



講座

パーソナルコンピュータで行うプラズマ解析

2. トカマクプラズマの平衡解析

新 谷 吉 郎
(株式会社 東芝)

Equilibrium Analysis of Tokamak Plasma

SHINYA Kichiro

Toshiba Corporation, Kawasaki 210-0862, Japan

(Received 14 February 2000)

Abstract

Personal computers can be used for the scientific numerical calculations due to their performance improvement based on the recent progress in MPU development, in addition to the UNIX based engineering workstations and the mainframe computers. As an example of the utilization of personal computers, equilibrium analysis of a tokamak plasma is presented. The objectives and general features of the equilibrium analyses are briefly reviewed and the Grad-Shafranov equation is deduced. Based on the physical interpretation of the equilibrium analyses, programming of the code is explained. Fortran is employed as a programming language and the Macintosh personal computer is selected as a platform. The program made for an exercise will be available on the home page of the Japan Society of Plasma Science and Nuclear Fusion Research.

Keywords :

personal computer, equilibrium analysis, Grad-Shafranov equation, numerical calculation, programming, Fortran, Power Macintosh

2.1 はじめに

トカマクプラズマの平衡解析は、実験結果の解析、MHD 安定性の理論解析やエネルギー輸送のシミュレーション、さらに炉設計に際しての基本ツールとして広汎に使用されています (例えば[1, 2])。また名前の知られた平衡解析コードは多数存在します (例えば[3-5])。それぞれのコードの目的とするところによって機能に多少の違いはありますが、基本的にグラッド・シャフラーノフ方程式[6]を解くことに変わりはありません。ただし解き方には2種類あって、一つはコイル電流や磁束ループの値を境界条件等として入力し、プラズマ形状を求めるもので、実験解析などで使用されています。実験で得られたいろいろなデータに基づいてプラズマ形状を求めることから、プラズマ再構成と呼ばれたりします。もう一つ

は設計などで使用されるもので、形成したいプラズマの形状を入力して、この形状を達成するためのコイル電流を求めるというものです。前者を Direct solution, 後者を Inverse solution と呼ぶこともあります。コイル電流が原因でプラズマ形状はその結果だと考えると、何となく Direct と Inverse の意味がつかめるかと思います。本講座では後者の目的での平衡解析について説明し、それに沿ってプログラムを作成します。また作成したソースプログラムをコンパイル、リンクして実行形式のファイルを作成し、テストデータで実行してみます。

言語は Fortran を使用し、コンパイルや実行は Power Macintosh で行います。どうして速度の遅い Macintosh で実行するのかという疑問を持たれるかも知れません。Macintosh, というよりは Macintosh の頭脳である Pow-

author's e-mail : shinya@pic.hama.toshiba.co.jp

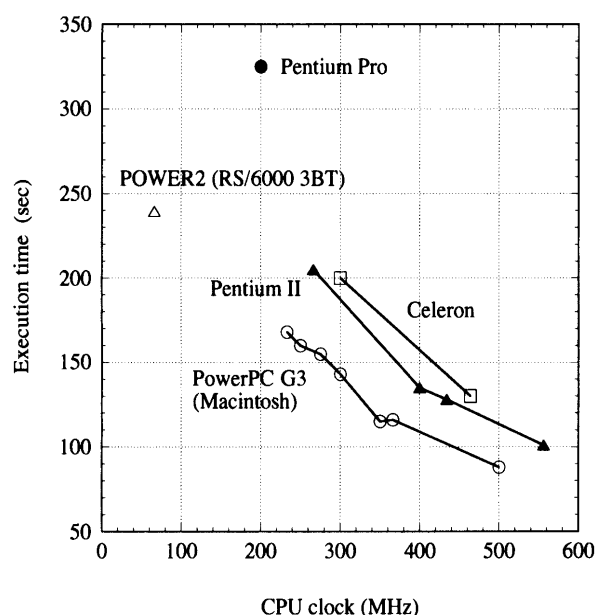


Fig. 1 Comparison of the execution times between PowerPC and Intel x 86 processors. Equilibrium analysis code TOSCA was used for the evaluation. A result of IBM workstation RS/6000 3BT is also plotted as reference. The execution time is inversely proportional to the cpu clock cycle.

erPC は決して遅くはないのです。Fig. 1 は筆者らが開発したトカマクプラズマの平衡解析コード TOSCA の計算時間を比較したものです。Intel の CPU に比べてかなりのアドバンスがありますね。これは PowerPC の優れた実数演算能力に起因するものです。少し古い機種になってしまいましたが、数値計算用に設計されていて筆者も使用している IBM 製ワークステーション RS/6000 3BT の結果を参考に載せてあります。最近のパーソナルコンピュータの性能がいかに優れているかがわかります。科学技術計算でよく使用されている Alpha プロセッサについては計測しておりませんが、他の数値計算結果から推測すると PowerPC G3 と同程度のようなようです。

UCLA の Appleseed というプロジェクトでは 4 機の Power Mac G3 をクラスタにすることで Cray T3E-900 と同程度の計算速度を達成しています。次の URL を参照してください (<http://exodus.physics.ucla.edu/appleseed/appleseed.html>)。Appleseed は John M. Dawson 教授が主唱している UCLA の Plasma Simulation Group の成果の一つです。下記ホームページでいろいろなシミュレーション結果を見ることができます (<http://exodus.physics.ucla.edu/UCLAPlasmaSimulation.html>)。本講座では Appleseed で使用しているものと同じ Absoft Pro Fortran を使用します。ちなみに Fig. 1 の結果はすべて

Absoft Pro Fortran を使用して計測したものです。Absoft Pro Fortran は Macintosh だけでなく、MS Windows と Linux 用が用意されています。

いろいろなプラットフォームでの各種 Fortran 90 コンパイラの性能を評価した結果が、次のホームページに掲載されています (<http://www.cs.rpi.edu/~szymansk/OOF90/eval.html>)。Fortran 90 のコンパイラにはまだまだバグが存在する可能性があるみたいです。Absoft Pro Fortran はその中では IBM, Fujitsu の Fortran と並んで最高の成績を示しています。本講座では Fortran 90 は使わずに Fortran 77 でプログラミングを行っています。

