Proyecto de Modelado Matemático I: Visualización 3D en VisIt

Paula Camila Wandurraga Sanabria

Maestría en Matemática Aplicada
Grupo de Investigación en Relatividad y Gravitación
Universidad Industrial de Santander

Planteamiento del problema

The Astrophysical Journal Supplement Series, 218:24 (35pp), 2015 June doi:10.1088/0067-0049/218/2/24

CAFE: A NEW RELATIVISTIC MHD CODE

F. D. Lora-Clavijo[1], A. Cruz-Osorio[1], and F. S. Guzmán[2]
[1] Instituto de Astronomía, Universidad Nacional Autónoma de México, AP 70-264, Distrito Federal 04510, México; fdlora@astro.unam.mx
[2] Instituto de Física y Matemáticas, Universidad Michoacana de San Nicolás de Hidalgo. Edificio C-3, Cd. Universitaria, 58040 Morelia, Michoacán, México; aosorio@astro.unam.mx, guzman@ifm.umich.mx
Received 2015 January 6; accepted 2015 April 30; published 2015 June 12

ABSTRACT

We introduce CAFE, a new independent code designed to solve the equations of relativistic ideal magnetohydrodynamics (RMHD) in three dimensions. We present the standard tests for an RMHD code and for the relativistic hydrodynamics regime because we have not reported them before. The tests include the one-dimensional Riemann problems related to blast waves, head-on collisions of streams, and states with transverse velocities, with and without magnetic field, which is aligned or transverse, constant or discontinuous across the initial discontinuity. Among the two-dimensional (2D) and 3D tests without magnetic field, we include the 2D Riemann problem, a one-dimensional shock tube along a diagonal, the high-speed Emery wind tunnel, the Kelvin–Helmholtz (KH) instability, a set of jets, and a 3D spherical blast wave, whereas in the presence of a magnetic field we show the magnetic rotor, the cylindrical explosion, a case of Kelvin–Helmholtz instability, and a 3D magnetic field advection loop. The code uses high-resolution shock-capturing methods, and we present the error analysis for a combination that uses the Harten, Lax, van Leer, and Einfeldt (HLLE) flux formula combined with a linear, piecewise parabolic method and fifth-order weighted essentially nonoscillatory reconstructors. We use the flux-constrained transport and the divergence cleaning methods to control the divergence-free magnetic field constraint.

Key words: magnetohydrodynamics (MHD) – methods: numerical – relativistic processes

1. INTRODUCTION

Models of high-energy astrophysics are closely related to relativistic fluid dynamics because most of the sources are identified with the dynamics of a gas or plasma associated with a high-energy source. In the most complex cases, the models

currently most used codes. The Cactus Einstein Toolkit, a multiusage package mounted on Cactus (Goodale et al. 2003), is capable of solving the general relativistic MHD (Mösta et al. 2014). Whisky, a code that in its most sophisticated version can evolve general relativistic resistive magnetohy-

The Astrophysical Journal, 844:57 (15pp), 2017 July 20 https://doi.org/10.3847/1538-4357/aa7a13

CrossMark

Magnus: A New Resistive MHD Code with Heat Flow Terms

Anamaría Navarro, F. D. Lora-Clavijo, and Guillermo A. González
Grupo de Investigación en Relatividad y Gravitación, Escuela de Física, Universidad Industrial de Santander, A. A. 678, Bucaramanga 680002, Colombia; ana.navarro1@correo.uis.edu.co, fadulora@uis.edu.co, guillego@uis.edu.co
Received 2017 March 28; revised 2017 June 13; accepted 2017 June 15; published 2017 July 21

Abstract

We present a new magnetohydrodynamic (MHD) code for the simulation of wave propagation in the solar atmosphere, under the effects of electrical resistivity—but not dominant—and heat transference in a uniform 3D grid. The code is based on the finite-volume method combined with the HLLE and HLLC approximate Riemann solvers, which use different slope limiters like MINMOD, MC, and WENO5. In order to control the growth of the divergence of the magnetic field, due to numerical errors, we apply the Flux Constrained Transport method, which is described in detail to understand how the resistive terms are included in the algorithm. In our results, it is verified that this method preserves the divergence of the magnetic fields within the machine round-off error ($\sim 1 \times 10^{-12}$). For the validation of the accuracy and efficiency of the schemes implemented in the code, we present some numerical tests in 1D and 2D for the ideal MHD. Later, we show one test for the resistivity in a magnetic reconnection process and one for the thermal conduction, where the temperature is advected by the magnetic field lines. Moreover, we display two numerical problems associated with the MHD wave propagation. The first one corresponds to a 3D evolution of a vertical velocity pulse at the photosphere–transition–corona region, while the second one consists of a 2D simulation of a transverse velocity pulse in a coronal loop.

