Zgoubi
A startup guide for the complete beginner

Annette Pressman and Kai Hock

May 21, 2014

Abstract

Zgoubi is a code which can be used to model accelerators and beam lines, comprised of magnetic and electrostatic elements. It has been extensively developed since the mid-1980s to include circular accelerators and related beam physics. It has been made freely available by its author on a code development site, including a Users' Guide, a data treatment/graphic interfacing tool, and many examples [1].

Contents

1 Getting started

1.1 Introduction to Zgoubi

Zgoubi began in the early 1970s as a code to be used for ray-tracing in large spectrometers for nuclear physics. Since then Zgoubi has been adapted such that it can be used to design beam optics and beam lines, and to simulate nonlinear dynamics. From the mid-1980's on, it has been developed and used extensively for the simulation of circular accelerators and storage rings, and related beam physics as spin dynamics, synchrotron radiation, short-lived beams, etc. It has been made freely available by its author on a code development site, including a Users' Guide, a data treatment/graphic interface program, zpop, and many examples [1–4].

The code is written in FORTRAN but users have developed additional interfaces for data handling and visualisation. In this guide reference will be made to MATLAB [5] scripts written for this purpose.

In the Zgoubi code, particle trajectories in electromagnetic fields are calculated through numerical integration of the Lorentz equations. As a result of the efficiency of the integration (through a Taylor-series based expansion) and the nature of the programming language used the computation time is very fast. The main program array is a linear description of the Zgoubi input file, named zgoubi.dat. This file contains a linear definition of the lattice and beam parameters, and is written for the specific simulation required.

1.2 Installation of zgoubi

Zgoubi can be run on Windows or UNIX systems. However the post-processing program, zpop, requires an xterm window so is better suited to a UNIX system. On most Windows systems it is possible to run a UNIX system such as Scientific Linux (`https://www.scientificlinux.org/`) using a product such as VirtualBox (`https://www.virtualbox.org/`), both of which are freely available. During the installation of Scientific Linux it is possible to select the developer options which should include the required FORTRAN compilers needed for Zgoubi.

The installation of Zgoubi on Windows and UNIX systems is covered in the following section. It is also possible to install Zgoubi through installation of PyZgoubi (see section 1.6) [6]. It is recommended that the new user gains some experience with Zgoubi to gain some understanding of how the code works before expanding to Zgoubi interfaces such as PyZgoubi.

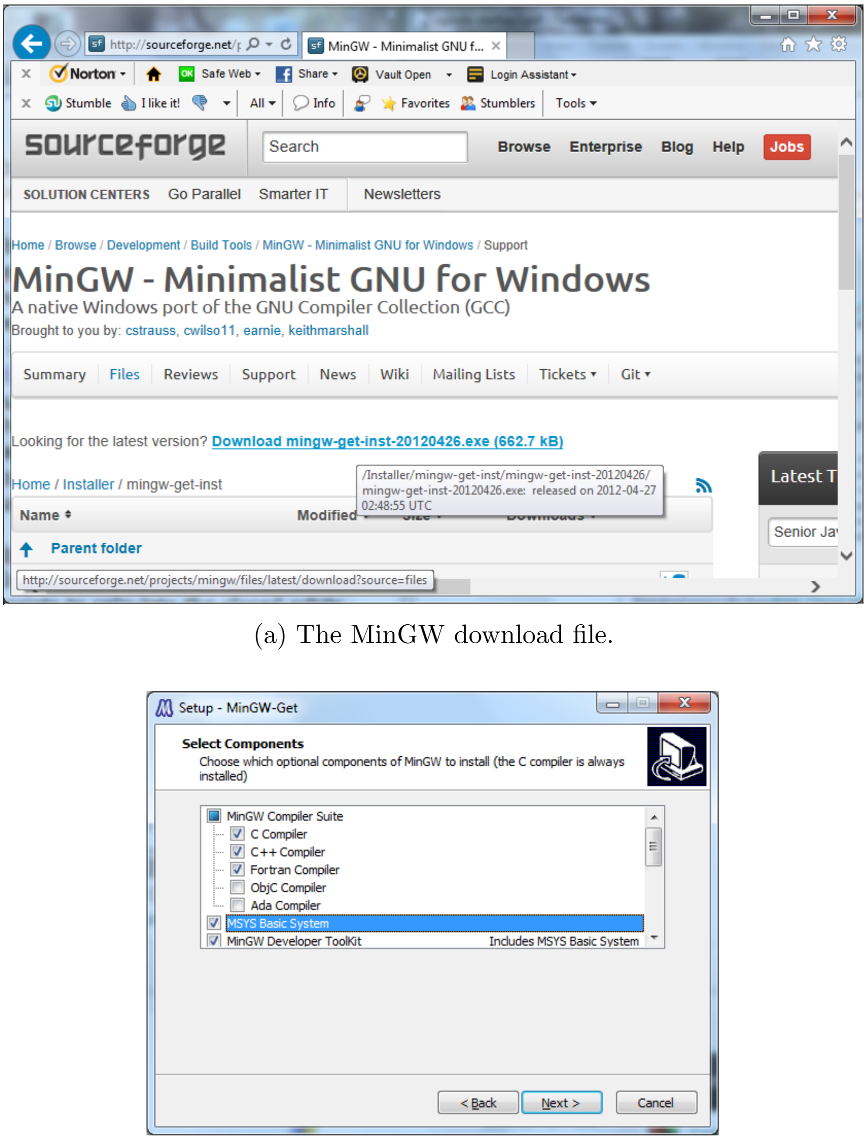
1.2.1 Required Software

To use Zgoubi you will need a FORTRAN compiler on your machine. Many UNIX systems will already have a compiler installed. On a Windows machine MinGW can be used [7].

To install MinGW;

1. Visit `http://www.mingw.org/wiki/Getting_Started`.

2. Click on ‘mingw-get-inst’, download and run the latest .exe version (figure 1a) to install to a directory of your choice, for example C:\MinGW. Ensure to select the FORTRAN option during the install process (figure 1b). Further information about this install is available on the ‘Getting Started’ webpage above.



MinGW - Minimalist GNU for Windows

A native Windows port of the GNU Compiler Collection (GCC)

Brought to you by: cstrauss, cwilso11, earnie, keithmarshall

Looking for the latest version? Download mingw-get-inst-20120426.exe (662.7 kB)

/Installer/mingw-get-inst/mingw-get-inst-20120426/
mingw-get-inst-20120426.exe: released on 2012-04-27 02:48:55 UTC

(a) The MinGW download file.

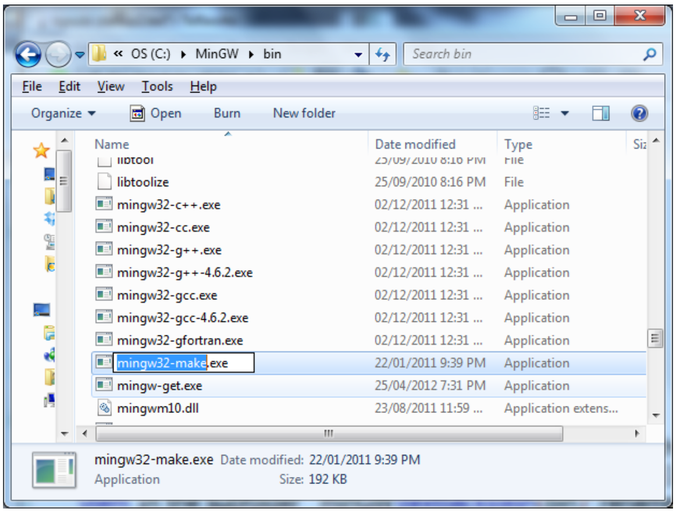
Select Components

Choose which optional components of MinGW to install (the C compiler is always installed)

- MinGW Compiler Suite
  - ☑ C Compiler
  - ☑ C++ Compiler
  - ☑ Fortran Compiler
  - ☐ ObjC Compiler
  - ☐ Ada Compiler
- ☑ MSYS Basic System
- ☑ MinGW Developer ToolKit — Includes MSYS Basic System

(b) Selecting MinGW install options.

Figure 1: Installing MinGW.



Name	Date modified	Type	Siz
libtool	25/09/2010 8:16 PM	File	
libtoolize	25/09/2010 8:16 PM	File	
mingw32-c++.exe	02/12/2011 12:31 ...	Application	
mingw32-cc.exe	02/12/2011 12:31 ...	Application	
mingw32-g++.exe	02/12/2011 12:31 ...	Application	
mingw32-g++-4.6.2.exe	02/12/2011 12:31 ...	Application	
mingw32-gcc.exe	02/12/2011 12:31 ...	Application	
mingw32-gcc-4.6.2.exe	02/12/2011 12:31 ...	Application	
mingw32-gfortran.exe	02/12/2011 12:31 ...	Application	
mingw32-make.exe	22/01/2011 9:39 PM	Application	
mingw-get.exe	25/04/2012 7:31 PM	Application	
mingwm10.dll	23/08/2011 11:59 ...	Application extens...	

mingw32-make.exe Date modified: 22/01/2011 9:39 PM
Application Size: 192 KB

Figure 2: Renaming to 'make.exe'.

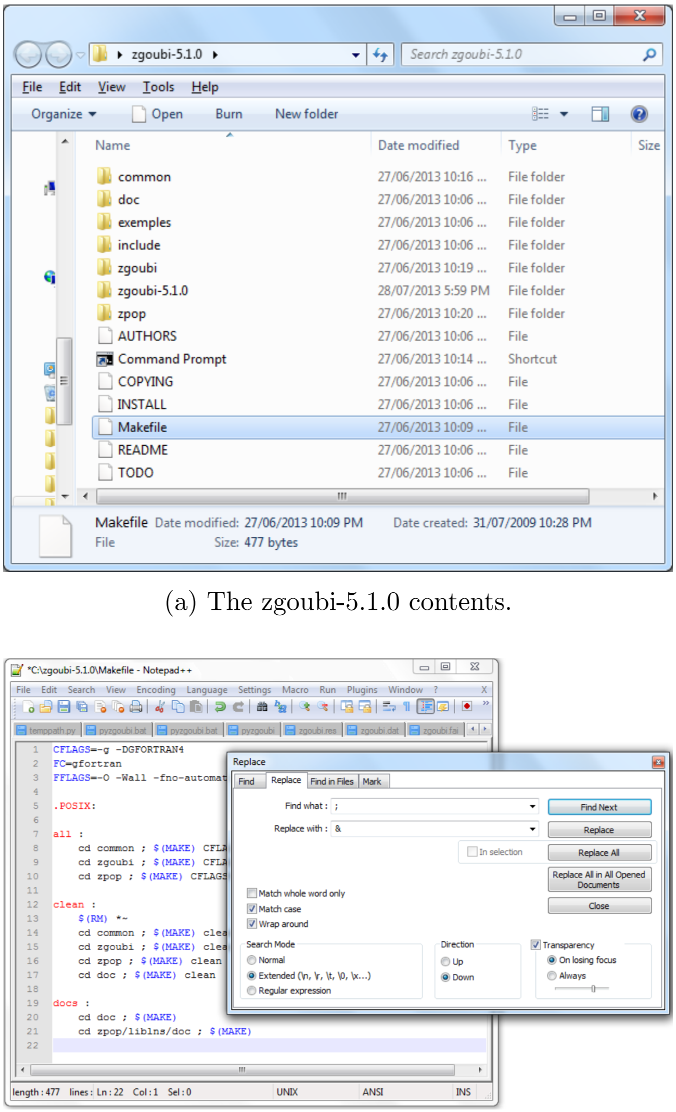
3. In the subfolder 'MinGW\bin' rename the file 'mingw32-make.exe' to 'make.exe' (figure 2).

4. Set the environmental variable with the address of the MinGW\bin directory (see section 1.2.2).

To install Zgoubi;

- Windows

  1. Download Zgoubi from source, e.g. `http://sourceforge.net/projects/zgoubi/` and unpack to a directory of your choosing, e.g. C:\zgoubi-5.1.0. Version 5.1.0 is stable but newer versions are available.
  2. To use Zgoubi in Windows; in the main directory of the unpacked folder and every subfolder edit the 'Makefile' to replace every occurrence of ';' with '&' (figure 3b): in UNIX the semicolon can be used separate two lines of command, but in Windows the ampersand performs this function.
  3. Set the environmental variable with the address of the file 'zgoubi.exe' (see section 1.2.2).
  4. Open a command prompt (cmd.exe) in the Zgoubi folder and type the command **make**. This should trigger an install of the program (see figure 4). In Windows the install is likely to complete with error messages for the install of 'zpop'. This graphics package needs to be run in an xterm window, which requires an X11 environment. This is not a trivial process for most Windows systems so it will not be covered here.



