DRAGONCELLO code

(fully anisotropic 2D CR transport equation)

Documentation

Daniele Gaggero, Silvio Sergio Cerri

Features of currently released version

Version 1.0:

➢ two-dimensional (**2D**) **CR-transport equation** in cylindrical coordinates

➢ **fully anisotropic diffusion tensor** (with respect to local Galactic magnetic-field direction)

➢ axis-symmetric version of state-of-the-art **Galactic magnetic-field (GMF) model**

➢ realistic source distribution of CRs

➢ simplified energy losses description (IC-loss model for leptons only)

DRAGON model equations

reference:

Cerri *et al.*, *J. Cosmol. Astropart. Phys.* **10**, 019 (2017)

Transport equation:

$$\frac{\partial N}{\partial t} = \boldsymbol{\nabla} \cdot (\mathbf{D} \cdot \boldsymbol{\nabla} N) + S = \frac{\partial}{\partial x_i}\left(D_{ij}\frac{\partial N}{\partial x_j}\right) + S$$

Diffusion tensor:

$$D_{ij} \equiv D_\perp \delta_{ij} + (D_\parallel - D_\perp) b_i b_j\,, \qquad b_i \equiv \frac{B_i}{|\mathbf{B}|}$$

Diffusion coefficients:

$$D_\parallel = D_{0\parallel}\left(\frac{p_{\mathrm{GeV}}}{Z}\right)^{\delta_\parallel} \quad \text{and} \quad D_\perp = D_{0\perp}\left(\frac{p_{\mathrm{GeV}}}{Z}\right)^{\delta_\perp} \equiv \epsilon_D\, D_{0\parallel}\left(\frac{p_{\mathrm{GeV}}}{Z}\right)^{\delta_\perp}$$

Source term:

$$S(R,z) = \left(\frac{R}{R_\odot}\right)^a \exp\left(-b\frac{R - R_\odot}{R_\odot} - \frac{|z|}{z_0}\right)$$

GMF model:

$$B_\phi^{\mathrm{disk}}(R,z) = \begin{cases} B_{D0}\, e^{-|z|/z_0} & (R < R_{cD}) \\ B_{D0}\, e^{-|z|/z_0}\, e^{-(R-R_\odot)/R_0} & (R > R_{cD}) \end{cases}$$

$$B_\phi^{\mathrm{halo}}(R,z) = B_{H0}\left[1 + \left(\frac{|z| - z_0^H}{z_1^H}\right)\right]^{-1} \frac{R}{R_O^H}\, e^{\left(1 - \frac{R}{R_0^H}\right)}$$

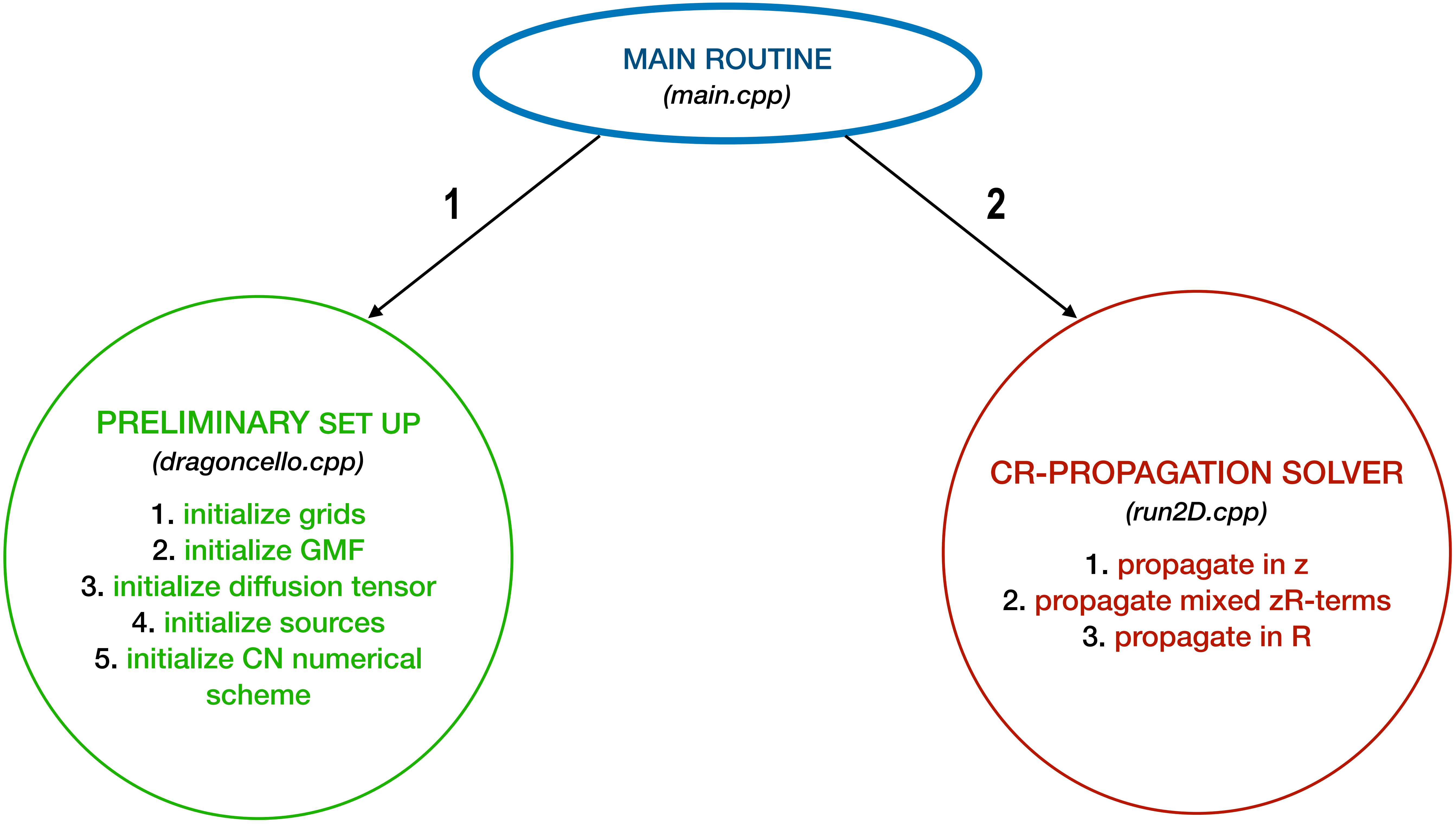
$$B_z^{\mathrm{pol}}(R,z) = B_{\mathrm{X}}(R,z) \cos\left[\Theta_{\mathrm{X}}(R,z)\right]$$

