Development of the Integrated System Design Code for Fusion Power Plants

Makoto NAKAMURA[1], Yuichi OGAWA[1], Naoki SHINJI[1], Yuya MIYOSHI[1], Ryoji HIWATARI[2], Kunihiko OKANO[2] and Youji SOMEYA[3]

[1]*Department of Advanced Energy, Graduate School of Frontier Sciences, the University of Tokyo, Kashiwanoha 5-1-5, Kashiwa, Chiba 277-8561, Japan*

[2]*Nuclear Technology Research Laboratory, Central Research Institute of Electric Power Industry (CRIEPI), Iwadokita 2-11-1, Komae, Tokyo 201-8511, Japan*

[3]*Atomic Energy Research Laboratory, Tokyo City University, Ozenji 971, Asao, Kawasaki 215-0013, Japan*

(Received: 29 October 2009 / Accepted: 22 January 2010)

The integrated system design code for fusion power plant design is now being developed. Such a code will bridge the gap between the huge number of operational parameter scans by system analysis codes and a final engineering design of each reactor component. In this paper we report current status of the development of the integrated system design code. This integrated design code consists of the main frame and detailed component design analysis modules. The former is based on the previous system analysis code, FUSAC, and its role is basic system analysis. The latters are based on existing, authorized calculation codes and intended for system component analysis more detailed than FUSAC analysis. As an example of integrated system design, we show the two-dimensional equilibrium calculation code integration into FUSAC.

Keywords: tokamak, reactor design, system code, integration, modular structure

1. Introduction

For fusion reactor design, a huge number of plasma-physics and fusion-engineering parameters must be evaluated with taking into account relations among them. System analysis codes are computer programs for consistent analysis of such parameters. Parameter scans by means of system analysis codes remove inappropriate parameter sets and bring fusion reactor operation windows with physics parameters compatible with engineering and economical constraints. So far several system analysis codes has been developed.

FUSAC [1, 2] is one of such system analysis codes. FUSAC consists of three parts. The first is plasma physics analysis, i.e. a zero-dimensional plasma analysis based on the ITER physics design guidelines [3]. The second is fusion engineering analysis, i.e. a simple engineering design program to determine the shape of toroidal field (TF) coils, the position and width for blankets, shields, central solenoid coils, bucking cylinder and so on. This part is based on TRESCODE [4, 5]. The last is economic analysis, i.e. estimation of the cost of electricity (COE) based on the Generomak model [6]. For more detailed description of FUSAC, see Ref. [2]. So far FUSAC has been used for several engineering and economical analysis for fusion power plants. Okano *et al.* analyzed the COE of commercial fusion reactors using FUSAC, and found that the normalized beta value is a crucially sensitive parameter to the COE [7]. On basis of FUSAC analysis, the compact reversed shear tokamak reactor (CREST) has been proposed [8]. Recently FUSAC has been used for design of the demonstration tokamak fusion power plant, Demo-CREST, for early realization of net electric power generation [9].

Apart from the development and applications of FUSAC, a number of numerical simulation codes has been developed for diverse areas of plasma physics and fusion engineering research. Integration of these detailed physics/engineering codes into FUSAC will bring refinement of design and performance evaluation of demonstration and commercial fusion power plants. Ultimately an integrated system design code will bridge the gap between the huge number of operational parameter scans by system analysis codes and a final engineering design of each reactor component. For quickness of such integration, a system analysis code should have a 'robust' structure, that is, the integration can be completed without large change of a basic system analysis code (such as FUSAC) and detailed physics/engineering calculation codes.

The purposes of this work are to develop the integrated system design code taking into account recent progress of diverse areas of plasma physics and fusion engineering, and to refine the previous designs and performance evaluations of demonstration and commercial fusion power plants. In this paper we report current status of the development of the integrated system design code.