Name	Date modified	Type	Size
common	27/06/2013 10:16 ...	File folder	
doc	27/06/2013 10:06 ...	File folder	
exemples	27/06/2013 10:06 ...	File folder	
include	27/06/2013 10:06 ...	File folder	
zgoubi	27/06/2013 10:19 ...	File folder	
zgoubi-5.1.0	28/07/2013 5:59 PM	File folder	
zpop	27/06/2013 10:20 ...	File folder	
AUTHORS	27/06/2013 10:06 ...	File	
Command Prompt	27/06/2013 10:14 ...	Shortcut	
COPYING	27/06/2013 10:06 ...	File	
INSTALL	27/06/2013 10:06 ...	File	
Makefile	27/06/2013 10:09 ...	File	
README	27/06/2013 10:06 ...	File	
TODO	27/06/2013 10:06 ...	File	

Makefile Date modified: 27/06/2013 10:09 PM Date created: 31/07/2009 10:28 PM
File Size: 477 bytes

(a) The zgoubi-5.1.0 contents.

*C:\zgoubi-5.1.0\Makefile - Notepad++

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
temppath.py
pyzgoubi.bat
pyzgoubi.bat
pyzgoubi
zgoubi.res
zgoubi.dat
zgoubi.fai

```
CFLAGS=-g -DGFORTRAN4
FC=gfortran
FFLAGS=-O -Wall -fno-automat

.POSIX:

all :
	cd common ; $(MAKE) CFLA
	cd zgoubi ; $(MAKE) CFLA
	cd zpop ; $(MAKE) CFLAGS

clean :
	$(RM) *~
	cd common ; $(MAKE) clea
	cd zgoubi ; $(MAKE) clea
	cd zpop ; $(MAKE) clean
	cd doc ; $(MAKE) clean

docs :
	cd doc ; $(MAKE)
	cd zpop/liblns/doc ; $(MAKE)

```

length : 477 lines : Ln : 22 Col : 1 Sel : 0 UNIX ANSI INS

Replace

Find what : ;

Replace with : &

□ In selection

□ Match whole word only
☑ Match case
☑ Wrap around

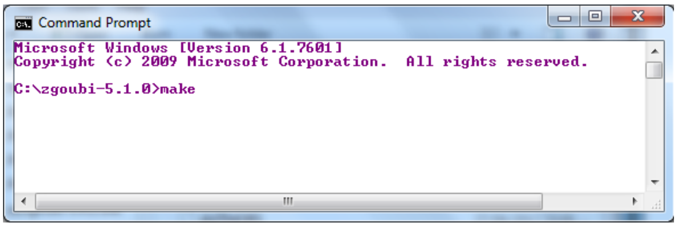
Search Mode
○ Normal
● Extended (\n, \r, \t, \0, \x...)
○ Regular expression

Direction
○ Up
● Down

☑ Transparency
● On losing focus
○ Always

(b) Editing 'Makefile'.

Figure 3: Editing the Zgoubi 'Makefile' to work in Windows.



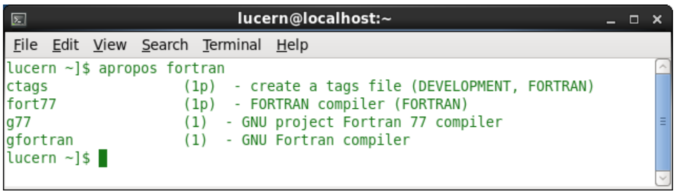
Command Prompt

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\zgoubi-5.1.0>make
```

Figure 4: Entering the command 'make' on the Windows command line.

- UNIX

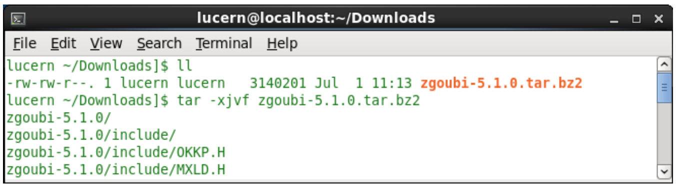
1. If you are unsure if there is a FORTRAN compiler on your system entering the command **apropos fortran** will show a list of the commands available and their use (figure 5). If there is no installer available it can be obtained using the command **sudo apt-get install gcc-gfortran**, or through the GUI software installer.



```
lucern ~]$ apropos fortran
ctags                  (1p)  - create a tags file (DEVELOPMENT, FORTRAN)
fort77                 (1p)  - FORTRAN compiler (FORTRAN)
g77                    (1)  - GNU project Fortran 77 compiler
gfortran               (1)  - GNU Fortran compiler
lucern ~]$ 
```

Figure 5: Expected response from the **apropos** command if FORTRAN compilers are installed.

2. Download source file from `http://sourceforge.net/projects/zgoubi/` (e.g. zgoubi-5.1.0.tar.bz2).
3. Unpack using the command **tar -xjvf zgoubi-5.1.0.tar.bz2** (figure 6).



```
lucern ~/Downloads]$ ll
-rw-rw-r--. 1 lucern lucern   3140201 Jul  1 11:13 zgoubi-5.1.0.tar.bz2
lucern ~/Downloads]$ tar -xjvf zgoubi-5.1.0.tar.bz2
zgoubi-5.1.0/
zgoubi-5.1.0/include/
zgoubi-5.1.0/include/OKKP.H
zgoubi-5.1.0/include/MXLD.H
```

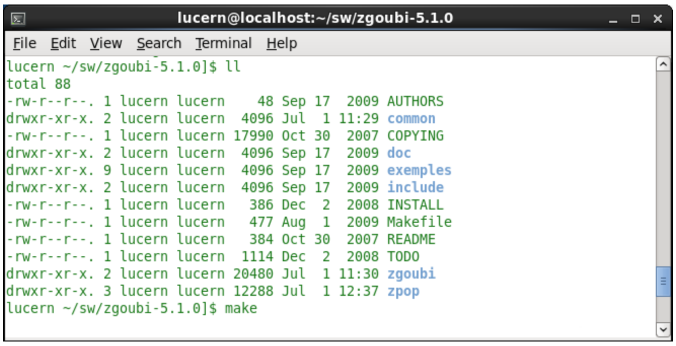
Figure 6: Unpacking Zgoubi download in Scientific Linux.

4. Move the unpacked directory 'zgoubi-5.1.0' to a directory of your choice, e.g. usr/local (moved to folder ~/sw in figure 7).
5. Within the directory 'zgoubi-5.1.0' enter the command **make** (figure 7).
6. Add the location of the folder 'zgoubi-5.1.0/zgoubi' to the Path - see section 1.2.2.

1.2.2 Setting Environmental Variables

Both MinGW and Zgoubi will need adding to the 'Path' variable. In order to do this, follow the instructions for the operating system used.

- Windows

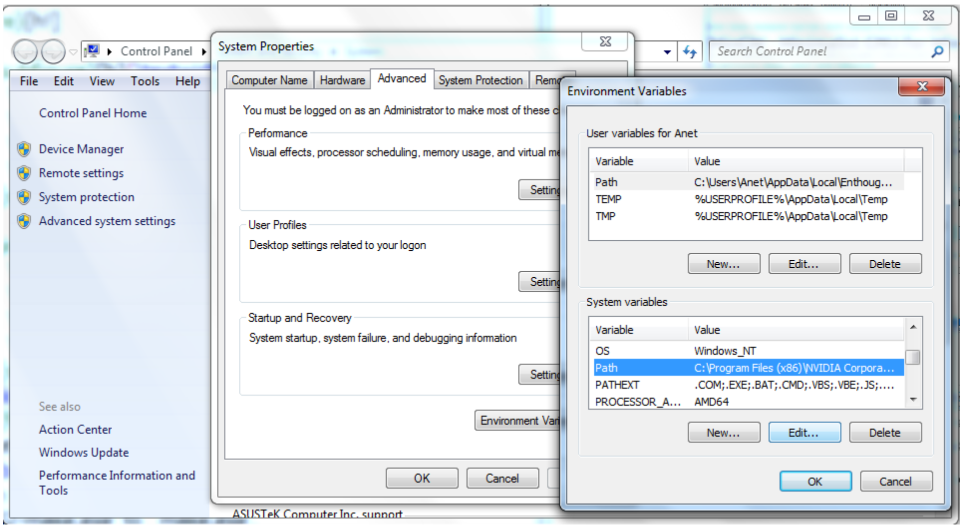


lucern@localhost:~/sw/zgoubi-5.1.0
File Edit View Search Terminal Help

```
lucern ~/sw/zgoubi-5.1.0]$ ll
total 88
-rw-r--r--. 1 lucern lucern    48 Sep 17  2009 AUTHORS
drwxr-xr-x. 2 lucern lucern  4096 Jul  1 11:29 common
-rw-r--r--. 1 lucern lucern 17990 Oct 30  2007 COPYING
drwxr-xr-x. 2 lucern lucern  4096 Sep 17  2009 doc
drwxr-xr-x. 9 lucern lucern  4096 Sep 17  2009 exemples
drwxr-xr-x. 2 lucern lucern  4096 Sep 17  2009 include
-rw-r--r--. 1 lucern lucern   386 Dec  2  2008 INSTALL
-rw-r--r--. 1 lucern lucern   477 Aug  1  2009 Makefile
-rw-r--r--. 1 lucern lucern   384 Oct 30  2007 README
-rw-r--r--. 1 lucern lucern  1114 Dec  2  2008 TODO
drwxr-xr-x. 2 lucern lucern 20480 Jul  1 11:30 zgoubi
drwxr-xr-x. 3 lucern lucern 12288 Jul  1 12:37 zpop
lucern ~/sw/zgoubi-5.1.0]$ make
```

Figure 7: Entering the **make** command in Scientific Linux.

1. From the Start menu right-click on ‘My Computer’ or ‘Computer’ and select ‘Properties’, and ’Advanced System Settings’.
2. Under the ‘Advanced’ tab select ‘Environmental Variables’ (figure 8).



You must be logged on as an Administrator to make most of these c

Visual effects, processor scheduling, memory usage, and virtual m

Desktop settings related to your logon

System startup, system failure, and debugging information

User variables for Anet

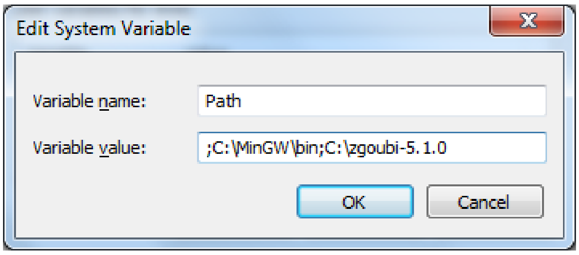
Variable	Value
Path	C:\Users\Anet\AppData\Local\Enthoug...
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

System variables

Variable	Value
OS	Windows_NT
Path	C:\Program Files (x86)\NVIDIA Corpora...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	AMD64

Figure 8: Setting the Path in Windows 7.

3. Here there are two boxes, the upper specifying user settings and the lower specifying settings for all users. Be careful editing these settings as a mistake can affect the correct operation of the computer.
4. In the lower box highlight ’Path’ and click ’Edit’.
5. At the end of the existing entry add the address for the MinGW bin directory and the Zgoubi directory, e.g. for the base directories described in the install above the add to the path ’;C:\MinGW\bin’ and ’;C:\zgoubi-5.1.0’ (figure 9).



Variable name: Path

Variable value: ;C:\MinGW\bin;C:\zgoubi-5.1.0

Figure 9: Inputing the path addresses.

- UNIX

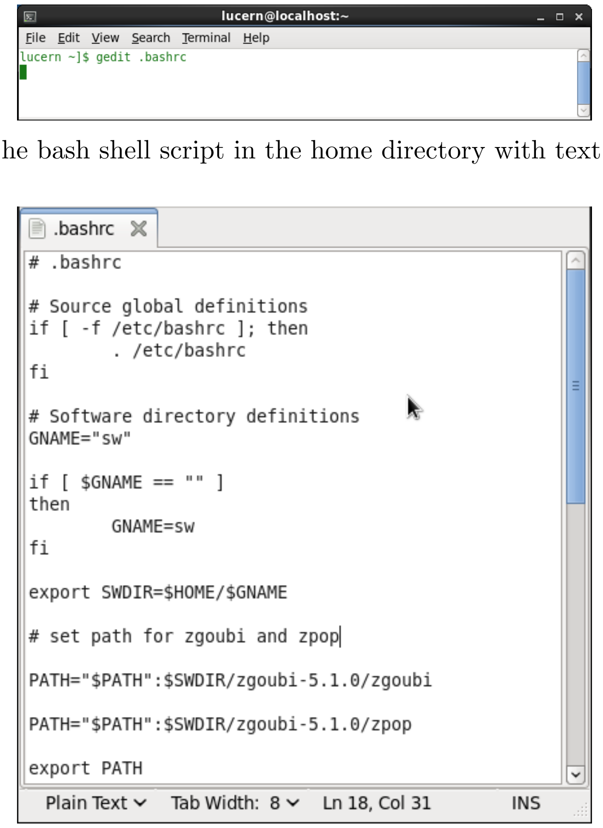
  1. Edit the startup file and modify the PATH variable to include the location of the file /zgoubi-5.1.0/zgoubi as follows (e.g. if this file is in the directory usr/local/);
     - bash shell
       (a) edit the '~/ .bashrc' file with the folowing;
       (b) PATH="$PATH":/usr/local/zgoubi-5.1.0/zgoubi
       (c) export PATH
     - C shell
       (a) edit the '~/ .cshrc' file with the following;
       (b) set path="$PATH":/usr/local/zgoubi-5.1.0/zgoubi
  2. Save and close the file.
  3. To use the graphics program 'zpop' another line can be added to the file using the location of the file '/zgoubi-5.1.0/zpop' in the same manner as above (inserted before the 'export PATH' line for the bash shell) as can be seen in figure 10b below.

1.2.3 Hints

Depending on your familiarity with Windows the following information may be unnecessary.

- You will need a text editor to edit FORTRAN files. 'Notepad++' (`http://notepad-plus-plus.org/`) is freely available and an easy to use option for Windows.

- In order to quickly open the command prompt in the correct folder on each use, paste a shortcut to the cmd.exe into the example folder, right click and under 'Properties', in the 'Start in' field copy and paste the address of the folder containing the zgoubi.dat file.