Key words: magnetohydrodynamics (MHD) – methods: numerical – Sun: atmosphere

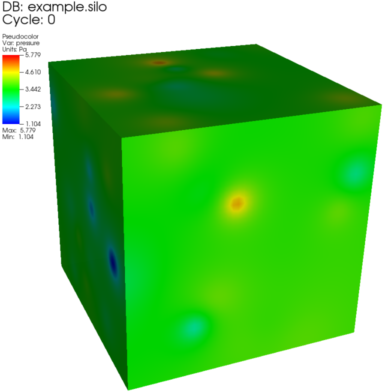
1. Introduction

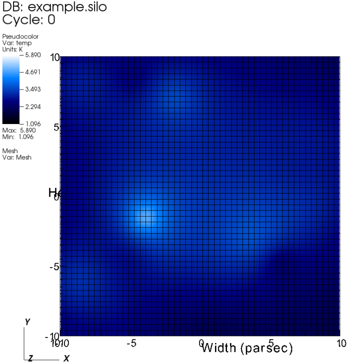
The theory of magnetohydrodynamics—the study of interactions between magnetic fields and conductive fluids in low frequencies—is of great importance for understanding the dynamics of the plasma in the solar atmosphere (Priest & Hood 1991). Since the plasma in this region is highly

atmosphere (Jess et al. 2012a). Complete reviews of observations of magnetohydrodynamic waves in solar regions like the corona, sunspots, prominences, coronal mass ejections, solar flares, and solar winds can be found in Nakariakov & Verwichte (2005), Khomenko & Collados (2015), Okamoto et al. (2007), Vršnak et al. (2013), Shibata & Magara (2011), and Ofman (2010), respectively.

```
(base) paula@PaulaCW:~/prom/Graphs_ASCII$ head -25 primitivas_1.xyzl
 #Time =    0.0000000000000000


   0.0000000000000000      -0.50000000000000000      -0.50000000000000000       3.0300707122200770E-006   9.9999758799911391E-015   1.0000023485319312E-014
  9.9999997540632175E-015   299.99983692554611        7.0897999763653063E-023   7.0897999763653063E-023   5.0000001419004654E-003
   0.0000000000000000      -0.50000000000000000      -0.47999999999999998       3.0300707122200770E-006   9.9998374590737456E-015   1.0000168665552601E-014
  1.0000000189603918E-014   299.99983692554611        7.0897999763653063E-023   7.0897999763653063E-023   5.0000001419004654E-003
   0.0000000000000000      -0.50000000000000000      -0.46000000000000002       3.0300707122200770E-006   9.9989808468348144E-015   1.0001107112894802E-014
  1.0000003004945944E-014   299.99983692554611        7.0897999763653063E-023   7.0897999763653063E-023   5.0000001419004654E-003
   0.0000000000000000      -0.50000000000000000      -0.44000000000000000       3.0300707122200770E-006   9.9941040676192648E-015   1.0006699245196090E-014
  1.0000019781342848E-014   299.99983692554611        7.0897999763653063E-023   7.0897999763653063E-023   5.0000001419004654E-003
   0.0000000000000000      -0.50000000000000000      -0.41999999999999998       3.0300707122200770E-006   9.9685720085807909E-015   1.0037413580362885E-014
  1.0000111924348348E-014   299.99983692554611        7.0897999763653063E-023   7.0897999763653063E-023   5.0000001419004654E-003
   0.0000000000000000      -0.50000000000000000      -0.40000000000000002       3.0300707122200770E-006   9.8456996948181114E-015   1.0192874667451618E-014
  1.0000578307609614E-014   299.99983692554611        7.0897999763653063E-023   7.0897999763653063E-023   5.0000001419004654E-003
   0.0000000000000000      -0.50000000000000000      -0.38000000000000000       3.0300707122200770E-006   9.3024279254236336E-015   1.0917857257959950E-014
  1.0002753255381139E-014   299.99983692554611        7.0897999763653063E-023   7.0897999763653063E-023   5.0000001419004654E-003
   0.0000000000000000      -0.50000000000000000      -0.35999999999999999       3.0300707122200770E-006   7.0968887834391687E-015   1.4032098153788638E-014
  1.0012095978068625E-014   299.99983692554611        7.0897999763653063E-023   7.0897999763653063E-023   5.0000001419004654E-003
   0.0000000000000000      -0.50000000000000000      -0.33999999999999997       3.0300707122200770E-006  -1.1186569161339387E-015   2.6350965269109879E-014
  1.0049052579414590E-014   299.99983692554611        7.0897999763653063E-023   7.0897999763653063E-023   5.0000001419004654E-003
   0.0000000000000000      -0.50000000000000000      -0.32000000000000001       3.0300707122200770E-006  -2.9173474951067055E-014   7.1208553797846292E-014
  1.0183625345000799E-014   299.99983692554611        7.0897999763653063E-023   7.0897999763653063E-023   5.0000001419004654E-003
   0.0000000000000000      -0.50000000000000000      -0.29999999999999999       3.0300707122200770E-006  -1.1690786257281141E-013   2.2151310344176631E-013
  1.0634538993932558E-014   299.99983692554611        7.0897999763653063E-023   7.0897999763653063E-023   5.0000001419004654E-003
   0.0000000000000000      -0.50000000000000000      -0.28000000000000003       3.0300707122200770E-006  -3.6783910560619354E-013   6.8471268769845656E-013
  1.2024137746702629E-014   299.99983692554611        7.0897999763653063E-023   7.0897999763653063E-023   5.0000001419004654E-003
   0.0000000000000000      -0.50000000000000000      -0.26000000000000001       3.0300707122200770E-006  -1.0231425810445261E-012   1.9968126549272344E-012
  1.5960437648388964E-014   299.99983692554611        7.0897999763653063E-023   7.0897999763653063E-023   5.0000001419004654E-003
   0.0000000000000000      -0.50000000000000000      -0.23999999999999999       3.0300707122200770E-006  -2.5823440342231712E-012   5.4107167369864601E-012
  2.6202149894566641E-014   299.99983692554611        7.0897999763653063E-023   7.0897999763653063E-023   5.0000001419004654E-003
   0.0000000000000000      -0.50000000000000000      -0.21999999999999997       3.0300707122200770E-006  -5.9528653076985586E-012   1.3561966607367233E-011
  5.0655899505708960E-014   299.99983692554611        7.0897999763653063E-023   7.0897999763653063E-023   5.0000001419004654E-003
   0.0000000000000000      -0.50000000000000000      -0.20000000000000001       3.0300707122200770E-006  -1.2546531168509553E-011   3.1401327920163171E-011
  1.0417398344409679E-013   299.99983692554611        7.0897999763653063E-023   7.0897999763653063E-023   5.0000001419004654E-003
   0.0000000000000000      -0.50000000000000000      -0.17999999999999999       3.0300707122200770E-006  -2.4154398808302749E-011   6.7133330021864337E-011
  2.1136998974920029E-013   299.99983692554611        7.0897999763653063E-023   7.0897999763653063E-023   5.0000001419004654E-003
   0.0000000000000000      -0.50000000000000000      -0.15999999999999998       3.0300707122200770E-006  -4.2387838862377795E-011   1.3250324644362160E-010
  4.0747973901447202E-013   299.99983692554611        7.0897999763653063E-023   7.0897999763653063E-023   5.0000001419004654E-003
   0.0000000000000000      -0.50000000000000000      -0.14000000000000001       3.0300707122200770E-006  -6.7587161875994187E-011   2.4142843526995708E-010
  7.3425530549347846E-013   299.99983692554611        7.0897999763653063E-023   7.0897999763653063E-023   5.0000001419004654E-003
   0.0000000000000000      -0.50000000000000000      -0.12000000000000000       3.0300707122200770E-006  -9.7447395985511123E-011   4.0608248327132344E-010
  1.2282174494975776E-012   299.99983692554611        7.0897999763653063E-023   7.0897999763653063E-023   5.0000001419004654E-003
(base) paula@PaulaCW:~/prom/Graphs_ASCII$
```

