

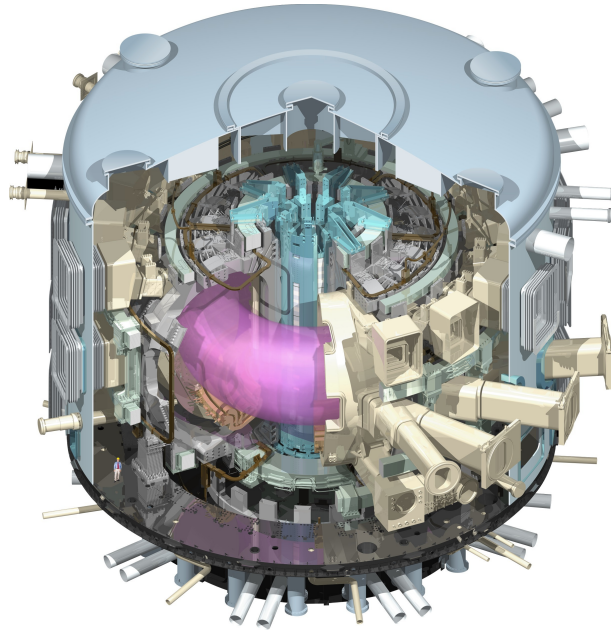
10401

Fusion Energy and Plasma Physics

Rasmus Kronborg Finnemann Wiuff (s163977)* and Nicklas Kihm (s143286)[†]

Technical University of Denmark[‡]

(Dated: January 24th 2019)



Abstract: Abstract

CONTENTS

I. Intro	1
II. Assignment 1: A simple reactor model	2
A. Freidberg's simple reactor model	2
B. Model sensitivity	4
C. Elliptic Cross section	4
Acknowledgments	5
References	6
List of Figures	7
List of Tables	7
Listings	7
Appendices	8
A. tokamakDTU_asign_1	8
B. IterateTokamakDTU	18

I. INTRO

In recent years a large and quickly growing collaboration between plasma physcists and engineers have materialised in one of the most ambitious energy producing projects ever seen. The proof-of-concept plasma fusion tokamak reactor ITER¹ in Cadarache is currently taking shape in order to adress the issue of growing energy demands and climate changes.

The mission is simple: Prove that plasma fusion is a viable source of electricity.

Whilst not being the most surmountable task, many researchers and institutions have gathered from across the world, including the Department of Physics at DTU.

In this paper, three assignments are solved as part of the course “10401 Fusion Energy and Fusion Plasma Physics”. Some key aspects of fusion plasma fueled reactors are adresssed and discussed in the assignments.

Symbol	Quantity
b	Blanket-and-shield thickness [m]
c	Magnet coil thickness [m]
a	Minor radius [m]
R_0	Major radius [m]
A	Aspect ratio []
A_p	Plasma surface [m ²]
V_p	Plasma volume [m ³]
P_{dens}	Power density [W·m ⁻¹]
p	Plasma pressure [Pa]
n	Particle density [m ⁻³]
B_0	Magnetic field at magnetic axis [T]
β	Normalised plasma pressure []
$\tau_{E_{\text{min}}}$	Min confinement time for satisfaction of $(p \times \tau_E)_{\text{min}}$ [s]
$C_{\text{per watt}}$	The cost of the powerplant [\$]

Table I: Quantities calculated by the model in Freidberg's².

II. ASSIGNMENT 1: A SIMPLE REACTOR MODEL

A. Freidberg's simple reactor model

In the 5th chapter of the textbook by Freidberg², he makes a simple model for designing a fusion reactor power plant. Table I shows a list of quantities necessary for the construction of a power plant. The model finds these quantities using simple geometric and electromagnetic assumptions with little involvement of plasma physics. The variables put into the model are shown in Table II. This model has been implemented in a matlab script provided for the course. The script is shown included in Appendix A. As an example, the model is run with the following parameters:

$$\begin{aligned}
 n_{\text{flux fraction}} &= 0.01 & P_E &= 1000 \text{ MW} \\
 P_W &= 4 \text{ MW} \cdot \text{m}^{-2} & B_{\text{max}} &= 13 \text{ T} \\
 \sigma_{\text{max}} &= 3000 \text{ atm} & \eta_t &= 0.4
 \end{aligned} \tag{1}$$