```
lucern ~]$ gedit .bashrc
```

(a) Opening the bash shell script in the home directory with text editor 'gedit'.

```
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# Software directory definitions
GNAME="sw"

if [ $GNAME == "" ]
then
        GNAME=sw
fi

export SWDIR=$HOME/$GNAME

# set path for zgoubi and zpop

PATH="$PATH":$SWDIR/zgoubi-5.1.0/zgoubi

PATH="$PATH":$SWDIR/zgoubi-5.1.0/zpop

export PATH
```

Plain Text ⌄ Tab Width: 8 ⌄ Ln 18, Col 31 INS

(b) The completed shell script.

Figure 10: Editing the shell script in Scientific Linux.

1.3 Running Zgoubi

In Windows the program is run from the command line (cmd.exe). In both Windows and Linux, at the command line within the folder containing the zgoubi.dat file to be used, type **zgoubi** and enter (see figure 11).

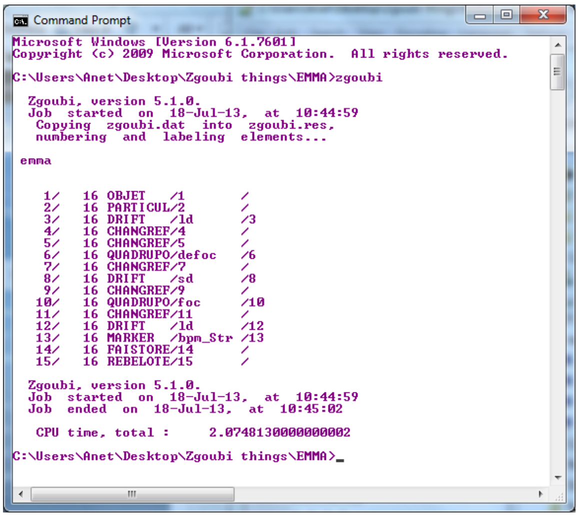
A successful run will output a list of elements as specified in the .dat file and produce a results file, 'zgoubi.res' alongside other files such as;

- zgoubi.fai - particle coordinates and spin information. This stores data at the end of each magnet - it is useful for analysing phase space.

- zgoubi.spn - spin coordinate information.

- zgoubi.plt - particle coordinates and fields experienced by the particles stepwise along the beamline. This stores data points within a magnetic field only, drift spaces don't store tracking data in this file.

- zgoubi.map - 2-D field map information.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Anet\Desktop\Zgoubi things\EMMA>zgoubi

  Zgoubi, version 5.1.0.
  Job  started  on  18-Jul-13,  at  10:44:59
   Copying  zgoubi.dat  into  zgoubi.res,
   numbering  and  labeling  elements...

 emma


    1/   16 OBJET   /1        /
    2/   16 PARTICUL/2        /
    3/   16 DRIFT   /ld       /3
    4/   16 CHANGREF/4        /
    5/   16 CHANGREF/5        /
    6/   16 QUADRUPO/defoc    /6
    7/   16 CHANGREF/7        /
    8/   16 DRIFT   /sd       /8
    9/   16 CHANGREF/9        /
   10/   16 QUADRUPO/foc      /10
   11/   16 CHANGREF/11       /
   12/   16 DRIFT   /ld       /12
   13/   16 MARKER  /bpm_Str /13
   14/   16 FAISTORE/14       /
   15/   16 REBELOTE/15       /

  Zgoubi, version 5.1.0.
  Job  started  on  18-Jul-13,  at  10:44:59
  Job  ended  on  18-Jul-13,  at  10:45:02

   CPU time, total :     2.0748130000000002

C:\Users\Anet\Desktop\Zgoubi things\EMMA>_
```

Figure 11: Running Zgoubi in Windows.

The results output overview of Zgoubi is written to the zgoubi.res file. The particle tracking and field data are stored to the other output files. The data can be visualised using the Zpop code for users with access to an xterm window. Otherwise extracting the required data can be done using other program interfaces. Dr Kai Hock has written MATLAB scripts to perform this function (see section 4.1). There are also add-on scripts such as PyZgoubi which uses a python interface to input lattice data and analyse the results (see section 1.6).

Further information about the Zgoubi output files can be found in Part D of the Zgoubi User's Guide.

1.4 Running zpop

The Zgoubi code comes with a graphics package called zpop. It can only be run in an xterm window which can be accessed on most UNIX systems with the command **xterm**. An example of use can be seen in section 3.2.1.

1.5 How to use the Zgoubi User's Guide

The Zgoubi User's Guide is split into four sections. The first section, Part A - 'Description of software contents', provides information about the physics processes considered, the procedures available and the mathematical methods used to perform the integrations. The section begins with a glossary of the keywords used in the program. The keywords are how the user defines the accelerator lattice in the input zgoubi.dat file. Further information about the keywords can be found in section 4 of Part A, with details about how to use the keywords to be found in section B.

Part B - ‘Keywords and input data formatting’ details all the keywords used in the program. To enter lattice information into the zgoubi.dat file a keyword is used, followed by further numerical or text data to define the use of the keyword. As a simple example a drift section is defined in zgoubi.dat with the keyword ‘DRIFT’ with the length of the section (in cm). A few other examples will be discussed in section 2 of this guide.

In the introduction to Part B a description is given of how to input the information given for each keyword. An important thing to note is the units expected for each keyword. As an example, both radians and degrees are used at different occasions. When defining a lattice careful reference to the use of the keywords as defined in this section is strongly recommended. The allowed values for each keyword and the units to be used can be found in part B.

In Part C - ‘Examples of input data files and output result files’ several zgoubi.dat files are given as examples of different accelerator definitions. The resultant zgoubi.res files are also shown.

The information regarding instillation and running of Zgoubi is contained within Part D - ‘Running Zgoubi and its post-processor/graphic interface zpop’.

1.6 PyZgoubi - a python interface to Zgoubi

PyZgoubi is a freely available interface to Zgoubi written in python (`http://www.hep.manchester.ac.uk/u/sam/pyzgoubi/`). It has been designed to make reading and writing input data files easier, and takes advantage of python’s utilities and analysis tools to interpret the output [8]. It requires an installation of python with the packages of NumPy, SciPy and Matplotlib. It is also possible to install Zgoubi from the PyZgoubi installation if the OS has the GFortran compiler installed.

1.6.1 Installing PyZgoubi on Windows

Python can be installed on Windows with the required packages and libraries, NumPy, SciPy, Matplotlib either as individual modules (see `http://www.scipy.org/`) or through the single package options such as Anaconda (`http://continuum.io/downloads`) or Enthought Canopy (`https://www.enthought.com/products/epd/free/`). The 32 bit version is free to all users. Links to these products and more are available on the PyZgoubi homepage above. The environmental variables need be amended to include the executable ’python.exe’ and also the library site packages, which may have an address such as ’`C:\Users\Username\AppData\Local\Enthought\Canopy32\User\Lib\site-packages`’.

After installing a python package with the above libraries PyZgoubi can be downloaded from sourceforge via the homepage (`http://www.hep.`

manchester.ac.uk/u/sam/pyzgoubi/). After downloading it can be unpacked to a directory such as 'C:\pyzgoubi' through a MinGW (see 1.2.1 for MinGW information) interface opened in the directory containing the tar.gz file (for example 'filename.tar.gz') using the following command;

```
        tar -zxvf filename.tar.gz
```

After unpacking PyZgoubi, through a command prompt open in the pyzgoubi folder enter the command **python setup.py install**, or to install to a specific destination (e.g. /python/install/) with the command **python setup.py install --prefix=~/python/install/**. The installation dialogue will offer suggestions of amendments to be added to the environmental Path variable which should be updated accordingly, along with the address of the PyZgoubi executable (see section 1.2.2).

Depending on the installation some small amendments may be necessary to run PyZgoubi. A batch file in the pyzgoubi folder (titled for example 'pyzgoubi.bat') can help pyzgoubi to locate the python.exe executable. An example is shown below for an Enthought installation of python.

```
@echo off
python C:\Users\Username\AppData\Local\Enthought\Canopy32\User\
    Scripts\pyzgoubi %1
```

In the python file 'core.py' within the 'run(...)' loop a temporary directory is made with a prefix. This can cause problems in Windows which can be patched by changing the definition of 'tmpdir = tempfile.mdktemp(...)' in this loop to 'tmpdir = tempfile.mkdtemp()'. On some systems the 'settings.ini' file in the .pyzgoubi folder may need to be edited to ensure the path names are correct.

1.6.2 Installing PyZgoubi on UNIX

The PyZgoubi homepage contains links and information about how to install the code. The required packages of SciPy, NumPy and Matplotlib can be installed through Anaconda (link from the homepage) or other sources. Anaconda will also suggest appropriate changes to the PATH variable or offer to adjust the shell script automatically.

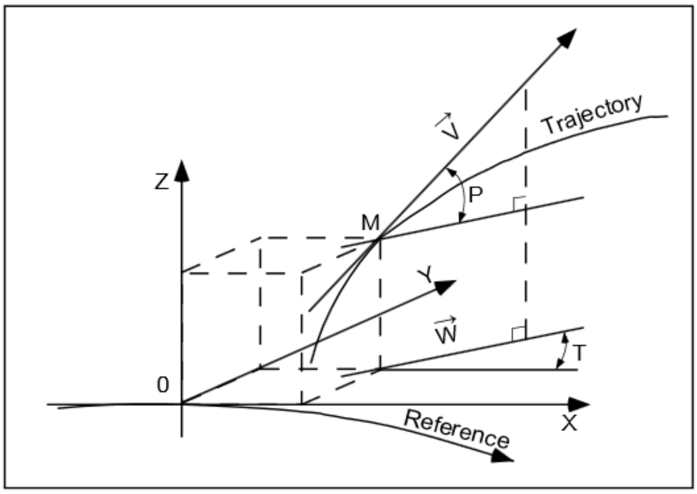
After a successful installation of python with these libraries on a system with a g77 compiler it should now be possible to download and install PyZgoubi following the advice on the homepage and in the README included with the source code. The shell environment needs to be configured to include the paths for the python executable and paths for the python libraries (PYTHONPATH) as advised during the installation procedure.

2 Defining a lattice

To define an accelerator lattice for Zgoubi the details need to be saved into a text file named ‘zgoubi.dat’. Keeping this input file for different accelerators in separate directories is essential to avoid accidentally overwriting one with another. In the file each element of the accelerator is specified in turn.

2.1 Coordinates

It is important to note that Zgoubi always works in local coordinates, not global coordinates. A right-handed system is used but the terminology may be slightly different to other commonly used systems, most notably in that the x axis is along the general direction of motion. The x axis is zeroed at the beginning of a component. In the figure below (see figure 12) the origin is in the median plane on a reference curve which coincides with the optical axis of optical elements.



Reference frame and coordinates (Y, T, Z, P) in **zgoubi**.
OX : in the direction of motion,
OY : normal to OX,
OZ : orthogonal to the (X, Y) plane,
$\vec{W}$: projection of the velocity, $\vec{v}$, in the (X, Y) plane,
T = angle between $\vec{W}$ and the X-axis,
P = angle between $\vec{W}$ and $\vec{v}$.

Figure 12: The Zgoubi coordinate reference frame. Taken from [2]

2.2 Keywords and labels

In the zgoubi.dat file a keyword is used (in capitals and in quotes) to input data about every step of the lattice. The keyword and the data following it is read into the Zgoubi program. Each keyword can have two 'labels' on the same line.

```
'KEYWORD' label1 label2
```

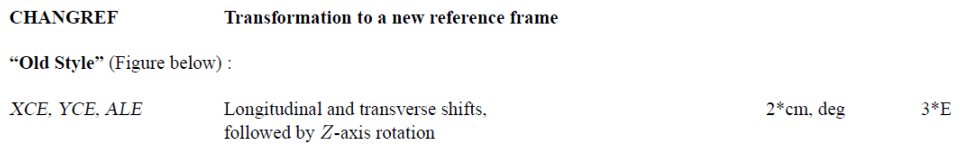
Although this label can be free text and used as an aide memoir, it can also on occasion be used to produce a result. For the keyword 'MARKER', if the second label is **.plt** the current coordinates will be stored to the file named 'zgoubi.plt'. This file can be used with Zpop to produce plots. As another example, if given the name of a label the keyword 'FAISTORE' will create a '.fai' file for storage of particle data at any element where the label is used. An example is given below. Here the output file will be called 'zgoubi.fai' and will store data at the exit of any element labelled **label1**.

```
 'FAISTORE'
 zgoubi.fai label1
```

The details about the available keywords and how to use them can be found in Part B of the Zgoubi User's Manual. Further information and examples of use can also be found in the Zgoubi Tutorial [9] available online.

