

**EFIT — AN INTERACTIVE APPROACH TO HIGH  
TEMPERATURE FUSION PLASMA MAGNETIC  
EQUILIBRIUM FITTING**

by  
**Q. PENG, J. SCHACHTER, D.P. SCHISSEL,  
and L.L. LAO**

**JUNE 1999**

## DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, 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 any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

# **EFIT — AN INTERACTIVE APPROACH TO HIGH TEMPERATURE FUSION PLASMA MAGNETIC EQUILIBRIUM FITTING**

by  
**Q. PENG, J. SCHACHTER, D.P. SCHISSEL,  
and L.L. LAO**

This is a preprint of a paper to be presented at the 11th IEEE/NPSS Real Time Conference, June 14–18, 1999, Santa Fe, New Mexico and to be published in *Transactions on Nuclear Science*.

**Work supported by  
the U.S. Department of Energy  
under Contract No. DE-AC03-99ER54463**

**GA PROJECT 30033  
JUNE 1999**

# IEFIT — An Interactive Approach to High Temperature Fusion Plasma Magnetic Equilibrium Fitting

Q. Peng, J. Schachter, D.P. Schissel, and L.L. Lao  
General Atomics, San Diego, California 92121

## Abstract

An interactive IDL based wrapper, IEFIT, has been created for the magnetic equilibrium reconstruction code EFIT written in FORTRAN. It allows high temperature fusion physicists to rapidly optimize a plasma equilibrium reconstruction by eliminating the unnecessarily repeated initialization in the conventional approach along with the immediate display of the fitting results of each input variation. It uses a new IDL based graphics package, GaPlotObj, developed in cooperation with Fanning Software Consulting, that provides a unified interface with great flexibility in presenting and analyzing scientific data. The overall interactivity reduces the process to minutes from the usual hours.

## I. INTRODUCTION

Fusion is the power source of the sun and other stars. It occurs when forms of the lightest atom, hydrogen, combine to make helium in a very hot (100 million degrees centigrade) ionized gas or “plasma.” In this process a small amount of the matter involved in the reaction is converted to a large amount of energy. On earth, fusion could provide a safe, environmentally attractive and affordable long-term energy source. One technique scientists have been using to contain this hot plasma on earth is called magnetic confinement. This concept, called a tokamak, employs magnetic fields in a doughnut shaped configuration to confine the plasma. Experiments on tokamaks have involved plasma pulses where the hot plasma exists for tens of seconds and the pulse is repeated with a tens of minutes cycle time. Fundamental to this type of research is the detailed knowledge of the magnetic field topology since the tokamak magnetic field is generated in part from currents flowing in the hot plasma. This analysis is important in between pulses to guide the research and to be performed in more detail at the conclusion of experimental operation for comprehensive analysis of the results.

The Magnetohydrodynamics (MHD) equilibrium fitting code EFIT [1] was first developed in 1985 to perform magnetic topology analysis for the Doublet III tokamak. It was later adapted for DIII-D (a National Fusion Facility) and many other tokamaks around the world. It is written in FORTRAN and translates measurements from plasma diagnostics such as external magnetic probes, external poloidal flux loops and the Motional Stark Effect, into useful information like plasma geometry, stored energy and current profiles. An example of reconstructed magnetic flux surfaces is shown in Fig. 1.

Back in 1980’s, a typical single-precision EFIT calculation on a VAX-11/780, with a relative error of  $\psi <$

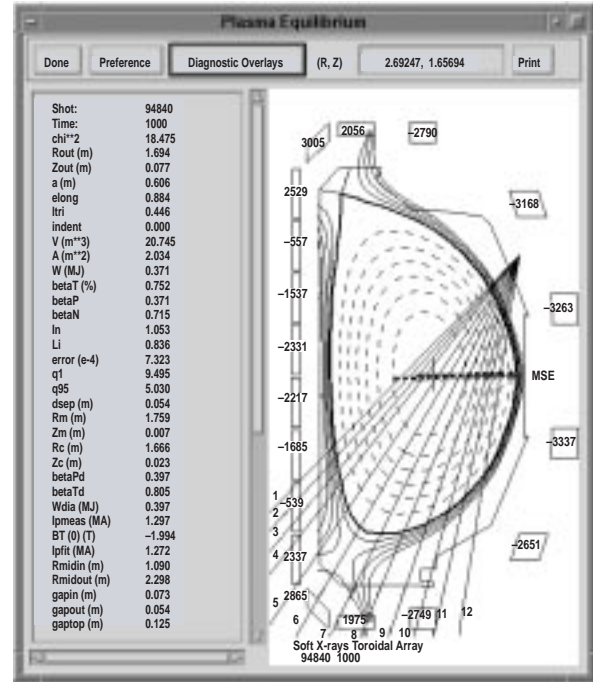


Fig. 1. Reconstructed magnetic flux surfaces with diagnostic overlay of a DIII-D pulse shown in the EFIT graphic viewer.