2.2 道具を揃える

最初に道具を揃えましょう。Power Macintosh が必要です。G3 か G4 が計算速度の面から最適です。プログラム開発環境はそれ自身が結構メモリを必要としますから、メモリは 128 MB くらいは積んでおいたほうが快適です。Macintosh 用の Fortran は現在 Absoft 社製のものしかありません。計算結果のグラフィック表示は、昔懐かしい Calcomp ルーチンを Postscript ファイルに出力するので、これを描画するために Ghostscript または Ghostview が必要です。プログラムやデータを作成したり編集するためにはエディタが必要となります。Macintosh ではフリーのエディタが多数ありますが、商用版の BBEdit がプログラミングにはお勧めです。これらについては以下の URL で詳細を調べてください。Ghostscript や BBEdit のフリー版をダウンロードできます。

<http://www.apple.com>

<http://www.absoft.com>

<http://www.cs.wisc.edu/~ghost/mac/>

<http://www.cs.wisc.edu/~ghost/index.html>

<http://www.kiffe.com/macghostview.html>

<http://www.barebones.com/>

Absoft Pro Fortran には C/C++ と Macintosh 用の開発環境 MPW が付属しています。本講座では MPW を積極的に使用することはありませんが、コンパイルから実行形式のファイルを作成する一連の動作を行うための makefile を作成する際には非常に重宝します。このあたりについては 2.5.2 でもう少し詳しく説明いたします。

2.3 グラッドーシャフラーノフ方程式

それでは最初に平衡計算の基礎となるグラッドーシャフラーノフ方程式を導出してみます。グラッドーシャフ

ノフ方程式の導出は多数のプラズマ物理の教科書で記述されていますが、ここで復習しておきましょう。使用する方程式は、プラズマを一流体と見なした場合の運動方程式とマクスウェル方程式の一部です。運動方程式では電磁力とプラズマ圧力の項だけを考え、重力は無視します。電場も考えません。プラズマ中に高速な流れは存在しないものとします。このような条件での平衡状態（時間微分が零となる）の運動方程式とマクスウェル方程式は以下のようになります。

$$\begin{aligned} \mathbf{j} \times \mathbf{B} - \nabla p &= 0 \\ \mu_0 \mathbf{j} &= \nabla \times \mathbf{B} \\ \nabla \cdot \mathbf{B} &= 0 \end{aligned}$$

トカマクプラズマは軸対称であるため、円筒座標系 (r, φ, z) を用いて成分に分解します。プラズマ電流密度とトロイダル磁場は、それぞれ j_φ , B_φ となります。運動方程式とマクスウェル方程式の r, z, φ 成分は

$$j_\varphi B_z - j_z B_\varphi - \frac{\partial p}{\partial r} = 0 \quad (1)$$

$$j_\varphi B_r - j_r B_\varphi - \frac{\partial p}{\partial r} = 0 \quad (2)$$

$$j_z B_\varphi - j_r B_z = 0 \quad (3)$$

$$\mu_0 j_r = -\frac{\partial B_\varphi}{\partial z} = -\frac{1}{r} \frac{\partial F}{\partial z} \quad (4)$$

$$\mu_0 j_z = \frac{1}{r} \frac{\partial}{\partial r} (r B_\varphi) = \frac{1}{r} \frac{\partial F}{\partial r} \quad (5)$$

$$\mu_0 j_\varphi = \frac{\partial B_r}{\partial z} - \frac{\partial B_z}{\partial r} \quad (6)$$

ただしトロイダル磁場 B_φ のかわりに $F = r B_\varphi$ を用いました。 F はポロイダル電流に対応するもので、真空磁場に対しては一定となります。したがって F の値が真空磁場に対する値と異なる分はプラズマによる常磁性または反磁性の効果を示しています。

次に磁場の発散の式は

$$\frac{1}{r} \frac{\partial}{\partial r} (r B_r) + \frac{\partial B_z}{\partial z} = \frac{1}{r} \frac{\partial}{\partial r} (r B_r) + \frac{1}{r} \frac{\partial}{\partial z} (r B_z) = 0$$

上式からは、よく知られているようにポロイダル磁束 Ψ (実際は $1/2\pi$) を導入することができ、 B_r , B_z は次式のようにポロイダル磁束 Ψ で表すことができます。

$$B_r = -\frac{1}{r} \frac{\partial \Psi}{\partial z}, \quad B_z = \frac{1}{r} \frac{\partial \Psi}{\partial r} \quad (7)$$

(7)式を(6)式に代入するとグラッド-シャフランド方程式の片割れが得られます。

$$r \frac{\partial}{\partial r} \left(\frac{1}{r} \frac{\partial \Psi}{\partial r} \right) + \frac{\partial^2 \Psi}{\partial z^2} = -\mu_0 j_\varphi \quad (8)$$

上式ではトロイダル電流 j_φ , つまりプラズマ電流がソース項として残っています。そこで(7)式と(5), (4)式をそれぞれ(1), (2)式に代入して整理すると

$$\mu_0 j_\varphi \frac{1}{r} \frac{\partial \Psi}{\partial z} - \frac{1}{r^2} \frac{\partial}{\partial z} \left(\frac{F^2}{2} \right) - \mu_0 \frac{\partial p}{\partial z} = 0 \quad (9)$$

$$\mu_0 j_\varphi \frac{1}{r} \frac{\partial \Psi}{\partial r} - \frac{1}{r^2} \frac{\partial}{\partial r} \left(\frac{F^2}{2} \right) - \mu_0 \frac{\partial p}{\partial r} = 0 \quad (10)$$