VisIt

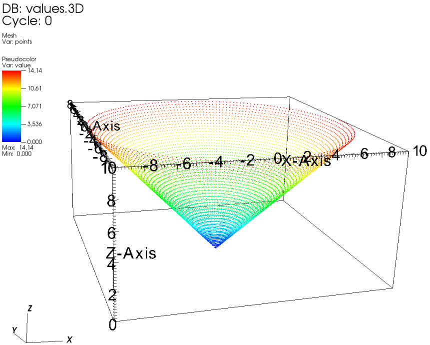

DB: example.silo
Cycle: 0
Pseudocolor
Var: pressure
Units: Pa
5.779
4.610
3.442
2.273
1.104
Max: 5.779
Min: 1.104


DB: example.silo
Cycle: 0
Pseudocolor
Var: temp
Units: K
5.890
4.691
3.493
2.294
1.096
Max: 5.890
Min: 1.096
Mesh
Var: Mesh
Width (parsec)
Y
X
Z

```python
from __future__ import print_function
import os
from astropy.io import ascii

def ascii2hdf5(inputfile, outputfile, clobber=False, overwrite=True,
               verbose=False):
    """Convert a file to hdf5 using compression and path set to data"""
    if verbose:
        print('converting {} to {}'.format(inputfile, outputfile))

    tbl = ascii.read(inputfile)
    try:
        tbl.write(outputfile, format='hdf5', path='data', compression=True,
                  overwrite=overwrite)
    except:
        print('problem with {}'.format(inputfile))
        return

    if clobber:
        os.remove(inputfile)
        if verbose:
            print('removed {}'.format(inputfile))

    return
```

