A Magnetic Equilibrium Reconstruction in Tokamak

J. Havlicek[1,2] and J. Urban[2,3]

[1] Charles University in Prague, Faculty of Mathematics and Physics, Prague, Czech Republic.

[2] Institute of Plasma Physics AS CR, v.v.i., Association EURATOM/IPP.CR, Za Slovankou 1782/3, 182 00, Prague 8, Czech Republic.

[3] Czech Technical University in Prague, FNSPE, Prague, Czech Republic.

Abstract. Equilibrium reconstruction determines the plasma shape, the current profile and pressure profile parameters in tokamaks. It is based on solving Grad-Shafranov equation describing MHD equilibrium in axially symmetric devices. This paper reports on the principles of the EFIT code and on the parallelization of the EFIT2006 code.

Introduction

Equilibrium reconstruction is used to obtain information about the plasma shape, the current and the pressure profile parameters in tokamaks [*Lao et al.*, 1985]. The reconstruction is performed by iterative solving the Grad-Shafranov equation [*Shafranov*, 1971; *Wesson et al.,* 1997]. Various tokamak experiments have their own unique equilibrium reconstruction codes. Some of these codes are particular versions of EFIT [*Lao et al.*, 1985]. Tokamak geometry, currents in the poloidal field coils and magnetic measurements are fundamental input data to the EFIT code. Other diagnostics, such as Thomson scattering, Motional Stark Effect (MSE) or Faraday rotation, might be used as well. EFIT2006 is a machine-independent version of the EFIT code. It adopts recommendations of the ITM (Integrated Tokamak Modelling) EFDA task force, such as object-oriented design and XML input files [*Appel et al*., 2006].

Grad-Shafranov equation

Plasma can be described by many different approaches. One of the models describing plasma is the MHD one-fluid model. This model describes plasma macroscopically as one fluid consisting of electrons and ions. It uses set of so called MHD equations.

In this section of the article we will derive the Grad-Shafranov equation, which describes equilibrium state of plasma and magnetic configuration in axially symmetric devices such as tokamak, from Maxwell equations and MHD equation of motion.

Maxwell equations in the MHD approximation are:

$$\nabla \cdot \vec{B} = 0 \tag{1}$$

$$\nabla \times \vec{B} = \mu \vec{j} \tag{2}$$

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}, \tag{3}$$

where $\vec{B}$ is magnetic induction, μ is magnetic permeability, $\vec{j}$ is current density and $\vec{E}$ is electric field. It should be noted that in MHD approximation Maxwell equation (2) does not have term $\mu \frac{\partial \vec{D}}{\partial t}$.

MHD equation of motion is:

$$\rho \frac{\mathrm{d}\vec{v}}{\mathrm{d}t} = \vec{j} \times \vec{B} - \nabla p + \rho \vec{g}, \tag{4}$$

where ρ is plasma density, $\vec{v}$ is velocity, p is plasma pressure and $\vec{g}$ is gravitational acceleration. Plasma density, velocity and pressure are defined as:

$$\rho = n_i m_i + n_e m_e \approx n(m_i + m_e) \tag{5}$$

$$\vec{v} = \frac{1}{\rho}(n_i m_i \vec{v_i} + n_e m_e \vec{v_e}) \tag{6}$$

$$p = p_i + p_e\,, \tag{7}$$

where n_i, m_i, $\vec{v_i}$ and p_i is ion concentration, mass, velocity and pressure. n_e, m_e, $\vec{v_e}$ and p_e is electron concentration, mass, velocity and pressure.

It is possible to assume that the equilibrium configuration in tokamak is axially symmetric (this assumption neglects toroidal ripple of magnetic field). It means that in the cylindrical coordinate system $(\mathrm{R},\varphi,\mathrm{Z})$ the magnetic induction $\vec{B}$ is independent on the toroidal angle φ. Therefore

$$\vec{B} = \vec{B}(\mathrm{R,Z}) \neq \vec{B}(\varphi)\,. \tag{8}$$

Hence, it is possible to define a poloidal flux function $\Psi(\mathrm{R,Z})$:

$$\Psi(\mathrm{R,Z}) = \frac{1}{2\pi}\int_D \vec{B}\cdot \mathrm{d}\vec{S}\,, \tag{9}$$

where D denotes the area of the disc at position Z with radius R perpendicular to the Z-axis. $\Psi(\mathrm{R,Z})$ is up to the factor 2π the flux of the poloidal magnetic field through area D.

