

This is a work of the United States Government. In accordance with 17 U.S.C. 105, no copyright protection is available for such works under U.S. Law.

Public Domain Mark 1.0

<https://creativecommons.org/publicdomain/mark/1.0/>

Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

Please provide feedback

Please support the ScholarWorks@UMBC repository by emailing scholarworks-group@umbc.edu and telling us what having access to this work means to you and why it's important to you. Thank you.



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-TR-221145

Conversion of NIMROD simulation results for graphical analysis using VisIt

C. A. Romero-Talamas

May 5, 2006

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

Conversion of NIMROD simulation results for graphical analysis using VisIt

Carlos A. Romero-Talamás

*Lawrence Livermore National Laboratory,
7000 East Avenue, Livermore, California, 94550**

(Dated: April 27, 2006)

Abstract

Software routines developed to prepare NIMROD [C. R. Sovinec et al., J. Comp. Phys. **195**, 355 (2004)] results for three-dimensional visualization from simulations of the Sustained Spheromak Physics Experiment (SSPX) [E. B. Hooper et al., Nucl. Fusion **39**, 863 (1999)] are presented here. The visualization is done by first converting the NIMROD output to a format known as legacy VTK and then loading it to VisIt, a graphical analysis tool that includes three-dimensional rendering and various mathematical operations for large data sets. Sample images obtained from the processing of NIMROD data with VisIt are included.

*Electronic address: romerotalamas1@llnl.gov; The author would like to thank Jeff Moller and Bill Meyer for their help with software and computer issues, to the VisIt support team for their fast response to questions about their software, and to Lynda Lodestro for her comments that helped improve this manuscript. Work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract W-7405-ENG-48.

I. INTRODUCTION

This report describes the procedures and routines used in the conversion of NIMROD [1] simulation results to files readable by VisIt, a visualization and graphical analysis tool package. NIMROD is a three-dimensional (3D) magnetohydrodynamic (MHD) code that is used to simulate plasma dynamics at the Sustained Spheromak Physics Experiment (SSPX) [2]. VisIt is a free (distribution and open source code) package created by the Department of Energy (DOE) Advanced Simulation and Computing Initiative (ASCI) to visualize and analyze large computational data sets. VisIt can read databases written in standard visualization formats and render two or three-dimensional images, as well as perform various mathematical operations on the data [3].

The NIMROD simulation data is stored in binary files called dump files. Each dump file contains information about a particular cycle, that is, a particular time step in the simulation. The nomenclature of each dump file is *dump.XXXXX*, where the five X represent the digits corresponding to the cycle number (including leading zeroes for numbers smaller than 10^5). To visualize the data from a particular dump cycle, a program called NIMPLOT was created by the NIMROD Team[6]. This program takes as input one or more dump files and the user may choose to plot NIMROD variables in different ways, such as vector plots and contour plots, among others. With NIMPLOT, it is also possible to create data output files in the American Standard Code for Information Interchange (ASCII) format. However, all of the plots are two-dimensional, and the user must choose which toroidal location of the simulation domain to plot.

At the time of writing of this report, the files output by NIMPLOT could not be loaded directly into VisIt, and therefore had to be translated into one of the formats recognizable by the visualization program. From these formats, the VTK simple legacy format [4] is relatively simple to work with, and thus a translator was made to rearrange the data in this format. Furthermore, multiple toroidal *slices* can be included in a single VTK file, conveniently grouping all the necessary data for 3D rendering using VisIt in a single ASCII file. Ideally, dump files would be loaded directly into VisIt to save processing time and storage space. This is indeed possible since VisIt supports database reader plugins. However, the development of a NIMROD dump file plugin for VisIt was not carried out for the present work.

This document is organized as follows. Section II describes the routines and inputs used to produce volumetric data in separate toroidal slice files by the running NIMPLOT automatically. Section III describes the process of merging toroidal slices to a single volumetric file per time step. The mesh and representation domain inferred by VisIt, along with some examples of 3D rendering and analysis, are shown in Section IV. The complete code and sample inputs used in the conversion process is shown in each corresponding section.