Symbol	Quantity
$n_{\text{flux fraction}}$	n flux in breeder end/n flux in breeder start []
C_F	Fixed cost propotionality constant [\$]
C_I	Nuclear island cost propotionality constant [$\$ \cdot \text{W} \cdot \text{m}^{-3}$]
P_E	Desired output power [MW]
P_W	Maximum wall load [$\text{MW} \cdot \text{m}^{-2}$]
B_{max}	Magnetic field at the edge of the coil [T]
σ_{max}	Tensile strenght of the magnetic field coils [atm]
η_t	Energy conversion efficiency []

Table II: Variables in the Freidberg's model

Symbol	Quantity
b	1.20 m
c	0.799 m
a	2.01 m
R_0	4.96 m
A	2.4670
A_p	393 m ²
V_p	395 m ³
P_{dens}	$4.97 \times 10^6 \text{ W} \cdot \text{m}^{-1}$
p	$7.37 \times 10^5 \text{ Pa}$
n	$1.53 \times 10^{20} \text{ m}^{-3}$
B_0	4.57 T
β	8.85%
$\tau_{E_{\text{min}}}$	1.14 s

Table III: Obtained parameters by running Freidberg's model.

Note that C_F and C_I has been ommitied as these serve no purpose for this assignment. It is not of interest how expensive the plant will be. Rather the geometries and physical quantities are of interest. The results from the model is given in Table [III](#).

B. Model sensitivity

At this point Freidberg has provided a model that produce some reasonable results for a powerplant. It could be interesting to see how this model behaves when some vital parameters are changed. In the last section the model used the parameters shown in Eq. (1). Now the model will be iterated over variations in the following parameters, while retaining the rest. The variable parameters are the electric power P_E , the maximum wall loading P_W , the maximum magnetic field B_{\max} and the maximum stress σ_{\max} . Appendix B includes the code for iterating the matlab model over various parameters.

C. Elliptic Cross section

Freidberg's model assumes a circular cross section of the plasma. In reality this is not the case, and as of such we will now make a more realistic, yet still approximate reactor for an elliptic plasma cross section. In describing the geometry one refers to the elongation ratio:

$$\kappa = \frac{a_{\max}}{a_{\min}} \quad (2)$$

This parameter ensure a true elliptic cross section as defined by the equation,

$$\frac{x^2}{a_{\max}^2} + \frac{y^2}{a_{\min}^2} = 1 \quad (3)$$

, even after increasing the radii to accomodate the larger ellipses describing the blanket-and-shield. This does however mean that the fixed thickness, b from Freidberg's model is variable. To find this variation the parametric ellipses defining the blanket-and-shield are defined with b being the length difference between the larger radii and the length difference between the smaller radii

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} a_{\min} \cos \phi \\ \kappa a_{\min} \sin \phi \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} (b + a_{\min}) \cos \phi \\ \kappa (b + a_{\min}) \sin \phi \end{bmatrix} \quad (5)$$

At a given angle, ϕ , the distance must be:

$$\mathbf{u} - \mathbf{v} = \begin{bmatrix} \cos \phi b \\ \kappa \sin \phi b \end{bmatrix} \quad (6)$$

$$|\mathbf{u} - \mathbf{v}| = \sqrt{\cos^2 \phi b^2 + \kappa^2 \sin^2 \phi b^2} \quad (7)$$

An example is given in Fig. 1. Here the parameters are:

$$\begin{aligned} a_{\min} &= 2 \\ \kappa &= 2 \\ b &= 1.2 \end{aligned} \tag{8}$$

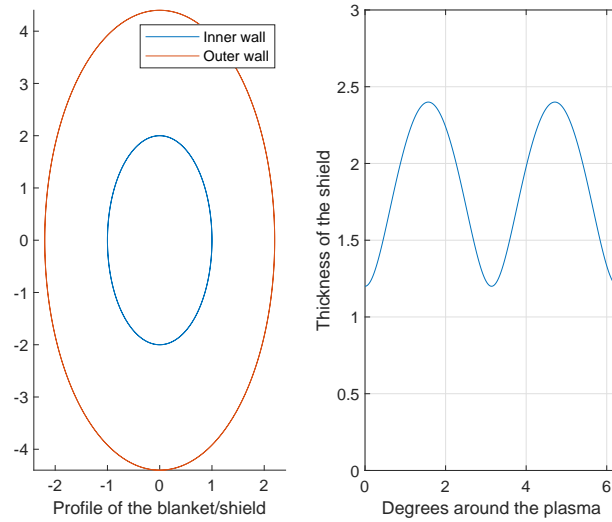


Figure 1: The profile of the blanket-and-shield and the thickness as a function of the angle around origo, ϕ . The parameters here are $a_{\min} = 2$, $\kappa = 2$ and $b = 1.2$

ACKNOWLEDGMENTS

The authors would like to thank...

