeta_dyn is used
Value : $\geq 0.$
Default : 0.

Name : eta_max
Description: Maximum resistivity when ifeta_dyn is used
Value : $\geq 0.$
Default : 0.

Name : rmcell

Description: Cell magnetic Reynolds number (2 is standard upwind)
Value : Real > 0
Default : 2.

Name : rmcell_etamax
Description: Maximum allowed value for cell resistivity
Value : Real > 0
Default : 10^{20}

Name : alpha_tc
Description: Exponent used in modified thermal conduction
Value : $\geq 0.$
Default : 2.5

Name : nfilt_kappa
Description: Number of times to filter the thermal conduction tensor
Value : Integer ≥ 0
Default : 0

Name : split_visc
Description: Split viscosity advance in momentum equation
Value : Logical
Default : .true.

Name : beta0
Description: Totally useless variable
Value : Any real
Default : 0.

Name : old_jxb_diag
Description: Totally useless variable
Value : Logical
Default : .true.

Name : use_f90_loops
Description: Choose whether to use F90 syntax or not in array assignments
Value : Logical
Default : .false. (Use rank-one assignments)

Name : hdf32
Description: Select precision in HDF output files

Value : Logical
Default : .true. (32-bit reals)

Name : use_old_jxb
Description: Use old formulation to calculate the Lorentz force
Value : Logical
Default : .false.

Name : use_old_vdgv
Description: Use old formulation to calculate advection of the velocity
Value : Logical
Default : .false.

Name : ros_equil
Description: Totally useless variable
Value : Logical
Default : .false.

Name : rs_final
Description: Write restart file and the end of the run
Value : Logical
Default : .true.

Name : hs
Description: Initialize special hydrostatic equilibrium (see routine initialize_hs_equilibrium)
Value : Structure
Default : See module vars

Name : zb
Description: Initialize special zero-beta equilibrium (see routine initialize_zb_equilibrium)
Value : Structure
Default : See module vars

Name : omega1
Description: Totally useless variable
Value : Any real
Default : 0.

Name : omega2
Description: Totally useless variable
Value : Any real

Default : 0.

Name : n_ssdip
Description: Number of sub-surface dipoles to add
Value : Integer ≥ 0
Default : 1

Name : r_ssdip
Description: Radial position of sub-surface dipoles
Value : Array of real $\leq$ r0
Default : 0.8 Code Units (1 CU = 6.96×10^{10} cm)

Name : t_ssdip
Description: Latitudinal position of sub-surface dipoles
Value : Array of real $0 \leq$ t_ssdip $\leq \pi$
Default : π

Name : p_ssdip
Description: Longitudinal position of sub-surface dipoles
Value : Array of real $0 \leq$ p_ssdip $< 2\pi$
Default : π

Name : alpha_ssdip
Description: Dipole moment rotation wrt N pole of sub-surface dipoles
Value : Array of real in degrees
Default : 0.

Name : b0_ssdip
Description: Dipole strength of sub-surface dipoles
Value : Array of real ≥ 0.
Default : 1. Code Unit (1 CU =2.205 G)

Name : print_matrix_pot2d
Description: In one processor runs, print matrix in 2D potential solver
Value : Logical
Default : .false.

Name : print_matrix_pot3d
Description: In one processor runs, print matrix in 3D potential solver
Value : Logical
Default : .false.

Name : print_matrix_adva
Description: In one processor runs, print matrix in resistivity advance
Value : Logical
Default : .false.

Name : print_matrix_advv
Description: In one processor runs, print matrix in semi-implicit advance
Value : Logical
Default : .false.

Name : ijk_flux0
Description: Specify mesh points where to calculate heat flux
Value : 2D integer array specifying first diagonal mesh point for box
Default : Undefined

Name : ijk_flux1
Description: Specify mesh points where to calculate heat flux
Value : 2D integer array specifying second diagonal mesh point for box
Default : Undefined

Name : n_flux_box,
Description: Number of boxes where to calculate heat flux
Value : Integer ≥ 0
Default : 0

Name : ijk_box0
Description: Specify mesh points where to add heating
Value : 2D integer array specifying first diagonal mesh point for box
Default : Undefined

Name : ijk_box1
Description: Specify mesh points where to add heating
Value : 2D integer array specifying second diagonal mesh point for box
Default : Undefined

Name : heatscale_box
Description: Specify heat scale in the box
Value : Real array $> 0.$
Default : 0.

Name : heat_box
Description: Specify heat constant in the box
Value : Real array $> 0.$
Default : 0. Code Units (1CU = 2.6806×10^{-4} erg cm^{-3} s^{-1})

Name : n_heat_box
Description: Number of boxes with heating
Value : Integer ≥ 0
Default : 0

Name : b_fac_box
Description: Magnetic field proportionality constant for heating in box
Value : Array of reals $geq0.$
Default : 0.