II. AUTOMATED RUN OF NIMPLOT AND THE CREATION OF DATA SLICES IN ASCII FORMAT

To convert dump file data to ASCII text files, a group of simple routines was created in *Mathematica* [5] in order to automate the process of running NIMPLOT and choosing the inputs. These routines take a list of dump cycles and the number of toroidal slices per dump cycle as input, and produce an input file for NIMPLOT. This input file mimics the human input for which NIMPLOT was designed. An excerpt of a NIMPLOT input file (*NimplotInput.txt*) is shown below.

```
xdraw

5
dump.01500

c
0
y
drawvec.in
lam07_dump.01500_0.00000fraction.txt

c
0.01388888902777776
y
drawvec.in
lam07_dump.01500_0.01388fraction.txt

c
0.02777777805555553
y
drawvec.in
lam07_dump.01500_0.02777fraction.txt
```

It should be noted that the file above can be created manually and fed into NIMPLOT with the Unix command `nimplot < NimplotInput.txt`. However, because of the large number of dump cycles and toroidal slices processed, and because *Mathematica* was installed in `grendel.llnl.gov` (the server where the NIMROD dump files are stored), it was more convenient to create the aforementioned routines and run them directly on the server. The code for these routines is shown below. [Note: line numbers are included here only for reference and are not part of the *Mathematica* file; comments and explanations about the routines are included in enclosed brackets of the form *(* some comments ... *)*, which are not evaluated by *Mathematica*.]

```

1  (* Begin file MatheNIMPLOTv2.txt *)
2
3  (*****
4  The following lines are to be evaluated in the command line version of Mathematica.
5
6  First, log on to grendel.llnl.gov, and change directories until you are in the directory
7  where nimplot and all the necessary input files are. Make sure this input file, and all
8  other files required by Mathematica, are also in the same directory.
9
10 Start Mathematica by typing (in the grendel command prompt): math
11
12 Now, to load this file, type: <<MatheNIMPLOTv2.txt. This is all that is needed to run the
13 lines below.
14
15 IMPORTANT: THIS PROGRAM IS MADE TO RUN FROM A TERMINAL WITHOUT
16 XTERM DISPLAY, OR RUN WITHOUT XDRAW INSTALLED. This program will not run
17 if plots can be displayed. Remove any directory paths pointing to any xdraw program.
18 *****)
19
20 Print["\n The Mathematica instructions for this run are found in the file MatheNIMPLOTv2.txt \n"];
21
22 (*****
23 (*****
24 (* Below, the input variables for the loop function *)
25
26 NumberOfToroidalSegments = 72;
27 NameOfFileWithDumpNumbersAndTimes =
28 "Log_of_NIMROD_lam07_dump_files_and_times_02.csv";
29 (*****
30 (*****
31
32 (* Function declaration *)
33

```

```

34  (*****)
35  (* Auxiliary function to have the dump file number as a string with leading zeroes
36  if necessary. Note that if the NIMROD dump files ever have more than 5 digits,
37  this function will have to be modified accordingly *)
38
39  ConvertDumpNumberToStringInFiveDigitFormat[DumpNumber_] :=
40      Module[{TextDumpNumber},
41          If[DumpNumber < 10000,
42              If[DumpNumber < 1000,
43                  If[DumpNumber < 100,
44                      If[DumpNumber < 10,
45                          TextDumpNumber = "0000"<>ToString[DumpNumber],
46                          TextDumpNumber = "000"<>ToString[DumpNumber]],
47                      TextDumpNumber = "00"<>ToString[DumpNumber]],
48                      TextDumpNumber = "0"<>ToString[DumpNumber]],
49              TextDumpNumber = ToString[DumpNumber]];
50      TextDumpNumber
51  ];
52
53  (*****)
54
55  ProcessMultipleTimeStepsAndToroidalLocationsFromNIMROD[V_] :=
56      Module[{NameOfFileWithDumpNumbersAndTimes, DumpFileNumbersAndTimes,
57          NumberOfToroidalSegments, NumberOfDumpFiles, i, DumpNumber,
58          NimplotInputFile, TextDumpNumber, AngleFraction, TextAngleFraction},
59
60      NameOfFileWithDumpNumbersAndTimes = V[[1]];
61      NumberOfToroidalSegments = V[[2]];
62
63      DumpFileNumbersAndTimes = Import[NameOfFileWithDumpNumbersAndTimes];
64
65      NumberOfDumpFiles = Dimensions[DumpFileNumbersAndTimes][[1]];
66
67      NimplotInputFile = OpenWrite["NimplotInput.txt"];
68
69      WriteString[NimplotInputFile, "xdraw \n"];
70
71      For[i=1, i<= NumberOfDumpFiles, i++,
72
73          WriteString[NimplotInputFile, "\n"];
74          WriteString[NimplotInputFile, "5 \n"];
75          DumpNumber = DumpFileNumbersAndTimes[[i,1]];
76          TextDumpNumber = ConvertDumpNumberToStringInFiveDigitFormat[DumpNumber];
77          WriteString[NimplotInputFile, "dump."<>TextDumpNumber<>" \n", "\n"];
78
79          For[j=0, j< NumberOfToroidalSegments, j++,
80