author's e-mail: mnakamura@ppl.k.u-tokyo.ac.jp

©2010 by The Japan Society of Plasma Science and Nuclear Fusion Research

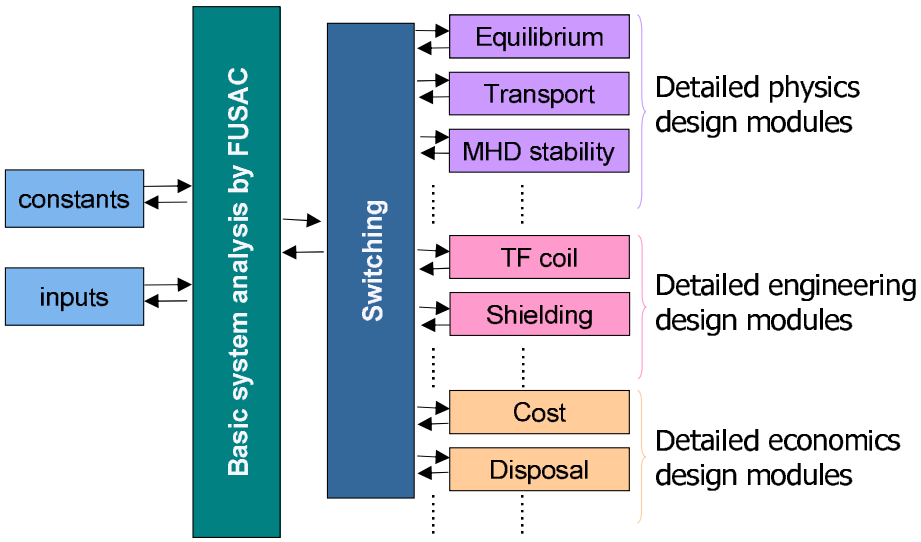

Fig.1 Schematic of the integrated system design code.

The paper is organized as follows. Section 2 describes the modular structure of the integrated system design code. As an example of development of detailed design modules, numerical calculation of maximum toroidal magnetic fields, i.e. the magnetic fields on the inner surfaces of the TF coils, is presented in Section 3. Summary and future prospects of code development and reactor design are presented in Section 4.

2. Structure of the integrated design code

We integrate a number of detailed plasma physics, fusion engineering and economics design modules into FUSAC. The detailed plasma physics design modules includes core plasma equlibrium analysis, core transport analysis, MHD stability analysis, current drive analysis, and so on. The detailed fusion engineering design modules include maximum toroidal field analysis, neutron shielding analysis, and so on. The economics design modules include COE analysis, radioactive disposal analysis, and so on. We are developing the integrated design code so that these modules can be selected and executed according to analysts' design philosophy. Existing, authorized numerical calculation codes are used as these modules. (Somewhat little modifications might be needed for integration into FUSAC.) In this section we describe the modular structure of the integrated code.

We use Fortran 90/95 as a programming language. For efficiency of the integrated design code development, we have established the following programming rules;

1. Physics constants and databases are allocated to a private module, and referred in the main frame of the integrated code.
2. Input physics, engineering and economical parameters are allocated to a private module, and referred in the main frame.
3. Variables interfacing with the main frame and detailed design modules are allocated to a private module and this module is referred in the main frame and these design modules.
4. These variables are interfaced with the main frame and detailed design modules via 'execution subroutines.'

The schematic of the integrated code following the above rules are shown in Fig. 1. The main frame is the basic system analysis performed by FUSAC. The detailed system component design modules are mediated by the 'switching module.' The role of the switching module is an interface for design parameters between FUSAC and the

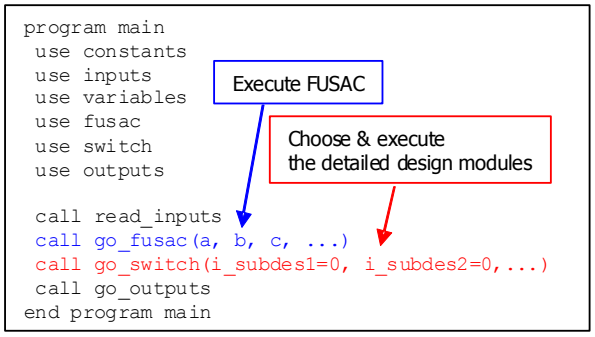

Fig.2 Example of the main frame of the integrated system analysis code.

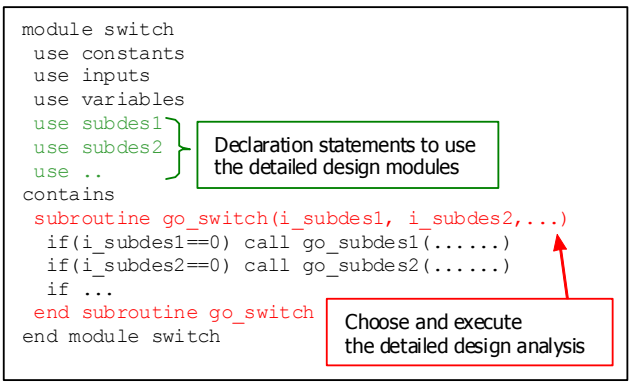

Fig.3 Example of the switching module of the integrated system analysis code.