Solución 2

```python
import math
n = 10000
f = open("values.3D", "wt")
f.write("x y z value\n");
for i in range(n):
    t = float(i) / float(n-1)
    angle = t * (math.pi * 2.) * 50.
    r = t * 10.
    x = r * math.cos(angle)
    y = r * math.sin(angle)
    z = t * 10.
    value = math.sqrt(x*x + y*y + z*z)
    f.write("%g %g %g %g\n" % (x,y,z,value))
f.close()
```



```python
import sys
import numpy as np

# Se importa el archivo con los datos correspondientes
# a las dimensiones y tiempos de las corridas
filedata = open('linput_plots.par', 'r')

lines = filedata.readlines()
# Se asignan la variables del archivo filedata en
# las variables correspondientes
xmin = float(lines[1].split('=')[1])
xmax = float(lines[2].split('=')[1])
ymin = float(lines[3].split('=')[1])
ymax = float(lines[4].split('=')[1])
zmin = float(lines[5].split('=')[1])
zmax = float(lines[6].split('=')[1])
Nxx = int(lines[7].split('=')[1])
Nyy = int(lines[8].split('=')[1])
Nzz = int(lines[9].split('=')[1])
Ntt = int(lines[10].split('=')[1])
courant = float(lines[11].split('=')[1])
every3D = float(lines[14].split('=')[1])

# Se cierra el archivo
filedata.close()

# Se calculan los pasos espaciales y temporales
dx  = (xmax - xmin)/float(Nxx)
dy  = (ymax - ymin)/float(Nyy)
dz  = (zmax - zmin)/float(Nzz)
dt = courant * min(dx,dy,dz)
```

```python
# Se importa el archivo con los datos de las corridas correspondientes a las variables
# densidad, velocidad en x, y, z, presion y campo magnetico en x, y, z
f = open('primitivas_1.xyzl', 'r')

# Se crea la funcion 'extraer' para tomar determinado tiempo y variable del archivo f
def extraer(parametro):

    # Se leen las entradas para reconocer la variable y tiempo a extraer
    variable = str(parametro.split('-')[0])
    bloque = format(int(parametro.split('-')[1]), '02')

    # Se calcula el tiempo real que se esta extrayendo
    t = dt*float(bloque)*every3D

    # Se valida la infomacion acerca de la variable, el bloque de tiempo y el tiempo
    # real que se esta extrayendo
    print 'Extrayendo la variable',variable,'en el bloque',bloque,'en',t, 'segundos.'

    # Se saltan las lineas correspondientes a los tiempos anteriores al que se requiere
    skip = int((((Nyy+2)*(Nxx+1)+1)*(Nzz+1)+4)*int(bloque))
    print 'Se saltan', skip, 'lineas.'
    for a in range (0,skip+3):
        f.readline()

    # Se crea un archivo nuevo llamado por el nombre de la variable y el bloque a extraer
    w = open('%s_b%s.3D'%(variable,bloque), 'wt')
    # Se escribe el encabezado necesario para que VisIt pueda leer los archivos ASCII
    w.write('x y z %s\n'%variable);
```

```python
# Se leen las lineas de la misma manera como fueron salvadas y se reescriben
# los datos requeridos en el archivo w
for i in range (0,Nzz+1):
    f.readline()
    for j in range (0,Nxx+1):
        f.readline()
        for k in range (0,Nyy+1):
            z, x, y, rho, vx, vy, vz, press, Bx, By, Bz = f.readline().split()
            if (variable == 'v'):
                v = np.sqrt(float(vx)**2+float(vy)**2+float(vz)**2)
                w.write('%g %g %g %g\n' %(float(x),float(y),float(z),float(v)))
            elif (variable == 'B'):
                B = np.sqrt(float(Bx)**2+float(By)**2+float(Bz)**2)
                w.write('%g %g %g %g\n' %(float(x),float(y),float(z),float(B)))
            elif (variable == 'rho'):
                w.write('%g %g %g %g\n' %(float(x),float(y),float(z),float(rho)))
            elif (variable == 'vx'):
                w.write('%g %g %g %g\n' %(float(x),float(y),float(z),float(vx)))
            elif (variable == 'vy'):
                w.write('%g %g %g %g\n' %(float(x),float(y),float(z),float(vy)))
            elif (variable == 'vz'):
                w.write('%g %g %g %g\n' %(float(x),float(y),float(z),float(vz)))
            elif (variable == 'press'):
                w.write('%g %g %g %g\n' %(float(x),float(y),float(z),float(press)))
            elif (variable == 'Bx'):
                w.write('%g %g %g %g\n' %(float(x),float(y),float(z),float(Bx)))
            elif (variable == 'By'):
                w.write('%g %g %g %g\n' %(float(x),float(y),float(z),float(By)))
            elif (variable == 'Bz'):
                w.write('%g %g %g %g\n' %(float(x),float(y),float(z),float(Bz)))

# Se cierran ambos archivos
w.close()
f.close()
```