Poloidal components of the magnetic induction vector $\vec{B}$ are then:

$$B_R = -\frac{1}{R}\frac{\partial\Psi}{\partial Z} \tag{10}$$

$$B_Z = \frac{1}{R}\frac{\partial\Psi}{\partial R}\,. \tag{11}$$

The poloidal flux function $\Psi(\mathrm{R,Z})$, given by equation (9), satisfies the Maxwell equation (1) for cylindrical coordinate system:

$$\nabla\cdot\vec{B} = \frac{1}{R}\frac{\partial}{\partial R}(RB_R) + \frac{1}{R}\frac{\partial B_T}{\partial\varphi} + \frac{\partial B_Z}{\partial Z} = 0\,, \tag{12}$$

where B_T is the toroidal component of the magnetic induction vector $\vec{B}$.

Let us now define a function F as:

$$\vec{B}_T = \frac{F}{R}\vec{e}_T\,, \tag{13}$$

where $\vec{e_T}$ is a unit vector in the direction of toroidal angle φ.

The magnetic induction $\vec{B}$ can then be written as:

$$\begin{gathered}\vec{B} = \vec{B}_p + \vec{B}_T\\ \vec{B}_p = \frac{1}{R}[\nabla\Psi\times\vec{e}_T]\\ \vec{B}_T = \frac{F}{R}\vec{e}_T\,,\end{gathered} \tag{14}$$

where $\vec{B}_p$ denotes the poloidal vector component of magnetic induction $\vec{B}$.

From the Maxwell equation (2) and the magnetic induction $\vec{B}$ defined by equation (14) it is possible to obtain expression for current density $\vec{j}$:

$$\begin{gathered}\vec{j} = \vec{j}_p + \vec{j}_T\\ \vec{j}_p = \frac{1}{R}[\nabla(F/\mu)\times\vec{e}_T]\\ \vec{j}_T = (L\Psi)\vec{e}_T\,,\end{gathered} \tag{15}$$

where $\vec{j}_p$ and $\vec{j}_T$ are the poloidal and the toroidal components of the current density $\vec{j}$ and the operator L is defined as:

$$L = -\frac{\partial}{\partial R}\left(\frac{1}{\mu R}\frac{\partial}{\partial R}\right) - \frac{\partial}{\partial Z}\left(\frac{1}{\mu R}\frac{\partial}{\partial Z}\right). \tag{16}$$

The expressions (14) and (15) for $\vec{B}$ and $\vec{j}$ are valid in the whole space of the tokamak (plasma, vacuum, vessel, coils) because they involve only the Maxwell equations and the assumption of axial symmetry.

For tokamaks it is possible to neglect $\rho\vec{g}$ in equation (4), and for an equilibrium state $\rho\frac{\mathrm{d}\vec{v}}{\mathrm{d}t} = 0$. Equation (4) then becomes so called equilibrium equation which reflects that the force due to ∇p is compensated by the $\vec{j} \times \vec{B}$ force:

$$\nabla p = \vec{j} \times \vec{B}. \tag{17}$$

From (17) it is possible to see that:

$$\vec{B} \cdot \nabla p = \vec{B} \cdot \vec{j} \times \vec{B} = 0 \tag{18}$$

$$\vec{j} \cdot \nabla p = 0, \tag{19}$$

which means that equilibrium magnetic field lines and current lines lie on isobaric surfaces (p=constant). These surfaces are called magnetic surfaces. These magnetic surfaces are defined by:

$$\Psi(R,Z) = \text{constant}. \tag{20}$$

Equations (14) and (18) indicate that ∇p is collinear with $\nabla\Psi$, therefore:

$$p = p(\Psi) \tag{21}$$

and equations (15) and (19) indicate that ∇p is collinear with $\nabla\Psi$, therefore:

$$F = F(\Psi). \tag{22}$$

The equilibrium equation (17) combined with equations (14) and (15) for $\vec{B}$ and $\vec{j}$ becomes:

$$\nabla p = \frac{L\Psi}{R}\nabla\Psi - \frac{F}{\mu R^2}\nabla F, \tag{23}$$

which can be written as:

$$L\Psi = R\frac{\partial p}{\partial \Psi} + \frac{1}{2\mu R}\frac{\partial (F^2)}{\partial \Psi}. \tag{24}$$