2.3 Handling rotations and translations

A translation and/or rotation may be required between different elements which can be achieved by using the 'CHANGREF' keyword. The CHANGREF keyword is used to define transverse and longitudinal shifts and a z-axis rotation. The entry for this keyword in Part B of the Zgoubi User's Guide is shown in figure 13 below.



CHANGREF **Transformation to a new reference frame**

"Old Style" (Figure below) :

XCE, YCE, ALE	Longitudinal and transverse shifts, followed by *Z*-axis rotation	2*cm, deg	3*E

Figure 13: An excerpt from the Zgoubi User's Guide [2].

The above entry gives an explanation of the function of the term, the units required (where '2*cm deg' means two separate entries in cm followed by one entry in degrees), and each entry to be in real numbers (as indicated by '3*E'). As such, a rotation of the frame of reference by 10 degrees with respect to the z axis of the original frame would be entered into zgoubi.dat as below.

```
'CHANGREF'
0 0 -10
```

2.4 Defining initial particle coordinates

The keyword for the initial particle is 'OBJET'. The input parameters are specified over the subsequent five lines of code as can be seen in the example code below. With reference to Part B of the User's Guide it can be found that the number on the second of these lines (with value '2' in the code below) selects options about the subsequent variables. It is again important to note the units used ($1\,\mathrm{G} = 10^{-4}\,\mathrm{T}$). The units for the particle coordinates can be in different units dependent on the number option used on line 2.

```
'OBJET'
-36.689 *Rigidity - note the units of kG.cm : 10.5 eV electron
2       *Selects menu option 2
1 1     *Total no. of particles, no. of distinct momenta
-0.9 45 0.001 0.01 1  e         *y, y', z, z', dl, p/p_0
1       *Switch to toggle ray-tracing or not
```

This defined object describes the starting coordinates at the beginning of a cell in the lattice. If these coordinates were all set to zero the particle would travel along the reference path. However this may not be the best starting path, and may result in the particle getting lost after travelling though a few components. Some trial and error is likely to be involved to check if a particle can travel through the lattice correctly. Careful selection of a starting point for a cell - often at a point of symmetry - may mean that the right starting coordinates can be found by varying the horizontal displacement (the Y value) alone.

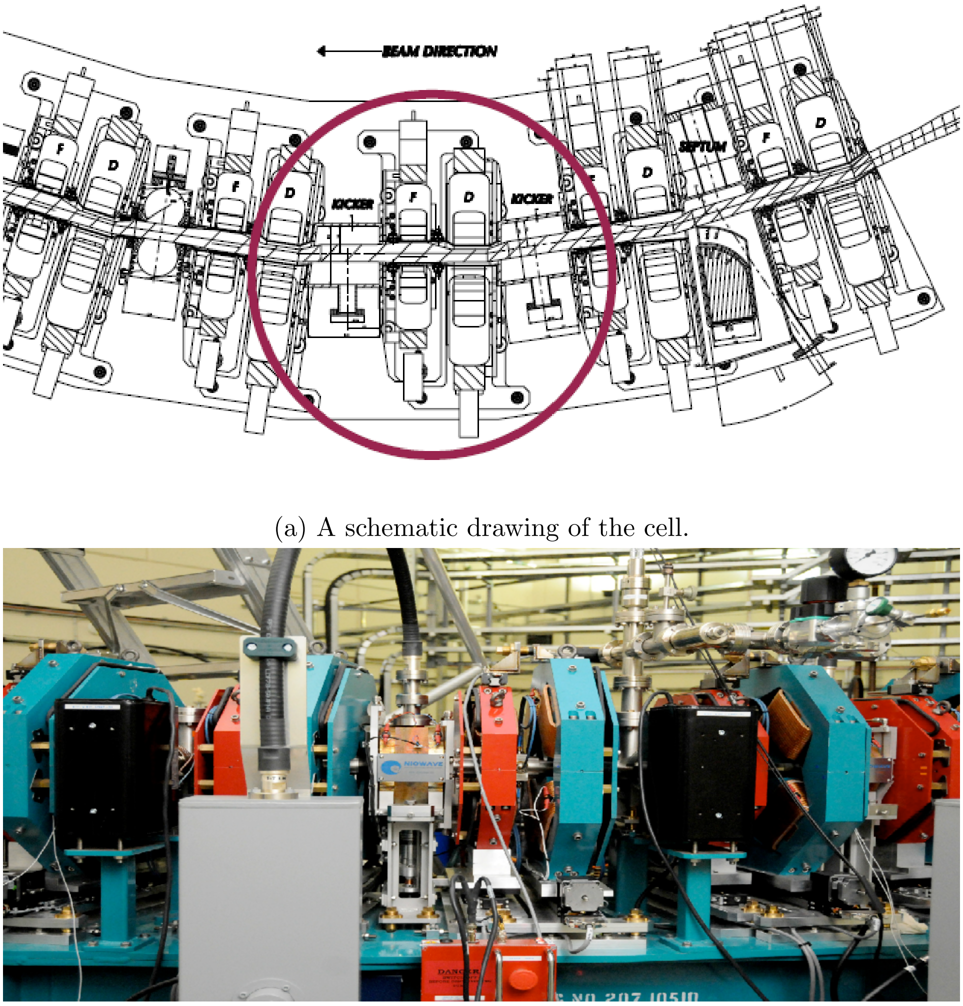
2.5 Extracting local particle coordinates

Printing out the local particle coordinates into zgoubi.res can be done with the keyword 'FAISCEAU'. However if the coordinates are to be stored (ie saved to file for later analysis) the keyword 'FAISTORE' should be used. This saves the particle data to a filename (filename.fai) to be specified under the keyword. The keyword 'MARKER' can be used to free-text name a point in the lattice. For example the following code would label a point in the lattice as **three** and produce a .fai file 'three_zgoubi.fai' which would record the particle data at every occurrence of the marker 'three'.

```
'MARKER'    three
'FAISTORE'
three_zgoubi.fai three
2      *This value indicates how frequently the data is
   recorded, i.e. every second lap around the ring
```

3 Example: The EMMA lattice

EMMA (Electron Model of Many Applications) is a non-scaling FFAG accelerator. It consists of 42 doublet quadrupole cells (see figure 14) in a 16.6 m circumference ring which accelerate electrons from $10 - 20$ MeV (kinetic energy) [10].



(a) A schematic drawing of the cell.

(b) A photograph of the cell.

Figure 14: The EMMA quadrupole cell [11].

As the EMMA ring contains 42 replications of the same doublet cell the lattice definition is comparatively straightforward (see figure 15) - after definition the cell can be repeated 42 times to produce the complete ring.

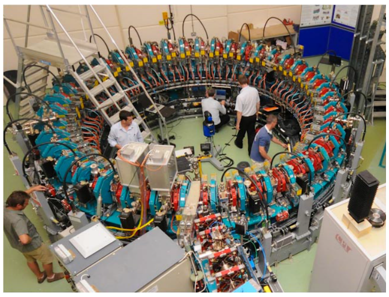
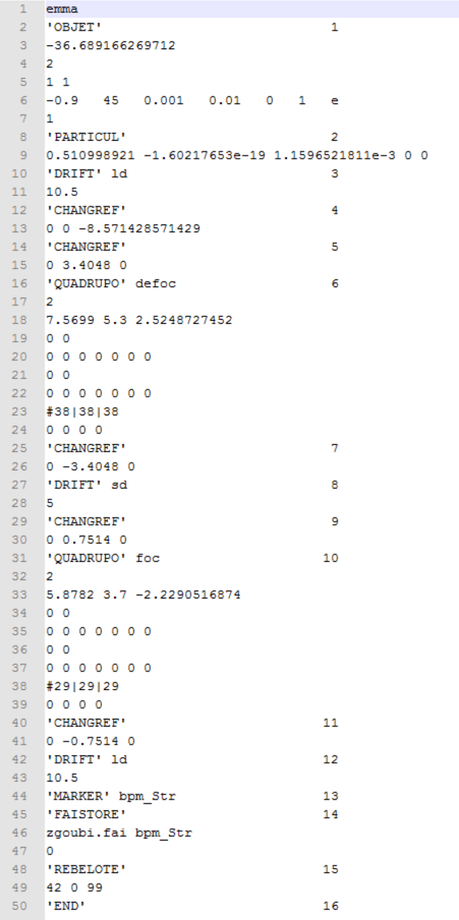


Figure 15: The complete EMMA ring [11]. Each blue and red module is the focusing/defocusing quadrupole doublet, separated from the next doublet by an RF cavity. There are no dipoles used.

3.1 The zgoubi.dat file

The 'zgoubi.dat' input file for the EMMA lattice can be seen in figure 16. This section will explain the content of this file line by line using some of the components as described in the previous section.

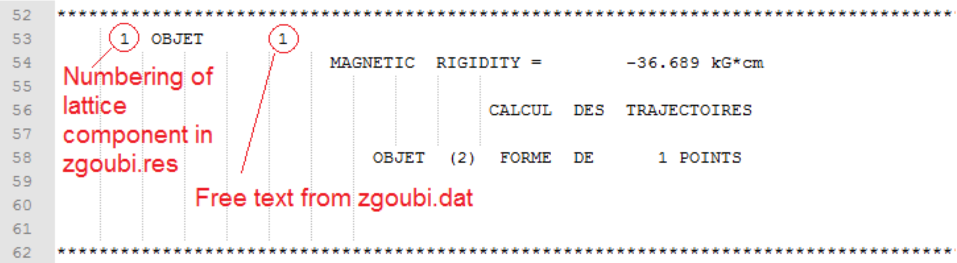
Along the right hand side of the data in this file it can be seen that each keyword has been labelled in numerical order. This is not an essential part of the file, but can make matching the keyword to the output in the zgoubi.res file more simple. It can be seen in figure 17 that the results file zgoubi.res numbers each lattice element (the number before the keyword). Labelling each element in the input file with the corresponding number can make it easier to locate an input element with its output. For this lattice, with only 16 keywords used, it is not very difficult to identify each element in the zgoubi.res file and match it to the corresponding element in the zgoubi.dat file but in more complicated lattices this is less straightforward.



```
 1  emma
 2  'OBJET'                                1
 3  -36.689166269712
 4  2
 5  1 1
 6  -0.9   45   0.001   0.01   0   1   e
 7  1
 8  'PARTICUL'                             2
 9  0.510998921 -1.60217653e-19 1.1596521811e-3 0 0
10  'DRIFT' ld                             3
11  10.5
12  'CHANGREF'                             4
13  0 0 -8.571428571429
14  'CHANGREF'                             5
15  0 3.4048 0
16  'QUADRUPO' defoc                       6
17  2
18  7.5699 5.3 2.5248727452
19  0 0
20  0 0 0 0 0 0 0
21  0 0
22  0 0 0 0 0 0 0
23  #38|38|38
24  0 0 0 0
25  'CHANGREF'                             7
26  0 -3.4048 0
27  'DRIFT' sd                             8
28  5
29  'CHANGREF'                             9
30  0 0.7514 0
31  'QUADRUPO' foc                        10
32  2
33  5.8782 3.7 -2.2290516874
34  0 0
35  0 0 0 0 0 0 0
36  0 0
37  0 0 0 0 0 0 0
38  #29|29|29
39  0 0 0 0
40  'CHANGREF'                            11
41  0 -0.7514 0
42  'DRIFT' ld                            12
43  10.5
44  'MARKER' bpm_Str                      13
45  'FAISTORE'                            14
46  zgoubi.fai bpm_Str
47  0
48  'REBELOTE'                            15
49  42 0 99
50  'END'                                 16
```

Figure 16: The zgoubi.dat file for the EMMA lattice.