* E-mail at s163977@student.dtu.dk

† E-mail at spacrone@live.dk

‡ Homepage of the Technical University of Denmark <http://www.dtu.dk/english/>

¹ <https://www.iter.org/>.

² Jeffrey P. Freidberg, *Plasma physics and fusion energy* (Cambridge University Press, 2007).

LIST OF FIGURES

- 1 The profile of the blanket-and-shield and the thickness as a function of the angle around origo, ϕ . The parameters here are $a_{\min} = 2$, $\kappa = 2$ and $b = 1.2$ 5

LIST OF TABLES

- I Quantities calculated by the model in Freidberg's²..... 2
II Variables in the Freidberg's model 3
III Obtained parameters by running Freidberg's model. 3

LISTINGS

Appendices

Appendix A: tokamakDTU_asign_1

```
1  %*****
2  % Name:          tokamakDTU
3  %
4  % Version:       1.0
5  %
6  % Purpose:       Contains the function 'tokamakDTU' which gives parameters
7  %                for a tokamak fusion power plant as output based on a
8  %                simplified model. The equations used are derived in
9  %                chapter 5 in Friedberg, Plasma physics and Fusion
10 %                Energy, 2007 (all references are referring thereto).
11 %
12 % To do (NOT for 10401 - Fusion Energi students):
13 %               1. Rewrite the code to a class (this is not done on purpose so
14 %               that the code is more readable for students not familiar
15 %               with classes). Can from that merge the files which takes
16 %               R_0/a and/or the ellipticity as an input into one file.
17 %
18 % Changelog:
19 %               1. December 2014:
20 %               Written by Michael Løiten based on a similar code written
21 %               as a bachelor project by Elias Pagh Sentius
22 %               mailto: mmag@fysik.dtu.dk
23 %*****
24
25 function [b, c, a, R_0, A, A_p, V_p, P_dens, p, n, B_0, beta, tau_E_min,...
26          C_per_watt] =...
27          tokamakDTU_asign_1(...
28          n_flux_fraction, C_F, C_I, P_E, P_W, B_max, sigma_max, eta_t)
```

```

29 %TOKAMAK_DTU Function which returns the parameters of a power plant
30 %
31 % Output parameters
32 %-----
33 % b          - Blanket/shield thickness [m]
34 % c          - Magnet coil thickness [m]
35 % a          - Minor radius [m]
36 % R_0        - Major radius [m]
37 % A          - Aspect ratio []
38 % A_p        - Plasma surface [m^2]
39 % V_p        - Plasma volume [m^3]
40 % P_dens     - Power density [W/m]
41 % p          - Plasma pressure [Pa]
42 % n          - Particle density [m^-3]
43 % B_0        - Magnetic field at magnetic axis [T]
44 % beta       - Plasma beta in the centre []
45 % tau_E_min  - Min confinement time for satisfaction of (p*tau_E)_min [s]
46 % C_per_watt - The cost of the powerplant [$]
47 %
48 % Input parameters
49 %-----
50 % n_flux_fraction - n flux in breeder end/n flux in breeder start []
51 % C_F           - Fixed cost propotionality constant [$]
52 % C_I           - Nuclear island cost propotionality constant [$W/m^3]
53 % P_E           - Desired output power [MW]
54 % P_W           - Maximum wall load [MW/m^2]
55 % B_max         - Magnetic field at the edge of the coil [T]
56 % sigma_max     - Tensile strenght of the magnetic field coils [atm]
57 % eta_t         - Energy conversion efficiency []
58
59
60 % The function starts by defining fixed constants

