Speeding up of Equilibrium Calculations and Real Time Forecasts of Tokamak Discharges

L. E. Zakharov

Princeton Plasma Physics Laboratory, MS-27 P.O. Box 451, Princeton NJ 08543-0451

PPPL Research Seminar,

June 16, 2004, PPPL, Princeton, NJ

[1] **This work is supported by US DoE contract No. DE–AC020–76–CHO–3073.**

Contents

1 Three components of the Real Time Forecast (RTF) code 3

2 Cb.ESC-ASTRA-DCON-BALLOON as a prototype of RTF 5

2.1 ESC-ASTRA interface 6

2.2 Reconstruction of JET hollow current equilibria with ESC 7

2.3 Equilibrium Spline Interface (ESI) 14

3 Speeding up of equilibrium calculations 17

3.1 Linearized Grad-Shafranov equation in ESC 18

3.2 Linearized Equilibrium Equations for r, z (LEE). 19

4 New algorithms required for stiff transport models 23

5 The CodeBuilder for RTF development 24

5.1 Rising entropy is unavoidable in large numerical projects 25

5.2 Cb approach for controlling the code and its communications 28

5.3 Organization of I/O in Cb 30

5.4 Code documentation and maintenance system 31

6 Summary 32

RTF $\equiv$ a transport simulation code running faster than the discharge

In contrast to transport analysis (i.e., "yesterday weather prediction") and to transport predictive code (i.e, "long term weather forecast") RTF challenges near term predictions (in a fraction of τ_E) during the tokamak discharge.

- *RTF does not exist yet, but seems to be realistic.*
- *It would extend EFIT/rtEFIT idea and experience (very successful) to more “intelligent” prediction and control of a tokamak regime.*
- *It would include an implementation of basic principle transport models with tuning up by the data flow.*
- *In its turn, the real time discharge simulations can serve as a "filter" in the data flow and a generator of new signals for feed-backing the regime.*

RTF is a challenge for the code development

It requires progress, at least, in three areas:

1. *Fast (< 100 msec/eq) equilibrium calculating algorithms and routines,*
2. *Fast (< 100 msec/step) transport calculation algorithms (especially for stiff transport models),*
3. *Special software environment for:*
   - *the code development, its maintenance and documentation*
   - *establishing a rigorous code control and control of its communications between both non- and synchronized processes*
   - *plugging in and out the RTF components with minimal (and non-intrusive) requirements for independently developed components (diametrically different approach, e.g., to NTCC philosophy).*

ESC-ASTRA-DCON-BALLOON system of codes controlled by Code-Builder can be considered as a prototype of RTF:

1. *Uses fastest so far equilibrium algorithms, but*
2. *still standard (and slow for stiff models) transport calculation algorithms*
3. *uses a special software, which is based on first principles, for*
   - *the code development, its maintenance and documentation (so far only for ESC)*
   - *controlling the codes, their communication (with interactive access to control parameters),*
   - *using its components*
     (a) *in conventional (subroutine call routine) manner as well as*
     (b) *parallel processes,*
     (c) *"code talking"*

System is open for interfacing with other codes.

ESC-ASTRA interface is based on RFT consistent principles:

Both codes have interactive access to all control parameters.

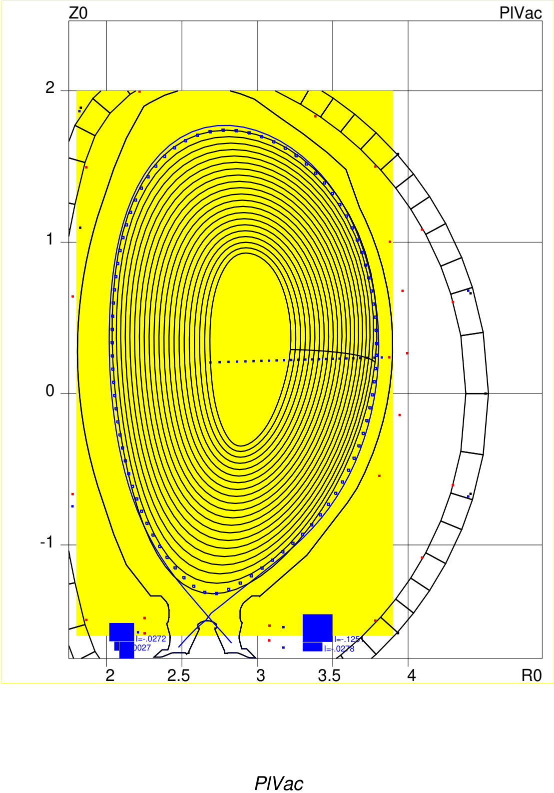
"Code talking" communication (through the shared memory) results in:

1. *A simple linking with 2 additional C-files for ASTRA source list containing*
   - *(a) one, with routines setting up the code talking and launch of ESC*
   - *(b) another, with reconstruction routines from equilibrium output data (ESI interface, covering transport, stability, particle orbits, and gyro-kinetic codes).*
2. *Elimination of name conflicts*
3. *Insensitivity to independent maintenance of two codes*
4. *Automatic consistency of executable of different versions of both codes*

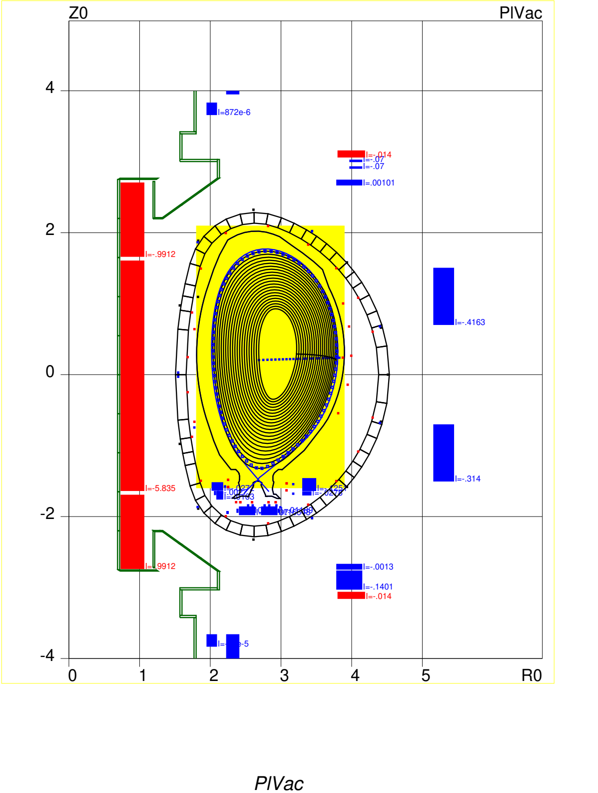
"Code talking" requires a special software for its implementation

The hollow current equilibrium option has been implemented into ESC in 2002.

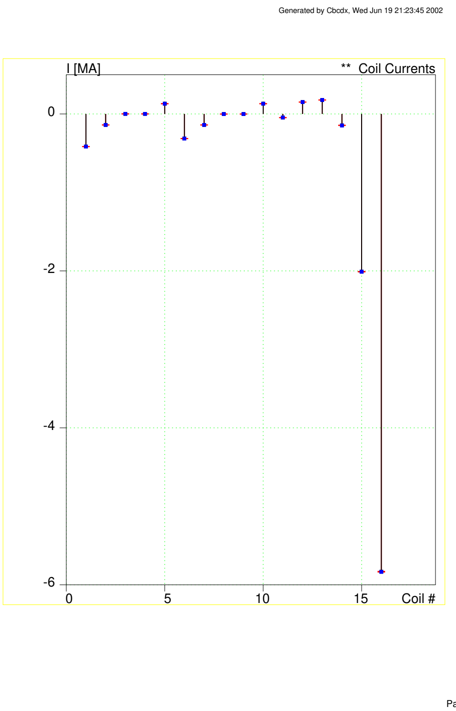
Generated by Cbcdx, Thu Jun 20 12:15:09 2002