$10^{-3}$ , would require 3-5 minutes of CPU time using Green’s function method [1] on a 26x30 grid closely enclosing the plasma. On a modern HP 9000/C180 workstation, a similar reconstruction takes only a few seconds of CPU time. The dramatic increase in computational power has allowed routine time-dependent EFIT analysis to be performed between plasma pulses. Additionally, this enhanced power has made fine-tuning an analysis possible via multiple EFIT runs on selected times in a pulse.

EFIT has been running in batch mode since its creation. For routine analysis, default inputs are provided. For specialized analysis, inputs have to be prepared manually. Post-processing is usually required for viewing the results. The approach is sufficient for routine processing of large amounts of experimental data. When fine-tuning a fit, however, where many cycles of pre-processing, execution of the code and post-processing may be involved, the conventional approach becomes time-consuming and labor-intensive. A typical example is during a Kinetic EFIT [2] where a good initial fit is essential to the final result of the analysis. The Kinetic EFIT uses measured plasma pressure and modeled fast ion pressure in conjunction with external magnetic data. This results in a more accurate equilibrium

\*Work supported by U.S. Department of Energy under Contract No. DE-AC03-99ER54463.

reconstruction of the plasma current and pressure profiles, which is crucial to transport and MHD stability analysis. The problem of the conventional approach is two-fold. One is the unnecessarily repeated initialization of the EFIT code. The other is the lack of good interactive graphics to provide efficient feedback. As a result, a large portion of the time is not spent directly in analysis. A more efficient approach is therefore highly desirable for high quality EFIT analysis. This paper will present an interactive approach to EFIT that addresses and solves both problems.

## II. EFIT AT BACK-END

Similar to most batch mode analysis codes, EFIT structurally contains three parts, initialization including data retrieval, computation and output generation. The EFIT code eliminates the repetition of data retrieval by providing an intermediate stage that caches retrieved experimental data along with fitting parameters in a file. However, every time EFIT is executed, the initializations of the Green's function tables, fitting weights, current density distribution and other variables are inevitable.

Based on the fact that the results of the initialization are essentially the same for one set of data using one grid size, the interactive EFIT takes the caching one step further. The back-end EFIT that is dispatched by the front-end starts as usual as an independent process. It carries out initialization, equilibrium and fitting iterations and writes outputs. Instead of exiting the process as in the batch mode, the EFIT sleeps in the background until it receives a re-fit or exit request. Upon receipt of a re-fit request, the process wakes up and reads in the new parameters, and starts iterations immediately using cached Green's function tables and other variables. The user is given an option on whether to re-initialize or use the current profile from the last solution for the subsequent fittings. After generating the outputs, it notifies the front-end of the

completion of the calculation and resumes its waiting status. The front-end will then display the results in an interactive graphic viewer (see Fig. 1). The original design was to allow the user to view the results on the graphic viewer as the iterations progress. But experience has shown that since the calculation runs fast enough, there is hardly any advantage for such complication. The iterations, however, may be interrupted by the user to correct input mistakes.

Typical cases have been tested and timed. The final parameters after a serial of interactive EFIT runs were saved for a batch mode EFIT execution for comparison. The equilibria were overlaid. No detectable difference was present. The execution time varies from case to case and also depends on grid size, tolerance and precision. But in general, the timing has shown that, in a batch mode execution using cached data, the initialization takes up about two thirds of the wall-clock time. In other words, every interactive EFIT run after the initial one only takes one third of the time, as is illustrated in Fig. 2.

## III. FRONT-END GRAPHICAL USER INTERFACE

The front end is written using the commercial product Interactive Data Language (IDL) [3]. It serves as an interface for both taking inputs and presenting results. Figure 3 shows the input window of the IEFIT. For presenting results as shown in Fig. 4, the interactive EFIT employs a new graphic package, GaPlotObj, developed in cooperation with Fanning Software Consulting. The GaPlotObj library is based on IDL Direct Graphics. While providing capabilities similar to IDL Object Graphics in converting plots from static end product to manipulatable objects, it does not sacrifice the performance of 1-D, 2-D and surface plots. This is important for a non OpenGL complaint X-terminal dominated environment, in which applications cannot take advantage of the Object Graphics while achieving a reasonable performance. The