Line 1 of the zgoubi.dat file is for free text, in order to identify the lattice. The first keywords needed for this, and most lattices, are 'OBJET' and 'PARTICUL'. OBJET defines the object, i.e. the initial particle coordinates and PARTICUL defines the

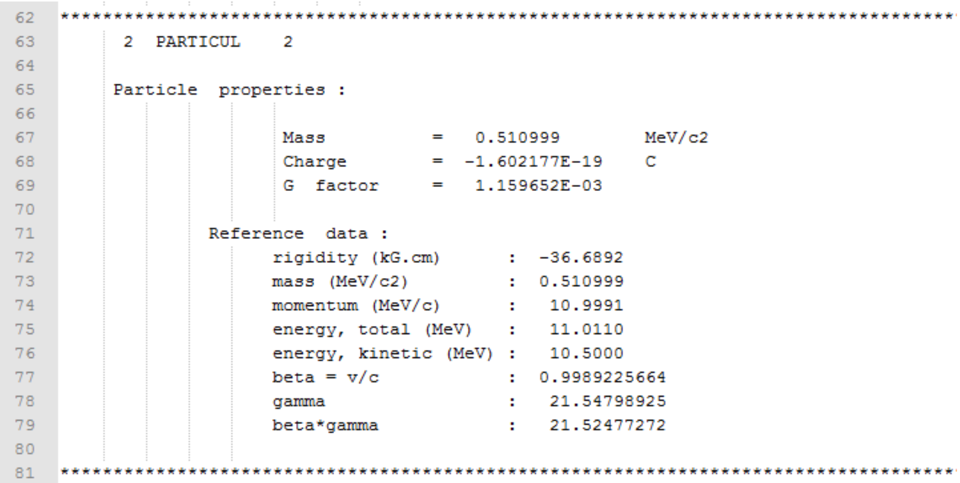


```
*******************************************************************************************
      1  OBJET              1
                              MAGNETIC  RIGIDITY =          -36.689 kG*cm

                                           CALCUL  DES  TRAJECTOIRES

                                  OBJET  (2)  FORME  DE       1 POINTS

*******************************************************************************************
```

Numbering of lattice component in zgoubi.res
Free text from zgoubi.dat

Figure 17: Except from zgoubi.res regarding keyword 'OBJET'.

characteristics of the particles in the beam. For EMMA the beam consists of electrons. The zgoubi.res file output gives further details about the particle from the input data from OBJET and PARTICUL (see figure 18). It is strongly advised to check this file after defining a lattice to ensure this output is correct, and the data has been input correctly.



```
*******************************************************************************************
      2  PARTICUL    2

    Particle  properties :

                     Mass          =   0.510999        MeV/c2
                     Charge        =  -1.602177E-19    C
                     G  factor     =   1.159652E-03

              Reference  data :
                    rigidity (kG.cm)       :  -36.6892
                    mass  (MeV/c2)         :  0.510999
                    momentum  (MeV/c)      :   10.9991
                    energy, total  (MeV)   :   11.0110
                    energy, kinetic  (MeV) :   10.5000
                    beta = v/c             :  0.9989225664
                    gamma                  :   21.54798925
                    beta*gamma             :   21.52477272

*******************************************************************************************
```

Figure 18: Except from zgoubi.res regarding keyword 'PARTICUL'.

At this point in the input file the cell can be defined. In EMMA the cell consists of the quadrupole doublet followed by a drift of 21 cm. The next cell is then at an angle to the first. As there are 42 cells in the complete ring this angle is $360°/42 = 8.57°$ (actually $-8.57°$ with respect to the x axis to describe a rotation towards the centre of the ring) . For this lattice the drift section has been described as two 10.5 cm sections at the beginning and end of the cell. This means the rotation of reference frame appears within the cell description rather than at the end as expected. The reason for this relates to the initial injection of particles in the OBJET definition. In order to obtain a stable orbit around the ring some trial and error with the initial particle coordinates is required. Using the middle of the drift space as the start of the cell allows a stable orbit to be found by varying the horizontal displacement (Y) alone. If the start/end of the cell involved the rotation the angle theta (T) would need to be varied too, making the process of finding a stable orbit more complicated. So it can be seen in figure 16 at line 10 the first drift section of 10.5 cm is defined, which is then followed by a change in reference frame of a rotation of $-8.57°$. The other half of the drift cavity is defined at the end of the cell on line 42. This defines a cell represented by the schematic picture in figure 19.

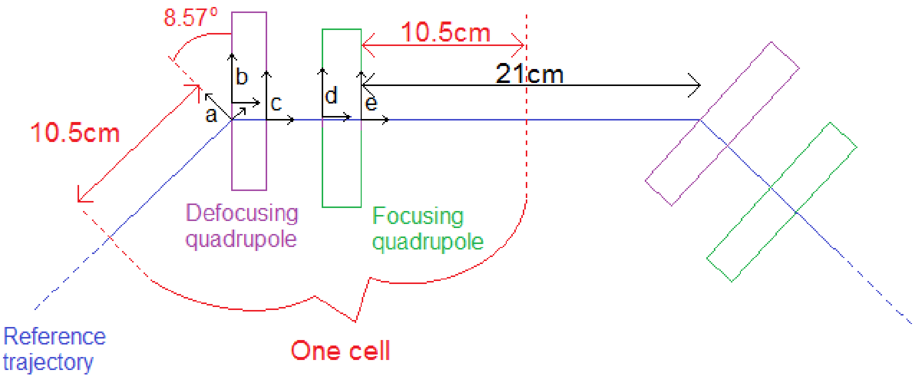


Figure 19: Schematic drawing of the cell as defined in zgoubi.dat.

The CHANGREF on line 12 of zgoubi.dat rotates the coordinate system from 'a' in figure 19 to one with the x-axis parallel to 'b'. The following CHANGREF on line 14 then performs a translation to move the coordinates to 'b', with the x-axis in line with the centre of the defocusing quadrupole. Line 16 then defines the defocusing quadrupole, with the label 'defoc' and is followed by a CHANGREF which reverses the previous translation and moves the coordinates to 'c'. After a further drift section there is another change of coordinates to 'd' before the definition of the focusing quadrupole on line 31. One final CHANGREF moves the coordinates to 'e' before the final drift section is defined at line 42.

The defocusing and focusing quadrupoles are defined at lines 16 and 31 respectively. The initial value of '2' is an instruction to print the field and coordinates along the trajectories. The following lines (18 and 33) define the length, radius and magnetic field at the pole tip for each quadrupole. The focusing magnet can be seen to have a negative value for this field. Both quadrupoles are then defined with four lines consisting of zero values. These lines can be used to define the fringe magnetic fields - the 'entrance' and 'exit' fringe fields. With zero values a 'sharp edge' magnetic field approximation is used.

On lines 23 and 38 three values are entered to define the 'XPAS' - this defines the number of steps for the integration process in the entrance field, main field and exit field of each component. If passed a single value a step length in cm would be used.

At line 44 a marker is placed. This itself has no impact on the beam (if given a second keyword of **.plt** data from this point in the lattice will be stored in the zgoubi.plt file) but marks the position on the lattice with the label **bpm_str**. The following keyword 'FAISTORE' stores particle information in a .fai file given as an argument, here **zgoubi.fai**. The information is stored at every occurence of the labels given on line 46 which in this case is bpm_str. So the net result of lines 44-47 is to produce a file output of particle data at the end of the cell where the marker is placed.

The 'REBLOTTE' keyword on line 48 of figure 16 is a command to repeat the zgoubi.dat file. Here it is given the values '42 0 99' which translates as 42 repetitions (i.e. one complete ring) with the '0' specifying a level of verbosity in the **.res** file and the '99' indicating that the particle coordinates at the end of one pass are used as initial coordinates for the next pass. To track the particle through several laps of the ring a multiple of 42 would be used for the first value.

3.2 Running EMMA and sample output

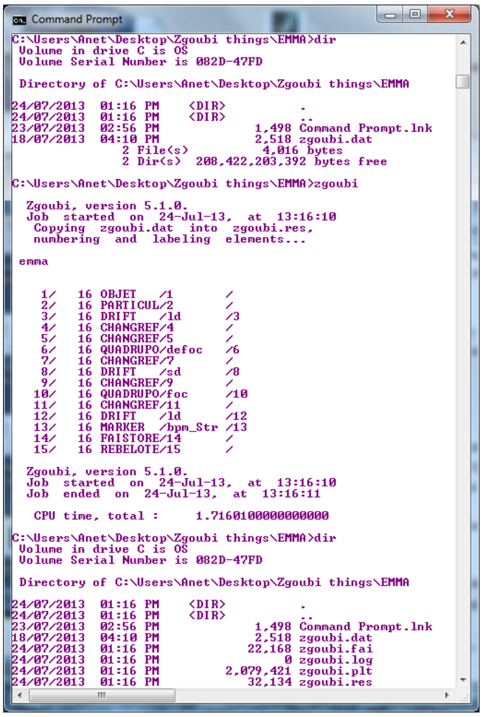
A run can be performed by typing the command **zgoubi** in the directory containing EMMA's zgoubi.dat file (within a command prompt in Windows); this can be seen in figure 20, the run produces four output files.

3.2.1 Zpop output

If used in UNIX the output can be visualised with Zpop which is part of the Zgoubi package. In the following example the same zgoubi.dat file is used as input as seen in figure 16.

In order to produce data plots first Zgoubi should be run as in the previous section to produce .fai and/or .plt files, as can be seen in figure 21a. An xterm window can then be opened with the command 'xterm'. In the xterm window the command **zpop** (see figure 21b) then opens the menu 21c and a blank tektronix window. Firstly option 1, 'Run Zgoubi' should be selected in the xterm window to produce a zpop.log file. After this has been done the other options can be selected to examine different data. A list of the available variables to plot can be seen in figure 22. There is also an analysis package as part of Zpop accessible through option 8 of the main menu. This offers options as can be seen in figure 23 below.

As an example of the possible options, it is possible to plot the track of the particle through the lattice quadrupole by loading the zgoubi.plt file (see figure 24). Plotting the lab X and Y coordinates then produces a plot showing one cell in the lab coordinates. The quadrupoles can be seen as trapezoidal rather than rectangular shapes as the axis have different scales. The reference track which passes through the centre of the quadrupoles can



```
C:\Users\Anet\Desktop\Zgoubi things\EMMA>dir
 Volume in drive C is OS
 Volume Serial Number is 082D-47FD

 Directory of C:\Users\Anet\Desktop\Zgoubi things\EMMA

24/07/2013  01:16 PM    <DIR>          .
24/07/2013  01:16 PM    <DIR>          ..
23/07/2013  02:56 PM             1,498 Command Prompt.lnk
18/07/2013  04:10 PM             2,518 zgoubi.dat
               2 File(s)          4,016 bytes
               2 Dir(s)  208,422,203,392 bytes free

C:\Users\Anet\Desktop\Zgoubi things\EMMA>zgoubi

  Zgoubi, version 5.1.0.
  Job  started  on  24-Jul-13,  at  13:16:10
   Copying  zgoubi.dat  into  zgoubi.res,
   numbering  and  labeling  elements...

 emma


    1/   16 OBJET   /1       /
    2/   16 PARTICUL/2       /
    3/   16 DRIFT   /ld      /3
    4/   16 CHANGREF/4       /
    5/   16 CHANGREF/5       /
    6/   16 QUADRUPO/defoc   /6
    7/   16 CHANGREF/7       /
    8/   16 DRIFT   /sd      /8
    9/   16 CHANGREF/9       /
   10/   16 QUADRUPO/foc     /10
   11/   16 CHANGREF/11      /
   12/   16 DRIFT   /ld      /12
   13/   16 MARKER  /bpm_Str /13
   14/   16 FAISTORE/14      /
   15/   16 REBELOTE/15      /

  Zgoubi, version 5.1.0.
  Job  started  on  24-Jul-13,  at  13:16:10
  Job  ended  on  24-Jul-13,  at  13:16:11

   CPU time, total :     1.7160100000000000

C:\Users\Anet\Desktop\Zgoubi things\EMMA>dir
 Volume in drive C is OS
 Volume Serial Number is 082D-47FD

 Directory of C:\Users\Anet\Desktop\Zgoubi things\EMMA

24/07/2013  01:16 PM    <DIR>          .
24/07/2013  01:16 PM    <DIR>          ..
23/07/2013  02:56 PM             1,498 Command Prompt.lnk
18/07/2013  04:10 PM             2,518 zgoubi.dat
24/07/2013  01:16 PM            22,168 zgoubi.fai
24/07/2013  01:16 PM                 0 zgoubi.log
24/07/2013  01:16 PM         2,079,421 zgoubi.plt
24/07/2013  01:16 PM            32,134 zgoubi.res
```