が得られます。ここから目的地まではあと一歩です。

(9) $\times \frac{\partial \Psi}{\partial z}$ - (10) $\times \frac{\partial \Psi}{\partial r}$ を作ると

$$\begin{aligned} -\frac{1}{r^2} \left[\frac{\partial \Psi}{\partial z} \frac{\partial}{\partial r} \left(\frac{F^2}{2} \right) - \frac{\partial \Psi}{\partial r} \frac{\partial}{\partial z} \left(\frac{F^2}{2} \right) \right] \\ - \mu_0 \left(\frac{\partial \Psi}{\partial z} \frac{\partial p}{\partial r} - \frac{\partial \Psi}{\partial r} \frac{\partial p}{\partial z} \right) = 0 \end{aligned}$$

となりますが、この式が恒久的に成立するには、 $F^2/2$ と Ψ および p と Ψ の Jacobian が零となることです。このことは $F^2/2$ と p がポロイダル磁束 Ψ のみの関数であることに対応しています。すなわち、

$$\frac{\partial}{\partial r} \left(\frac{F^2}{2} \right) = \frac{\partial \Psi}{\partial r} \frac{d}{d\Psi} \left(\frac{F^2}{2} \right),$$

$$\frac{\partial}{\partial z} \left(\frac{F^2}{2} \right) = \frac{\partial \Psi}{\partial z} \frac{d}{d\Psi} \left(\frac{F^2}{2} \right)$$

$$\frac{\partial p}{\partial r} = \frac{\partial \Psi}{\partial r} \frac{dp}{d\Psi}, \quad \frac{\partial p}{\partial z} = \frac{\partial \Psi}{\partial z} \frac{dp}{d\Psi}$$

$F^2/2$ と p がポロイダル磁束 Ψ のみの関数であることは、磁気面の性質を考えると容易に推測がつくと思います。上の関係を用いると(9), (10)式は一つにまとめられてグラッド-シャフランド方程式のもう一方が得られます。

$$j_\varphi = r \frac{dp}{d\Psi} + \frac{1}{r} \frac{d}{d\Psi} \left(\frac{F^2}{2\mu_0} \right) \quad (11)$$

これで終わりと安心してはいけません。まだ $F^2/2$ と p が具体的に Ψ のどのような関数になるかの問題が残っ

ています。実は平衡計算では $F^2/2$ と p の関数形を確定することはできません。このためにはプラズマのエネルギー輸送方程式を解く必要があります。したがって、通常は取り扱おうとしている平衡に対して妥当と考えられる関数形を想定して解きます。プラズマ電流分布は、中心ピークしたものから平坦に近いもの、さらには最近のトカマク研究の重要な対象となっている負磁気シア分布プラズマで実現しているホロー電流分布まで、多種多様なものが考えられますが、磁束分布はいずれの電流分布に対しても放物線に近い形を取ることがわかっています。すなわち、 $\sqrt{1-\tilde{\psi}}$ をプラズマの小半径と見なして電流分布形を考えればよいということになります。ただし、 $\tilde{\psi}$ は磁気軸で1、プラズマ表面で0と規格化したポロイダル磁束です。

本講座では、次式で示されるような関数形を採用します。

$$j_{\psi} = j_0 \left[\beta_j \frac{r}{R_p} + (1-\beta_j) \frac{R_p}{r} \right] [1 - (1-\tilde{\psi})^n]^m \quad (12)$$

(12)式で j_0 はプラズマ電流値で決まります。 R_p はプラズマ大半径で、 β_j はポロイダルベータ値に対応するパラメータです。 n と m の値を調節することによって内部インダクタンスと磁気軸での安全係数を調整することができます。(12)式の関数形ではホロー電流分布は取り扱えません。いろいろな関数形を試してホロー電流分布を作り出してみてください。

2.4 数値解法

2.4.1 解法の概略

グラッド・シャフラノフ方程式が解析的に解けるのは特殊な関数形のプラズマ電流分布の場合だけです。一般的には数値解法でのみ解くことができます。もちろん、この講座でも数値解法によって解を求めます。プラズマ断面が十分に収まるような矩形の領域を考えて r 方向と z 方向をそれぞれ nr, nz のメッシュに分割し、このメッシュで(8)式を差分化します。 nr と nz は後述するように Buneman の Poisson solver を使用する場合には $nr = 2^k + 1$, $nz = 2^l + 1$ の形の整数値である必要があります。グリーン関数を用いて解く場合にはこのような制限はありません。

数値解法の概略は以下のとおりです。まず設定されたプラズマ形状 (prescribed plasma shape) に一致する、あるいは近い領域のメッシュ点にプラズマ電流分布の初期値を分布させます。分布形は一定値でも放物型でもよ

いでしょう。次にこの電流値に対して(8)式を解いて磁束 ψ の分布を求めます。この磁束はプラズマ電流に起因するものだけなので、ここでは ψ^{plasma} と表すこととします。一方、コイル電流の初期値もあらかじめ設定しておき、コイル電流のつくる磁束 ψ^{coil} を計算します。双方の和として得られた磁束分布を(12)式に代入して新しいプラズマ電流分布を計算します。新しいプラズマ電流分布が許容誤差の範囲で初期値として設定した値と一致しているならば、それを解とします。一致していない場合は、上記の作業を許容誤差の範囲に収まるまで繰り返します。簡単に言ってしまうと、平衡計算で行っていることはこれだけです。後はアクセサリをたくさん付けて格好よくしているだけです、と言うと少々乱暴な言い方になりますが。

上述の解法の中でコイル電流値はどのようにして決めるのかははっきりしません。実はコイル電流は、設定プラズマ形状を離散化した複数の点において、磁束の値が等しくなるように最小自乗法によって毎繰り返しにおいて修正されます。この修正されたコイル電流値を次のコイルによる磁束を計算するために使用するわけです。このようにして得られたプラズマ形状は、あらかじめ設定されたプラズマ形状にほぼ一致し、得られたコイル電流はその形状を平衡維持できる電流値となっています。