$$B_R^{\mathrm{pol}}(R,z) = B_{\mathrm{X}}(R,z) \sin\left[\Theta_{\mathrm{X}}(R,z)\right]$$

$$B_{\mathrm{X}}(R,z) = \begin{cases} B_{\mathrm{X}}^0 \left(\frac{R_p}{R}\right)^2 e^{-R_p/R_{\mathrm{X}}} & (R \le R_{\mathrm{X}}^c) \\ B_{\mathrm{X}}^0 \left(\frac{R_p}{R}\right) e^{-R_p/R_{\mathrm{X}}} & (R > R_{\mathrm{X}}^c) \end{cases},$$

$$\Theta_{\mathrm{X}}(R,z) = \begin{cases} \tan^{-1}\left(\frac{|z|}{R - R_p}\right) & (R \le R_{\mathrm{X}}^c) \\ \Theta_{\mathrm{X}}^0 & (R > R_{\mathrm{X}}^c) \end{cases},$$

Schematic structure of the code



Simulation parameters

(constants.h — to be modified ***before*** compilation)

☞ Source selection *(un-comment desired one, comment the others)*

- GREEN_FUNCTION
- ANALYTIC_SOLUTION_TEST
- REALISTIC_SOURCE

☞ GMF selection *(un-comment desired one, comment the others)*

- USER_DEFINED_BFIELD (to be defined in *initBfield* within dragoncello.cpp)
- FARRAR_SIMPLE_2D

☞ Parallel execution

- comment/un-comment #define PARALLEL
- NUM_THREADS = select number of threads to use for parallel option

☞ Timestep implementation

- the general algorithm starts with a timestep dt = dtmax, and after Nrept cycles the dt is reduced by a factor dtfactor. The algorithm stops if dt reaches dtmin OR after a multiple of Nrept iterations set by interruptAfter. ***NOTICE***: since dtmax should not exceed diffusion timescale across grid cells anyway for numerical stability, we reccomend to use dtfactor = 1 (so that the algorithm will perform exactly Nrept*interruptAfter time steps).