```

```

61 % Note that this is inefficient if we are looping over the function, but it
62 % makes the code easier to use, as these are not needed as input parameters
63
64 % Fixed constants
65 %#####
66 % Nuclear
67 %-----
68 % Energies
69 E_t      = 2.5e-8; % [MeV] Energy of slow (thermal) neutron (eq 5.6)
70 E_n      = 14.1;  % [MeV] Neutron energy after fusion (eq 2.17)
71 E_a      = 3.5;   % [MeV] alpha energy after fusion (eq 2.17)
72 E_Li     = 4.8;   % [MeV] Heat produced by breeding Li (under eq 4.31)
73 % Cross section and main free paths
74 sigma_v_avg = 3.0e-22; % [m^3/s] DT fusion cross section @ 15keV (table 5.2)
75 lambda_br   = 0.0031; % [m] Breeding mean free path (under eq 5.7)
76 lambda_sd   = 0.055;  % [m] Mean free path from sigma_sd (eq 5.3)
77
78 % Plasma physics
79 %-----
80 % Parameters for infinity gain at the minimum of p tau_E (eq 4.20)
81 T          = 15.0;  % [keV] Temperature for obtaining min tripple product
82 tripple_min = 8.3;  % [atm s] Min tripple prod to obtain Q=inf @ T=15 keV
83
84 % Natural constants
85 %-----
86 mu_0 = 4.0*pi*1e-7; % Vacuum permeability [T*m/A]
87 e    = 1.602176565e-19; % Elementary charge [C]
88 %#####
89
90
91 % Secondly we convert everything to SI units, so that the variables are
92 % easier to handle

```

```

93 % Again, this is computationally inefficient, but it suffices for our use
94 % Conversion to SI-units
95 %#####
96 % Conversion factors
97 W_per_MW      = 1.0e6;
98 Pa_per_atm    = 1.01325e5;
99 eV_per_keV    = 1.0e3;
100 eV_per_MeV    = 1.0e6;
101 J_per_eV      = e;
102 J_per_keV     = J_per_eV * eV_per_keV;
103 J_per_MeV     = J_per_eV * eV_per_MeV;
104 % Conversions
105 P_E           = P_E * W_per_MW;           % Desired output power
106 P_W           = P_W * W_per_MW;           % Wall Loading limit on first wall
107 E_t           = E_t * J_per_MeV;           % Energy of slow (thermal) neutron
108 E_n           = E_n * J_per_MeV;           % Neutron energy after fusion
109 E_a           = E_a * J_per_MeV;           % alpha energy after fusion
110 E_Li          = E_Li * J_per_MeV;           % Heat produced by breeding Li
111 sigma_max     = sigma_max * Pa_per_atm;    % Max allowable structural stress
112 T             = T * J_per_keV;             % Temperature for minimum p*tau_E
113 tripple_min   = tripple_min * Pa_per_atm;  % The Lawson parameter (p*tau_E)
114 %#####
115
116
117 % Calculate the geometrical factors
118 %-----
119 % Find the breeder thickness
120 b = get_b(lambda_sd, E_n, E_t, lambda_br, n_flux_fraction);
121 % Find the minor plasma radius and the coil thickness
122 [a, c] = get_a_and_c(B_max, mu_0, sigma_max, b);
123 % Find the major radius
124 R_0 = get_R_0(a, eta_t, E_n, E_a, E_Li, P_E, P_W);

```

```

125 % Find the resulting geometrical factors
126 A = R_0/a; % Aspect Ratio
127 A_p = (2.0*pi*a)*(2.0*pi*R_0); % Plasma surface area
128 V_p = (pi*a^(2.0))*(2.0*pi*R_0); % Plasma volume
129
130
131 % Calculate the plasma physics parameters
132 %-----
133 % Find the power density in the plasma
134 P_dens = get_P_dens(E_a, E_n, E_Li, P_E, eta_t, V_p);
135 % Find the plasma pressure
136 p = get_p(E_a, E_n, P_dens, T, sigma_v_avg);
137 % Calculate the density from the definition of p under eq 5.36
138 n = p/(2.0*T);
139 % Find the magnetic field strength on the magnetic axis
140 B_0 = get_B_0(R_0,a,b,B_max);
141 % Find the plasma beta on the magnetic axis
142 beta = get_beta(p, B_0, mu_0);
143 % Find the minimum required confinement time from the definition of the
144 % minimum tripple product.
145 % NOTE: A higher confinement time is advantageous, and could in principle
146 % yield a smaller (and cheaper) reactor. However, the effect is not
147 % included in this model
148 tau_E_min = tripple_min/p;
149
150
151 % Calculate the cost
152 % (details about the cost can be found in the function get_a_and_c)
153 %-----
154 % Find the volume of the nuclear island
155 % (the material surrounding the plasma)
156 V_I = get_V_I(R_0,a,b,c);