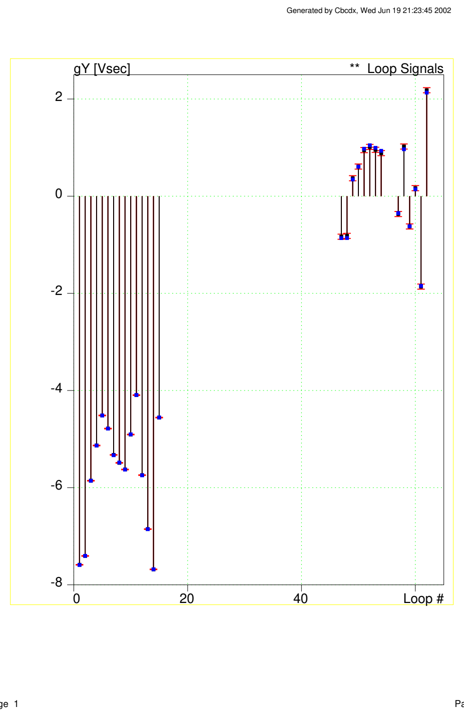


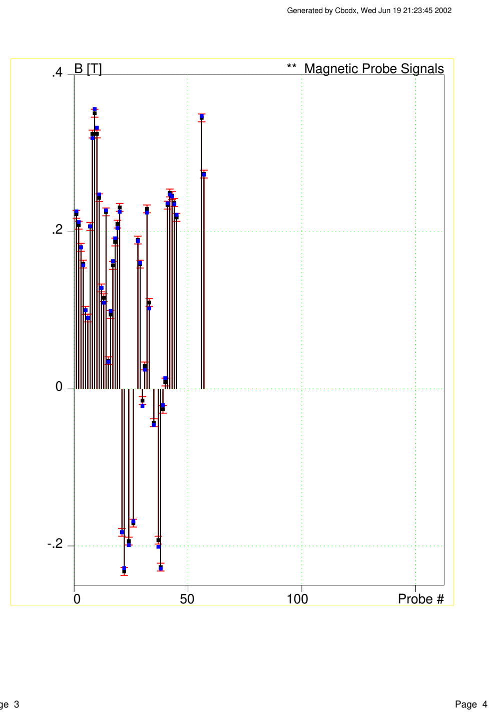
Generated by Cbcdx, Thu Jun 20 12:16:02 2002



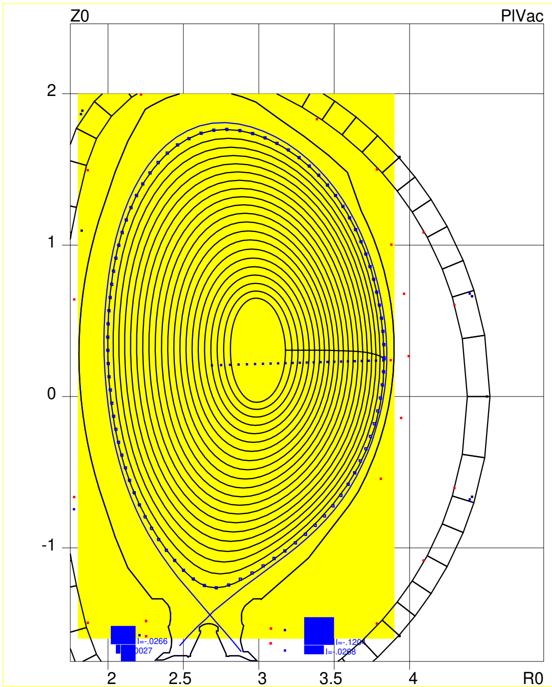
A perfect fit to magnetic data has been obtained (with freeing "iron" core surface currents)





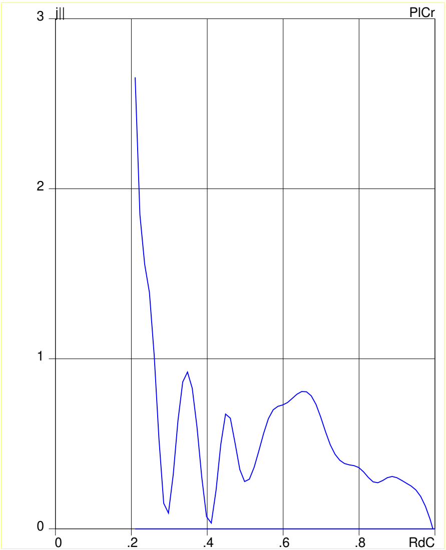


With a hole (22 % of the minor radius) the MSE signal was reproduced well. (A slightly enhanced pressure was used to position the hole).



Even with "crazy" current profile ESC produces a converged equilibrium.

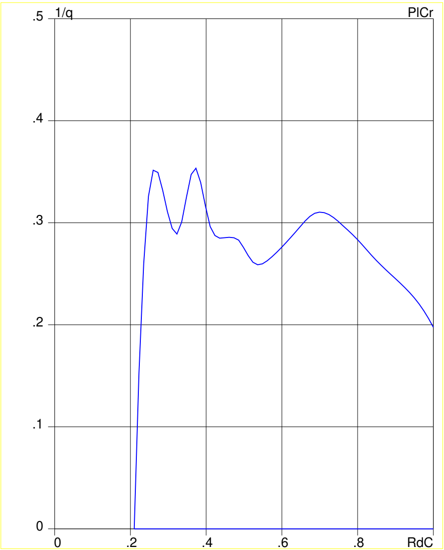
Generated by Cbcdx, Wed Jun 19 21:21:52 2002



PlCr

Page 1

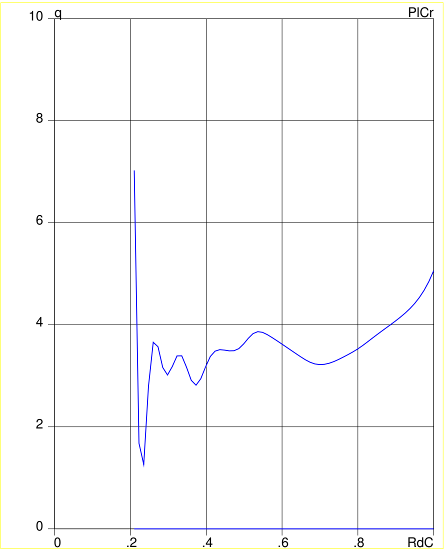
Generated by Cbcdx, Wed Jun 19 21:22:08 2002