The equation (24) is the Grad-Shafranov equation and L is the elliptic operator (16). Right-hand side of the Grad-Shafranov equation represents toroidal component of the plasma current density $\vec{j}_{T,pl}$.

EFIT2006

EFIT algorithm

The EFIT code iteratively solves the Grad-Shafranov equation using many different constraints [*e.g. Lao et al.*, 1990; *Appel et al.*, 2001]. When we write the Grad-Shafranov equation with all terms representing toroidal current density, we get:

$$L\Psi e_T = \vec{j}_{T,C} + \vec{j}_{T,V} + \vec{j}_{T,pl}, \tag{25}$$

where $\vec{j}_{T,C}$ is current density in the poloidal field coils, $\vec{j}_{T,V}$ is toroidal component of current density in the vacuum vessel and other passive conductive structures and $\vec{j}_{T,pl}$ is toroidal component of the plasma current density defined by right-hand side of equation (24). $\vec{j}_{T,C}$ might be expressed as:

$$\vec{j}_{T,C} = \sum_{i=1}^{N_c} \frac{I_i^{pf}}{|c_i|} \vec{e}_T \ , \tag{26}$$

where I_i^{pf} is the current in i-th poloidal field coil and $|c_i|$ is geometric factor transforming the current in the i-th coil to current density.

In the first iterative step EFIT gets new iteration of poloidal flux function Ψ by solving equation (25) with selected representation of the term $\vec{j}_{T,pl} = R\frac{\partial p}{\partial \Psi} + \frac{1}{2\mu R}\frac{\partial (F^2)}{\partial \Psi}$. The most common selection is representation in the form of polynomial functions:

$$\frac{\partial p}{\partial \Psi} = \sum_{i=1}^{N_\alpha} \alpha_i \Psi^i \tag{27}$$

$$\frac{\partial (F^2)}{\partial \Psi} = \sum_{i=1}^{N_\gamma} \gamma_i \Psi^i \ . \tag{28}$$

This first iterative step is performed by Green's function method:

$$\Psi^{t+1}(\vec{r})\vec{e}_T = \sum_{i=1}^{N_C} G_i^{pf}(\vec{r},\vec{r}_i) I_i^{pf} \vec{e}_T + \sum_{i=1}^{N_V} G_i^{V}(\vec{r},\vec{r}_i) I_i^{V} \vec{e}_T + \int G(\vec{r},\vec{r}^{\,'}) \vec{j}_{T,pl}(\vec{r}^{\,'},\Psi^t) \mathrm{d}R' \mathrm{d}Z' \ , \tag{29}$$

where Ψ^{t+1} is new iteration of poloidal flux function, G_i^{pf} is Green's function for determination of poloidal flux function at position $\vec{r}$ from the current I_i^{pf} in the i-th poloidal field coil at the position $\vec{r}_i$, G_i^V and G are Green's functions for i-th vacuum vessel element current and plasma current density respectively.

In the second iterative step EFIT gets new iteration of parameters α_i, γ_i, I_i^{pf} and I_i^V. In this step the physical measurements are involved. Any measured value a_M^k (k=1,…,number of measurements) which is dependent on any of the parameters α_i, γ_i, I_i^{pf} or I_i^V (therefore dependent on the Ψ, $\vec{j}_T$, $\vec{B}$ or any derivative) might be used as a constraint for EFIT. In order to do so, this dependence $a^k(\alpha_i, \gamma_i, I_i^{pf}, I_i^V)$ is linearized in the parameters α_i, γ_i, I_i^{pf} and I_i^V:

$$a^k\Big|_{a_0} + \sum_{i=1}^{N_\alpha} \frac{\partial a^k}{\partial \alpha_i}\Bigg|_{a_0} \alpha_i + \sum_{i=1}^{N_\gamma} \frac{\partial a^k}{\partial \gamma_i}\Bigg|_{a_0} \gamma_i + \sum_{i=1}^{N_C} \frac{\partial a^k}{\partial I_i^{pf}}\Bigg|_{a_0} I_i^{pf} + \sum_{i=1}^{N_V} \frac{\partial a^k}{\partial I_i^{V}}\Bigg|_{a_0} I_i^{V} = a_P^k \tag{30}$$

and a_P^k is the predicted value of a_M^k.