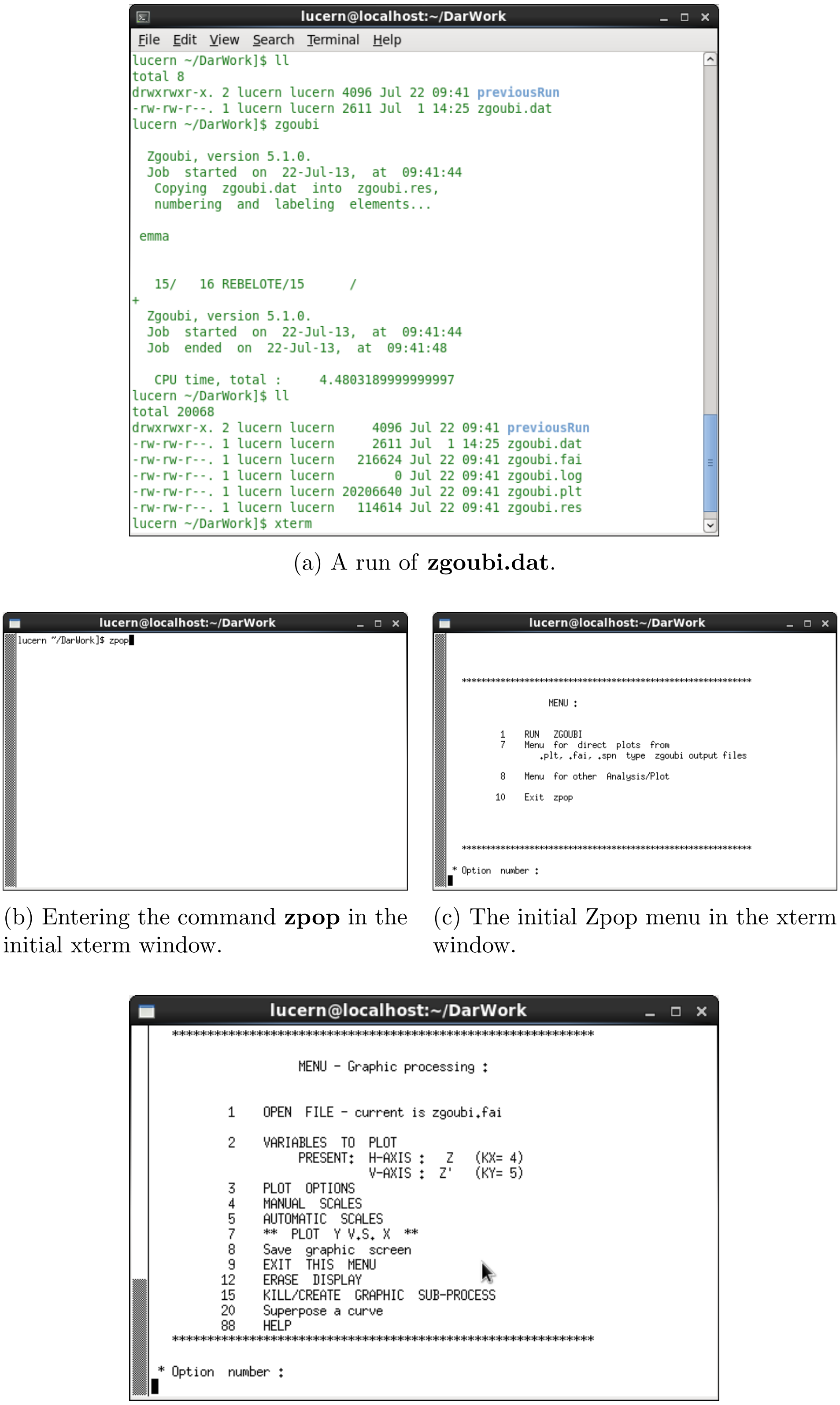
Figure 20: A run of Zgoubi in a Windows command prompt showing the directory contents before and after the run.

be seen changing direction in line with the 'CHANGREF' commands from zgoubi.dat. The particle tracking data can be seen as two arcs of data through the quadrupoles. The trapezoids representing the quadrupoles do not define the physical extent of the magnetic field, so it can be seen that although the particle track happens outside of the defocusing quadrupole trapezoid the track is still curved by its magnetic field.

The zgoubi.fai file is useful for analysing the phase space. From the Zpop main menu choosing the menu options to open the file 'zgoubi.fai' allows plotting of a choice of variables. If the variables **Z** and **Z'** are chosen (see figure 21d) and then plotted the following phase space chart can be produced (see figure 25b).

3.2.2 Analysis in Windows

As the running of Zpop requires an xterm window it it not ideally suited to a Windows system. In this situation a MATLAB script can be used to extract the data from the zgoubi.fai or zgoubi.plt file (see Appendix 4.1). The output from these files can be seen in figure 26.



lucern@localhost:~/DarWork
File Edit View Search Terminal Help

```
lucern ~/DarWork]$ ll
total 8
drwxrwxr-x. 2 lucern lucern 4096 Jul 22 09:41 previousRun
-rw-rw-r--. 1 lucern lucern 2611 Jul  1 14:25 zgoubi.dat
lucern ~/DarWork]$ zgoubi

  Zgoubi, version 5.1.0.
  Job  started  on  22-Jul-13,  at  09:41:44
   Copying  zgoubi.dat  into  zgoubi.res,
   numbering  and  labeling  elements...

 emma


   15/   16 REBELOTE/15      /
+
  Zgoubi, version 5.1.0.
  Job  started  on  22-Jul-13,  at  09:41:44
  Job  ended  on  22-Jul-13,  at  09:41:48

   CPU time, total :     4.4803189999999997
lucern ~/DarWork]$ ll
total 20068
drwxrwxr-x. 2 lucern lucern     4096 Jul 22 09:41 previousRun
-rw-rw-r--. 1 lucern lucern     2611 Jul  1 14:25 zgoubi.dat
-rw-rw-r--. 1 lucern lucern   216624 Jul 22 09:41 zgoubi.fai
-rw-rw-r--. 1 lucern lucern        0 Jul 22 09:41 zgoubi.log
-rw-rw-r--. 1 lucern lucern 20206640 Jul 22 09:41 zgoubi.plt
-rw-rw-r--. 1 lucern lucern   114614 Jul 22 09:41 zgoubi.res
lucern ~/DarWork]$ xterm
```

(a) A run of **zgoubi.dat**.

```
lucern ~/DarWork]$ zpop
```

(b) Entering the command **zpop** in the initial xterm window.

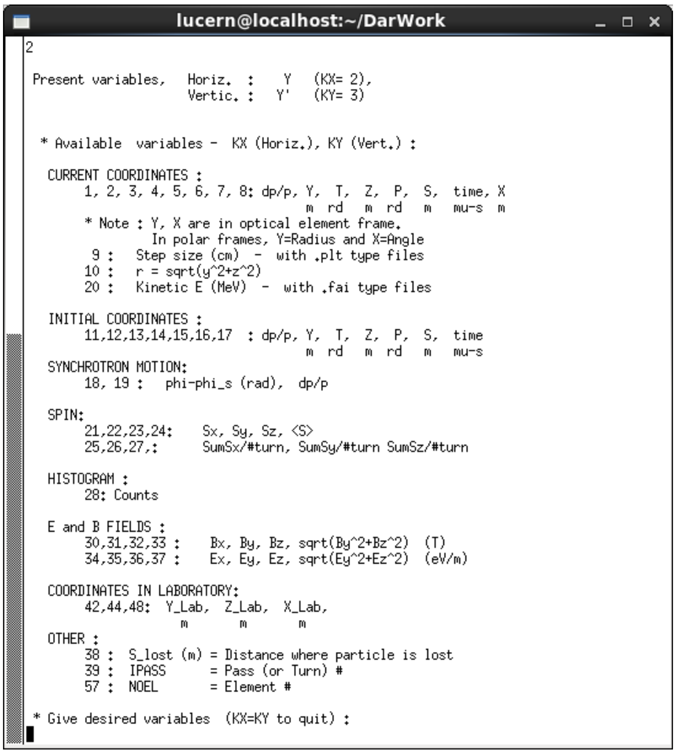
```
 ************************************************************

               MENU :


        1    RUN   ZGOUBI
        7    Menu  for  direct  plots  from
                .plt, .fai, .spn  type  zgoubi output files

        8    Menu  for other  Analysis/Plot

       10    Exit  zpop



 ************************************************************
* Option  number :
```

(c) The initial Zpop menu in the xterm window.

```
  ************************************************************

                    MENU - Graphic processing :


           1    OPEN  FILE - current is zgoubi.fai

           2    VARIABLES  TO  PLOT
                     PRESENT:  H-AXIS :   Z   (KX= 4)
                               V-AXIS :  Z'   (KY= 5)
           3    PLOT  OPTIONS
           4    MANUAL  SCALES
           5    AUTOMATIC  SCALES
           7    **  PLOT  Y V.S. X  **
           8    Save  graphic  screen
           9    EXIT  THIS  MENU
          12    ERASE  DISPLAY
          15    KILL/CREATE  GRAPHIC  SUB-PROCESS
          20    Superpose a curve
          88    HELP
  ************************************************************

* Option  number :
```

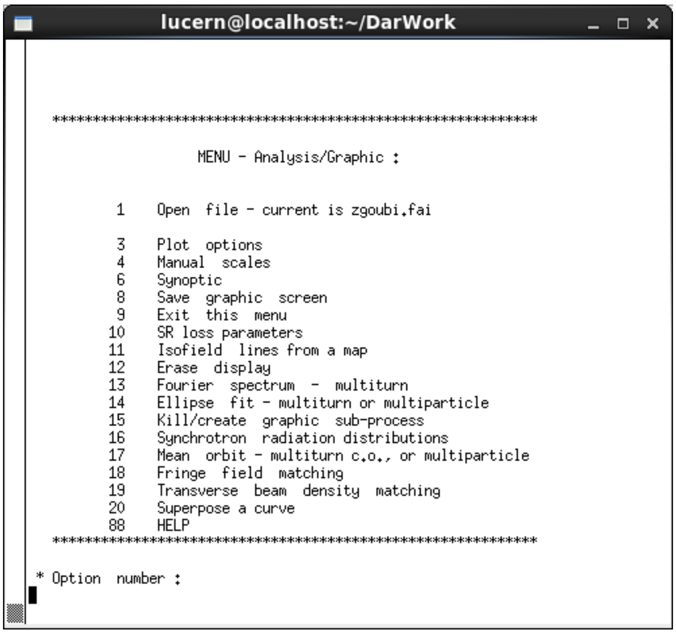
(d) The menu ready to plot **Z** vs **Z'** in the xterm window.

Figure 21: Output from Zgoubi on a Scientific Linux OS.



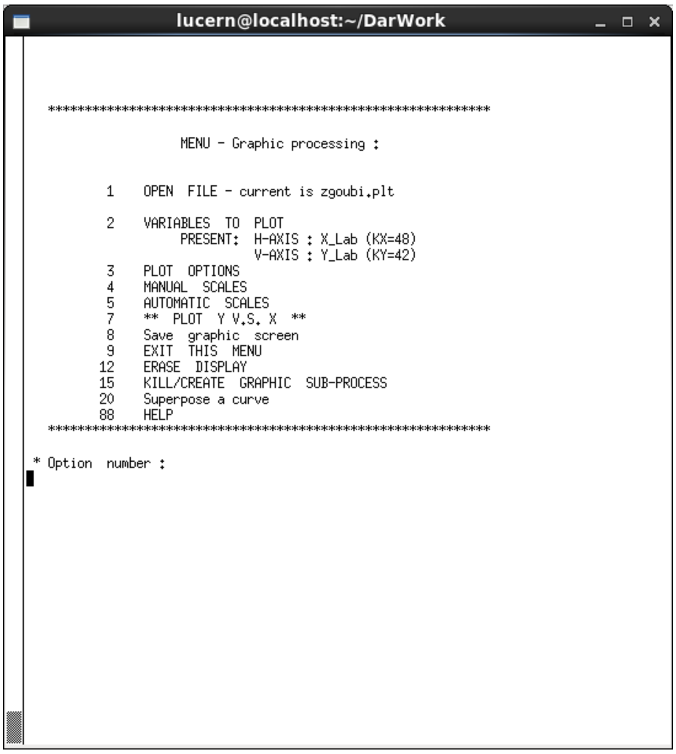
```
2

Present variables,   Horiz. :    Y   (KX= 2),
                     Vertic. :   Y'   (KY= 3)


 * Available  variables -  KX (Horiz.), KY (Vert.) :

  CURRENT COORDINATES :
       1, 2, 3, 4, 5, 6, 7, 8: dp/p, Y,  T,  Z,  P,  S,  time, X
                                     m  rd   m  rd   m   mu-s  m
       * Note : Y, X are in optical element frame.
                In polar frames, Y=Radius and X=Angle
        9 :   Step size (cm)  -  with .plt type files
       10 :   r = sqrt(y^2+z^2)
       20 :   Kinetic E (MeV)  -  with .fai type files

  INITIAL COORDINATES :
       11,12,13,14,15,16,17  : dp/p, Y,  T,  Z,  P,  S,  time
                                     m  rd   m  rd   m   mu-s
  SYNCHROTRON MOTION:
       18, 19 :   phi-phi_s (rad),  dp/p

  SPIN:
       21,22,23,24:    Sx, Sy, Sz, <S>
       25,26,27,:      SumSx/#turn, SumSy/#turn SumSz/#turn

  HISTOGRAM :
       28: Counts

  E and B FIELDS :
       30,31,32,33 :    Bx, By, Bz, sqrt(By^2+Bz^2)  (T)
       34,35,36,37 :    Ex, Ey, Ez, sqrt(Ey^2+Ez^2)  (eV/m)

  COORDINATES IN LABORATORY:
       42,44,48:  Y_Lab,  Z_Lab,  X_Lab,
                    m       m       m
  OTHER :
       38 :  S_lost (m) = Distance where particle is lost
       39 :  IPASS      = Pass (or Turn) #
       57 :  NOEL       = Element #

* Give desired variables  (KX=KY to quit) :
```

Figure 22: The variables available to plot within Zpop for the zgoubi.fai file.