2.4.2 コイル電流値の最適化

実際の炉設計においては、コイル電流の大きさは炉の経済性に大きく影響します。したがって、設定されたプラズマ形状を達成するコイル電流値は、できるだけ小さいほうが望ましいことになります。正確にはコイル電流値だけではなく、コイル電源のパワーについても評価しなければならぬので、電流の時間変化もできるだけ小さくしたいのですが、嬉しいことと言うべきか悩ましいことと言うべきか、コイル電流値がかなり違っていてもプラズマ形状はあまり変化しないコイル電流値の組み合わせが存在するという特徴が、トカマクプラズマの平衡にはあります。コイルの数が不必要に多くなると、この性質が顕著になります。このような事情から最小自乗法でコイル電流値を求める際に、コイル電流値の絶対値を抑制するような拘束条件を付加します。本講座では次式で示される量を最小化するようにコイル電流を求めています。

$$E = \sum_{k=1}^{N_p} W_k \left[\sum_{j=1}^{N_c} \alpha_{kj} I_j - (\psi_{\text{surface}} - \psi_k^{\text{plasma}}) \right]^2$$

$$+ \sum_{j=1}^{N_c} \gamma_j I_j^2 + \eta \left(\sum_{j=1}^{N_c} f_{mj} I_j - \Psi_{\text{center}} \right)^2 \quad (13)$$

(13)式で、右辺第一項がプラズマ表面を設定形状に合わせるための項、第二項はコイル電流値の絶対値を抑制する項、第三項はコイルがプラズマに供給する磁束を指定する項です。炉設計でプラズマの運転シナリオを考える際には、この供給磁束が重要になります。 α_{kj} は k 番目の設定プラズマ表面点と j 番目のコイルの、 f_{mj} はプラズマ中心と j 番目のコイルのグリーン関数です。 Ψ_{surface} はプラズマ表面の磁束値、 Ψ_k^{plasma} は k 番目のプラズマ表面点でのプラズマ電流値のみによる磁束値です。 Ψ_{center} はコイルからの供給磁束ですから、プラズマ電流による磁束は含まれていません。各項の重み W_k, γ_j, η は個々の平衡計算に際して目的に応じて設定します。ダイバータプラズマの場合は、設定ヌル点位置に確実に実際のヌル点が存在するような仕掛けが必要です。これは設定ヌル点がプラズマ表面であることと、そこでポロイダル磁場が消失するという条件を(13)式に付加することで実現できます。この条件の定式化は、皆さんが試みてください。

最小自乗法を用いてコイル電流を求めるには、コイル電流を未知数とした連立一次方程式を解くことに帰着します。例題のプログラムでは一番簡単な数値解法であるガウス消去法を用いて解いています。

2.4.3 プラズマ表面の決定

次にプラズマ表面の磁束 Ψ_{surface} を決める方法を説明します。プラズマ形状は、ダイバータプラズマとリミタプラズマに大別できます。ダイバータプラズマでは磁気軸での磁束の値に一番近い磁束値を有するセパトリックスをプラズマ表面とします。リミタプラズマでは複数あるリミタ点で、磁束の値が磁気軸での磁束の値に一番近いものを実際のリミタとし、この点を通る磁力線でプラズマ表面を決めます。もちろんダイバータプラズマの計算に際してリミタを設置することができます。この場合は、セパトリックスとリミタ点でのすべての磁束の値の中で一番磁気軸での値に近いものをプラズマ表面の磁束値とします。したがって、ダイバータプラズマの平衡を求めるつもりで、実際はリミタプラズマになってしまったということが起こりえます。当然その反対のケースもあります。ダイバータプラズマの計算を収束させるために、時々仮想的なリミタ点を入力することがありますが、このような場合は、重々、上に述べたようなことが起こらないように注意しましょう。

プラズマ形状は次式で示されるものを36分割していま

す。

$$\begin{aligned} r &= R_p + a \cos(\theta + \delta_1 \sin \theta), \quad \delta_1 = \sin^{-1} \delta \\ z &= Z_p + \kappa a \sin \theta \end{aligned} \quad (14)$$

(14)式でプラズマ断面の非円形度 κ と三角度 δ が指定できます。 θ の原点は大半径外側の赤道面上で、ポロイダル角の反時計方向を正とします。(14)式で記述されるプラズマ表面は滑らかな曲線に載っているため、ダイバータプラズマのヌル点近傍を表現できません。このためダイバータプラズマの計算を行う場合には、ヌル点を中心とした11点はプラズマ表面点から省いています。

2.4.4 グリーン関数と Buneman の Poisson solver

計算に際しては、計算メッシュ点間、メッシュ点とコイル、設定プラズマ表面とコイル等のグリーン関数が必要になります。トカマクは軸対称であるため、これらのグリーン関数はすべて完全楕円積分を用いて表すことができます。電気を専攻する人にとっては、グリーン関数というよりは2個の同軸コイル間の相互インダクタンスといったほうがわかりやすいかも知れません。

$$\alpha_{ij} = \frac{\mu_0 \sqrt{r_i r_j}}{\pi k} \left[\left(1 - \frac{k^2}{2} \right) K(k) - E(k) \right],$$

$$k^2 = \frac{4r_i r_j}{(r_i + r_j)^2 + (z_i - z_j)^2}$$

計算メッシュの数が多くなると（本講座では $nr = 65$, $nz = 129$ としています）グリーン関数の数は膨大な量になります。このような膨大な量のグリーン関数を毎回計算することは大変な時間の浪費になりますから、通常は収束計算を開始する前に1回だけ計算してテーブルとして記憶しておきます。テーブルのサイズも結構大きくなるのですが、今ではメモリは潤沢に使用できるので問題ないでしょう。それでも膨大な量のグリーン関数の四則演算は残るわけですから、これに要する計算時間は馬鹿になりません。Buneman の Poisson solver[7-11]はこの点を大幅に改良してくれます。メッシュ数が多くなればなるほど効果を発揮します。

軸対称でのポアソン方程式は次式で表されます。

$$\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial \Psi}{\partial r} \right) + \frac{\partial^2 \Psi}{\partial z^2} = -\frac{\rho_c}{\epsilon_0}$$