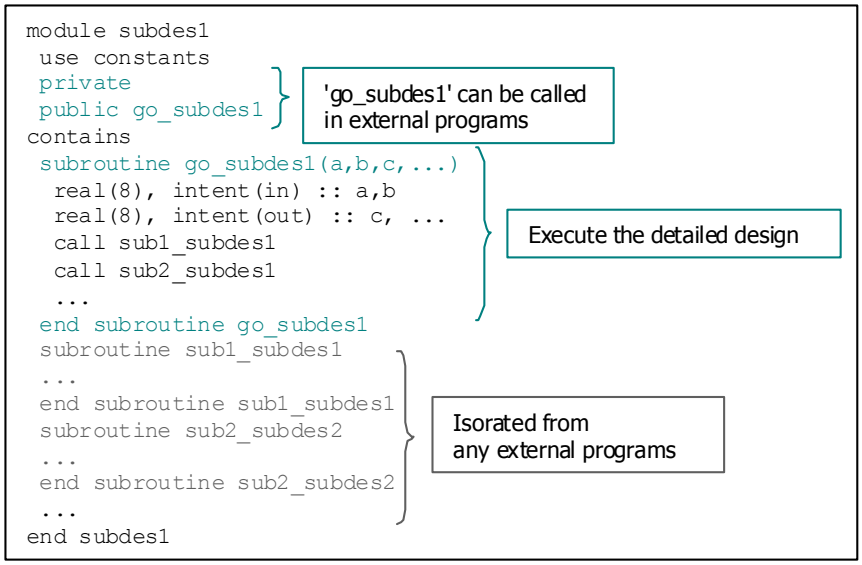

Fig.4 Example of the 'module-packaged' detailed system component design code

detailed design modules. Some detailed design modules are selected and executed via the switcihng modules; the others are not used according to analysts' design philosophy.

Examples of the main frame and switching module are shown in Figs. 2 and 3, respectively. The subroutine 'go_switch' in Fig. 2 select and execute the detailed system component design modules. The detailed design modules, i.e. 'subdes1' and 'subdes2' in Fig. 3, are not referred explicitly in the main frame, but called in the switching module (i.e. the 'module switch' in Fig. 3). The role of the subroutines 'go_subdes1' and 'go_subdes2' in Fig. 3 is to execute the detailed design analysis contained in the modules 'subdes1' and 'subdes2,' respectively.

Existing, authorized numerical calculation codes are used as detailed system component design modules. Modification of such calculation codes for integration should be as little as possible in order to make coupling of these codes with FUSAC efficient and avoid coding errors. We propose modification of these detailed system component design codes so that they have modular structures *with* using an 'private-public system.' An example of the 'module-packaged' detailed design code is shown in Fig. 4. The main program of the original code is changed into a subroutine form like the execution subroutine 'go_subdes1' in Fig. 4. The role of the 'go_subdes1' is to execute the detailed component design. It is noted that the subroutine 'go_subdes1' is accompanied with dummy variables. These variables represent input and output parameters of the original code. The 'sub1_subdes1' and 'sub2_subdes1' are the subroutines contained in the original code. In the module-packaged design code 'subdes1,' it is only the execution subroutine 'go_subdes1' that can be called in the main program. The 'sub1_subdes1' and 'sub2_subdes1' are isolated from any external programs.

The modular structure described above has two important features. One is the 'private-public system.' This system brings isolation of the subroutines of the original code from any other external programs. When integrating detailed design modules, there would be the case where a subroutine in a module has the same name as another subroutine in another module. Such a case will cause coding errors. The 'private-public system' makes it possible to avoid such errors. The other feature is the dummy arguments accompanying with the executing subroutine. When integrating the modules, there would be the case where an input or output parameter has a different name from another input or output parameter in another module. Such a case will also cause coding errors. Explicit notion of the dummy variables make it possible to avoid such errors.

3. Example of detailed design module integration

As described in the previous section, we use existing, authorized numerical codes as detailed component design modules. For example, the TOSCA code [10] could be used as a plasma equilibrium calculation module. The DRIVER88 code [11] could be used as a driven and bootstrap current calculation module. We are now integrating the plasma equilibrium calculation code into FUSAC as an example of integration of a complicated heavy code with system analysis. In this section we report current status of the equilibrium calculation code integration into FUSAC.

The example of the integrated code is shown in Fig. 5. For the integration, we have re-developed FUSAC so that it has a modular structure. The module 'equilibrium' is the 'module-packaged' two-dimensional plasma equilibrium calculation. Here the module 'equilibrium' is not mediated by the 'switching module,' because it is obvious that the module 'equilibrium' is the only detailed design module we use here. The subroutines 'go_fusac' and 'go_equilibrium'

are to execute FUSAC and the equilibrium module, respectively. Here, their dummy arguments are abbreviated.