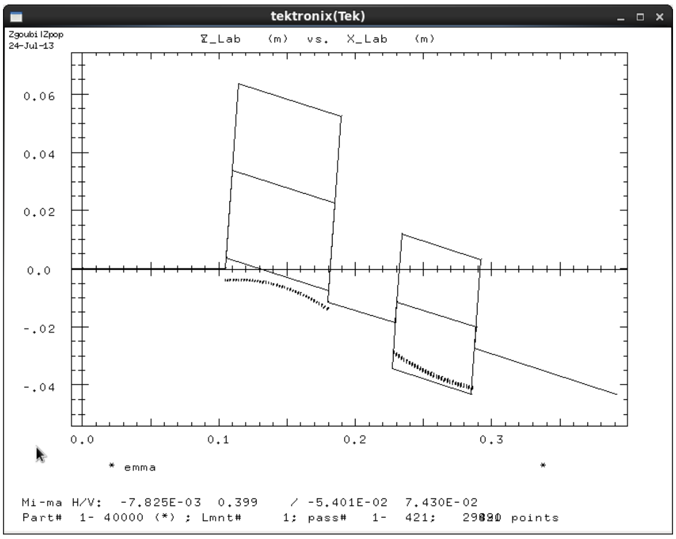
```
   ***************************************************************

                    MENU - Analysis/Graphic :


           1   Open  file - current is zgoubi.fai

           3   Plot  options
           4   Manual  scales
           6   Synoptic
           8   Save  graphic  screen
           9   Exit  this  menu
          10   SR loss parameters
          11   Isofield  lines from a map
          12   Erase  display
          13   Fourier  spectrum  -  multiturn
          14   Ellipse  fit - multiturn or multiparticle
          15   Kill/create  graphic  sub-process
          16   Synchrotron  radiation distributions
          17   Mean  orbit - multiturn c.o., or multiparticle
          18   Fringe  field  matching
          19   Transverse  beam  density  matching
          20   Superpose a curve
          88   HELP
   ***************************************************************

 * Option  number :
```

Figure 23: The analysis menu in zpop.



```
 *************************************************************

                  MENU - Graphic processing :


         1    OPEN  FILE - current is zgoubi.plt

         2    VARIABLES  TO  PLOT
                   PRESENT:  H-AXIS : X_Lab (KX=48)
                             V-AXIS : Y_Lab (KY=42)
         3    PLOT  OPTIONS
         4    MANUAL  SCALES
         5    AUTOMATIC  SCALES
         7    **  PLOT  Y V.S. X  **
         8    Save  graphic  screen
         9    EXIT  THIS  MENU
        12    ERASE  DISPLAY
        15    KILL/CREATE  GRAPHIC  SUB-PROCESS
        20    Superpose a curve
        88    HELP
 *************************************************************

* Option  number :
```

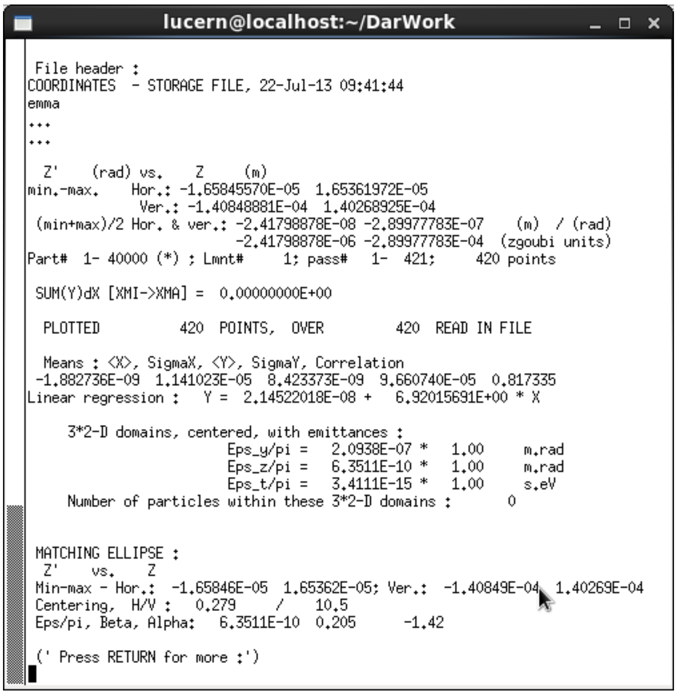
(a) The menu showing the selected options for the plot below using the zgoubi.plt file.



```
Mi-ma H/V:  -7.825E-03  0.399    / -5.401E-02  7.430E-02
Part#  1- 40000 (*) ; Lmnt#     1; pass#   1-   421;   29820 points
```

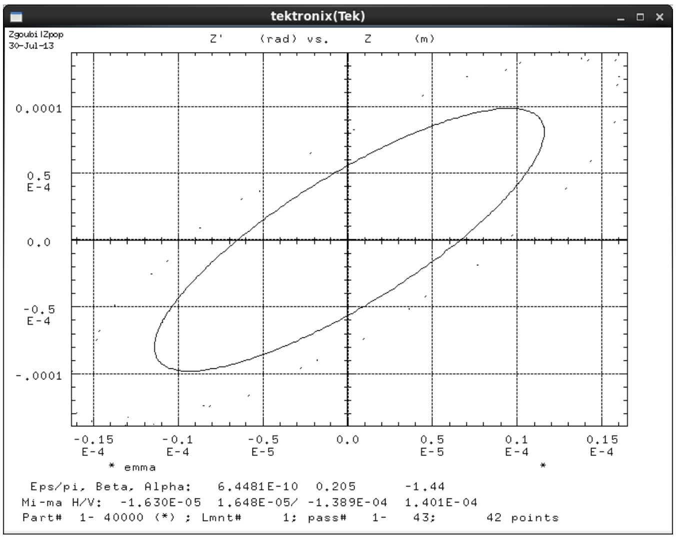
(b) A plot to show the particle track (in lab coordinates) through the magnetic fields of the quadrupoles of one cell. The .plt file only contains tracking data though magnetic fields.

Figure 24: An example of using zgoubi.plt in zpop.



```
 File header :
COORDINATES  - STORAGE FILE, 22-Jul-13 09:41:44
emma
...
...

  Z'    (rad) vs.    Z     (m)
min.-max.    Hor.: -1.65845570E-05  1.65361972E-05
              Ver.: -1.40848881E-04  1.40268925E-04
 (min+max)/2 Hor. & ver.: -2.41798878E-08 -2.89977783E-07    (m)  / (rad)
                          -2.41798878E-06 -2.89977783E-04  (zgoubi units)
Part#  1- 40000 (*) ; Lmnt#     1; pass#   1-  421;     420 points

 SUM(Y)dX [XMI->XMA] =  0.00000000E+00

  PLOTTED         420  POINTS,  OVER        420  READ IN FILE

  Means : <X>, SigmaX, <Y>, SigmaY, Correlation
 -1.882736E-09  1.141023E-05  8.423373E-09  9.660740E-05  0.817335
Linear regression :   Y =  2.14522018E-08 +   6.92015691E+00 * X

     3*2-D domains, centered, with emittances :
                         Eps_y/pi =   2.0938E-07 *   1.00     m.rad
                         Eps_z/pi =   6.3511E-10 *   1.00     m.rad
                         Eps_t/pi =   3.4111E-15 *   1.00     s.eV
     Number of particles within these 3*2-D domains :       0


 MATCHING ELLIPSE :
  Z'    vs.    Z
 Min-max - Hor.:  -1.65846E-05  1.65362E-05; Ver.:  -1.40849E-04  1.40269E-04
 Centering,  H/V :   0.279     /    10.5
 Eps/pi, Beta, Alpha:   6.3511E-10  0.205      -1.42

 (' Press RETURN for more :')
```

(a) The xterm output from plotting **Z** vs **Z'**.



```
 Eps/pi, Beta, Alpha:   6.4481E-10  0.205      -1.44
Mi-ma H/V:  -1.630E-05  1.648E-05/ -1.389E-04  1.401E-04
Part#  1- 40000 (*) ; Lmnt#     1; pass#   1-   43;      42 points
```

(b) The graphical output of **Z** vs **Z'** in the tektronix window with a 'matched ellipse'.

Figure 25: Output from Zpop on a Scientific Linux OS.

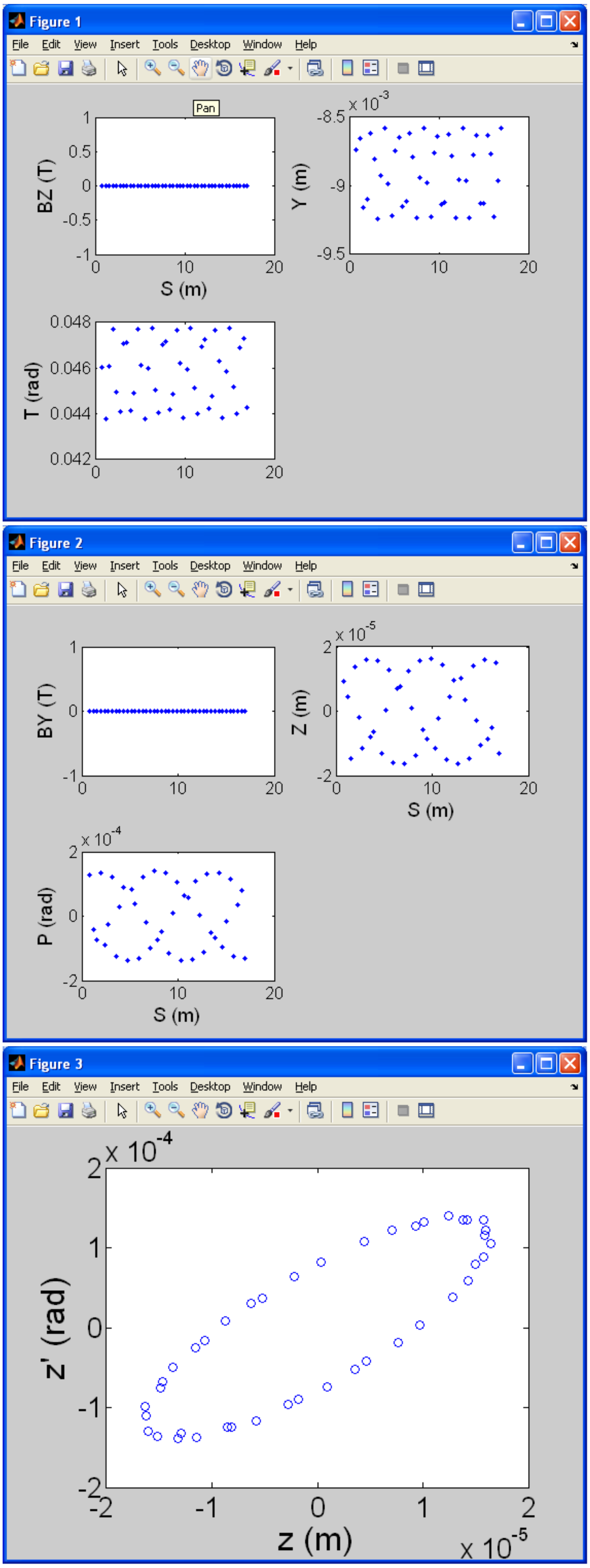
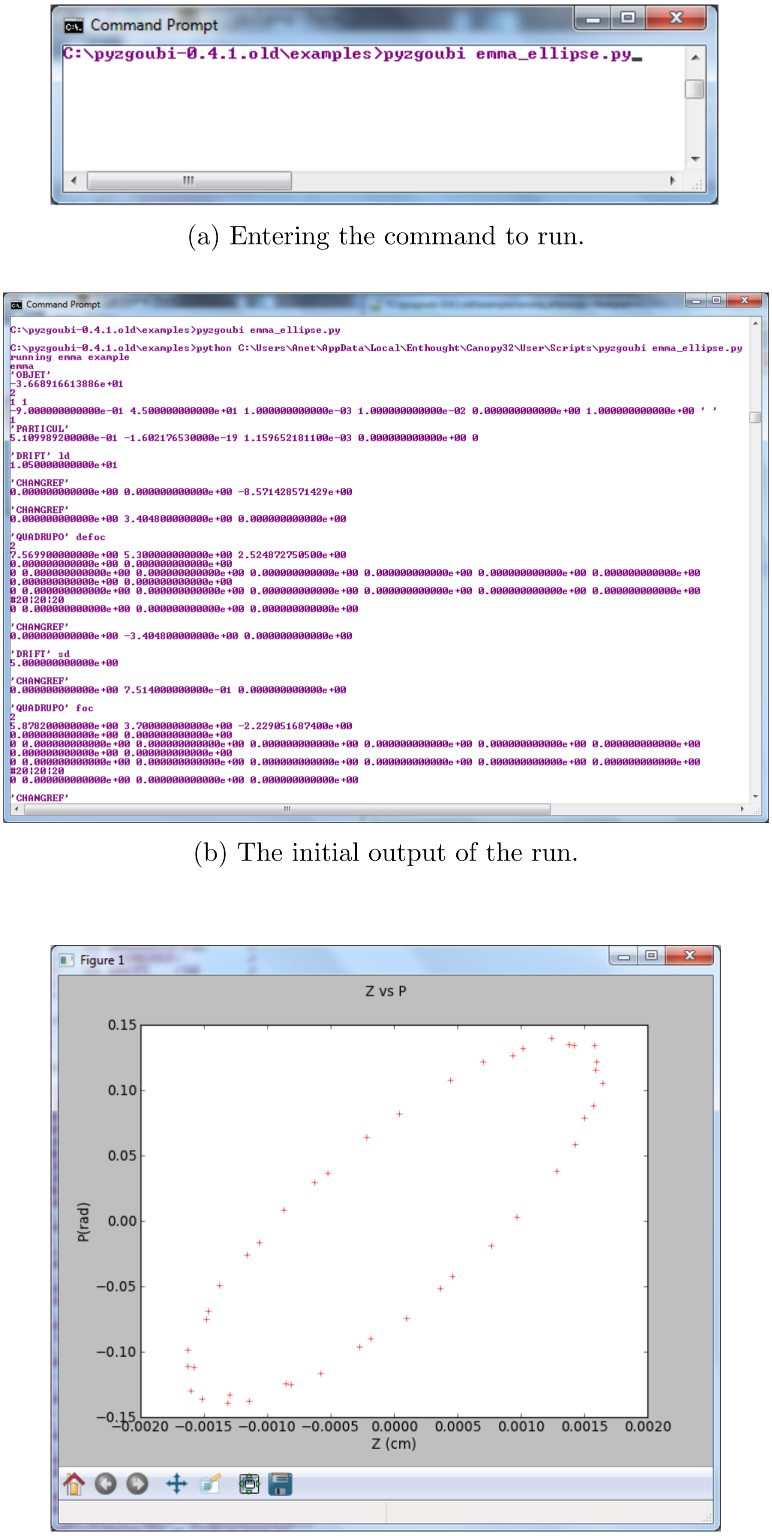