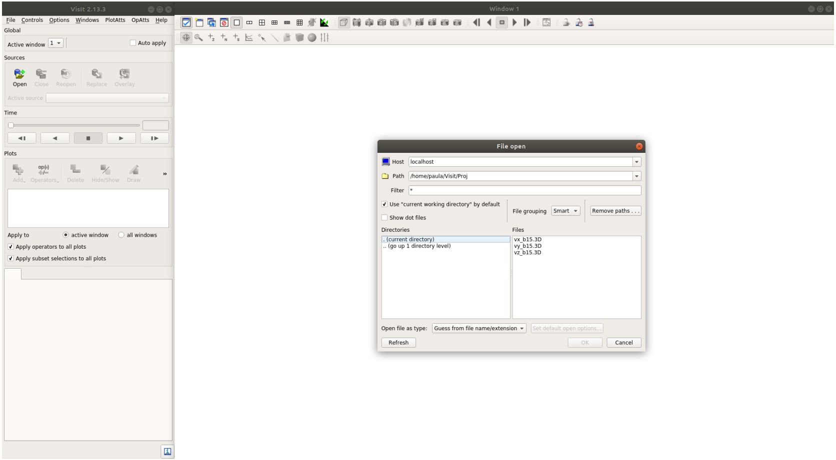
```python
# Se utilizan estas 3 lineas para llamar la funcion
# e ingresar la variable y bloque a salvar desde la terminal
method_name = sys.argv[1]
parameter_name = sys.argv[2]
getattr(sys.modules[__name__], method_name)(parameter_name)
```

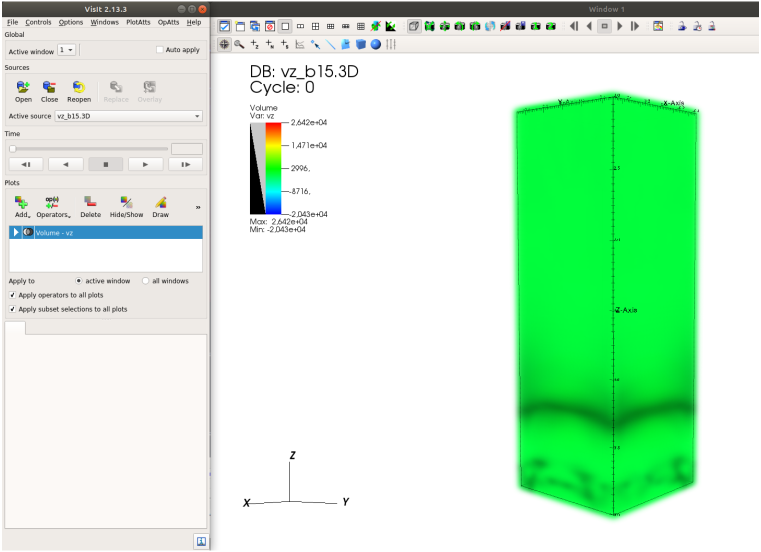
Archivos resultantes

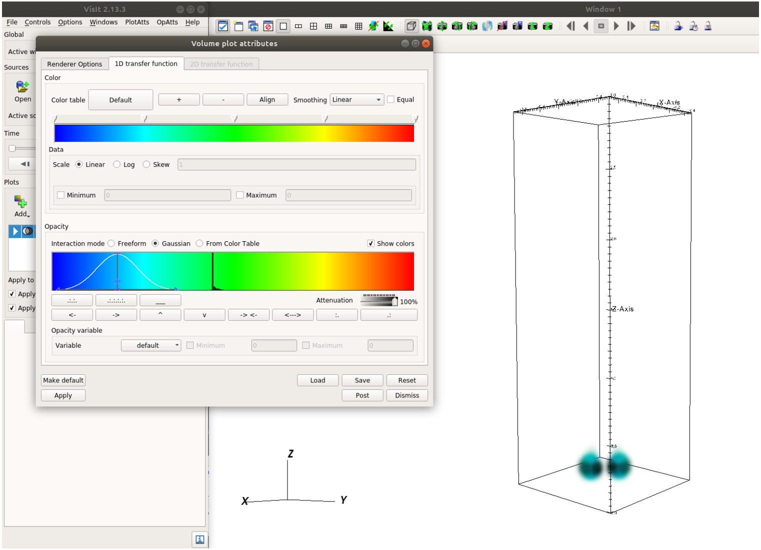
```
(base) paula@PaulaCW:~/prom/Graphs_ASCII$ head -25 vz_b15.3D
x y z vz
-0.5 -0.5 0 1e-14
-0.5 -0.48 0 1e-14
-0.5 -0.46 0 1e-14
-0.5 -0.44 0 1.00001e-14
-0.5 -0.42 0 1.00008e-14
-0.5 -0.4 0 1.00041e-14
-0.5 -0.38 0 1.00195e-14
-0.5 -0.36 0 1.00859e-14
-0.5 -0.34 0 1.03482e-14
-0.5 -0.32 0 1.13036e-14
-0.5 -0.3 0 1.45048e-14
-0.5 -0.28 0 2.437e-14
-0.5 -0.26 0 5.23151e-14
-0.5 -0.24 0 1.25024e-13
-0.5 -0.22 0 2.9863e-13
-0.5 -0.2 0 6.78572e-13
-0.5 -0.18 0 1.43959e-12
-0.5 -0.16 0 2.83184e-12
-0.5 -0.14 0 5.15173e-12
-0.5 -0.12 0 8.65852e-12
-0.5 -0.1 0 1.34386e-11
-0.5 -0.08 0 1.92576e-11
-0.5 -0.06 0 2.54771e-11
-0.5 -0.04 0 3.11156e-11
```