```



```

81      WriteString[NimplotInputFile, "c \n"];
82      AngleFraction = (1.00000001*j/NumberOfToroidalSegments);
83      Write[NimplotInputFile, AngleFraction];
84      WriteString[NimplotInputFile, "y \n", "drawvec.in \n"];
85      If[AngleFraction==0,
86        TextAngleFraction = "0.00000";,
87        TextAngleFraction = StringTake[ToString[SetPrecision[AngleFraction,8]],7];
88        (* The 1.00000001 factor ensures the string has 7 digits *)
89      ];
90      WriteString[NimplotInputFile,
91        "lam07_dump."<> TextDumpNumber<>"_."<> TextAngleFraction<>"fraction.txt \n","\n"];
92    ];
93
94  ];
95
96  WriteString[NimplotInputFile, "\n", "\n","0"]; (* Exits nimplot *)
97  Close[NimplotInputFile];
98
99 ];
100
101  (*****
102
103  V={NameOfFileWithDumpNumbersAndTimes, NumberOfToroidalSegments};
104
105  ProcessMultipleTimeStepsAndToroidalLocationsFromNIMROD[V];
106
107  Run["nimplot < NimplotInput.txt", " "]
108
109  (* End file MatheNIMPL0Tv2.txt *)

```

In the program above, between lines 23 and 29, there are two inputs required from the user: the number of toroidal segments in which to divide the simulation domain, and the name of the file containing the dump cycle numbers that the user wants processed including the simulation time they represent. For each dump cycle in the input file, the program would produce 72 text files with the name structure lam07_dump.XXXXX_0.YYYYYfraction.txt, where X represents the dump cycle number and Y a fraction between 0 and 1. The number 72 was chosen in order to maximize the toroidal resolution (in this case every 5 degrees) while having manageable VTK file sizes (less than 100 MB, see Section III) and processing time for a personal computer (tens of minutes per dump cycle).

The file containing the dump cycle numbers and times is written in comma separated value (CSV) format. The number of lines in this file, as well as the particular cycle numbers

are chosen by the user. Note that the times included in this file are not used for the file processing presented in this report, but are used in other automated processes which will be reported elsewhere. The contents of `Log_of_NIMROD_lam07_dump_files_and_times_02.csv` are shown below.

```
1500,2.37E-05
2000,2.99E-05
2070,3.07E-05
2500,3.56E-05
3000,3.88E-05
3500,4.07E-05
4000,4.24E-05
4500,4.41E-05
5000,4.57E-05
5500,4.74E-05
6500,5.04E-05
7000,5.17E-05
7500,5.32E-05
7880,5.44E-05
8500,5.64E-05
9000,5.80E-05
9500,5.96E-05
10000,6.13E-05
10500,6.31E-05
11000,6.48E-05
11500,6.67E-05
12431,6.93E-05
13000,7.10E-05
13500,7.28E-05
14000,7.44E-05
14500,7.51E-05
15000,7.63E-05
15500,7.78E-05
16000,7.92E-05
16500,8.08E-05
17000,8.25E-05
17500,8.42E-05
18000,8.60E-05
18500,8.78E-05
18803,8.89E-05
19000,8.96E-05
19500,9.13E-05
20000,9.33E-05
20500,9.54E-05
21000,9.72E-05
21500,9.88E-05
```

```

22000,1.00E-05
22500,1.02E-05
23000,1.03E-04
23500,1.04E-04
24000,1.06E-04
24500,1.08E-04
25000,1.09E-04
25500,1.11E-04
26000,1.12E-04
26500,1.14E-04
27000,1.15E-04
27500,1.16E-04
28000,1.17E-04
28500,1.18E-04
29000,1.19E-04
29500,1.20E-04