Figure 26: Output from MATLAB scripts analysing zgoubi.fai (see Appendix 4.1) in Windows.

3.2.3 EMMA with PyZgoubi

The identical lattice can be defined using the PyZgoubi script found in appendix 4.2. A version of this script and other EMMA examples are included in the PyZgoubi downloaded package. As with Zgoubi, the program is run from a command line (cmd.exe in Windows) open in the directory containing the particular script. The 'emma_ellipse.py' script is run by entering the command **pyzgoubi emma_ellipse.py** (see figure 27a). The output of the script begins with a print to screen of the zgoubi.res file produced (figure 27b), followed by other instructions from the code - in this case a list of the Z,P values and a graphical output (see figure 27c).

An advantage of PyZgoubi is the ability to use python analysis tools on the resultant data. Included in the PyZgoubi downloaded package are sample EMMA scripts to calculate the closed orbits, tune, Twiss profiles and magnetic apertures for the lattice. Futher information about the use of PyZgoubi can be found at `http://www.hep.manchester.ac.uk/u/sam/pyzgoubi/doc/0.3/doc.html`.



```
C:\pyzgoubi-0.4.1.old\examples>pyzgoubi emma_ellipse.py_
```

(a) Entering the command to run.

```
C:\pyzgoubi-0.4.1.old\examples>pyzgoubi emma_ellipse.py

C:\pyzgoubi-0.4.1.old\examples>python C:\Users\Anet\AppData\Local\Enthought\Canopy32\User\Scripts\pyzgoubi emma_ellipse.py
running emma example
emma
'OBJET'
-3.668916613886e+01
2
1 1
-9.000000000000e-01 4.500000000000e+01 1.000000000000e-03 1.000000000000e-02 0.000000000000e+00 1.000000000000e+00 ' '
1
'PARTICUL'
5.109989200000e-01 -1.602176530000e-19 1.159652181100e-03 0.000000000000e+00 0

'DRIFT' ld
1.050000000000e+01

'CHANGREF'
0.000000000000e+00 0.000000000000e+00 -8.571428571429e+00

'CHANGREF'
0.000000000000e+00 3.404800000000e+00 0.000000000000e+00

'QUADRUPO' defoc
2
7.569900000000e+00 5.300000000000e+00 2.524872750500e+00
0.000000000000e+00 0.000000000000e+00
0 0.000000000000e+00 0.000000000000e+00 0.000000000000e+00 0.000000000000e+00 0.000000000000e+00 0.000000000000e+00
0.000000000000e+00 0.000000000000e+00
0 0.000000000000e+00 0.000000000000e+00 0.000000000000e+00 0.000000000000e+00 0.000000000000e+00 0.000000000000e+00
#20|20|20
0 0.000000000000e+00 0.000000000000e+00 0.000000000000e+00

'CHANGREF'
0.000000000000e+00 -3.404800000000e+00 0.000000000000e+00

'DRIFT' sd
5.000000000000e+00

'CHANGREF'
0.000000000000e+00 7.514000000000e-01 0.000000000000e+00

'QUADRUPO' foc
2
5.878200000000e+00 3.700000000000e+00 -2.229051687400e+00
0.000000000000e+00 0.000000000000e+00
0 0.000000000000e+00 0.000000000000e+00 0.000000000000e+00 0.000000000000e+00 0.000000000000e+00 0.000000000000e+00
0.000000000000e+00 0.000000000000e+00
0 0.000000000000e+00 0.000000000000e+00 0.000000000000e+00 0.000000000000e+00 0.000000000000e+00 0.000000000000e+00
#20|20|20
0 0.000000000000e+00 0.000000000000e+00 0.000000000000e+00

'CHANGREF'
```

(b) The initial output of the run.

(c) The graphical output of the script.

Figure 27: Running **emma_ellipse.py** in Windows.

4 Appendix

4.1 MATLAB scripts

The following codes were written by Dr. Kai Hock to extract and plot data from the Zgoubi output files using MATLAB.

4.1.1 get_zgoubi_data.m

This code extracts data from zgoubi.plt.

```
function [Y, T, Z, P, X, S, BX, BY, BZ] = get_zgoubi_data(
    filename)
% filename = 'zgoubi.plt';
%         s1 = 'e';     % particle label

fid = fopen (filename, 'r');

fgets(fid);
fgets(fid);
fgets(fid);
fgets(fid);

s = fscanf(fid, '\%s', 1);
fgets(fid);

a = fscanf(fid, '\%f', 3);
fgets(fid);
c = fscanf(fid, '\%f', 3);
fgets(fid);
b = fscanf(fid, '\%f', 8);

n = 1;

while feof(fid) == 0,
Y(n) = a(2);
T(n) = a(3);
Z(n) = c(1);
P(n) = c(2);
S(n) = c(3);
X(n) = b(5);
BX(n) = b(6);
BY(n) = b(7);
BZ(n) = b(8);
n = n + 1;

fgets(fid);
fgets(fid);
s = fscanf(fid, '\%s', 1);
fgets(fid);

a = fscanf(fid, '\%f', 3);
fgets(fid);
```

```
c = fscanf(fid, '\%f', 3);
fgets(fid);
b = fscanf(fid, '\%f', 8);

end

fclose (fid);
```

4.1.2 plt_zgoubi_plt_2b.m

In this script the previous code is called to extract the data which is then plotted.

```
clear

filename = 'zgoubi.plt';
[Y, T, Z, P, X, S, BX, BY, BZ] = get_zgoubi_data(filename);

close all
figure

subplot(221), plot(S/100, BZ/10, '.')
set(gca,'fontsize',12)
xlabel('S␣(m)', 'fontsize', 14)
ylabel('BZ␣(T)', 'fontsize', 14)

subplot(222), plot(S/100, Y/100, '.')
set(gca,'fontsize',12)
%xlabel('S (m)', 'fontsize', 14)
ylabel('Y␣(m)', 'fontsize', 14)

subplot(223), plot(S/100, T/1000, '.')
set(gca,'fontsize',12)
%xlabel('S (m)', 'fontsize', 14)
ylabel('T␣(rad)', 'fontsize', 14)

figure

subplot(221), plot(S/100, BY/10, '.')
set(gca,'fontsize',12)
%xlabel('S (m)', 'fontsize', 14)
ylabel('BY␣(T)', 'fontsize', 14)

subplot(222), plot(S/100, Z/100, '.')
set(gca,'fontsize',12)
xlabel('S␣(m)', 'fontsize', 14)
ylabel('Z␣(m)', 'fontsize', 14)

subplot(223), plot(S/100, P/1000, '.')
set(gca,'fontsize',12)
xlabel('S␣(m)', 'fontsize', 14)
ylabel('P␣(rad)', 'fontsize', 14)
```

```
figure

plot(Z/100, P/1000, ’o’)
set(gca, ’fontsize’, 20)
xlabel(’Z␣(m)’, ’fontsize’, 24)
ylable(’Z’’␣(rad)’, ’fontsize’, 24)
```

4.2 PyZgoubi EMMA script

This script, emma_ellipse.py, will write the zgoubi.dat file for the EMMA lattice, print the ’zgoubi.res’ output to screen and plot a chart of Z vs Z’ (or Z vs P).

```
print "running emma example"
import pylab as plt
emma = Line(’emma’)

xpas = (20,20,20)

cells = 42
angle = -360/cells
d_offset = 34.048 * mm
f_offset = 7.514 * mm

#lengths
ld = 210 * mm
sd = 50 * mm

fq = 58.782 * mm
dq = 75.699 * mm

# quad radius
fr = 37 * mm
dr = 53 * mm

fb = -6.02446402 * fr * T
db = 4.76391085 * dr * T

ob = OBJET2()
emma.add(ob)

emma.add(ELECTRON())

emma.add(DRIFT(’ld’, XL=ld*cm_/2))
emma.add(CHANGREF(ALE=angle))

emma.add(CHANGREF(YCE=d_offset*cm_))
emma.add(QUADRUPO(’defoc’, XL=dq*cm_, R_0=dr*cm_, B_0=db*
    kgauss_, XPAS=xpas, IL=2))
emma.add(CHANGREF(YCE=-d_offset*cm_))

emma.add(DRIFT(’sd’, XL=sd*cm_))
emma.add(CHANGREF(YCE=f_offset*cm_))
emma.add(QUADRUPO(’foc’, XL=fq*cm_, R_0=fr*cm_, B_0=fb*kgauss_,
     XPAS=xpas, IL=2))
emma.add(CHANGREF(YCE=-f_offset*cm_))

emma.add(DRIFT(’ld’, XL=ld*cm_/2))

emma.add(FAISCNL(FNAME=’zgoubi.fai’))

emma.add(REBELOTE(K=99, NPASS=42))

emma.add(END())

rigidity = ke_to_rigidity(10.5e6, 0.51099892e6)
ob.set(BORO=-rigidity)
ob.add(Y=-0.9, T=45, Z=0.001, P=0.01, D=1)

print emma.output()
res = emma.run()#xterm = True)

fai_data =  res.get_all(’fai’)
print "fai"
print "PASS Z P"
z=[]
q=[]
for p in fai_data:
        z.append(p[’Z’])
        q.append(p[’P’])
        print p[’PASS’], p[’Z’], p[’P’]

fig = plt.figure()
plt.plot(z,q,’r+’)
fig.suptitle(’Z␣vs␣P’)
plt.xlabel(’Z␣(m)’)
plt.ylabel(’P(rad)’)
plt.show()

print res.res()
res.clean()
```

References

[1] ”Zgoubi”, F. Méot, http://sourceforge.net/projects/zgoubi/

[2] ”Zgoubi User’s Guide”, F. Méot, 2012, https://www.bnl.gov/isd/documents/80278.pdf

[3] ”Zgoubi”, C. Lapoire, R. Basset, F. Zimmermann and F. Ruggiero, https://oraweb.cern.ch/pls/hhh/code_website.disp_code?code_name=Zgoubi

[4] http://www.scienceaccelerator.gov/dsa/result-list/fullRecord:zgoubi/

[5] Mathworks, 2013, http://www.mathworks.co.uk/

[6] ”PyZgoubi - a python interface to Zgoubi”, S.Tygier, http://www.hep.manchester.ac.uk/u/sam/pyzgoubi/

[7] MinGW, 2013, http://www.mingw.org/

[8] ”PyZgoubi Documentation”, S. Tygier, http://www.hep.manchester.ac.uk/u/sam/pyzgoubi/doc/0.3/doc.html

[9] ”Zgoubi Tutorial - A course for (potential) users”, F. Méot, 2009, ftp://ftp3.ie.freebsd.org/pub/sourceforge/z/zg/zgoubi/exemples/tutorial/lecture.pdf

[10] ”EMMA, The World’s First Non-Scaling FFAG Accelerator”, S.L. Smith, 2009, http://accelconf.web.cern.ch/AccelConf/PAC2009/papers/we4pbi01.pdf

[11] ”Fixed Field Alternating Gradient Accelerators (FFAG)”, S.Machida, 2012, http://cas.web.cern.ch/cas/Granada-2012/Lectures/GranadaLectures/Machida.pdf