PlCr

Page 2

Generated by Cbcdx, Wed Jun 19 21:22:25 2002

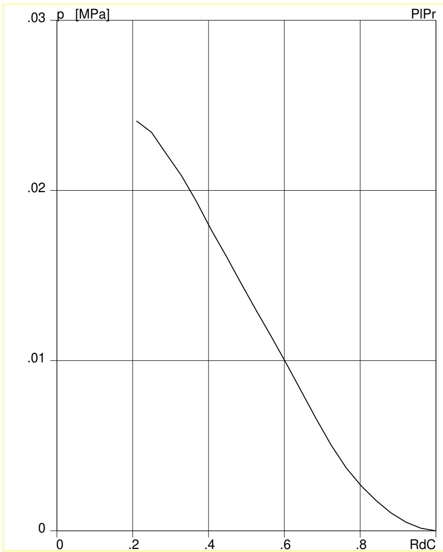


PlCr

Page 3

Pressure profile is slightly higher than the magnetic data would suggest.

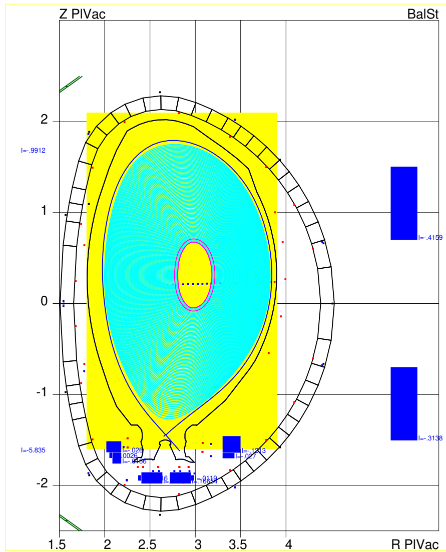
Generated by Cbcdx, Wed Jun 19 21:23:33 2002



PlPr

Page 1

Generated by Cbcdx, Thu Jun 20 13:38:37 2002

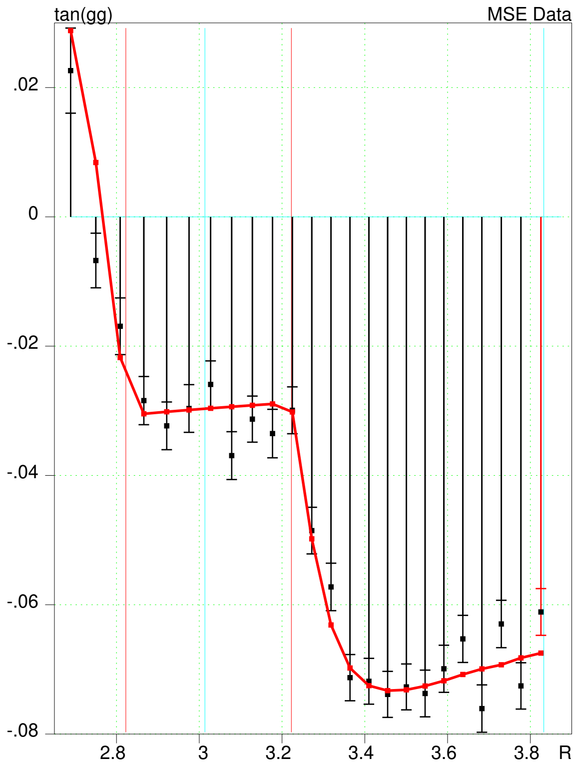


BalSt

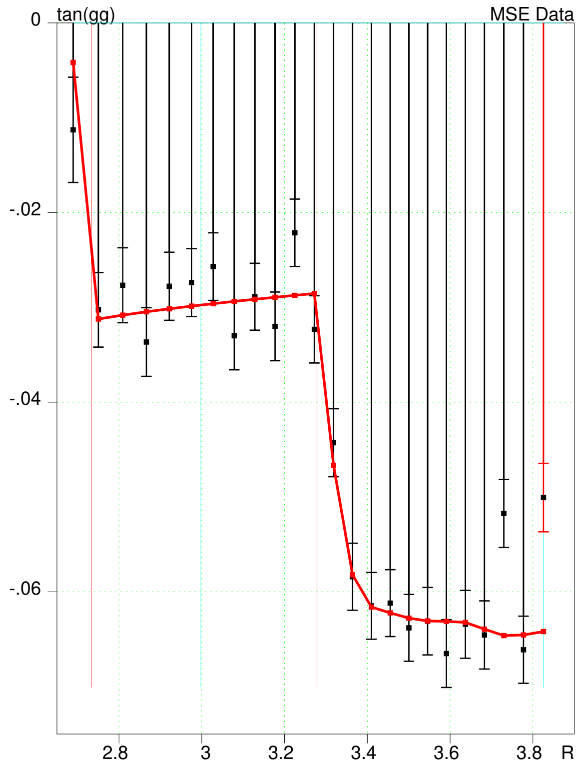
Page 1

Ballooning test (right picture) indicates no instability (light blue color). Presence of couple "red" surfaces near the hole is probably due to large numerical errors which is reported by BALLOON)