Name : bexp_box
Description: Magnetic field exponent for heating in box
Value : Array of reals $geq0.$
Default : 0.

Name : ifheat_mask
Description: Flag to use a mask file for the heating (see routine heating)
Value : Logical
Default : .false. (no mask)

Name : heat_mask_file
Description: Mask file to be read for the heating (see routine heating)
Value : Characters
Default :

Name : ifcheck0temp
Description: Check whether the temperature is negative
Value : Logical
Default : .true. (check)

Name : ifeta_phot
Description: Switch to use photospheric diffusion
Value : Logical
Default : .false. (do not use)

Name : eta_phot_file

Description: File name from which to read the photospheric diffusion profile
Value : Characters
Default :

Name : rotation_flux
Description: Calculate component due to differential rotation in boundary electric field
Value : Logical
Default : .false. (do not use)

Name : bt_photo_in
Description: Transverse magnetic field advected in by normal flow, B_θ
Value : Real
Default : 0.

Name : bp_photo_in
Description: Transverse magnetic field advected in by normal flow, B_ϕ
Value : Real
Default : 0.

Name : eflux_phi
Description: Emerging flux electric field time profile factor
Value : Structure
Default : No emerging flux

Name : nfiltphi
Description: Number of times to filter the boundary potential
Value : Integer ≥ 0
Default : 5

Name : v_eflux_max
Description: When emerging flux, set characteristic velocity to zero if larger
Value : Real $\geq 0.$
Default : 1. Code Units (1 CU = 481.37 km/s)

Chapter 4

Example Runs

4.1 Overview

In this chapter we describe a suite of test runs that can be made using the code exactly as set up in Chapter 2. The input parameters for the test runs can be downloaded at:

`http://www.predsci.com/MAS/userguide/testruns/`

Currently, we provide two test cases: Carrington rotations 1913 and 1961. In the future, we will add more benchmark solutions, including 2- and 3-dimensional, time-dependent CME eruptions as well as thermodynamic solutions.

4.2 Test Case 1: Ambient Solar Corona for Carrington Rotation 1913

The input files and solutions for this run can be found at:

`http://www.predsci.com/MAS/userguide/testruns/cr1913-low/corona_mas/corona/`

The input file for this run is called "imas" and the input magnetogram is the file: br_m9_filt3.hdf.

The logfile for the run is located in the directory above this, i.e.,

```
../corona_mas/corhel.log
```

This file summarizes exactly how the magnetogram was processed prior to running the simulation, as well as details on the input parameters, the specific version of the code being run, and some basic timing information.

Once complete, you can compare your solutions, such as the output HDF files with the ones located in the directory:

```
http://www.predsci.com/MAS/userguide/testruns/cr1913-low/corona_mas/corona/
```

The numbers should match to within 10^{-6}. In the near future, we are planning to release a tool that will compare the HDF files directly.

4.3 Test Case 2: Ambient Solar Corona for Carrington Rotation 1961

The previous test case represented the solar corona under solar minimum conditions. Test case 2 was chosen because it occurred at the maximum of the solar activity cycle. The solution is located at:

```
http://www.predsci.com/MAS/userguide/testruns/cr1913-low/corona_mas/
```

The input files and log files are in the corresponding locations, as described in the previous section.

Chapter 5

The Physics Behind MAS

5.1 Overview

In this chapter, we summarize the main physical principles used to guide our development of the MAS code. In the next chapter we describe how these principles were implemented in the code and how the equations were solved numerically.

5.2 The MHD Equations

The SAIC coronal code MAS solves the following equations in spherical coordinates:

$$\nabla \times \mathbf{B} = \frac{4\pi}{c}\mathbf{J}, \tag{5.1}$$

$$\nabla \times \mathbf{E} = -\frac{1}{c}\frac{\partial \mathbf{B}}{\partial t}, \tag{5.2}$$

$$\mathbf{E} + \frac{1}{c}\mathbf{v} \times \mathbf{B} = \eta \mathbf{J}, \tag{5.3}$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \tag{5.4}$$