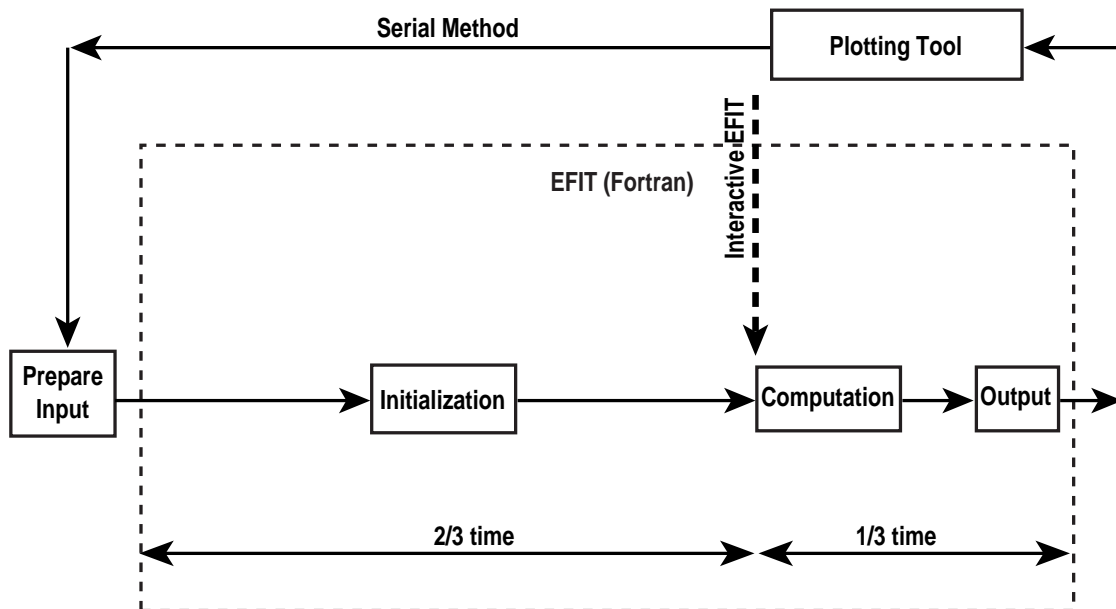


Fig. 2. Schematic of interactive EFIT.

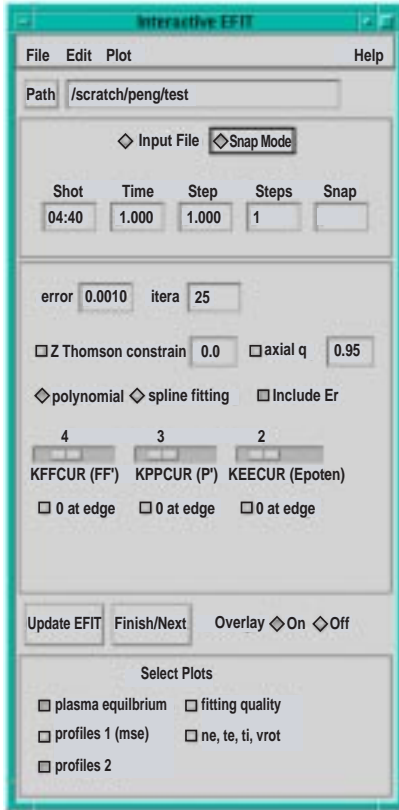


Fig. 3. Main EFIT input window dispatches and communicates with EFIT.

package provides a unified interface with great flexibility in presenting and analyzing scientific data. Taking advantage of object-orientation, the functionality is self-contained, readily expandable and may be used in various applications.

An EFIT graphic viewer using GaPlotObj library was developed first as an independent tool and has been widely used for post-processing EFIT results. It allows a user to select an EFIT from files or from MDSplus [4] database, and provides the graphics of equilibrium, fitting qualities, profiles and time history. The comparisons between pulses and diagnostic overlays, and switching among  $R$ ,  $\rho$  and  $\psi$  coordinates are as simple as a mouse-click. The general interactive capabilities such as data selection, zooming within a plot or switching between single and multiple plots, data picking and cursor tracing, layout and other configurations are automatically available from the GaPlotObj library. While the viewer has been in existence and will continue to exist as an EFIT post-processing tool, it has also become an integral part of the interactive EFIT.