Negative current is not excluded in JET hollow current regimes

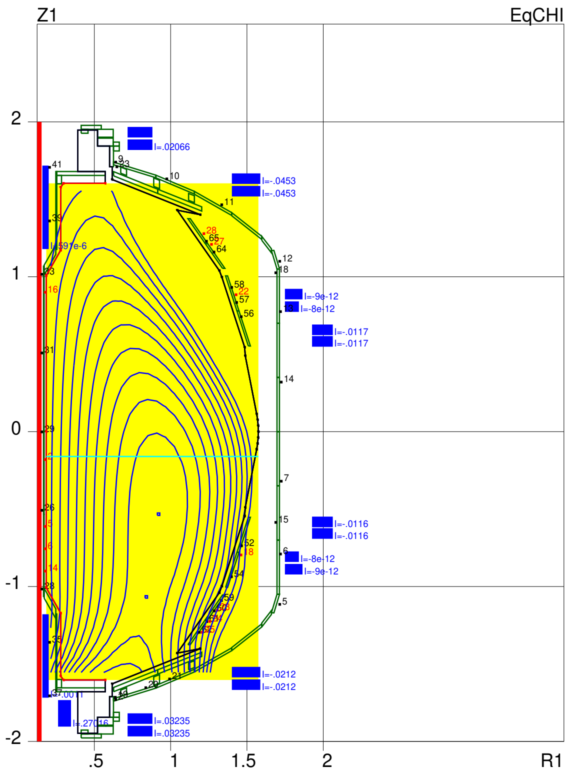


JET #55722



JET #55755

Reconstruction of open field lines was implemented into ESC in 2002



NTX #106488

ESI $\equiv$ data file + `escZ.c`-file with reconstruction routines

Basic functions of ESI and its routines provide reconstruction capability of all equilibrium data

for (a) transport, (b) stability, (c) particle gyro-orbits, and (d) kinetic codes

at any point in the plasma.

Set of 2D output Basic functions		
C- or FORTRAN	Math	Comment
`r[n],ra[n],rq[n]`	r, r'_a, r'_θ	major radius and its derivatives in [m]
`z[n],za[n],zq[n]`	z, z'_a, z'_θ	z-coordinate and its derivatives in [m]
`B[n],Ba[n],Bq[n]`	B, B'_a, B'_θ	Arrays for module of $\|\vec{B}\|$ and its derivatives
`gh[n],gha[n],ghq[n]`	$\eta, \eta'_a, \eta'_\theta$	Arrays of η function for transformation into straight field line coordinates

Set of output Basic radial profiles		
F[n],Fa[n]	$\bar{F}, \bar{F}'$	$\bar{F} \equiv rB_{tor}$ in [m T]
gFa[n],gFaa[n]	$\bar{\Phi}', \bar{\Phi}''$	Toroidal flux $\bar{\Phi}(a) \equiv \Phi/(2\pi)$ in [Vsec]
gYa[n],gYaa[n]	$\bar{\Psi}', \bar{\Psi}''$	Poloidal flux $\bar{\Psi}(a) \equiv \Psi/(2\pi)$ in [Vsec], $\bar{\Psi} = 0$ at the plasma boundary
T[n],Ta[n]	$\bar{F}\bar{F}'/\bar{\Psi}', (\bar{F}\bar{F}'/\bar{\Psi}')'$	Grad-Shafranov function in the RHS
P[n],Pa[n]	$\bar{p}'/\bar{\Psi}', (\bar{p}'/\bar{\Psi}')'$	Grad-Shafranov function in the RHS, where $\bar{p} \equiv \mu_0 p$ is the "rationalized" plasma pressure.

Having the present state of 4-code prototype it can be concluded that

Significant progress toward RTF is realistic and possible

(at least, in a technical sense).

Linearization is the fastest method of solving equilibrium equations

Grad-Shafranov equation

$$G(\bar{\Psi}) \equiv \frac{1}{r^2}\Delta^*\bar{\Psi} + P + \frac{T}{r^2} \to 0, \qquad P = P(\bar{\Psi}) \equiv \frac{d\bar{p}}{d\bar{\Psi}}, \quad T = T(\bar{\Psi}) \equiv \bar{F}\frac{d\bar{F}}{d\bar{\Psi}}, \tag{3.1}$$

where $\bar{F} \equiv rB_{tor}$, $\bar{p} \equiv \mu_0 p$.