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = \frac{1}{c}\mathbf{J} \times \mathbf{B} - \nabla p - \nabla p_w + \rho \mathbf{g} + \nabla \cdot (\nu \rho \nabla \mathbf{v}), \tag{5.5}$$

$$\frac{1}{\gamma-1}\left(\frac{\partial T}{\partial t}+\mathbf{v}\cdot\nabla T\right)=-\,T\nabla\cdot\mathbf{v}+S, \tag{5.6}$$

$$S=\frac{1}{2kn_e}\left(-\nabla\cdot\mathbf{q}-n_e n_p Q(T)+H+H_d+D\right) \tag{5.7}$$

where $\mathbf{B}$ is the magnetic field, $\mathbf{J}$ is the current density, $\mathbf{E}$ is the electric field, ρ, $\mathbf{v}$, p, and T are the plasma mass density, velocity, pressure, and temperature, n_e and n_p are the electron and proton density, respectively, and p_w is the wave pressure and represents the acceleration due to waves. The gravitational acceleration is $\mathbf{g}=-g_0\hat{\mathbf{r}}R_s^2/r^2$, R_s is the solar radius, $\gamma=5/3$ is the ratio of specific heats, η is the resistivity, and ν is the viscosity. The plasma pressure is $p=(n_e+n_p)kT$, where for a hydrogen plasma, $n_e=n_p$, and $\rho=m_p n_p$, where m_p is the proton mass. In practice, the vector potential $\mathbf{A}$ is used to advance the equations. These equations are solved on nonuniform meshes that allow us to concentrate grid points in regions of interest. The method of solution, including the boundary conditions, has been described previously (**??????**).

The model can be applied to both ideal MHD, in which case the plasma resistivity, η is set to zero, as well as resistive MHD, where η is finite.

Although MAS is a fully 3-D, global model, it can be applied to 2-D configurations by truncating the number of points in longitude, ϕ. In the serial version of MAS, because of the pseudo-spectral treatment in ϕ, the number of points in longitude can be reduced to 1. In the parallel version, however, because a finite difference approach is used in all three dimensions, a minimum of 3 points must be retained in ϕ. MAS cannot be used to model two dimensions in either the equatorial plane (r,$\theta=\pi/2$,ϕ) or on a spherical shell ($r=r_o$,θ,ϕ).

While MAS is typically used to solve the full MHD equations, it is often constructive to simplify the problem by implementing the so-called zero-beta model. In this case, plasma pressure and gravitational forces are neglected ($P=\mathbf{g}=0$), and we assume a fixed plasma density profile, $\rho(r)$. This ensures that equilibria are force-free ($\mathbf{J}\times\mathbf{B}=0$) and the modeler can compare the numerical results with both analytical and numerical solutions of force-free fields.

In the energy equation (5.6)–(5.7), H is the coronal heating source, $H_d=\eta J^2+\nu\nabla\mathbf{v}\colon\nabla\mathbf{v}$ is the heating due to resistive and viscous dissipation, D is the heating due to dissipation of Alfvén waves, and $Q(T)$ is the radiation loss function (e.g., **??**). In the collisional regime (below $\sim 10R_s$), the heat flux is given by $\mathbf{q}=-\kappa_\parallel\hat{\mathbf{b}}\hat{\mathbf{b}}\cdot\nabla T$, where $\hat{\mathbf{b}}$ is the unit vector along $\mathbf{B}$, and $\kappa_\parallel=9\times10^{-7}T^{5/2}$ is the Spitzer value of the parallel thermal conductivity (in cgs units). In the collisionless regime (beyond $\sim 10R_s$), the heat flux is given by $\mathbf{q}=\alpha n_e kT\mathbf{v}$, where α is a dimensionless parameter of order 1 (**?**). We refer to our model that solves this more detailed energy equation as the "thermodynamic" MHD model.

The (unknown) coronal heating source H is a parameterized function. **?** describe how we have evaluated different coronal heating models to see which can yield a reasonable match to EUV and X-ray emission.

Since the acceleration of the solar wind by Alfvén waves occurs on spatial and time scales below those of our global numerical model, the wave pressure p_w is evolved using an equation for the time-space averaged Alfvén wave energy density ε (**??**).

A principal difficulty in solving the equations above is that extremely steep temperature and density gradients arise in the transition region, a consequence of the balance between conduction of heat from the hot corona and radiation loss in the transition region. Since typical scale lengths for the transition region can be as small as 1 km, this makes calculations with a transition region very challenging, even on massively parallel computers. To make these calculations feasible, we have developed a way of artificially broadening the transition region. This is accomplished by modifying the thermal conductivity $\kappa(T)$ and radiation loss function $Q(T)$ in the lower part of the transition region in such a way that the coronal solution is not modified. This artificial broadening of the transition region affects the detailed temperature and density profiles within the lower transition region, and hence the emission characteristics there, but it does not significantly change coronal emission. We have demonstrated this in 1D models of static and dynamic loops, and 2D MHD models, by comparing with calculations performed with the true $\kappa(T)$ and $Q(T)$.