```

III. MERGING ALL THE TOROIDAL SLICES INTO A SINGLE VTK FILE

Once all the required dump cycles have been processed with the routines shown in the previous section, the toroidal slices produced must be lumped into a single VTK file in order to load them easily into VisIt. The routines shown below were created in *Mathematica* to open the toroidal slice files, extract only one variable of interest, and rearrange the data in the VTK STRUCTURED_GRID format.

```

1  (* Begin file Conversion_from_NIMROD_ascii_dump_(tecplot)_to_VTK_v09_(version_for_moller1m4).txt *)
2
3  (* For this version, use the terminal in MacOS X and change to the directory where you want the files
4  to be read and written, then start Mathematica. In my computer is the following line:
5
6  '/Volumes/Data/Users/romero/Applications/Mathematica/Mathematica 5.0.app/Contents/MacOS/MathKernel'
7
8  Once Mathematica is running, type the following in the command prompt:
9
10 <<"Conversion_from_NIMROD_ascii_dump_(tecplot)_to_VTK_v04_(version_for_moller1m4).txt *)
11
12 (*****
13
14 (* The function below is required to change from a segmented (19*150*4) structured grid output by
15 NIMPLOT, to a uniform and structured grid (76*150), such that VTK can read the grid and automatically form
16 hexahedral cells *)
17
18 ReorganizeStructuredGrid[V_] :=

```

```

19 Module[{InputList, temp1, i, j, k, temp2},
20   InputList = V[[1]];
21   temp1 = Transpose[Partition[InputList, 150*19]];
22   temp1 = Partition[Flatten[temp1, 1], 76];
23   temp2 = temp1;
24   For[i = 1, i 150, i++,
25     For[k = 1, k 4, k++,
26       For[j = 1, j 19, j++,
27         temp2[[i, (j + (k - 1)*19)]] = temp1[[i, 4*(j + k - 1) - 3*k]];
28       ];
29     ];
30   ];
31   Flatten[temp2, 1]
32 ];
33
34 (*****
35
36 (* The function below assumes that all files contain a three letter extension (e.g. txt) and that the last
37 digits before the extension contain the angle (a fraction between 0 and 1) information and the characters "fraction".
38 -This function differs from the one found in Conversion_from_NIMROD_ascii_dump_(tecplot)_to_VTK_v01.nb in
39 that reads the input angle from the .txt file as a fraction between 0 and 1 (that is, 0 -> 1 is equivalent to 0 -> 2Pi). *)
40
41 CreateSingleVTKfileFromMultipleToroidalLocations[V_] :=
42 Module[{i, FileListFromSingleDumpTime, FieldToPlot, Rind,
43   ThetaInd, Zind, AllFieldNIMROD, DumpFile, DumpNIMRODdata, AllVariables, TransposedVariablesNIMRODdata, s1,
44     Theta, DataPoints, AllFieldNIMRODAllAngles, DataPointsAllAngles, MergedHeader1AndData, R, X, Y, Z, DumpIdentifier},
45
46   FileListFromSingleDumpTime = V[[1]];
47   FieldToPlot = V[[2]];
48
49   (* Below, choose the
50     columns in the array depending on the field to plot *)
51   If[FieldToPlot == "B",
52     Rind = 16;
53     ThetaInd = 18;
54     Zind = 17;,
55     If[FieldToPlot == "J",
56       Rind = 22;
57       ThetaInd = 24;
58       Zind = 23;,
59       If[FieldToPlot == "V",
60         Rind = 28;
61         ThetaInd = 30;
62         Zind = 29;
63       ];
64     ];
65 ];