During iterations G represents the discrepancy.

The fastest practical way to eliminated G is the use the linearized equation

$$\bar{\Psi} = \bar{\Psi}_0 + \psi, \quad \frac{1}{r^2}\Delta^*\psi + \frac{dP}{d\bar{\Psi}}\psi + \frac{1}{r^2}\frac{dT}{d\bar{\Psi}}\psi = -G(\bar{\Psi}_0). \tag{3.2}$$

One iteration in Eq.(3.2) is as fast as in Eq.(3.1) but provides much faster convergence.

Linearized GSh equation (1994) is used in ESC since 1996

GSh equation should be complemented with the coordinate advancing. In flux coordinates a, θ the final solution

$$\bar{\Psi} = \bar{\Psi}(a). \tag{3.3}$$

At every iteration there is a representation for flux coordinates

$$r = r(a, \theta), \quad z = z(a, \theta). \tag{3.4}$$

Advancing coordinates seems to be very simple

$$a \rightarrow a + \xi, \quad \bar{\Psi}_0(a + \xi) + \psi = \bar{\Psi}_0(a), \quad \xi \equiv -\frac{\psi}{\bar{\Psi}_0'} \tag{3.5}$$

and gives

$$r \rightarrow r + r_a'\xi + r_\theta'\sigma, \quad z \rightarrow z + z_a'\xi + z_\theta'\sigma. \tag{3.6}$$

(Here, σ is an arbitrary function redistributing the poloidal angle).

The caveat is that it involves differentiation.

Getting linearized equations for flux coordinates is a challenge

In flux coordinates

$$G \equiv \frac{1}{r^2}\Delta^*\bar{\Psi} = \frac{1}{\sqrt{g}}\left(\frac{g_{\theta\theta}}{\sqrt{g}}\bar{\Psi}'_a\right)'_a - \frac{1}{\sqrt{g}}\left(\frac{g_{a\theta}}{\sqrt{g}}\bar{\Psi}'_a\right)'_\theta + P + \frac{T}{r^2} = 0. \tag{3.7}$$

Here,

$$\begin{aligned} \bar{\Psi} = \bar{\Psi}(a), \quad D \equiv r'_a z'_\theta - z'_a r'_\theta, \quad \sqrt{g} \equiv rD, \\ g_{\theta\theta} \equiv r'^2_\theta + z'^2_\theta, \quad g_{a\theta} \equiv r'_a r'_\theta + z'_a z'_\theta. \end{aligned} \tag{3.8}$$

It is a highly non-linear equation for unknown functions

$$r(a,\theta), \quad z(a,\theta). \tag{3.9}$$

Even its linearization is difficult and was never preformed.

Also, one of the numerical problems is a freedom in the poloidal angle.

VMEQ tried to address it for both 2- and 3-D cases.

Prescribed $\sqrt{g}$, rather than θ dramatically simplifies the situation

With a given

$$\sqrt{g} = J(a, \theta), \tag{3.10}$$

the most difficult combination in equilibrium equation now is easy to linearize

$$Q \equiv \frac{\bar{\Psi}'}{\sqrt{g}} \to \frac{\bar{\Psi}'}{J}, \quad \delta Q = \frac{\delta\bar{\Psi}'}{J}. \tag{3.11}$$

Moreover, as a result of reformulation

LEE become a set of the first order equations for $\delta r, \delta z$.

$$\delta\left[(r_\theta'^2 + z_\theta'^2)Q\right]_a' - \delta\left[(r_a' r_\theta' + z_a' z_\theta')Q\right]_\theta' + J\delta P + J\delta\frac{T}{r^2} = -JG,$$

$$\delta r(r_a' r_\theta' - r_\theta' z_a') = J - r(r_a' r_\theta' - r_\theta' z_a'). \tag{3.12}$$

In Hamada coordinates, $J = J(a)$, LEE are extremely simple

$$
\begin{aligned}
&\left[g_{\theta\theta}\delta Q + Q\delta(r_\theta'^2 + z_\theta'^2)\right]_a' + (g_{a\theta}\delta Q + Q\delta g_{a\theta})_\theta' \\
&\quad + JP - 2\frac{JT}{r^3}\delta r = -J\bar{G}, \quad Q \equiv \frac{\bar{\Psi}}{J}, \\
&(r_a' z_\theta' - r_\theta' z_a')\delta r + r\delta(r_a' z_\theta' - r_\theta' z_a') = \\
&\quad J - r(r_a' z_\theta' - r_\theta' z_a').
\end{aligned}
\tag{3.13}
$$

Use of Hamada coordinates in equilibrium calculations makes them immediately consistent with stability codes.