この式の r の項を展開すると、グラッド-シャフラーノフ方程式とは r の一階微分の項の符号が異なるだけということがわかります。したがってPoisson solverがほんの少しの変更でそのままグラッド-シャフラーノフ方程式の

解法に適用できるのです。残念ながら例題で作成するプログラムではBunemanのPoisson solverは使用していません。ソースコードが公開されていますから[7], 自分で組み込んでみてください。

例題のプログラムでは使用していませんがBunemanのPoisson solverについて簡単に説明します。(8)式を中央差分で差分化すると

$$\frac{\psi_{i+1,j} + \psi_{i-1,j} - 2\psi_{i,j}}{\Delta r^2} - \frac{1}{r_i} \frac{\psi_{i+1,j} - \psi_{i-1,j}}{2\Delta r} + \frac{\psi_{i,j+1} + \psi_{i,j-1} - 2\psi_{i,j}}{\Delta z^2} = -\mu_0 r_i J_{i,j}$$

上式を $(j-1, j, j+1)$ について整理すると, $s = (\Delta r / \Delta z)^2$ と置いて

$$A = \begin{pmatrix} -2\left(1 + \frac{1}{s}\right) & \frac{1}{s}\left(1 - \frac{\Delta r}{2r_2}\right) & 0 & \dots & \dots & 0 \\ \frac{1}{s}\left(1 + \frac{\Delta r}{2r_3}\right) & -2\left(1 + \frac{1}{s}\right) & \frac{1}{s}\left(1 - \frac{\Delta r}{2r_3}\right) & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & -2\left(1 + \frac{1}{s}\right) & \frac{1}{s}\left(1 - \frac{\Delta r}{2r_{nr-2}}\right) \\ 0 & \dots & \dots & 0 & \frac{1}{s}\left(1 + \frac{\Delta r}{2r_{nr-1}}\right) & -2\left(1 + \frac{1}{s}\right) \end{pmatrix}$$

$$Q_j = \begin{pmatrix} \frac{1}{2}\Delta z^2 \mu_0 r_{2,j} + \frac{1}{2s}\left(1 + \frac{\Delta r}{2r_2}\right)\psi_{1,j} \\ \frac{1}{2}\Delta z^2 \mu_0 r_{3,j} \\ \vdots \\ \frac{1}{2}\Delta z^2 \mu_0 r_{nr-2,j} \\ \frac{1}{2}\Delta z^2 \mu_0 r_{nr-1,j} + \frac{1}{2s}\left(1 + \frac{\Delta r}{2r_{nr-1}}\right)\psi_{nr,j} \end{pmatrix}$$

で表示されるベクトルと行列です。

いま中央のメッシュから出発して, (15)式の ψ_{j-1} を ψ_{j-2} と ψ_j で, ψ_{j+1} を ψ_{j+2} と ψ_j で表示し整理すると $\psi_{j-2}, \psi_j, \psi_{j+2}$ からなる式が得られます。これはさらに $\psi_{j-4}, \psi_j, \psi_{j+4}$ で表され, という操作を中央のメッシュから出発して繰り返すと, 最終的には両端のメッシュと中央メッシュの関係に帰着します。ちょうど両端のメッシュに到達するには $nz = 2l + 1$ の関係が必要です。いったん, 中央メッシュでの値が求まれば, 中央メッシュと両端メッシュの中央のメッシュの値が求まり, という形ですべてのメッシュの値が求まります。BunemanのPois-

$$\psi_{i,j-1} + \left[\frac{1}{s} \left(1 + \frac{\Delta r}{2r_i} \right) \psi_{i-1,j} - 2 \left(1 + \frac{1}{s} \right) \psi_{i,j} + \frac{1}{s} \left(1 - \frac{\Delta r}{2r_i} \right) \psi_{i+1,j} \right] + \psi_{i,j+1} = -\Delta z^2 \mu_0 r_i J_{i,j}$$

上式を $i=2, \dots, nr-1$ に対して作成し, $i=2$ に対する $i=1$ の項と, $i=nr-1$ に対する $i=nr$ の項は右辺のソース項に繰り込みます。そうすると最終的に次式が得られます。

$$\psi_{j-1} + A \psi_j + \psi_{j+1} = -2Q_j \quad (15)$$

ここで ψ_j, A, Q_j は

$$\psi_j = (\psi_{2,j}, \psi_{3,j}, \dots, \psi_{nr-1,j})$$

son solverでは端のメッシュの磁束の値が必要です。これはメッシュ点の数がそれほど多くはないので, グリーン関数を用いて求めればよいでしょう。

2.5 プログラミング

2.5.1 プログラム概要

1章での解説に基づいて平衡解析プログラムを実際につけてみます。ただし, プログラムの詳細を説明することは本講座の紙数では不可能ですから, 要点と主要なSubroutineの機能の説明のみを行います。例題で作成したプログラムの機能はかなり単純で制限されたものです。したがって, 本プログラムで縦横無尽にトカマク炉の設計を行うには少し無理があります。また例題のプログラムは以下のような機能制限があります。

- (1) 一回の実行で一つの平衡計算のみが可能。複数の計算が可能のように改良することはそれほど困難ではないはずです。
- (2) 平衡コイル数に制限はないが, 個々のコイルは独立な

電流を有しており、複数のコイルを直列接続して一つのコイルブロックにまとめることはできない。

- (3)使用できるプラズマ電流分布は中心ピークのものだけで、ホロー電流分布を有するプラズマ平衡は計算できない。
- (4)リミタプラズマとダイバータプラズマが計算できるが、ダイバータプラズマについては下方にヌル点を有するシングルヌルダイバータのみ計算可能である。ヌル点位置は外部入力する。炉設計でよく行われているセパトリックスのダイバータ板上のストライク点を指定した計算はできない。
- (5)プラズマ電流値は設定するが、ポロイダルベータ値、内部インダクタンス、磁気軸での安全係数の値は、(12)式のパラメータで間接的に指定できるのみである。プラズマ電流分布を決定する収束ループの外に、ポロイダルベータ値、内部インダクタンス等を収束させるループを付加することによって、上記パラメータについても設定値に一致するようにプログラムを改良することはそれほど難しくはないでしょう。