These predicted values are compared with measured values:

$$a_P^k = a_M^k \ . \tag{31}$$

Equations (31) constitute a system of linear algebraic equations for α_i, γ_i, I_i^{pf} and I_i^V. This set of equations is typically overdetermined, i.e. the number of equations is greater than the number of unknowns. Singular Value Decomposition method is used to find the least-square solution, while each equation is weighted by a factor σ_k, corresponding generally to the measurement accuracy:

$$\left| \sum_k \left(\frac{a_M^k - a_P^k}{\sigma_k} \right)^2 \right|_{\min} \ . \tag{32}$$

This second iterative step supplies new iteration of parameters α_i, γ_i, I_i^{pf} and I_i^V into the first iterative step.

EFIT2006 parallelization

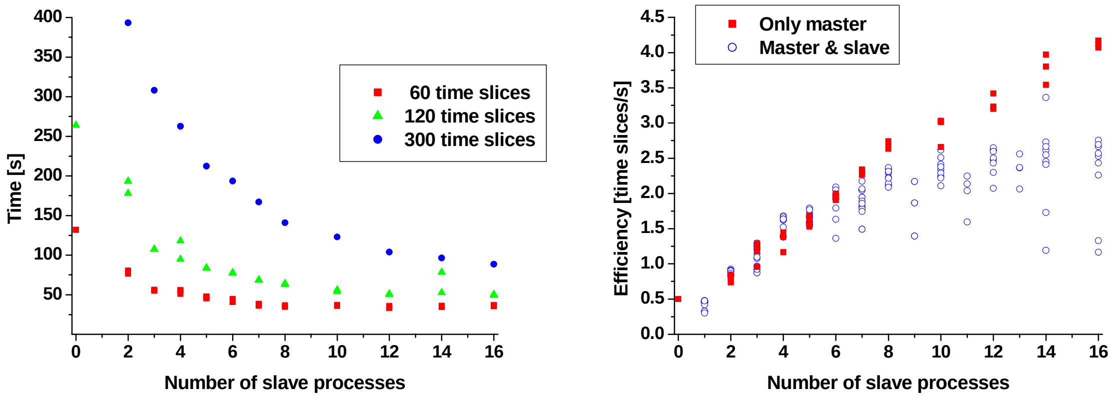
This section of the article describes our contribution to the further development of the EFIT2006 code. Small tokamaks (such like CASTOR or COMPASS-D) might have usual time duration between two consequent shots 5-10 minutes. In order to quickly analyze the last shot (and decide what changes should be taken to achieve desired plasma configuration in the next shot) results from the EFIT code should be available as quickly as possible. The EFIT2006 code is written in the C++ with computational core of the code in the Fortran 95 and is capable of solving equilibrium reconstruction of one time slice in approximately 2 seconds (PC with CPU Intel P4 2GHz Dual Core, 100Mbit Ethernet connection, grid 65×65, measured data from ≈ 60 magnetic sensors). To obtain time evolution of the plasma configuration several dozens or hundreds of time slices should be computed in time substantially smaller than 5-10 minutes. In order to achieve such performance we decided to parallelize the EFIT2006 code.

Computation of each consequent time slice does not require data input from the last time slice. Therefore the problem of parallelization of the EFIT2006 code is so called “embarrassingly parallelizable” problem, i.e. it is possible to divide computation into many subtasks (one subtask for one time slice), which might be solved independently on each other and without mutual communication.

The parallelization was carried out using C++ Boost Serialization Library [*Ramey*, 2002] and Boost MPI library [*Gregor et al*., 2005]. Boost Serialization Library is used for transforming various C++ data structures (i.e. classes) to and from a sequence of bytes and the Boost MPI library is used for sending these data between computer processes.