```

```
157 % Find the reactor volume per power out
158 % In the current model, this is the only non-constant in the expression for
159 % cost per watt
160 V_I_per_P_E = V_I/P_E;
161 C_per_watt = get_C_per_watt(C_F, C_I, V_I_per_P_E);
162 end
163
164
165
166 function [b] = get_b(lambda_sd, E_n, E_t, lambda_br, n_flux_fraction)
167 %GET_B Calculates b from the need of slowing down and breeding neutrons
168
169 % Thickness of the moderator-breeding region so that 1 - n_flux_fraction
170 % have slowed down and undergone a breeding reaction
171 % [m]
172 % Equation 5.10
173 delta_x = 2.0*lambda_sd*...
174         log( 1.0-(1.0/2.0)*(E_n/E_t)^(1.0/2.0)*...
175             (lambda_br/lambda_sd)*log( n_flux_fraction )...
176         );
177
178 % Set b from delta_x
179 % Friedberg argues above equation 5.11 that b should be between 1 and 1.5 m
180 % Therefore a self chose constant is set to 0.38
181 self_chosen_constant = 0.32;
182 b = delta_x + self_chosen_constant;
183 end
184
185
186
187
188 function [a,c] = get_a_and_c(B_max, mu_0, sigma_max, b)
```

```

189 %GET_A_AND_C Calculates a and c
190
191 % c is obtained from requiring that the magnets are so thin that they are
192 % on the limit of the tensile strenght
193 % a is obtained from minimizing the costs
194
195 % xi defined when making the magnetic coil c as thin as possible
196 % Under equation 5.27
197 xi = B_max^(2.0) / (4.0*mu_0*sigma_max);
198
199 % a is found from optimization of the cost, where
200 % total cost = fixed cost + nuclear island cost
201 %.....
202 % Fixed cost
203 %.....
204 % K_F = Fixed cost for building, turbines, generators etc (also applies to
205 % fusion, fission, fossil)
206 % Assumption: The fixed cost is proportional to power output:
207 % Equation 5.13
208 % K_F = C_F*P_E;
209 %.....
210 % Nuclear island cost (mainly cost of magnets, blanket and shield)
211 %.....
212 % Assumption: The proportional to reactor volume:
213 % Equation 5.14
214 % K_I = C_I*V_I;
215 % Equation 5.15
216 % V_I = 2.0*pi^(2.0) * R_0 * ( (a+b+c)^(2.0) - a^(2.0) ); % Reactor volume
217 %.....
218 % Cost per watt:
219 %.....
220 % Defined as C_p_watt = (K_F + K_I)/P_E, rewritten to

```

```

221 % C_p_watt = C_F + C_I*(V_I/P_E);
222 % Since the cost per watt contains two constants, we can minimize the
223 % V_I/P_E in order to optimize the cost
224 % Given by equation 5.20 inserted in 5.17
225 % Equation 5.21
226 % V_I_per_P_E = V_I/P_E; % Reactor volume per power out
227 % a is found by setting the derivative of V_I_per_P_E = 0
228 % Equation 5.29
229 a = ((1.0 + xi)/(2.0*xi^(1.0/2.0))) * b;
230
231 % Knowing xi, a, and, b, we can calculate c
232 % c found by comparing tensile force and magnetic force working on the coil
233 % Equation 5.27
234 c = 2*xi/(1-xi)*(a+b);
235 end
236
237
238
239 function [R_0] = get_R_0(a, eta_t, E_n, E_a, E_Li, P_E, P_W)
240 %GET_R_0 Calculate the major radius
241
242 % Divide eq 5.18 (electric power out) by
243 % eq 5.19 (wall loading * area = total neutron production) and solve for R0
244 % Equation 5.20
245 R_0 = (1.0/(4.0*pi^(2.0)*eta_t))*(E_n/(E_n + E_a + E_Li))*(P_E/(a*P_W));
246 end
247
248
249
250 function [P_dens] = get_P_dens(E_a, E_n, E_Li, P_E, eta_t, V_p)
251 %GET_P_DENS Calculate the power density
252