The inputs of FUSAC include the major radius R, the minor radius a, the elongation κ, the triangularity δ, the normalized beta value β_N the surface safety factor q_{95}, and so on. The FUSAC outputs relevant to the equilibrium module include the position and size of a TF coil. The equilibrium module inputs includes the position and size of a TF coil, R, a. Thus, some FUSAC outpus are inputs of the equilibrium module. The outputs of the equilibrium module includes locations of TF, CS and PF coils, the two-dimensional profile of flux surfaces, and the current density of each coil.

As shown in Fig. 5, we have developed the basic integration frame work. We are now benchmarking the integrated FUSAC by comparing the previous fusion reactor design studies such as [9].

Finally we discuss prospects of integrated reactor design. In the previous reactor design studies, two-dimensional analyses were performed *offline* based on a one-dimensional reactor configuration determined by one- or zero-dimensional system analyses. Generally such an offline design studies include a lot of tedious trials and errors. For example, in [9] the number, positions and current densities of the PF coils were determined *offline* by the equilibrium code analysis, which was based on the one-dimensional radial build determined by FUSAC. The present development of the integrated system design code will demonstrate reactor design in which zero-, one- and two-dimensional analyses are integrated consistently and *online*. The equilibrium-FUSAC integration discussed in this section will enable to *online* evaluate and optimize the number of PF coils and the position and current density of each TF, CS, PF and divertor coils so that the magnetic stored energy W_{mag} is minimized.

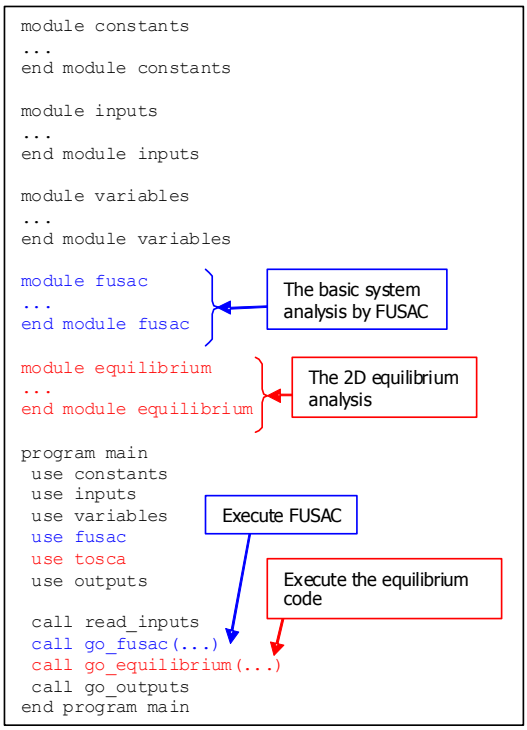

Fig.5 Example of the system design code integrated with the equilibrium calculation module.

4. Summary

The integrated system design code for fusion power plant design is now being developed. This system design code will bridge the gap between the huge number of operational parameter scans by system analysis codes and a final engineering design of each reactor component. In this paper we have reported the unique, modular structure of this code. It consists of the main frame and detailed component design analysis modules. The former is based on FUSAC, and its role is basic system analysis. The latters are based on existing, authorized calculation codes. As an example of detailed design module integration, we report current status of the two-dimensional equilibrium calculation code integration into FUSAC. The equilibrium-FUSAC integration discussed in this paper will enable to *online* evaluate the number of PF coils and the position and current density of each TF, CS, PF and divertor coils so that the magnetic stored energy W_{mag} is minimized.

[1] T. Yoshida *et al.*, CRIEPI Rep. No. T94001 (Central Research Institute of Electric Power Industry, Tokyo, 1994).
[2] R. Hiwatari *et al.*, Nucl. Fusion **44**, 106 (2004).
[3] N. Uckan and ITER Physics Group, ITER physics design guidelines: 1989 ITER Documentation Series No. 10 (IAEA, Vienna).
[4] T. Mizoguchi *et al.*, JAERI-M 98-120 (Naka Fusion Research Establishment, JAERI, Naka, 1987).
[5] K. Tobita *et al.*, Fusion Eng. Des. **81**, 1151 (2006).
[6] J. Sheffield *et al.*, Fusion Technol. **9**, 1986 (1986).
[7] K. Okano *et al.*, Fusion Eng. Des. **51-52**, 1025 (2000).
[8] K. Okano *et al.*, Nucl. Fusion **40**, 635 (2000).
[9] R. Hiwatari *et al.*, Nucl. Fusion **45**, 96 (2005).
[10] H. Fujieda *et al.*, JAERI-M 08-256 (Naka Fusion Research Establishment, JAERI, Naka, 1996).
[11] K. Okano *et al.*, Plasma Phys. Control. Fusion **32**, 225 (1990).