Complete theory of LEE is now developed (including all variational formulations) and will be published.

LEE can be a backbone for equilibrium calculations in RTF

LEE are unique for 3-D stellarator equilibrium calculations

not only in expected significant speeding up but also in

Eliminating resonant terms from the RHS in magnetic equation

$$\left(\bar{\Psi}'\frac{\partial}{\partial\theta} - \bar{\Phi}'\frac{\partial}{\partial\zeta}\right)\nu = p'J(a,\theta,\zeta) \tag{3.14}$$

So far, we only started tackling the problem (using IFS-PPPL model)

A single heat diffusion equation

$$\frac{T^n - T^{n-1}}{\tau} - \frac{\Gamma'_x}{x} = S, \quad \Gamma = \Gamma(x, T^n, T'^n_x) \tag{4.1}$$

was solved with a shooting technique (with simplest 3-point approximation of the second term) was tested.

Works well (independent of stiffness) for $\tau > 0.1\tau_E$ with 5-6 iterations for resolving

$$\Gamma(x, T^n, T'^n_x) = \int_0^x \left(\frac{T^n - T^{n-1}}{\tau} - S\right) x dx \tag{4.2}$$

at each grid points $\times$ 5-6 iterations for shooting into boundary condition (30-40 times slower than solving linear equation, vs $10^3 - \infty$).

E.g., iterations in solving Eq.(4.2) can be eliminated by a transition ($\times$ 5 in speed)

$$\Gamma(x, T^n, T'^n_x) \quad \rightarrow \quad T'_x(x, T^n, \Gamma) \tag{4.3}$$

The progress is possible in speeding up of the transport codes.

Special software is required for RTF development

Too much dependent and independent components, control parameters, communications, documentations in sophisticated codes to be handle solely by a human.

At the same time, to be easy understandable and acceptable

The software should be non-intrusive and based on first principles

and should avoid

“bird” languages and sporadic or hidden conventions.

Control param user has in mind	typical FORTRAN namelist
promotion to AL	igrid
promotion to group leader	rleft
major monetary award	rright
promotion within the rank	zbotto
...	ifcoil
...	iecoil
...	iacoil
...	...
...	af2
...	fcturn
minor disciplinary actions	he
money withholding	ecturn
layoff	vsid
torture	rvs
electric chair	zvs
elimination at the genetic level	we

Unstructured lists is a source of entropy

The total number N_0 of possible matches

$$N_0 = n!, \quad S_0 \equiv \ln N_0 \simeq n\left(\ln n - 1/2\right). \tag{5.1}$$

Now Suppose both sides subdivide each set on n/k matching sections with k elements in each.

Two cases are possible.

1. Sporadic (independent) permutations in each of sections

$$N_1 = \underbrace{k!k!\dots k!}_{n/k \text{ times}} = (k!)^{\frac{n}{k}},$$

$$S_1 = \ln N_1 \simeq \frac{n}{k}\left(k \ln k - k/2\right) = n\left(\ln k - 1/2\right) \simeq \frac{\ln k}{\ln n} S_0. \tag{5.2}$$

Simple grouping of variables has a little effect on entropy

Nested structures reduce the entropy in a crucial manner

2. Matching one section after another in sequence

$$
\begin{aligned}
N_2 &= \underbrace{k! + k! \ldots + k!}_{n/k \text{ times}} = (k!)\frac{n}{k}, \\
S_2 &= \ln N_2 \simeq (k-1)\ln(k-1) + \ln n \simeq \frac{k}{n} S_0 \ll S_0.
\end{aligned}
\tag{5.3}
$$

Nested structures

type_name{parameter list;

  ......{parameter list;

  ......}

...

} automatically provide consistency in processing control parameters.

(a) People always think in terms of nested structures

(b) People are limited in maintaining nested structures

CodeBuilder $\equiv$ structure recognition + drivers for controlling its elements