5.3 Initial and Boundary Conditions for Ambient Corona Calculations

The lower boundary of the polytropic model is defined as the "photospheric" boundary at $r = R_S$. This should be regarded as the base of the corona, that is just above the transition region, since this simplified model cannot hope to resolve the scale height of the photosphere. Since our primary concern is to model large-scale structure and evolution in the solar corona, this approximation, is likely justified.

The resistive MHD equations require specification of five of the 8 parameters (three components of both magnetic field and velocity, density, and pressure). Typically we choose to specify B_r, $\mathbf{E}_t$, rho, and T). The remaining parameters are computed self-consistently.

The two external boundaries are: (1) a sphere at $r = R_o$, typically a some point above the photosphere, say at the base of the corona, in the chromosphere, or at the base of the transition region; and (2) a sphere at $r = R_{max}$, where r_{max} is some

point outside the critical wave surfaces (i.e., the sound, Alfvén, and magnetosonic points). This could be as little as $20R_S$ or as much as $5AU$. Beyond this, the effects of pick-up ions (which are not presently included in the code) may play a non-negligible role. In practice, we rarely run a single model beyond $50 - 100R_S$ because of computational efficiency. We have found it better to split the region into two distinct regions: the corona and heliosphere, with a boundary typically set to $30R_S$.

The internal boundaries lie at $\theta = 0$ and 180°. Appropriate geometric conditions are applied using analytic functions.

Typical values for the plasma and magnetic field parameters at $r = R_S$ and the equator are: Magnetic field strength, $B_o = 2.2$ Gauss; mass density, $\rho_o = 1.67 \times 10^{-16}$ g cm^{-3}, which corresponds to an ion and electron number density, $n_o = 10^8$ cm^{-3}; temperature, $T_o = 1.4 \times 10^6$ K (defined by the ideal gas law, $P_o = 2n_o k T_o$). For polytropic runs, the ratio of specific heats, γ, is set to 1.05. Although this cannot be justified on any physical grounds (γ is clearly 5/3 in reality), it leads to a corona that is relatively isothermal, as is observed. To relax this limitation requires incorporation of various heating terms, which are included in the thermodynamic model. Using these values, we can compute the following parameters at the equator, at $r = R_S$: Sound speed, $c_s = 157$km s^{-1}; Alfvén speed; $v_A = 470$ km s-1; and $\beta = 0.2$. Similarly, at the poles: Sound speed, Alfvén speed; $v_A = 940$ km s-1; and $\beta = 0.05$. The Alfvén time, $\tau = 1,449$ s (24.2 minutes). It is worth noting that the plasma and magnetic field parameters in the code have all been renormalized using these values. Thus speeds in the code must be multiplied by 470 to convert to km s^{-1} and magnetic fields must be multiplied by 2 to convert to Gauss. The conversion of all parameters into cgs and rationalized MKS units is given in Appendix B.

In our thermodyamic model the lower radial boundary coincides with the upper chromosphere.

The initial and boundary conditions are described in detail by **?**. The key boundary condition required from observations is the radial magnetic field at the solar surface, B_{r0}. This must be supplemented with conditions on the plasma temperature and density at $r = R_s$, where we typically choose a temperature $T_0 = 20{,}000$ K and a number density $n_0 = 2 \times 10^{12}$ cm^{-3}, representative of the upper chromosphere. In the thermodynamic MHD model, the choices for coronal heating determine the properties of the solutions, and the exact choice of T_0 and n_0 are not crucial as long as n_0 is large enough to maintain a chromosphere in the presence of the specified heating. The velocity parallel to the magnetic field cannot be specified at $r = R_s$ in a well-posed problem; the boundary conditions on the velocity are determined from the characteristic equations along $\mathbf{B}$. The surface magnetic flux is specified from observations. At the upper radial boundary, which is placed beyond the sonic and

Alfvén points, the characteristic equations are used as well.

For the initial condition, we start by computing a potential magnetic field in the corona that matches B_{r0} at the solar surface. We impose a spherically symmetric solar wind solution and integrate the time-dependent MHD equations in time until the solution settles down to an equilibrium. Helmet streamers with closed field lines form, surrounded by open field lines along which the solar wind flows outward.

5.4 Initial and Boundary Conditions for prominence and CME Eruptions

The initial conditions for CME computations is similar to the preceding discussion. The equations are advanced forward in time using suitable boundary conditions until a steady state is reached. At this point, time-dependent variations are applied at the boundary to build up stresses in the coronal magnetic field, which lead to the eruption of a CME and/or prominence.

Chapter 6

The Numerical Techniques Implemented in MAS

6.1 Introduction

In this chapter we briefly describe the numerical techniques used to solve the time-dependent MHD equations discussed in the previous chapter.

