10401
Fusion Energy and Plasma Physics

Rasmus Kronborg Finnemann Wiuff (s163977)*
Technical University of Denmark†

(Dated: January 24th 2018)

Abstract: Abstract

CONTENTS

I. INTRO

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

ACKNOWLEDGMENTS

The authors would like to thank...

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

† Homepage of the Technical University of Denmark http://www.dtu.dk/english/

LIST OF FIGURES

LIST OF TABLES

LISTINGS

Appendices

Appendix A: tokamakDTU_asign_1

```
%***************************************************************************
% Name:          tokamakDTU
%
% Version:       1.0
%
% Purpose:       Contains the function 'tokamakDTU' which gives parameters
%                for a tokamak fusion power plant as output based on a
%                simplified model. The equations used are derived in
%                chapter 5 in Friedberg, Plasma physics and Fusion
%                Energy, 2007 (all references are referring thereto).
%
% To do (NOT for 10401 - Fusion Energi students):
%             1. Rewrite the code to a class (this is not done on purpose so
%                that the code is more readable for students not familiar
%                with classes). Can from that merge the files which takes
%                R_0/a and/or the ellipticity as an input into one file.
%
% Changelog:
%             1. December 2014:
%                Written by Michael Løiten based on a similar code written
%                as a bachelor project by Elias Pagh Sentius
%                mailto: mmag@fysik.dtu.dk
%***************************************************************************

function [b, c, a, R_0, A, A_p, V_p, P_dens, p, n, B_0, beta, tau_E_min,...
          C_per_watt] =...
    tokamakDTU_asign_1(...
        n_flux_fraction, C_F, C_I, P_E, P_W, B_max, sigma_max, eta_t)
```

```
%TOKAMAK_DTU Function which returns the parameters of a power plant
%
% Output parameters
%------------------
% b           - Blanket/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]
% p           - Plasma pressure [Pa]
% n           - Particle density [m^-3]
% B_0         - Magnetic field at magnetic axis [T]
% beta        - Plasma beta in the centre []
% tau_E_min   - Min confinement time for satisfaction of (p*tau_E)_min [s]
% C_per_watt - The cost of the powerplant [$]
%
% Input parameters
%-------------------
% n_flux_fraction - n flux in breeder end/n flux in breeder start []
% C_F             - Fixed cost propotionality constant [$]
% C_I             - Nuclear island cost propotionality constant [$W/m^3]
% P_E             - Desired output power [MW]
% P_W             - Maximum wall load [MW/m^2]
% B_max           - Magnetic field at the edge of the coil [T]
% sigma_max       - Tensile strenght of the magnetic field coils [atm]
% eta_t           - Energy conversion efficiency []


% The function starts by defining fixed constants
```

```
% Note that this is inefficient if we are looping over the function, but it
% makes the code easier to use, as these are not needed as input parameters

% Fixed constants
%########################################################################
% Nuclear
%--------
% Energies
E_t         = 2.5e-8; % [MeV] Energy of slow (thermal) neutron (eq 5.6)
E_n         = 14.1;   % [MeV] Neutron energy after fusion (eq 2.17)
E_a         = 3.5;    % [MeV] alpha energy after fusion (eq 2.17)
E_Li        = 4.8;    % [MeV] Heat produced by breeding Li (under eq 4.31)
% Cross section and main free paths
sigma_v_avg = 3.0e-22;% [m^3/s] DT fusion cross section @ 15keV (table 5.2)
lambda_br   = 0.0031; % [m] Breeding mean free path (under eq 5.7)
lambda_sd   = 0.055;  % [m] Mean free path from sigma_sd (eq 5.3)

% Plamsa physics
%--------------
% Parameters for infinity gain at the minimum of p tau_E (eq 4.20)
T            = 15.0;  % [keV] Temparature for obtaining min tripple product
tripple_min  = 8.3;   % [atm s] Min tripple prod to obtain Q=inf @ T=15 keV

% Natural constants
%-----------------
mu_0 = 4.0*pi*1e-7;        % Vacuum permeability [T*m/A]
e    = 1.602176565e-19;    % Elementary charge [C]
%########################################################################



% Secondly we convert everything to SI units, so that the variables are
% easier to handle
```

```
% Again, this is computationally inefficient, but it suffices for our use
% Conversion to SI-units
%##########################################################################
% Conversion factors
W_per_MW     = 1.0e6;
Pa_per_atm   = 1.01325e5;
eV_per_keV   = 1.0e3;
eV_per_MeV   = 1.0e6;
J_per_eV     = e;
J_per_keV    = J_per_eV * eV_per_keV;
J_per_MeV    = J_per_eV * eV_per_MeV;
% Conversions
P_E          = P_E * W_per_MW;          % Desired output power
P_W          = P_W * W_per_MW;          % Wall Loading limit on first wall
E_t          = E_t * J_per_MeV;         % Energy of slow (thermal) neutron
E_n          = E_n * J_per_MeV;         % Neutron energy after fusion