```

```

66
67 AllFieldNIMRODAllAngles = {};
68 DataPointsAllAngles = {};
69 Print["Comencing data extraction from dump files"];
70
71 For[i = 1, i <= Dimensions[FileListFromSingleDumpTime][[1]], i++,
72
73     DumpFileName = FileListFromSingleDumpTime[[i]];
74
75     sl = StringLength[DumpFileName]; (* Measures the file name length *)
76     Theta = 2*Pi *ToExpression[StringTake[DumpFileName, {sl - 18, sl - 12}]];
77     (* Extracts angle information from file name *)
78
79     DumpFile = Import[DumpFileName, "Table"];
80     Print["File " <> DumpFileName <> " opened"];
81     DumpNIMRODdata = Take[DumpFile, -(Dimensions[DumpFile][[1]] - 1)]; (* removes the file header *)
82     AllVariables = Partition[Flatten[DumpNIMRODdata], 45]; (* 45 is the number of columns in the NIMROD output file *)
83     TransposedVariablesNIMRODdata = Transpose[AllVariables];
84     (* Transpose ALLVariables to obtain only the variables we want *)
85
86     AllFieldNIMROD =
87     Module[{temp1, temp2, Xcomp, Ycomp, Zcomp, Rcomp, Thetacomp},
88         Rcomp = TransposedVariablesNIMRODdata[[Rind]];
89         Thetacomp = TransposedVariablesNIMRODdata[[ThetaInd]];
90         Zcomp = - TransposedVariablesNIMRODdata[[Zind]];
91         (* Multiply by - 1 to have a right handed system, since the Z coord. is multiplied by - 1 *)
92
93         Xcomp = Rcomp Cos[Theta] - Thetacomp Cos[(Pi/2) - Theta];
94         Ycomp = Rcomp Sin[Theta] + Thetacomp Sin[(Pi/2) - Theta];
95
96
97         temp1 = {Xcomp, Ycomp, Zcomp};
98         temp2 = Flatten[Transpose[temp1]];
99         Partition[temp2, 3]
100     ];
101
102     R = TransposedVariablesNIMRODdata[[1]];
103     X = R Cos[Theta ]; (* Note that Theta changed from the v01 of this function *)
104     Y = R Sin[Theta ];
105     Z = -TransposedVariablesNIMRODdata[[2]];
106
107
108     DataPoints = Transpose[{X, Y, Z}];
109
110     DataPoints = ReorganizeStructuredGrid[{DataPoints}];
111     AllFieldNIMROD = ReorganizeStructuredGrid[{AllFieldNIMROD}];
112

```

```

113     AllFieldNIMRODAllAngles = Join[AllFieldNIMRODAllAngles, AllFieldNIMROD];
114
115     DataPointsAllAngles = Join[DataPointsAllAngles, DataPoints];
116
117 ];
118 Print["VTK file creation in progress"];
119
120     NumberOfVertices = ToString[Dimensions[DataPointsAllAngles][[1]]];
121
122     DumpIdentifier = StringTake[DumpFileName, {12,16}]; (* Extracts angle information from file name *)
123
124     MergedHeader1AndData = Prepend[DataPointsAllAngles, {"POINTS " <> NumberOfVertices <> " double"}];
125     MergedHeader1AndData =
126     Prepend[MergedHeader1AndData, {"DIMENSIONS 76 150 " <> ToString[Dimensions[FileListFromSingleDumpTime][[1]]]};
127     MergedHeader1AndData = Prepend[MergedHeader1AndData, {"DATASET STRUCTURED_GRID"}];
128     MergedHeader1AndData = Prepend[MergedHeader1AndData, {"ASCII"}];
129     MergedHeader1AndData = Prepend[MergedHeader1AndData, {"NIMROD output for dump " <> DumpIdentifier}];
130     MergedHeader1AndData = Prepend[MergedHeader1AndData, {"# vtk DataFile Version 2.0"}];
131
132     MergedHeader1AndData = Append[MergedHeader1AndData, {"POINT_DATA " <> NumberOfVertices}];
133     MergedHeader1AndData = Append[MergedHeader1AndData, {"VECTORS " <> FieldToPlot <> "field double"}];
134     MergedHeader1AndData = Join[MergedHeader1AndData, AllFieldNIMRODAllAngles];
135
136
137     Export["NIMROD_Volume_File_" <> FieldToPlot <> "_Field_From_dump." <> DumpIdentifier <> ".vtk",
138     MergedHeader1AndData, "CSV", ConversionOptions -> {"FieldSeparator" -> " " }];
139
140     Print["Finished exporting: ", "NIMROD_Volume_File_" <> FieldToPlot <> "_Field_From_dump." <> DumpIdentifier <> ".vtk"];
141
142 ];
143
144 (* **** *)
145
146 Print["\n","Comencing Mathematica functions","\n"];
147
148 DumpCycleList={"00500","01000","01500","02000", "02070", "02500", "03000", "03500", "04000",
149 "04500", "05000", "05500", "06000", "06500", "07000", "07500", "07880", "08500", "09000", "09500",
150 "10000", "10500", "11000", "11500", "12431", "13000", "13500", "14000", "14500", "15000", "15500",
151 "16000", "16500", "17000", "17500", "18000", "18500", "18803", "19000", "19500", "20000", "20500",
152 "21000", "21500", "22000", "22500", "23000", "23500", "24000", "24500", "25000", "25500", "26000",
153 "26500", "27000", "27500", "28000", "28500", "29000", "29500"};
154
155 For[cont=1,cont<=Dimensions[DumpCycleList][[1]],cont++,
156
157     DumpCycleNumber = DumpCycleList[[cont]];
158
159     f =OpenWrite["sftpInput.txt"];