6.2 Meshes

The use of staggered meshes is a common technique in computational fluid dynamics and electrodynamics. It facilitates the specification of well-posed boundary conditions; at the lower boundary R_0, typically only the magnetic field normal to R_0 and the electric field tangential to R_0 are allowed to be specified. Other components are computed at boundary in the same way as interior grid points. Staggered meshes preserve the vector identities $\nabla \cdot \nabla \times = 0$ and $\nabla \times \nabla = 0$, which makes it possible to maintain $\nabla \cdot \mathrm{B} = 0$ $\nabla \cdot \mathrm{J} = 0$ to machine accuracy. The implementation of second order accurate derivatives or first order upwind derivatives is straightforward as well.

The original MAS code (a serial, vectorized code) uses a staggered mesh in r and θ. Grid points in r and θ can be spaced nonuniformly. Derivatives in the ϕ direction are computed using a pseudo-spectral method, which requires a uniform ϕ mesh. Figure 6.2 shows the location of quantities on the staggered r and θ.

The parallel MAS code (allows 3D domain decomposition with MPI) uses a staggered mesh in all three dimensions; Figure 6.2 shows the location of quantities for two cells of the staggered 3D mesh. All quantities map to the same locations as

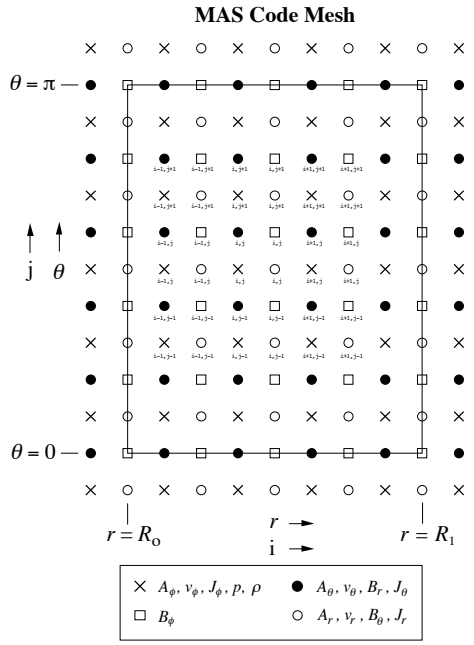

Figure 6.1: Staggering of the original MAS mesh in the (r,θ) plane.

in Figure 6.2 if the 3D mesh is collapsed into a plane.

6.3 Temporal Approximation

The right-hand sides are equations (1) - (4) have advective, dissipative, and wave-like terms that are treated using predictor-corrector, implicit, and semi-implicit techniques. A leap-frog scheme is implemented to discretize the various fields (**A**, ρ, p, and **v**) in time.

6.4 Self-adjoint Representation of the Diffusive and Semi-Implicit Terms

The differential operators in the MHD equations are self-adjoint. As such, advancing the equations in time implicitly requires solving the following algebraic equation: $Ax = b$, where A is the coefficient matric, x is the unknown vector, and b is the known term. It turns out that A is both self-adjoint and positive-definite for the

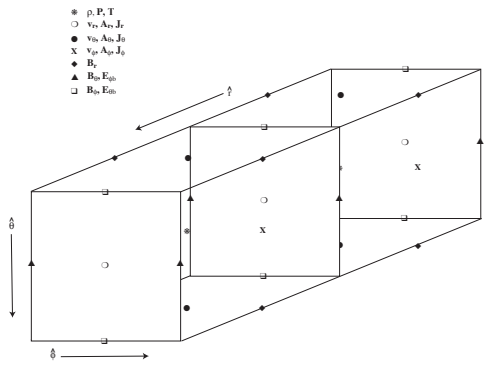

Figure 6.2: The 3D staggered mesh used in the parallel MAS code.

operators in our equations. As such, we can implement the conjugate gradient algorithm to efficiently compute the solution x. In comparison to the biconjugate gradient method, we have found it to be an order of magnitude faster.

6.5 Renormalization

To maximize the numerical precision of the result, the MHD variables are renormalized into non-dimensional quantities. The renormalization constants are chosen such that they are indicative of the typical value of that parameter in the solar corona. Hence most variables in "code" units have values near one. The conversion from "code" units to both cgs and rationalized MKS units is summarized in Table xxx. It is worth emphasizing that it is easy to recognize some parameters as being in code units if the renormalization constant is large. Speed, for example has to be multiplied by 481 to obtain values in kilometers per second. On the other hand, the conversion of magnetic field from "code" units to Gauss involves only a multiplication of 2.205. Thus one should be careful to keep track of whether one has converted to real units or not.