E_a          = E_a * J_per_MeV;         % alpha energy after fusion
E_Li         = E_Li * J_per_MeV;        % Heat produced by breeding Li
sigma_max    = sigma_max * Pa_per_atm;  % Max allowable structural stress
T            = T * J_per_keV;           % Temparature for minimum p*tau_E
tripple_min  = tripple_min * Pa_per_atm;  % The Lawson parameter (p*tau_E)
%##########################################################################


% Calculate the geometrical factors
%----------------------------------
% Find the breeder thickness
b = get_b(lambda_sd, E_n, E_t, lambda_br, n_flux_fraction);
% Find the minor plasma radius and the coil thickness
[a, c] = get_a_and_c(B_max, mu_0, sigma_max, b);
% Find the major radius
R_0 = get_R_0(a, eta_t, E_n, E_a, E_Li, P_E, P_W);
```

```
% Find the resulting geometrical factors
A   = R_0/a;                        % Aspect Ratio
A_p = (2.0*pi*a)*(2.0*pi*R_0);     % Plasma surface area
V_p = (pi*a^(2.0))*(2.0*pi*R_0);   % Plasma volume


% Calculate the plasma physics parameters
%---------------------------------------
% Find the power density in the plasma
P_dens = get_P_dens(E_a, E_n, E_Li, P_E, eta_t, V_p);
% Find the plasma pressure
p = get_p(E_a, E_n, P_dens, T, sigma_v_avg);
% Calculate the density from the definition of p under eq 5.36
n = p/(2.0*T);
% Find the magnetic field strength on the magnetic axis
B_0 = get_B_0(R_0,a,b,B_max);
% Find the plasma beta on the magnetic axis
beta = get_beta(p, B_0, mu_0);
% Find the minimum reqiured confinement time from the definition of the
% minimum tripple product.
% NOTE: A higher confinement time is advantegous, and could in principle
% yield a smaller (and cheaper) reactor. However, the effect is not
% included in this model
tau_E_min = tripple_min/p;


% Calculate the cost
% (details about the cost can be found in the function get_a_and_c)
%-------------------
% Find the volume of the nuclear island
% (the material surrounding the plasma)
V_I = get_V_I(R_0,a,b,c);
```

```
% Find the reactor volume per power out
% In the current model, this is the only non-constant in the expression for
% cost per watt
V_I_per_P_E = V_I/P_E;
C_per_watt = get_C_per_watt(C_F, C_I, V_I_per_P_E);
end



function [b] = get_b(lambda_sd, E_n, E_t, lambda_br, n_flux_fraction)
%GET_B Calculates b from the need of slowing down and breeding neutrons

% Thickness of the moderator-breeding region so that 1 - n_flux_fraction
% have slowed down and undergone a breeding reaction
% [m]
% Equation 5.10
delta_x = 2.0*lambda_sd*...
          log( 1.0-(1.0/2.0)*(E_n/E_t)^(1.0/2.0)*...
               (lambda_br/lambda_sd)*log( n_flux_fraction )...
              );

% Set b from delta_x
% Friedberg argues above equation 5.11 that b should be between 1 and 1.5 m
% Therefore a self chose constant is set to 0.38
self_chosen_constant = 0.32;
b = delta_x + self_chosen_constant;
end




function [a,c] = get_a_and_c(B_max, mu_0, sigma_max, b)
```

```
%GET_A_AND_C Calculates a and c

% c is obtained from requiring that the magnets are so thin that they are
% on the limit of the tensile strenght
% a is obtained from minimizing the costs

% xi defined when making the magnetic coil c as thin as possible
% Under equation 5.27
xi = B_max^(2.0) / (4.0*mu_0*sigma_max);

% a is found from optimization of the cost, where
% total cost = fixed cost + nuclear island cost
%...........
% Fixed cost
%...........
% K_F = Fixed cost for building, turbines, generators etc (also applies to
% fusion, fission, fossil)
% Assumption: The fixed cost is proportional to power output:
% Equation 5.13
% K_F = C_F*P_E;
%...........
% Nuclear island cost (mainly cost of magnets, blanket and shield)
%............
% Assumption: The proportional to reactor volume:
% Equation 5.14
% K_I = C_I*V_I;
% Equation 5.15
% V_I = 2.0*pi^(2.0) * R_0 * ( (a+b+c)^(2.0) - a^(2.0) ); % Reactor volume
%..............
% Cost per watt:
%..............
% Defined as C_p_watt = (K_F + K_I)/P_E, rewritten to
```

```
% C_p_watt = C_F + C_I*(V_I/P_E);
% Since the cost per watt contains two constants, we can minimize the
% V_I/P_E in order to optimize the cost
% Given by equation 5.20 inserted in 5.17
% Equation 5.21
% V_I_per_P_E = V_I/P_E;  % Reactor volume per power out
% a is found by setting the derivative of V_I_per_P_E = 0
% Equation 5.29
a = ((1.0 + xi)/(2.0*xi^(1.0/2.0))) * b;

% Knowing xi, a, and, b, we can calculate c
% c found by comparing tensile force and magnetic force working on the coil
% Equation 5.27
c = 2*xi/(1-xi)*(a+b);
end



function [R_0] = get_R_0(a, eta_t, E_n, E_a, E_Li, P_E, P_W)
%GET_R_0 Calculate the major radius

% Divide eq 5.18 (electric power out) by
% eq 5.19 (wall loading * area = total neutron production) and solve for R0
% Equation 5.20
R_0 = (1.0/(4.0*pi^(2.0)*eta_t))*(E_n/(E_n + E_a + E_Li))*(P_E/(a*P_W));
end



function [P_dens] = get_P_dens(E_a, E_n, E_Li, P_E, eta_t, V_p)
%GET_P_DENS Calculate the power density