```

```

160   WriteString[f, "cd nimplot\n"];
161   WriteString[f, "get lam07*dump."<> DumpCycleNumber<>]*fraction.txt\n"];
162   WriteString[f, "exit"];
163   Close[f];
164
165   Print["get lam07*dump."<> DumpCycleNumber<>]*fraction.txt\n"];
166
167   Run["sftp grendel < sftpInput.txt",""];
168
169   DumpFileFilterCriterion = "lam*" <> DumpCycleNumber <> "*fract*.txt";
170
171   FileListFromSingleDumpTime = FileNames[DumpFileFilterCriterion];
172
173   (* InputVectorForVTKfileCreator = {FileListFromSingleDumpTime, "J"};
174
175   CreateSingleVTKfileFromMultipleToroidalLocations[InputVectorForVTKfileCreator];
176
177   InputVectorForVTKfileCreator = {FileListFromSingleDumpTime, "B"};
178
179   CreateSingleVTKfileFromMultipleToroidalLocations[InputVectorForVTKfileCreator]; *)
180
181   InputVectorForVTKfileCreator = {FileListFromSingleDumpTime, "V"};
182
183   CreateSingleVTKfileFromMultipleToroidalLocations[InputVectorForVTKfileCreator];
184
185   Run["rm "<> DumpFileFilterCriterion,""];
186 ];
187
188 (* End of file Conversion_from_NIMROD_ascii_dump_(tecplot)_to_VTK_v09_(version_for_moller1m4).txt *)

```

Note that between lines 51 and 65 the choice of variables are the magnetic field \mathbf{B} , the current density \mathbf{J} , and the fluid velocity \mathbf{V} . Any of these variables comprises an array with their three field components. These components are taken from the columns of the NIMPLOT ASCII files. For example, the B_r , B_θ , and B_z components are taken from columns 16, 18, and 17, respectively (shown in lines 52 through 54). If it is desired to obtain scalar variables, the above function would have to be modified considerably to have the option of writing scalar array rather than vector arrays to the VTK file.

The list of dump cycles to process is shown in line 148. The length and content of this list are the choice of the user. The double quotes indicate that each element in the list is a string. Thus, if the same dump cycle list as the one in `Log_of_NIMROD_lam07_dump_files_and_times_02.csv` is desired, a simple routine that

extracts the cycle numbers and turns them into strings can be created (using, for example, the function shown in the code of Section II, line 39). Here the list elements were input manually in order to leave the computer process a few elements at a time - typically overnight or over weekends, and commenting out the rest of the list.

Line 167 shows the command for a secure file transfer protocol (*sftp*) operation with the server where the processed NIMPLOT files are stored. This allows the conversion of files at a terminal rather than the server. For this command to be successful without the intervention of the user at every cycle of the *For* loop (starting at line 155), passwordless secure shell (ssh) login should be setup beforehand. Alternatively, the *sftp* operation can be omitted if the conversion and VTK file storage is done in the same server.

The typical size of a VTK file that contains all the toroidal slices from a single dump file is greater than 80 MB. The files used here consists of 72 toroidal slices, each with a mesh of 76×150 data points . An excerpt of one of these VTK files produced with the code described in this section is shown below.

```
# vtk DataFile Version 2.0
NIMROD output for dump 06000
ASCII
DATASET STRUCTURED_GRID
DIMENSIONS 76 150 72
POINTS 820800 double
0.35 0. 0.85
0.3520833 0. 0.85
0.3541667 0. 0.85
0.35625 0. 0.85
.
.
.
POINT_DATA 820800
VECTORS Bfield double
-0.011474306000000012 0.1987628 -0.018406222
-0.010525617000000011 0.1986517 -0.018182434
-0.009598083800000013 0.1977903 -0.017962826
-0.008700034600000012 0.1972733 -0.017747452
.
.
.
```

Note how the node x, y, z locations have to be specified first (under the line "POINTS 820800 double") and then vectors corresponding to each node.