Recall that upon the completion of the calculation, the back-end EFIT immediately notifies the front-end. The front-end quickly accesses, processes and displays the EFIT results onto the viewer, with or without a comparison to those of the early settings. The data flow from EFIT to its viewer is transparent to the user. After carefully examining the results, a user may choose to adjust the parameters on the fly, either from the input windows or on the plot windows (see Figs. 3 & 4). The editing capability of the GaPlotObj library is activated for this purpose while it is usually disabled in display-only applications. A re-fit request will be sent to the EFIT in the background. This manipulation may be repeated to user's satisfaction.

#### IV. COMMUNICATION

The FORTRAN back-end and the IDL front-end are synchronized by UNIX signals and IDL timer events. The data

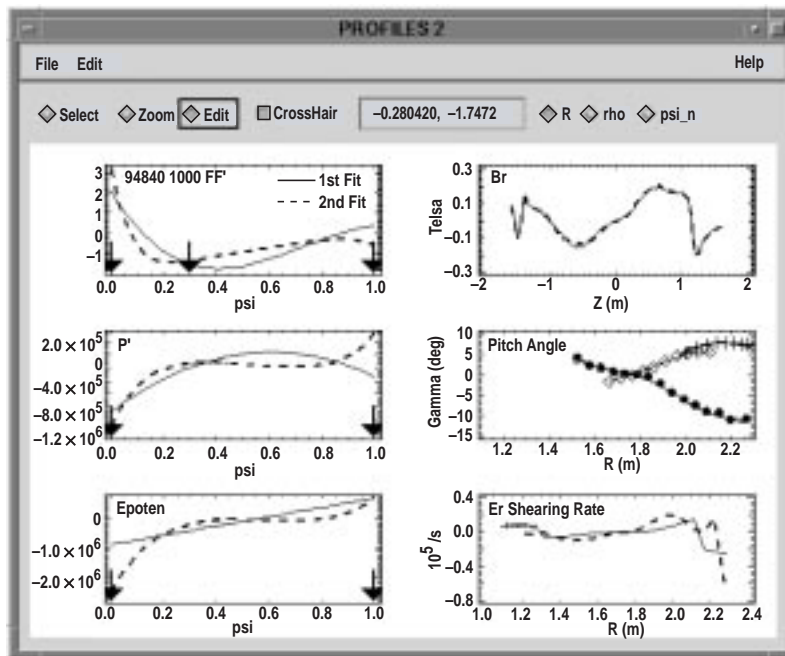


Fig. 4. An overlay of two consecutive EFIT runs shown in one of the output windows. The arrows on the axes indicate knot locations of spline fitting. They may be edited directly in this window.

transfer is carried out by I/O but without intervention from user. This simplified approach keeps the change to the EFIT code down to the minimum. Without the front-end, EFIT can be executed in its conventional batch mode. The IDL front-end remains active and responsive during EFIT computation. This is achieved by IDL timer events, that check signals periodically during the idle moments. The possible delay is insignificant.

## V. DISCUSSION

The elimination of repeated EFIT initializations, the more functional interactive graphical display for presenting results, and a more direct and efficient feedback system have greatly improved the efficiency of complex EFIT analysis. The results of the analysis are used to rapidly control the re-generation of the data. The processing time has been reduced from hours to minutes.

With further growth in computational power in workstation environment, more complex analyses will benefit from such improvement. A closely related analysis is Kinetic-EFIT, which is required to accurately map the inner flux surface. The interactive EFIT should be able to take one step further to include the profile fitting, and computation of the fast ion pressure.

Since the current implementation depends on UNIX signals, this limits the interactive EFIT to the UNIX platform and confines two relatively independent parts of the application on one host. While EFIT presently runs on UNIX and VMS systems, the front-end runs on all platforms that IDL supports. With a fast network, a client-server implementation may better take advantage of the increasing power on the personal computers, and a distributed computing environment.

## REFERENCES

- [1] L.L. Lao, H. St. John, R.D. Stambaugh, A..G., Kellman, W. Pfeiffer, "Reconstruction of Current Profile Parameters and Plasma Shapes in Tokamaks," Nucl. Fusion, Vol. **25** p. 1611, 1985.
- [2] L.L. Lao, J.R. Ferron, R.J. Geobner, W. Howl, H.E. St. John, E.J. Strait, T.S. Taylor, "Equilibrium Analysis of Current Profiles in Tokamaks," Nucl. Fusion, Vol. **30**, p. 1035, 1990.
- [3] Interactive Data Language (IDL), Research Systems, Inc. (<http://www.rsinc.com>).
- [4] J.A. Stillerman, T.W. Fredian, K.A. Klare, G. Manduchi, "MDSplus Data Acquisition System," Rev. Sci. Instr. **68** 1 (1997).