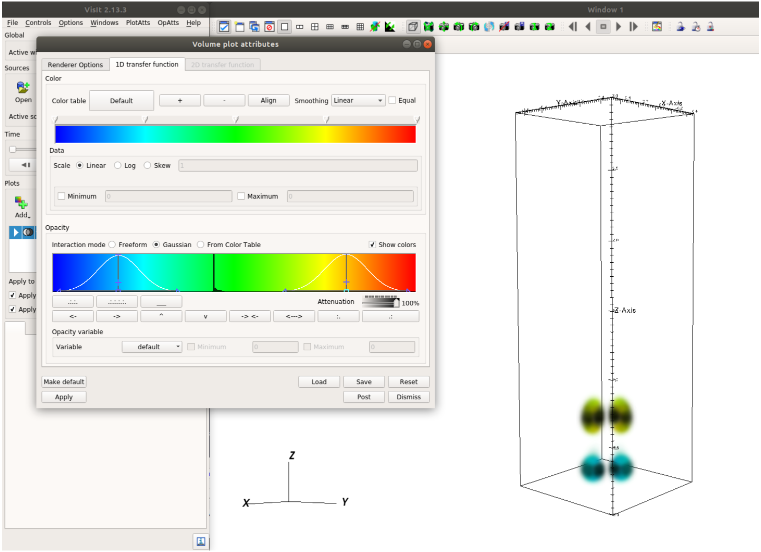
Script en bash

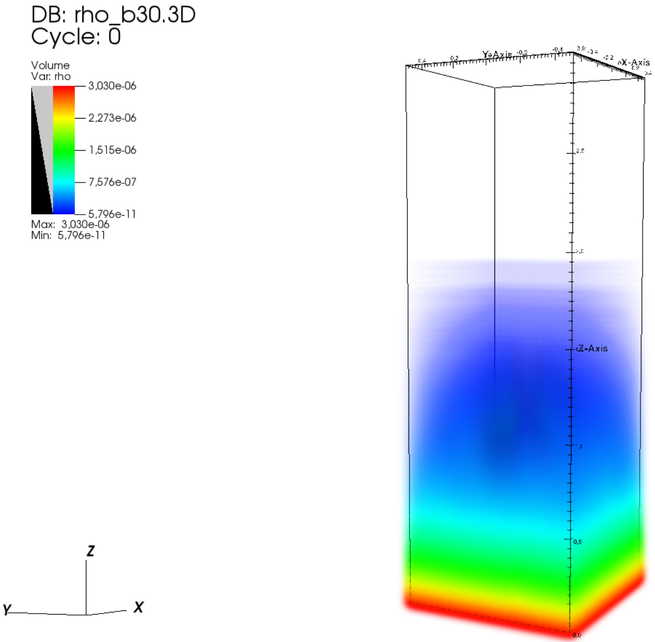
```
(base) paula@PaulaCW:~/prom/Graphs_ASCII$ cat myscript.sh
#!/usr/bin/env bash

rm *.3D

v=${1?Error: Ingrese otra variable}
b=${2?Error: Ingrese otro bloque}

python extractor.py extraer $v-$b

cp *.3D ../../../../../Visit/Proj/

cd
./visit
```