```



```
253 % The power density is found by the sum of the power from the alphas plus
254 % the power from the neutrons, divided by the plasma volume
255 % Equation 5.35
256 P_dens = (E_a + E_n)/(E_a + E_n + E_Li)*P_E/(eta_t*V_p);
257 end
258
259
260
261 function [B_0] = get_B_0(R_0, a, b, B_max)
262 %GET_B_0 Calculte the magnetic field strength on the magnetic axis
263
264 % B_max is found in the edge of the magnet (at R = R_0-a-b)
265 % B0 is the magnetic field at R0
266 % As B propto 1/R. we have that B_0/B_max = (R_0-a-b)/R_0, which leads to
267 % Equation 5.42
268 B_0 = ((R_0-a-b)/R_0)*B_max;
269 end
270
271
272
273 function [beta] = get_beta(p, B_0, mu_0)
274 %GET_beta Calculte the magnetic field strength on the magnetic axis
275
276 % Plasma beta in the center (kinetical pressure over magnetical pressure):
277 % Equation 5.43
278 beta = p / (B_0^2/(2.0*mu_0));
279 end
280
281
282
283 function [p] = get_p(E_a, E_n, P_dens, T, sigma_v_avg)
284 %GET_P Calculate the plasma pressure
```

```
285
286 % Found from solving the sum of neutron and alpha power for n, and multiply
287 % the result with T
288 % Equation 5.37
289 p = ( (16.0/(E_a + E_n)) * P_dens )^(1.0/2.0)*...
290     (T^(2.0)/sigma_v_avg)^(1.0/2.0);
291 end
292
293
294
295 function [V_I] = get_V_I(R_0, a,b,c)
296 %GET_V_I Calculate the volume of the material surrounding the plasma
297
298 % Equation 5.15
299 V_I = 2.0*pi^(2.0) * R_0 * ( (a+b+c)^(2.0) - a^(2.0) );
300 end
301
302
303
304 function [C_per_watt] = get_C_per_watt(C_F, C_I, V_I_per_P_E)
305 %GET_C_PER_WATT Calculates the cost for one watt out from the power plant
306
307 % For details in how the cost is derived, see comments in the function
308 % get_a_and_c
309
310 C_per_watt = C_F + C_I*(V_I_per_P_E);
311 end
```

Appendix B: IterateTokamakDTU

```
1 close;
2 clear;
3
4 titl = ["Desired output power [MW]", "Maximum wall load [MW m^-2]",...
5        "Magnetic field at the edge of the coil [T]",...
6        "Tensile strenght of the magnetic field coils [atm]"];
7 foldertitl = ["DesiredOutputPower", "MaximumWallLoad",...
8              "MagneticFieldAtTheEdgeOfTheCoil",...
9              "TensileStrenghtOfTheMagneticFieldCoils"];
10 l = 5;
11 p1 = [];
12 x = [];
13 mkdir('../MatlabFigures', foldertitl(1))
14
15 for i = 2000:5000
16     [b, c, a, R_0, A, A_p, V_p, P_dens, p,...
17      n, B_0, beta, tau_E_min, C_per_watt] =...
18     tokamakDTU_asign_1(0.01, 1, 2, 1000, 4, 13, 3000, 0.4);
19     q=[b, c, a, R_0, A, A_p, V_p, P_dens, p,...
20        n, B_0, beta, tau_E_min, C_per_watt];
21     p1=cat(1,p1,q);
22     x=cat(1,x,i);
23 end
24
25 T = ["Blanket-shield thickness [m]", "Magnet coil thickness [m]"...
26      "Minor radius [m]", "Major radius [m]", "Aspect ratio []"...
27      "Plasma surface [m^2]", "Plasma volume [m^3]", "Power density [W m^-1]"...
28      "Plasma pressure [Pa]", "Particle density [m^-3]",...
29      "Magnetic field at magnetic axis [T]", "Plasma beta in the centre []"...
30      "Min confinement time for satisfaction of (p tau_E)_min [s]",...
```

```
31     "The cost of the powerplant [$]");
32 fileT = ["Blanket-shieldThickness", "MagnetCoilThickness"...
33     "MinorRadius", "MajorRadius", "AspectRatio"...
34     "PlasmaSurface", "PlasmaVolume", "PowerDensity"...
35     "PlasmaPressure", "ParticleDensity",...
36     "MagneticFieldAtMagneticAxis", "PlasmaBetaInTheCentre"...
37     "MinConfinementTime",...
38     "TheCostOfThePowerplant"];
39 for k = 1:14
40     q = figure;
41     y = p1(:,k);
42     CM = jet(14);
43     plot(x, y, 'color', CM(k,:));
44     ylabel(T(k));
45     xlabel(titl(1));
46     ytickformat('%.2f');
47     epsfilename = sprintf('%s.eps',fileT(k));
48     foldername = sprintf('../MatlabFigures/%s', foldertitl(1));
49     fullfilename = fullfile(foldername,epsfilename);
50     saveas(q, fullfilename, 'epsc')
51 end
```