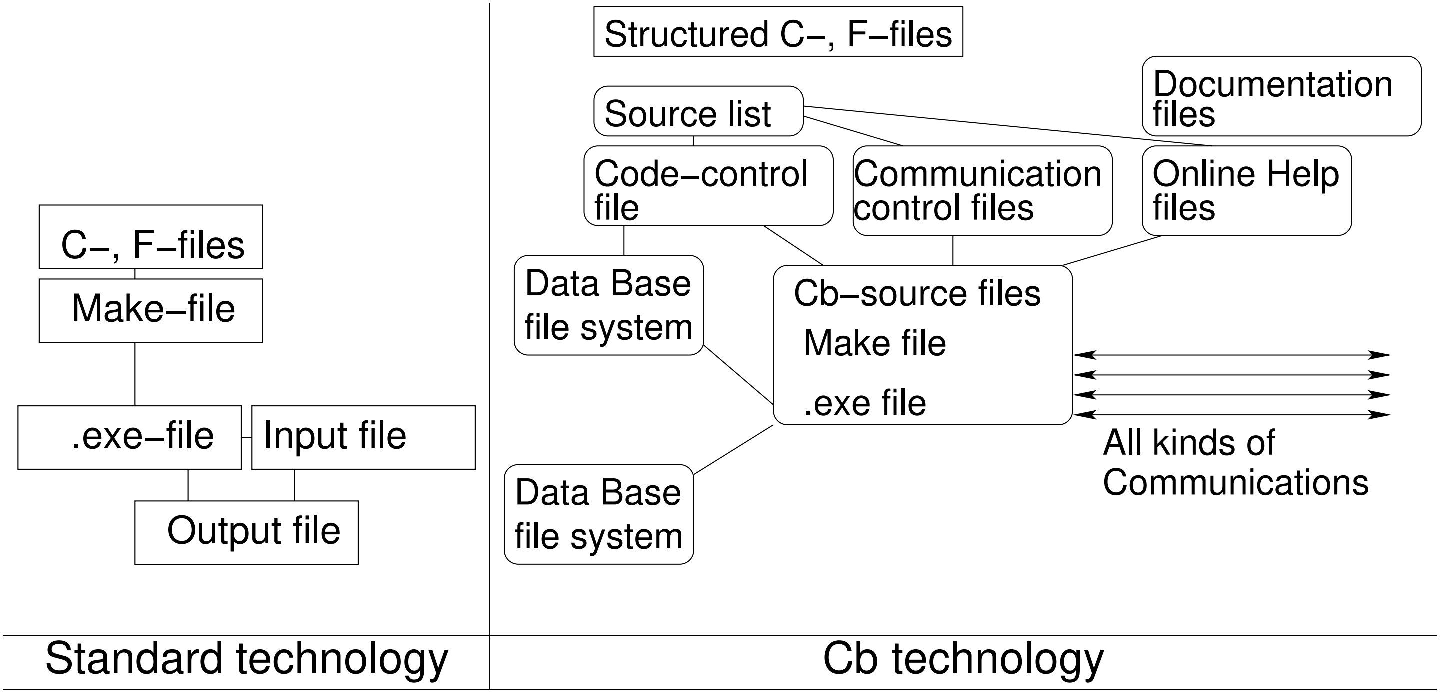
Cb is based on 3 types of structural elements reflecting nested structure and virtual parallel processes.

Initially was oriented on C- and FORTRAN:

1. *Structuring of the code with formalized comment-lines by the author.*
2. *Generating code control, communication control and On-Line Help files (automatic).*
3. *Editing templates of dB structure and of objects for different types of communications (interactive regime, graphics, I/O, messaging).*
4. *Processing the make-file to create the code launch command (make-file is replaced by a structured source list to avoid an extra "bird" language).*
5. *Generation of code maintenance documentation (at present, only for C).*

Cb is intrinsically a non-intrusive technology based on first principles consistent with RFT

Cb, while being non-intrusive, requires an extended source code



Cb is intrinsically a non-intrusive technology based on first principles consistent with RFT

Cb Database structure is automatically mapped to the code structure

Internally

1. ASCII files are structured, human understandable and allow free commenting.
2. Binary files represent unstructured steams of data with their structure saved in a separate human understandable ASCII format.

Communication and maintenance documentation are separated in Cb

On-Line Help

1. is distributed and attached to each section of the code,
2. has ASCII, dvi and PostScript versions
3. generated from the corresponding ASCII and LaTex files, starting from automatically generated templates
4. Automatically processed the Control parameter and I/O sections.

This provide the comprehensiveness and easy access with

No one control parameter hidden from the user and no one missed

Code maintenance documentation is mapped on a separate structure for code routines and global variables resulting in

Clean source code and comprehensive documentation of each routine

RTF is a necessary element in the next step transport studies

RTF is a challenging but realistic goal for equilibrium-transport-stability codes:

1. Equilibrium calculations, including equilibrium reconstruction, can be speed up to the necessary real time level.
2. Solving transport with stiff models can be significantly accelerated, if different solvers will be used for fast ($t \ll \tau_E$) and slow ($t \simeq \tau_E$) processes.
3. Zero banana width approximation can be used for heating sources from NB and alphas (thus, eliminating time consuming particle trajectory calculations).
4. At least, the ballooning stability analysis can be done practically at very time step (or in parallel in an asynchronous mode).
5. Low-n stability (including resistive) calculations can be done in parallel asynchronously.

RTF has already a well established prototype open for extension