IV. THE MESH RECONSTRUCTED BY VISIT

One important property of the VTK `STRUCTURED_GRID` file is that mesh cells are inferred from the layout of the node points in the text file. That is, there is no need to specify the type of cells to be constructed in order to create an interconnected mesh. For the case of the VTK files constructed from 72 slices, the mesh inferred by VisIt is shown in Figure 1. Note the gap in the mesh in frames (b) and (c). This is an unfortunate property of the `STRUCTURED_GRID` format, since there is no information in the file on how to connect the two ends of the mesh. However, the lack of interpolation available in that gap does not affect the information in the rest of the volume.

Once the mesh is loaded into VisIt, the user can proceed to take advantage of all the plot functions and analysis tools in VisIt. A sample of the processing possible with VisIt is shown in Figure 2.

-
- [1] C. R. Sovinec, A. H. Glasser, T. A. Gianakon, D. C. Barnes, R. A. Nebel, S. E. Kruger, D. D. Schnack, S. J. Plimpton, A. Tarditi, and M. S. Chu. *J. Comp. Phys.*, 195:355, 2004.
 - [2] E. B. Hooper, L. D. Pearlstein, and R. H. Bulmer. *Nucl. Fusion*, 39:863, 1999.
 - [3] Lawrence Livermore National Laboratory, <http://www.llnl.gov/visit/manuals.html>. *VisIt User's Manual*, 2003.
 - [4] Kitware Inc., <http://public.kitware.com/VTK/pdf/file-formats.pdf>. *VTK File Formats*.
 - [5] Stephen Wolfram. *The Mathematica Book Online*. Wolfram Media, Inc., <http://documents.wolfram.com/mathematica/>.
 - [6] <http://nimrodteam.org/>

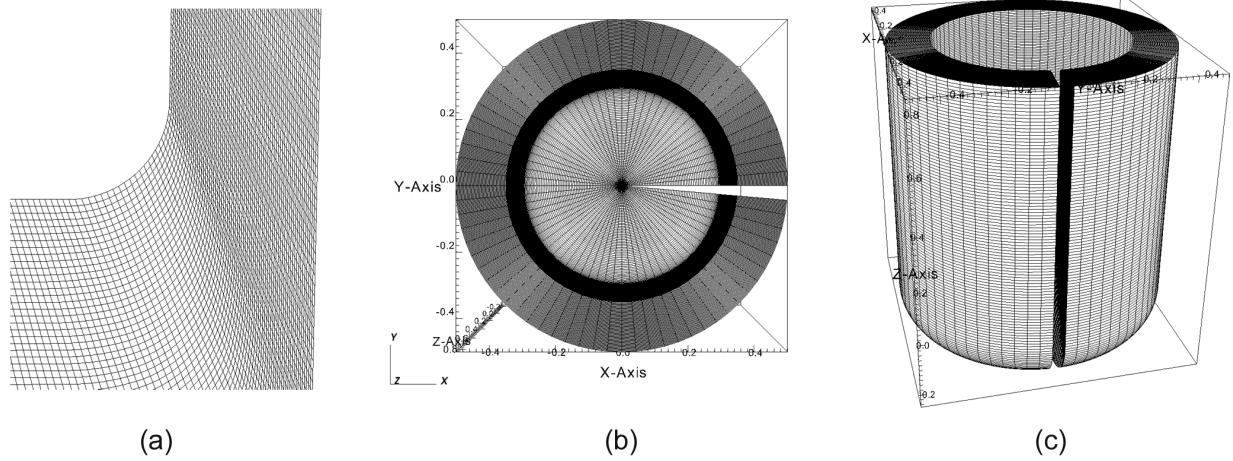


FIG. 1: Mesh inferred by VisIt from the VTK files. (a) Detail of the mesh. (b) Top view of the 3D mesh. (c) Side view.