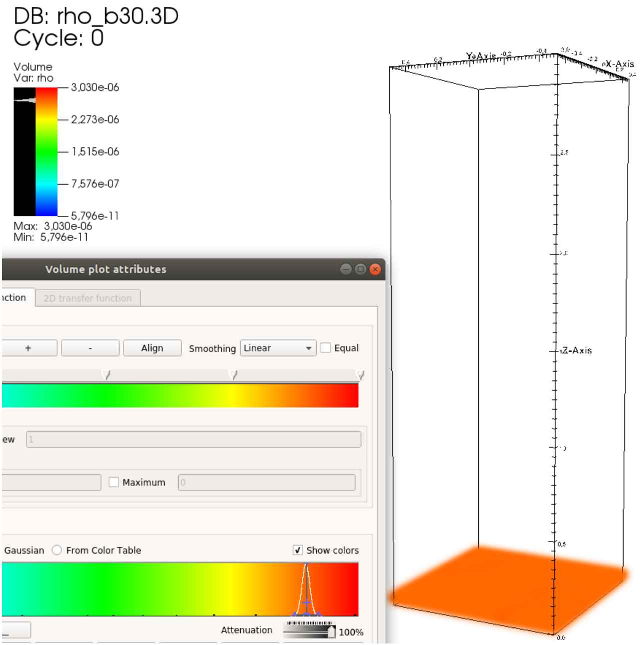

VisIt 2.13.3
File Controls Options Windows PlotAtts OpAtts Help
Global
Active window 1
Auto apply
Sources
Open Close Reopen Replace Overlay
Active source
Time
Plots
Add Operators Delete Hide/Show Draw
Apply to active window all windows
Apply operators to all plots
Apply subset selections to all plots
Window 1
File open
Host localhost
Path /home/paula/Visit/Proj
Filter *
Use "current working directory" by default
Show dot files
File grouping Smart
Remove paths . . .
Directories
. (current directory)
.. (go up 1 directory level)
Files
vx_b15.3D
vy_b15.3D
vz_b15.3D
Open file as type: Guess from file name/extension
Set default open options...
Refresh
OK
Cancel

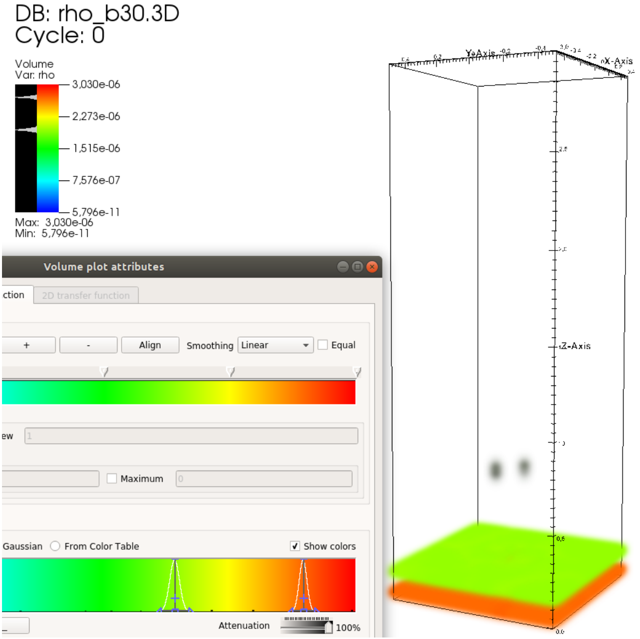

VisIt 2.13.3
File Controls Options Windows PlotAtts OpAtts Help
Global
Active window 1
Auto apply
Sources
Open Close Reopen Replace Overlay
Active source vz_b15.3D
Time
Plots
Add Operators Delete Hide/Show Draw
Volume - vz
Apply to active window all windows
Apply operators to all plots
Apply subset selections to all plots
Window 1
DB: vz_b15.3D
Cycle: 0
Volume
Var: vz
2,642e+04
1,471e+04
2996,
-8716,
-2,043e+04
Max: 2,642e+04
Min: -2,043e+04
Y-Axis
X-Axis
Z-Axis
Z
X
Y