プログラムのフローチャートを Fig. 2 に示します。プログラムは容易に改造できるように、できるだけ整合配列を使用しています。また整合配列のサイズは入力データから自動的に決まるようになっています。ソースプログラムの作成や改造にはエディタが必要です。間違っても英数字を全角で入力しないようにしてください。日本語ワープロソフトを使ってプログラミングしたため、すべて全角になってしまい、コンパイルエラーの原因をつきとめるのに苦労したという、笑えない実話もあります。英語版のエディタの方がこのような間違いを冒さないのでも使いやすいかも知れません。簡単に主要な Subroutine の説明をしていきましょう。

main	整合配列の大元となる配列のサイズを宣言
sbmain	プログラムの全体の流れを制御
preinp	整合配列のサイズを自動決定するために必要なデータの読み込み
setpar	整合配列のサイズを自動決定するために必要な数値や、平衡計算に使用する最大繰り返し数等のパラメータを設定
adrset	整合配列のサイズの自動決定
input	平衡計算に必要なデータ読み込み。読み込むデータは preinp で読み込んだものと同じものです。rewind 文を使用してデータを最初から読み直しています。
output	入力データの出力

pshape	プラズマ形状の設定
grntab	グリーン関数テーブルの作成
setini	プラズマ電流分布、コイル電流値の初期値設定
getsur	プラズマ表面を求める
getcur	最小自乗法を用いてコイル電流値を算出
gauss	Gauss 消去法により連立一次方程式を解く
calpsi	グリーン関数を用いてプラズマ電流による磁束の計算
calcur	磁束からプラズマ電流密度を算出
siconv	収束の判定
search	プラズマ表面および内部の磁気面座標の計算
eqpara	平衡諸量の計算
pltout	プロット出力(Postscript ファイルが作成される)

整合配列は main で宣言した一つの固定長配列を隙間なく連続して使用するようになっています。最近の CPU はメモリからのデータの読み込みは、連続した塊で一挙に行うようになっていますから、プログラミングに際しては、配列をできるだけ連続して読み書きするようにして計算時間短縮を図っています。

プロット出力は、昔懐かしい Calcomp の命令を使用しています。ただし名前は同じでも、実際は Postscript ファイルを作成するようになっています。したがって、Ghostscript や Ghostview を用いてディスプレイ上に描画することができます。Postscript プリンタがつながっていれば、Send PS などでもプリンタ出力が可能です。

プログラムのソースコードは Macintosh だけではなく、MS Windows, AIX 等の UNIX や Linux でも修正なしで使用できるように作成されています。

2.5.2 コンパイルとリンク

ソースプログラムが完成したら、いよいよコンパイルとリンクを行って実行ファイルを作成します。コンパイラは Absoft Pro Fortran を用います。プログラムのコンパイルやリンクの方法は、ソフトウェアによって少しずつ異なります。幸いなことに Macintosh では Absoft Pro Fortran しか現在は市販されていないため、以下の説明がほぼ普遍的な説明になります。MS Windows 環境では Fujitsu Fortran, Lahay Fortran, Visual Fortran, Watcom Fortran 等、多数のフォートランコンパイラがありますが、これらは IDE (Integrated Development Environment) という非常にグラフィカルな環境で行っています。ここでは IDE の説明はしませんので、興味のある方は独習してみてください。なお以下の方法は MS Win-

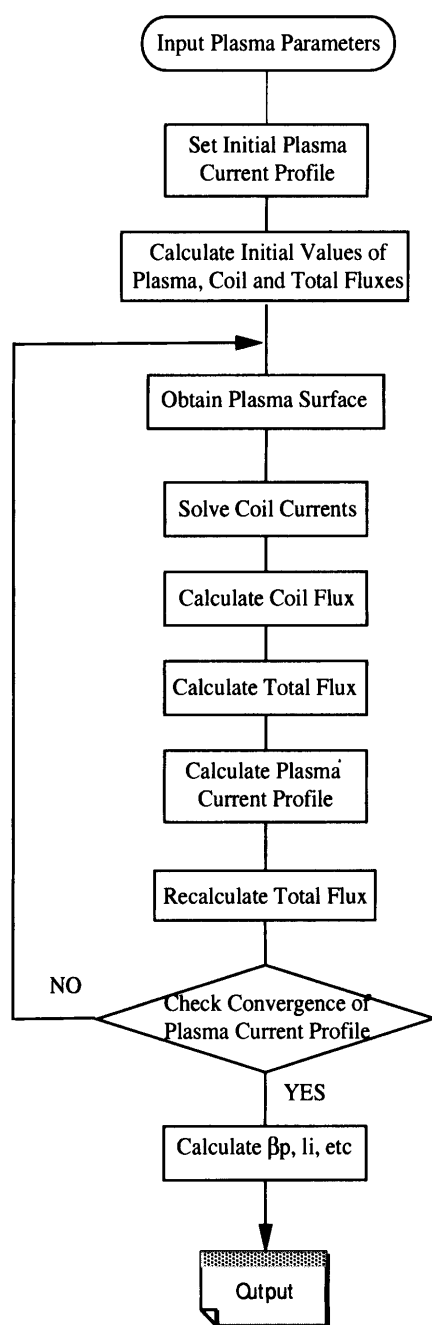


Fig. 2 Flow chart of the equilibrium analysis code made as an example.

dows用のAbsoft ProFortranを使用する場合でも基本的には同じです。

Absoft Pro Fortran はインストールすると MPW という名前のフォルダにまとめられています。MPW は Macintosh Programmer's Workshop の頭文字を取って名づけられたものです。このフォルダの中の MPW Shell というアプリケーションがコンパイルやリンクを行って