```

```
% The power density is found by the sum of the power from the alphas plus
% the power from the neutrons, divided by the plasma volume
% Equation 5.35
P_dens = (E_a + E_n)/(E_a + E_n + E_Li)*P_E/(eta_t*V_p);
end



function [B_0] = get_B_0(R_0, a, b, B_max)
%GET_B_0 Calculte the magnetic field strength on the magnetic axis

% B_max is found in the edge of the magnet (at R = R_0-a-b)
% B0 is the magnetic field at R0
% As B propto 1/R. we have that B_0/B_max = (R_0-a-b)/R_0, which leads to
% Equation 5.42
B_0 = ((R_0-a-b)/R_0)*B_max;
end



function [beta] = get_beta(p, B_0, mu_0)
%GET_beta Calculte the magnetic field strength on the magnetic axis

% Plasma beta in the center (kinetical pressure over magnetical pressure):
% Equation 5.43
beta = p / (B_0^2/(2.0*mu_0));
end



function [p] = get_p(E_a, E_n, P_dens, T, sigma_v_avg)
%GET_P Calculate the plasma pressure
```

```
% Found from solving the sum of neutron and alpha power for n, and multiply
% the result with T
% Equation 5.37
p = ( (16.0/(E_a + E_n)) * P_dens )^(1.0/2.0)*...
     (T^(2.0)/sigma_v_avg)^(1.0/2.0);
end



function [V_I] = get_V_I(R_0, a,b,c)
%GET_V_I Calculate the volume of the material surrounding the plasma

% Equation 5.15
V_I = 2.0*pi^(2.0) * R_0 * ( (a+b+c)^(2.0) - a^(2.0) );
end



function [C_per_watt] = get_C_per_watt(C_F, C_I, V_I_per_P_E)
%GET_C_PER_WATT Calculates the cost for one watt out from the power plant

% For details in how the cost is derived, see comments in the function
% get_a_and_c

C_per_watt = C_F + C_I*(V_I_per_P_E);
end
```