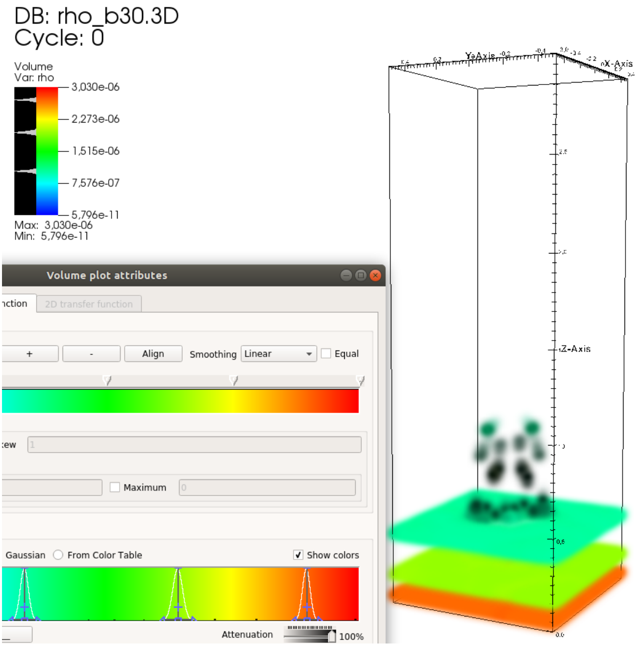

VisIt 2.13.3
File Controls Options Windows PlotAtts OpAtts Help
Window 1
Volume plot attributes
Renderer Options
1D transfer function
2D transfer function
Color
Color table
Default
Align
Smoothing
Linear
Equal
Data
Scale
Linear
Log
Skew
Minimum
Maximum
Opacity
Interaction mode
Freeform
Gaussian
From Color Table
Show colors
Attenuation
100%
Opacity variable
Variable
default
Make default
Apply
Load
Save
Reset
Post
Dismiss
Global
Sources
Open
Time
Plots
Add
Apply to
Y-Axis
X-Axis
Z-Axis
Z
X
Y

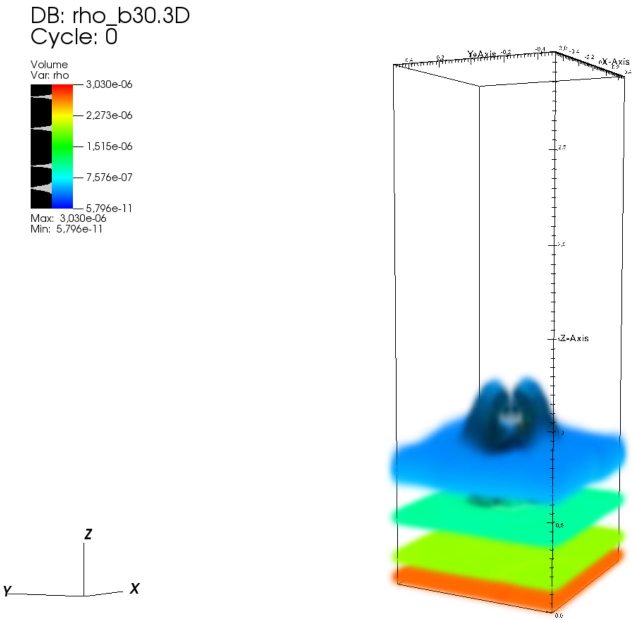

VisIt 2.13.3
Window 1
File Controls Options Windows PlotAtts OpAtts Help
Volume plot attributes
Renderer Options
1D transfer function
2D transfer function
Color
Color table
Default
Align
Smoothing
Linear
Equal
Data
Scale
Log
Skew
Minimum
Maximum
Opacity
Interaction mode
Freeform
Gaussian
From Color Table
Show colors
Attenuation
100%
Opacity variable
Variable
default
Make default
Apply
Load
Save
Reset
Post
Dismiss
Z-Axis
Y-Axis
X-Axis
Z
X
Y


DB: rho_b30.3D
Cycle: 0
Volume
Var: rho
3,030e-06
2,273e-06
1,515e-06
7,576e-07
5,796e-11
Max: 3,030e-06
Min: 5,796e-11
Y-Axis
X-Axis
Z-Axis
Z
Y
X


DB: rho_b30.3D
Cycle: 0
Volume
Var: rho
3,030e-06
2,273e-06
1,515e-06
7,576e-07
5,796e-11
Max: 3,030e-06
Min: 5,796e-11
Volume plot attributes
2D transfer function
Align
Smoothing
Linear
Equal
Maximum
Gaussian
From Color Table
Show colors
Attenuation
100%
Y-Axis
X-Axis
Z-Axis


DB: rho_b30.3D
Cycle: 0
Volume
Var: rho
3,030e-06
2,273e-06
1,515e-06
7,576e-07
5,796e-11
Max: 3,030e-06
Min: 5,796e-11
Volume plot attributes
ction
2D transfer function
+
-
Align
Smoothing
Linear
Equal
ew
Maximum
Gaussian
From Color Table
Show colors
Attenuation
100%
Y-Axis
X-Axis
Z-Axis


DB: rho_b30.3D
Cycle: 0
Volume
Var: rho
3,030e-06
2,273e-06
1,515e-06
7,576e-07
5,796e-11
Max: 3,030e-06
Min: 5,796e-11
Volume plot attributes
nction
2D transfer function
+
-
Align
Smoothing
Linear
Equal
kew
Maximum
Gaussian
From Color Table
Show colors
Attenuation
100%
Y-Axis
X-Axis
Z-Axis


DB: rho_b30.3D
Cycle: 0
Volume
Var: rho
3,030e-06
2,273e-06
1,515e-06
7,576e-07
5,796e-11
Max: 3,030e-06
Min: 5,796e-11
Y-Axis
X-Axis
Z-Axis
Z
Y
X

Referencias

1. Página oficial de VisIt
2. Convertidor de ASCII a HDF5
3. Generador de archivos en formato permitido

¡GRACIAS!