実行ファイルを作成するための開発環境（コマンドシェル）になっています。UNIX におけるシェルと同じものと考えていいでしょう。MPW Shell を立ち上げると Worksheet が現れます。MPW に堪能な方はこの上でコマンドを打ち込んで作業できますが、今回はコマンドを知らなくてもできる方法でコンパイルとリンクを行います。手順は以下のとおりです。

- (1)メニューで Directory をプルダウンして Set Directory を選びます。ダイアログボックスでソースプログラムが入っているフォルダを開き、Select Current Directory をクリックします。この操作はソースプログラムが入っているフォルダを選択して、右側の Directory をクリックしても結果は同じです。もう一度メニューで Directory をプルダウンして、ソースプログラムが入っているフォルダが一覧に入っていることを確認しましょう。
- (2)メニューで Tools をプルダウンして Create Build Commands... をクリックします。ACM Options というダイアログボックスが現れます。ここで Source Files のボタンを押すと、ソースファイルを選択するダイアログボックスが出てきますからそこでソースファイルを選択して下さい。ACM Options に戻って、Output filename に作成したい実行ファイルの名前を入力します。この欄はデフォルトでは link.out となっています。次に F77 options... のボタンを押して F77 options... のダイアログボックスを出します。ここではさらに Compiler Control..., Optimizations..., Compatibility... の3つのボタンがありますが、その中の option については皆さんでいろいろと試してみてください。ただし、Optimizations... で必ず PowerPC 604/G3 Optimizations には必ずチェックして下さい。またプログラムを小文字で書いたり、大文字と小文字を混ざって書いている場合には、Compatibility... で Fold to Uppercase のチェックをしたほうがよいでしょう。

デバッグ段階ではプログラムがよく暴走することがあります。このような場合は強制的にプログラムを終了したいことが起こります。Compiler Control... でさらに Miscellaneous をクリックして、Frequent Command + period check をチェックしておいたほうがよいでしょう。最後に再び ACM Options に戻って MRWE Application がチェックされていることを確認します。このオプションは、作成した実行ファイルを通常の Macintosh のアプリケーションのように動作させるためのものです。最後に ACM ボタンをクリックすると

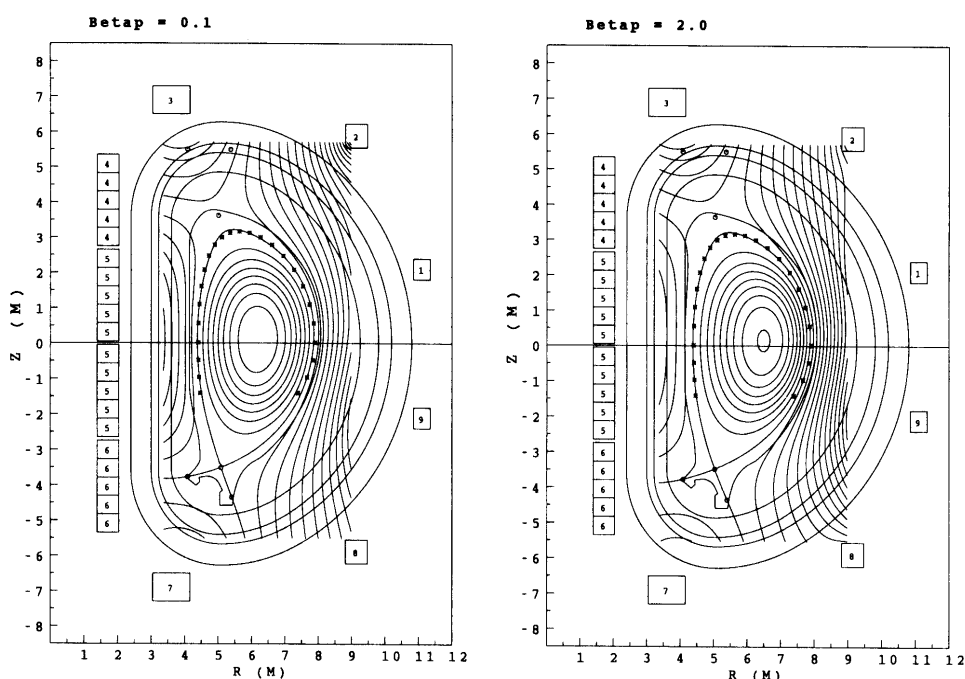


Fig. 3 An example of the equilibrium configurations with different values of poloidal beta calculated by TOSCA. Although the plasma shape is unchanged, the magnetic axis and the associated magnetic surfaces are shifted outward in the higher poloidal beta plasma.

makefile が作成されます。Makefile の名前は Output filename で指定した名前に、make がついたものになります。Set Directory で指定したフォルダ内に作成されているはずです。

(3)(2)で作成した makefile を実行するには、再び MPW Shell に戻って Tools メニューから Build... を選択します。Program Name ? と聞いてくるので、makefile を作成する際に Output filename で指定した名前を入れて OK をクリックすればコンパイルとリンクが実行されます。makefile を作成する際に Compiler Control... で Procedure Trace がチェックされていれば、実行の様子は MPW Shell 上に表示されます。最後に Done のメッセージが表示されれば無事実行ファイルが作成されています。正常に終了しなかった場合はエラーメッセージが表示されますので、それによってプログラムを修正しましょう。

2.5.3 実行

実行は通常の Macintosh のアプリケーションと同様に、作成した実行ファイルのアイコンをダブルクリックして行います。そうすると入力ファイル名を入力するダイアログボックスが現れますから、ファイル名を入力するかリストの中から選択するかして下さい。ただしこの入力ダイアログボックスは、プログラム内の open 文でファイル名を空白にした場合にのみ現れます。ファイ