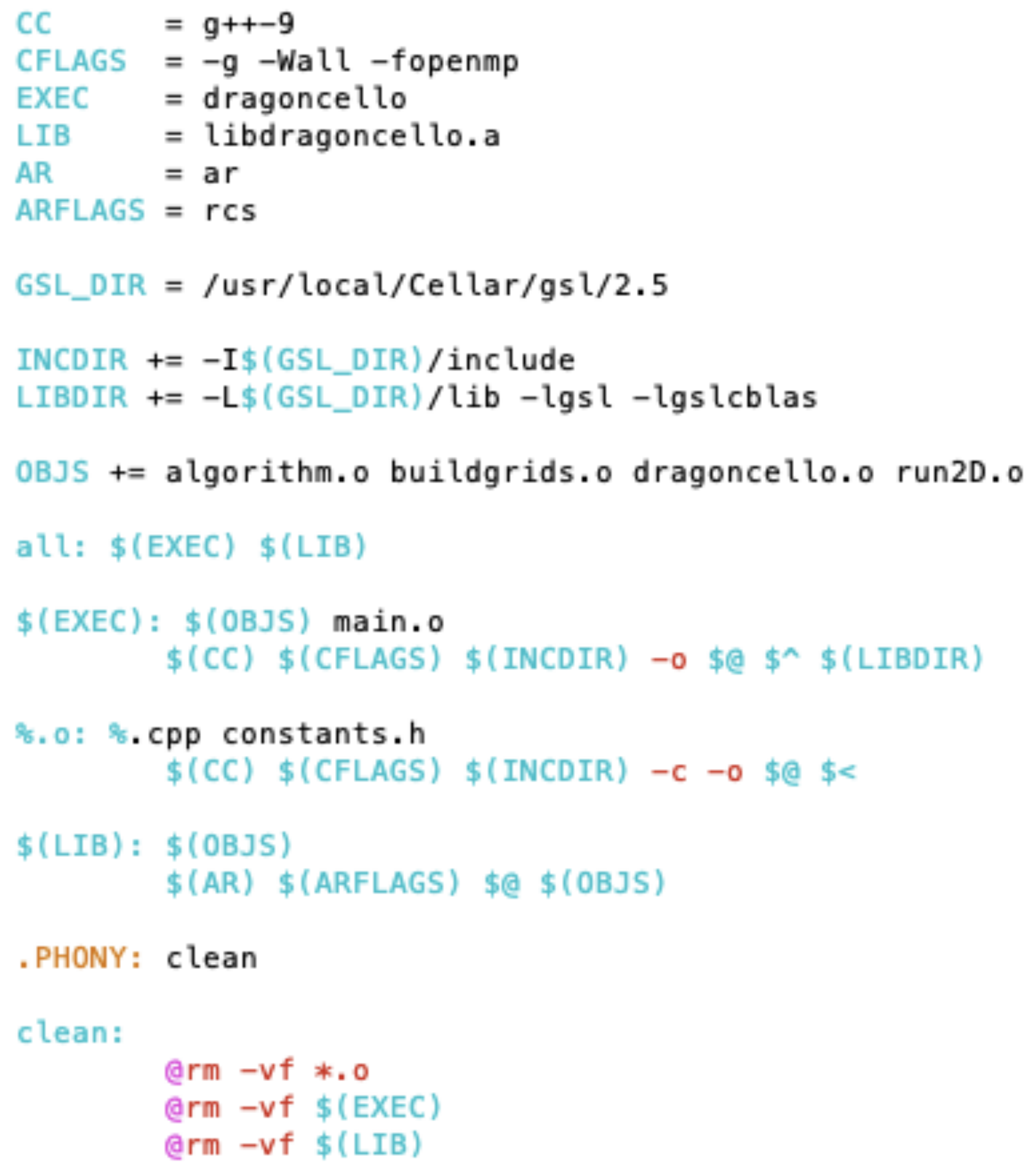
How to compile

Required libraries: **GSL**

Makefile configuration: edit line “GSL_DIR = ” with your GSL directory path

☞ run “make” command and check that executable “**dragoncello**” is created.

Makefile example:



```
CC       = g++-9
CFLAGS   = -g -Wall -fopenmp
EXEC     = dragoncello
LIB      = libdragoncello.a
AR       = ar
ARFLAGS  = rcs

GSL_DIR = /usr/local/Cellar/gsl/2.5

INCDIR += -I$(GSL_DIR)/include
LIBDIR += -L$(GSL_DIR)/lib -lgsl -lgslcblas

OBJS += algorithm.o buildgrids.o dragoncello.o run2D.o

all: $(EXEC) $(LIB)

$(EXEC): $(OBJS) main.o
        $(CC) $(CFLAGS) $(INCDIR) -o $@ $^ $(LIBDIR)

%.o: %.cpp constants.h
        $(CC) $(CFLAGS) $(INCDIR) -c -o $@ $<

$(LIB): $(OBJS)
        $(AR) $(ARFLAGS) $@ $(OBJS)

.PHONY: clean

clean:
        @rm -vf *.o
        @rm -vf $(EXEC)
        @rm -vf $(LIB)
```

How to run

Step #1: in the folder where “dragoncello” will be executed, create a folder named “**output**”

Step #2: run the executable (in background) with “`./dragoncello > log.txt &`”

☞ check execution status in log.txt file and/or output status in “output/“ folder

➥ check in log.txt that “diffusion timescales” are not larger than chosen timestep!

Questions and bug reports

Contacts:

Silvio Sergio Cerri (scerri@astro.princeton.edu)
Department of Astrophysical Sciences, Princeton University, USA

Daniele Gaggero (daniele.gaggero@uam.es)
Instituto de Física Teórica, UAM-CSIC, Madrid, Spain