Principle of the parallelized EFIT2006 code is quite simple. There is one master process running on one computer. This master process reads input data and beforehand computed Green’s response functions and distributes response functions to slave processes. Then it distributes input data for individual time slices to individual slave processes (one time slice to one slave process). After that the master process awaits for the first slave process to finish its job, receives the result and if there are any unassigned time slices it sends a new time slice to the slave process for computation. This is repeated until all the time slices are computed. Afterwards, the master process writes results into an output file.

The slave process only awaits an input data, then processes them and sends the result to the master process and again awaits new input data until it is terminated by a specific message tag from the master process.



a) b)

Figure 1: a) Dependence of the total time required to broadcast response functions, compute equilibria for 60, 120 and 300 time slices and write results to the output file. Non-parallel EFIT2006 run for 300 time slices lasts ≈ 660 s. **b)** Efficiency of computing phase. “Only master” means that there was only the master process and no slave process running on the master computer (2 CPUs per one computer, usually two processes per one computer). Master & slave means that the master and one slave process were running on the master computer. (Points with number of slave processes = 0 are for non-parallel EFIT2006 run.)

Performance gain is depicted in Figure 1. It is possible to see that the performance gain for lower number of time slices is quite small (~ 3 × smaller total time than in the non-parallel EFIT2006 run for optimal number of slave processes ~ 8). This is caused by long period of time necessary for broadcasting the response functions to the slave processes. The performance gain for a higher number of time slices is better.

Total time for computing all time slices for grid 65 × 65 consists of:
1) initialization of MPI environment ~ 6 seconds
2) broadcasting response functions ~ [number of slave processes] seconds
3) computing all time slices ~ [3 × number of time slices / number of slave processes] seconds
4) writing output data to the output file ~ [number of time slices / 12] seconds.

Conclusion

We have described the importance of computing equilibrium reconstruction in tokamak. We have presented a derivation of the Grad-Shafranov equation and its use in the EFIT algorithm. We have also described our contribution to the development of the EFIT2006 code - its parallelization – and the performance gains achieved by running the parallelized EFIT2006 on multiple computers. We plan to run EFIT2006 with data from COMPASS-D tokamak provided by UKAEA, Culham in the near future. We also plan to continue our participation in further development of the EFIT2006 code.

Acknowledgments. The authors thank to Lynton Appel for his support with understanding EFIT2006. The work was financially supported by Czech Science Foundation, grants 202/03/H162 and 202/07/0044, by grant AV0Z20430508 and by EURATOM.

References

Appel, L. C., M. K. Bevir and M. J. Walsh, Equilibrium reconstruction in the START tokamak, *Nuclear Fusion*, 41(2), 169-180, 2001.

Appel, L. C., G. T. A. Huysmans, L. L. Lao, P. J. McCarthy, D. G. Muir, E. R. Solano, J. Storrs, D. Taylor and W. Zwingmann, A Unified Approach to Equilibrium Reconstruction, 33rd EPS Conference on Plasma Phys., *Europhysics Conference Abstract*, 30I, P-2.184, 2006.

Gregor, D., M. Troyer, http://www.generic-programming.org/~dgregor/boost.mpi/doc/, 2005.

Lao, L. L., H. St. John, R. D. Stambaugh, A. G. Kellman and W. Pfeiffer, Reconstruction of current profile parameters and plasma shapes in Tokamaks, *Nuclear Fusion*, 25(11), 1611, 1985.

Lao, L. L., J. R. Ferron, R. J. Groebner, W. Howl, H. St. John, E. J. Strait and T. S. Taylor, Equilibrium analysis of current profiles in tokamaks, *Nuclear Fusion*, 30(6), 1035, 1990.

Ramey, R., http://www.boost.org/libs/serialization/doc/, 2002.

Shafranov, V. D., Determination of the parameters β_I and l_I in a Tokamak for arbitrary shape of plasma pinch cross-section, *Plasma Phys*., 13, 757-762,1971.

Wesson, J. *et al*., Tokamaks, *Clarendon Press*, Oxford, 1997.