6.6 The Zero-Beta Model

The model can be "short-circuited" to approximate the behavior of a low-beta plasma by setting $p = 0$ and $\mathbf{g} = 0$ and further assuming a fixed plasma density profile $\rho(\mathbf{r}$. In this configuration, all magnetic phenomena in the MHD equations are still retained. In addition to simplifying the physical processes, making the

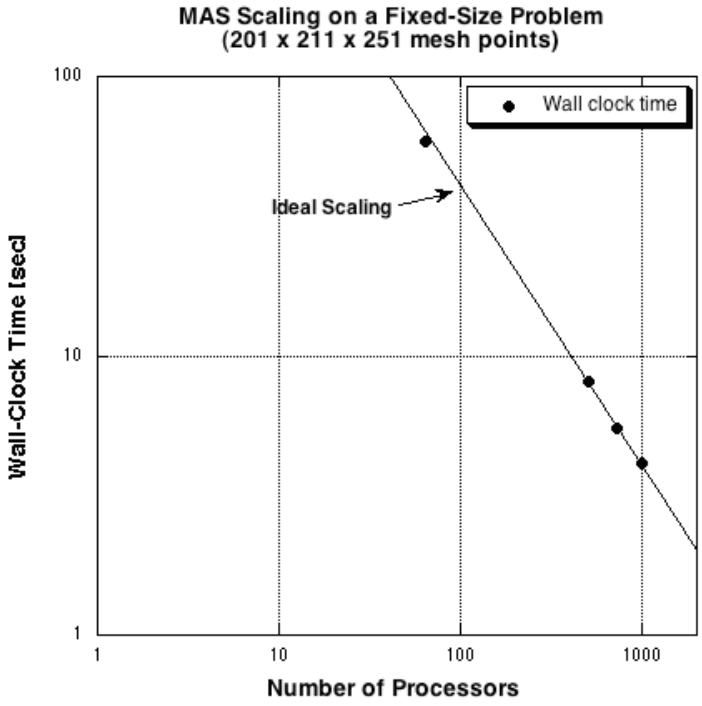

Figure 6.3: MAS scaling for a fixed-size problem.

code more robust, and speeding up the time of execution, this approximation has the property that equilibria formed are force free, that is, $\mathbf{J} \times \mathbf{B} = 0$. Thus one can compare MAS zero-beta equilibria with analytic or numerical calcuations of force-free fields.

6.7 The Performance of MAS

Figure 6.7 shows how MAS scales for a fixed-size problem as the number of processors increases from 64 to 1000. The solid line denotes what perfect scaling would look like. At least for problem sizes of the order of 200^3, the code scales extremely well.

Bibliography

Appendix A

Peer-Reviewed Articles about the MAS Code

- Z. Mikic and J. A. Linker, Disruption of coronal magnetic field arcades, Astrophys. J., 430, 898, 1994.
- J. A. Linker and Z. Mikic, Evolution and disruption of magnetic arcades, in Solar Active Region Evolution: Comparing Models with Observations (K. S. Balasubramaniam and G. W. Simon, Eds.) A.S.P. Conf. Series, 68, 251, 1994.
- J. A. Linker, Z. Mikic, and D. D. Schnack, Modeling coronal evolution, in Proc. 3rd SOHO Workshop-Solar Dynamic Phenomena and Solar Wind Consequences, Estes Park, Colorado, ESA SP-373, 249, 1994.
- J. A. Linker and Z. Mikic, Disruption of a helmet streamer by photospheric shear, Astrophys. J., 438, L45, 1995.
- R Lionello,., Z. Mikic, and J. A. Linker, Stability of algorithms for waves with large flows," J. Comp. Phys. 152, 346, 1999.
- Z. Mikic, J. A. Linker, D. D. Schnack, R. Lionello, and A. Tarditi, "Magnetohydrodynamic modeling of the global solar corona," Phys. Plasmas, 6, 2217, 1999.
- R. Lionello, J. A. Linker, and Z. Mikic, Including the Transition Region in Models of the Large-Scale Solar Corona, Astrophys. J., 546, 542 (2001).

Peer-Reviewed Articles demonstrating the Application of MAS to Solar and Heliospheric Problems

- Z. Mikic, J. A. Linker, and D. D. Schnack, Modeling of active-region magnetic fields, in Solar Drivers of Interplanetary and Terrestrial Disturbances, (K. S.

Balasubramaniam, S. L. Keil, and R. N. Smartt, eds.), Astron. Soc. Pac. Conf., 95, 108, 1996.

- Z. Mikic and J. A. Linker, Large-scale structure of the solar corona and inner heliosphere, Solar Wind 8, in AIP Conf. Proceedings, 382, 104, 1996.

- J. A. Linker, Z. Mikic, and D. D. Schnack, Global coronal modeling and space weather prediction, in Solar Drivers of Interplanetary and Terrestrial Disturbances, (K. S. Balasubramaniam, S. L. Keil, and R. N. Smartt, eds.), Astron. Soc. Pac. Conf., 95, 208, 1996.