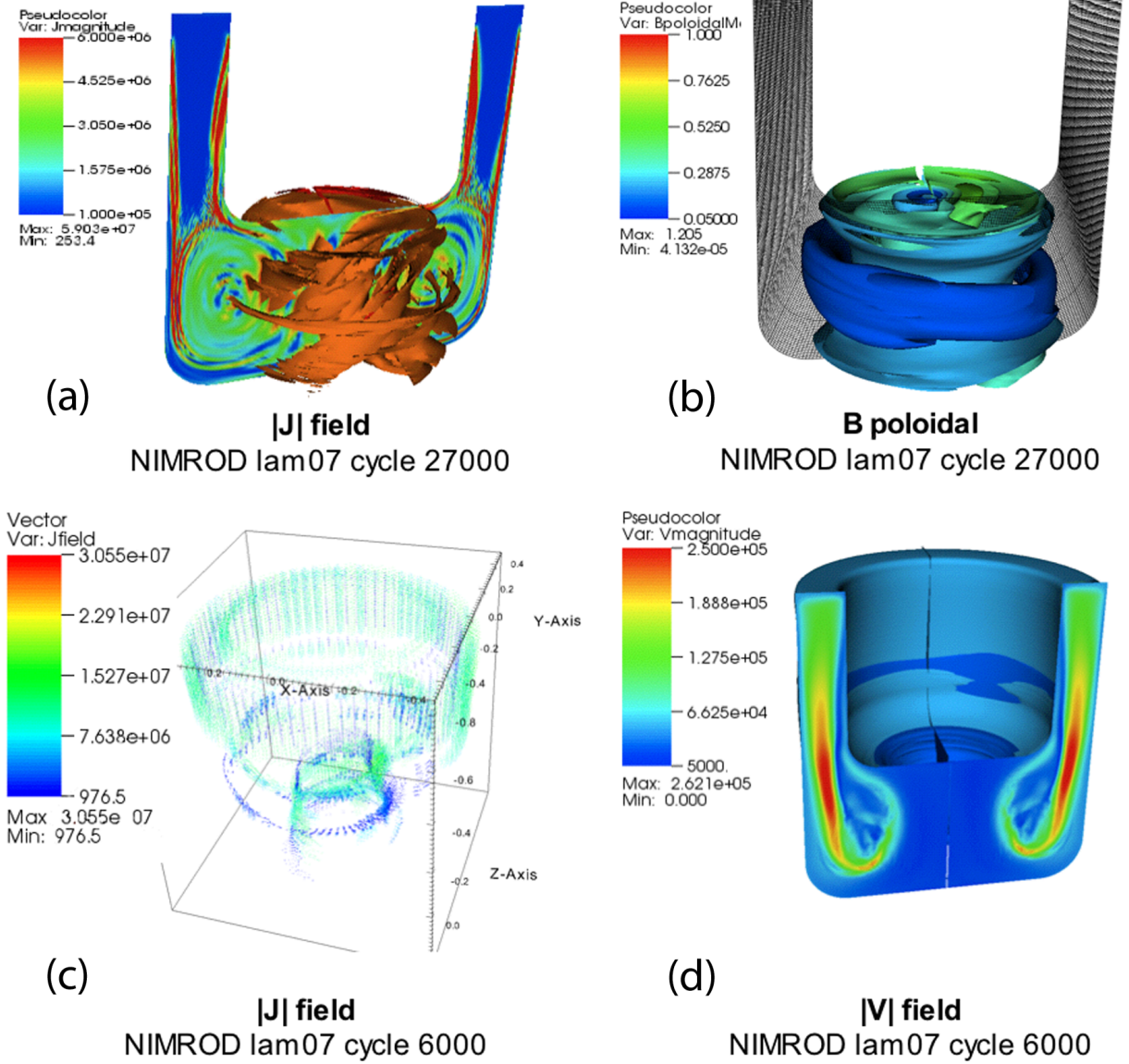


FIG. 2: Sample processing of VTK files with VisIt. (a) Magnitude of the current density J for cycle 27000. The volume plot shows thresholded values of J clipped near the geometrical and magnetic axes. (b) Magnitude of B poloidal (obtained through mathematical operations in VisIt) thresholded and clipped to show only selected values. (c) Thresholded view of J at cycle 6000 showing almost two revolutions in the current path at those values. (d) Magnitude of the velocity field for cycle 6000.