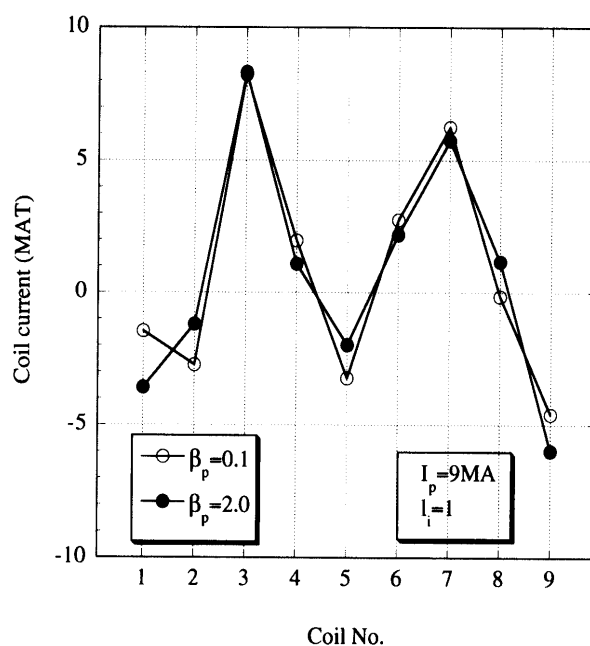


Fig. 4 Comparison of coil current distribution between different poloidal beta plasmas.

ル名を明示すると現れません。

ダイアログボックスが現れる場合

```
OPEN(UNIT=5, FILE="", STATUS="OLD", FILE-
TYPE="TEXT")
```

ダイアログボックスが現れない場合

```
OPEN(UNIT=5,FILE="test.dat",STATUS="OLD",FI-
      LETYPE="TEXT")
```

次に出力ファイル名を入力するダイアログボックスが現れます。このダイアログボックスも入力ファイルの場合と同様に open 文でファイル名を明示すると現れません。実行を開始するとファイル名に output と付け加えられたウインドウが開きます。出力ファイルを open しない場合は、このウインドウ内に結果が出力されます。実行時のエラーをデバッグする場合はこちらの方が便利かも知れませんが、プログラムの実行が正常に終了するとこのウインドウは消えます。

プログラムを実行して作成されたテキストファイルはエディタで内容を見ることができます。プロッタ出力に対応した Postscript ファイルは Postscript の命令を書いたファイルなので、Postscript の文法を知らない人は何が書いてあるかわからないかも知れません。この内容を表示するには Ghostscript または Ghostview を使用します。Ghostscript または Ghostview のアイコンに Postscript ファイルをドラッグアンドドロップして下さい。Postscript プリンタへの出力は Drop PS 等のユーティリティを使用します。Ghostscript の Ghostscript メニューで ps2pdf... を使用すれば PDF ファイルが作成できます。

無事計算できたでしょうか。正常に計算できた場合は Fig. 3 のようなプラズマ形状の出力が得られるはずです。また Fig. 4 はコイル電流値をプロットしたものです。ただし、Fig. 3 と Fig. 4 は TOSCA を用いて計算結果なので、今回の例題プログラムの結果とは少し異なっています。プラズマ電流値は 9 MA です。内部インダクタンスは 1 としたので、プラズマ電流分布はほぼ放物型です。ポロイダルベータ値でコイル電流分布がどのように異なるかがわかります。この講座の目的は解析結果を議論するものではありませんので、異なる理由は自分で考えてみてください。データを変えていろいろなトカマクプラズマの平衡計算ができます。入力データの簡単な説明はデータファイルに書き込んであります。参考にしてください。

2.6 まとめ

最近のパーソナルコンピュータの性能向上を最大限利用するために、パーソナルコンピュータを用いたプラズマ解析を試みてみました。一例として、Macintosh 上でト

カマクプラズマの平衡解析を行ってみました。しかしソースプログラムは MS Windows や UNIX でも使用できるように作成してありますので、手慣れた環境で実行することがよいと思います。

平衡解析に馴染みのない人のためにグラッド-シャフラーノフ方程式の導出から、実際のプログラム作成、実行までひととおりの説明を行いました。紙数の制限から省略せざるを得なかったところが多々あります。1999年12月号から本学会誌に連載された講座「核融合装置におけるプラズマ制御」には多くの平衡解析に関する記述があります。参考にされるとよいでしょう。

実際に例題で作成したプログラムを実行してみると、最近のパーソナルコンピュータが強力な解析ツールであることが実感できます。大いに多方面での解析に応用したいものです。例題で作成したプログラムは、実際に炉設計等に使用するには機能不足です。練習問題としてどんどん機能を拡張すれば、皆さん独自のプログラムが作成できるでしょう。拡張しやすいようには作成したつもりです。ただし、皆さんが拡張したプログラムを再配布する場合は、くれぐれも GNU (General Public License) に従うことを忘れないでください。

参考文献

- [1] M.S. Chu, D. Dobrot, T.H. Jensen and T. Tamano, *Phys. Fluids* **17**, 1183 (1974).
- [2] F.J. Helton, T.S. Wang, *Nucl. Fusion* **18**, 1523 (1978).
- [3] F.W. McClain and B.B. Brown, GAQ, A Computer Program to Find and Analyze Axisymmetric MHD Plasma Equilibria, GA-A 14490 (1977).
- [4] 二宮博正, 亀有昭久, 新谷吉郎: トロイダルプラズマの MHD 平衡計算コード[EQUICIR version 1], JAERI-M 9127 (1980).
- [5] 新谷吉郎, 西尾 敏: トロイダルプラズマの MHD 平衡計算コード[EQUICIR version 3] JAERI-M 87-133 (1987).
- [6] V.D. Shafranov, *Soviet Physics JETP* **37**, 775 (1960).
- [7] O. Buneman, A Compact Non-Iterative Poisson Solver, SUIPR Report No. 294 (1969).
- [8] B.L. Buzbee, G.H. Golub and C.W. Nielson, *SIAM J. Numer. Anal.* **7**, 627 (1970).
- [9] D.M. Burrige and C. Temperton, *J. Comp. Phys.* **30**, 145 (1979).
- [10] C. Temperton, *J. Comp. Phys.* **32**, 1 (1979).
- [11] R.W. Hockney, *Methods of Computational Physics* (Academic Press, New York, 1969) Vol. 9, p. 135.