- J. A. Linker and Z. Mikic, Extending coronal models to Earth orbit, in Coronal Mass Ejections: Causes and Consequences, Geophysical Monograph, 99, 269, 1997.

- Z. Mikic and J. A. Linker, The initiation of coronal mass ejections by magnetic shear, in Coronal Mass Ejections: Causes and Consequences, Geophysical Monograph, 99, 57, 1997.

- M. R. Neugebauer, J. Forsyth, A. B. Galvin, K. L. Harvey, J. T. Hoeksema, A. J. Lazarus, R. P. Lepping, J. A. Linker, Z. Mikic J. T. Steinberg, R. von Steiger, Y. M. Wang, and R. Wimmer-Schweingruber, The Spatial Structure of the Solar Wind and Comparisons with Solar Data and Models, J. Geophys. Res., 103, 14587, 1998.

- J. A. Linker, Z. Mikic, D. A. Biesecker, R. J. Forsyth, S. E. Gibson, A. J. Lazarus, A. Lecinski, P. Riley, A. Szabo, and B. J. Thompson, Magnetohydrodynamic Modeling of the Solar Corona During Whole Sun Month, J. Geophys. Res., 104, 9809, 1999.

- Z. Mikic, J. A. Linker, P. Riley, and R. Lionello, Predicting the Structure of the Solar Corona During the 11 August 1999 Total Solar Eclipse, in The Last Total Solar Eclipse of the Millennium (W. Livingston and A. O zg, eds.), Astronomical Society of the Pacific Conference Series, Vol. 205, p. 162 (2000).

- P. Riley, J. A. Linker, Z. Mikic, and R. Lionello, MHD Modeling of the Solar Corona and Inner Heliosphere: Comparison with Observations, in Space Weather (P. Song, G. Siscoe, and H. Singer, eds.), American Geophysical Union, Geophysical Monograph 125, 159 (2001).

- P. Riley, J. A. Linker, and Z. Mikic, An Empirically-Driven Global MHD Model of the Solar Corona and Inner Heliosphere, J. Geophys. Res., 106, 15,889 (2001).

- J. A. Linker, R. Lionello, Z. Mikic, and T. Amari, Magnetohydrodynamic Modeling of Prominence Formation Within a Helmet Streamer, J. Geophys. Res., 106, 25,165, (2001).

- P. Riley, J. A. Linker, and Z. Mikic, Modeling the Heliospheric Current Sheet: Solar Cycle Variations, J. Geophys. Res., 107, 1136, 10.1029/2001JA000299 (2002).

- P. Riley, J. A. Linker, Z. Mikic, D. Odstrcil, V. J. Pizzo, and D. F. Webb, Evidence of Post-Eruption Reconnection Associated with Coronal Mass Ejections in the Solar Wind, Astrophys. J., 578, 972 (2002).

- R. Lionello, Z. Mikic, J. A. Linker, and T. Amari, Magnetic Field Topology in Prominences, Astrophys. J., 581, 718 (2002).

- R. Lionello, J. A. Linker, and Z. Mikic, Three-Dimensional Magnetohydrodynamics of the Solar Corona and of the Solar Wind with Improved Energy Transport, in Solar Wind Ten: Proc. Tenth Intl. Solar Wind Conf. (M. Velli, R. Bruno, and F. Malara, eds.), AIP Conf. Proceedings 679, AIP Press, NY, pp. 222225 (2003).

- P. Riley, Z. Mikic, J. A. Linker, and T. H. Zurbuchen, Understanding the Solar Sources of in situ Observations, in Solar Wind Ten: Proc. Tenth Intl. Solar Wind Conf. (M. Velli, R. Bruno, and F. Malara, eds.), AIP Conf. Proceedings 679, AIP Press, NY, pp. 7982 (2003).

- J. A. Linker, Z. Mikic, P. Riley, R. Lionello, and D. Odstrcil, Models of Coronal Mass Ejections: A Review with a Look to the Future, in Solar Wind Ten: Proc. Tenth Intl. Solar Wind Conf. (M. Velli, R. Bruno, and F. Malara, eds.), AIP Conf. Proceedings 679, AIP Press, NY, pp. 703710 (2003).

- P. Riley, J. A. Linker, Z. Mikic, D. Odstrcil, T. H. Zurbuchen, D. A. Lario, and R. P. Lepping, Using an MHD Simulation to Interpret the Global Context of a Coronal Mass Ejection Observed by Two Spacecraft, J. Geophys. Res., 108, 1272, 10.1029/2002JA009760 (2003).

- P. Riley, Z. Mikic, and J. A. Linker, Dynamical Evolution of the Inner Heliosphere Approaching Solar Activity Maximum: Interpreting Ulysses Observations Using a Global MHD Model, Annales Geophysicae, 21, 1347 (2003).

- J. A. Linker, Z. Mikic, R. Lionello, P. Riley, T. Amari, and D. Odstrcil, Flux Cancellation and Coronal Mass Ejections, Phys. Plasmas, 10, 1971 (2003).

- J. G. Luhmann, S. C. Solomon, J. A. Linker, J. G. Lyon, Z. Mikic, D. Odstrcil, W. Wang, and M. Wiltberger, Coupled Model Simulation of a Sun-to-Earth Space Weather Event, J. Atmos. Solar-Terrestrial Phys., 66, 1243 (2004).

- P. Riley, J. A. Linker, R. Lionello, Z. Mikic, D. Odstrcil, M. A. Hidalgo, Q. Hu, R. P. Lepping, B. J. Lynch, and A. Rees, Fitting Flux Ropes to a Global MHD Solution: A Comparison of Techniques, J. Atmos. Solar-Terrestrial Phys., 66, 1321 (2004).

- P. Riley, J. A. Linker, Z. Mikic, and D. Odstrcil, Magnetohydrodynamic Modeling of Interplanetary CMEs, IEEE Trans. Plasma Sci., 32, 1415, (2004).

- R. Lionello, P. Riley, J. A. Linker, and Z. Mikic, The Effects of Differential Rotation on the Magnetic Structure of the Solar Corona: Magnetohydrodynamic Simulations, Astrophys. J., 625, 463 (2005).

- J. A. Linker, R. Lionello, Z. Mikic, and P. Riley, Time-Dependent Response of the Large-Scale Solar Corona, in Proc. of the International Scientific Conference on Chromospheric and Coronal Magnetic Fields (ESA SP-596). 30 Aug. - 2 Sept., 2005, Katlenburg-Lindau, Germany (D.E. Innes, A. Lagg & S.K. Solanki, eds), published on CDROM, p.28.1, (2005).

- P. Riley, Modeling interplanetary CMEs, Coronal and Stellar Mass Ejections, IAU Symposium Proceedings of the International Astronomical Union 226, 389-402, (2005).

- P. Riley et al., A Comparison between Global Solar Magnetohydrodynamic and Potential Field Source Surface Model Results, Ap. J., 653, 1510, (2006).

- P. Riley, J. A. Linker, Z. Mikic, and D. Odstrcil, Modeling interplanetary coronal mass ejections, Adv. Space. Res., 38, 535, (2006).

- P. Riley, and J. T. Gosling, On the origin of near-radial magnetic fields in the heliosphere: Numerical simulations, J. Geophys. Res., 112, A06115, doi:10.1029/2006JA012210, (2007).

- P. Riley et al., "Bursty" reconnection during solar eruptions: MHD Simulations and comparison with observations, Ap. J., 655, 591, (2007).

- P. Riley, Modeling corotating interaction regions: From the Sun to 1 AU, JASTP, 69, 32, (2007).

- P. Riley, R. Lionello, Z. Mikic, and J. A. Linker, Using Global Simulations to Relate the Three-part Structure of Coronal Mass Ejections to in situ Signatures, Ap. J., in press, 672, 1221, (2008).

Relevant Websites

- Predictive Science's Solar and Heliospheric Group's Main Page: http://www.predsci.com
- The Predictive Science STEREO Modeling Support Page: http://www.predsci.com/stereo
- The Predictive Science MAS Modeling Page: http://www.predsci.com/modeling.html

Appendix B

Conversion Table for MAS parameters

The following table provides the necessary constants to convert from MAS's internal code units to something more meaningful. To use them, simply multiply the raw MAS output by the appropriate constant.

Table 6.1: Converting from MAS code units to cgs and MKS.

Parameter	to convert to cgs	to convert to MKS
Length, l	6.96×10^{10} cm	6.96×10^{8} m
Time, t	1445.87 s	1445.87 s
Velocity, v	481.3711 km/s	481.3711 km/s
Electron density, n_e	10^{8} cm^{-3}	10^{14} m^{-3}
Mass density, ρ	1.6726×10^{-16} g/cm^3	1.6726×10^{-13} kg/m^3
Pressure, p	0.3875717 dyn/cm^2	3.875717×10^{-2} Pa
Temperature, T	2.807067×10^{7} K	2.807067×10^{7} K
Magnetic field, B	2.2068908 Gauss	2.2068908×10^{-4} T
Resistivity, η	0.0467796 s	—
Electrical Diffusivity, η/μ_o	—	3.3503427×10^{8} km^2/s
Current Density, J	.07558 statamp/cm^2	2.267×10^{4} A/m^2