Newton-Raphson

Name: Shafayat Alam
Instructor: Foluso Ladiende
TA: Taiwo Alare and Xueting Deng
Date: 3/19/2025

Table of Contents

I. Summary

This report focuses on the numerical modeling and analysis of a turbine cold-end system using the Newton-Raphson method. The cold-end is critical for the system since it rejects residual thermal energy and system's stability, which comprises components such as turbine exhaust, condenser, and cooling tower. [Equations 1-6] illustrate the mathematical model used in the analysis of the system. Due to the non-linear nature of the equations the Newton-Raphson method was employed as the numerical method to approximate the unknown quantities iteratively. However, convergence challenges were encountered which were resolved through a staged modular approach that was successfully algorithmically implemented in MATLAB. The final results were promising being under a stringent error tolerance level of 10^{-12}%, validating the effectiveness of Newton-Raphson method despite its limitations in thermal system modeling.

II. Introduction

A turbine is a mechanical device that illustrates thermodynamics principles of energy that converts the energy of a fluid into mechanical energy that can be used to generate electricity. For instance, in thermal power plants steam turbines are the core of the energy conversion process. However, this generation of energy requires management of residual heat.

The portion of the system after this generation of energy and residual heat is dissipated is referred to as "turbine cold-end" where this remaining heat is handled/rejected. This cold-end generally consists of turbine exhaust, condenser, and cooling tower. The cold-end plays a significant role in the overall efficiency and stability of for instance a power plant. The cooling process from the condenser removes the waste heat from for instance steam from the turbine and transfers it to the cooling tower to

reject it to the environment. Thus, the effectiveness of the cold-end system must be considered when investigating the performance of the turbine extracting energy.

Accurately modelling the cold-end of this turbine system is critical for optimizing energy extraction/conversion and ensuring reliable system function. This report focuses on accurately modelling the cold-end of a turbine system [Figure 1] and producing approximate numerical values of key variables that investigate the overall efficiency/performance of the system. The mathematical/numerical model is produced by utilizing the interdependence of the components for use in implementing the Newton-Raphson method that approximates numerical values within a desired accuracy.

III. Mathematical Model

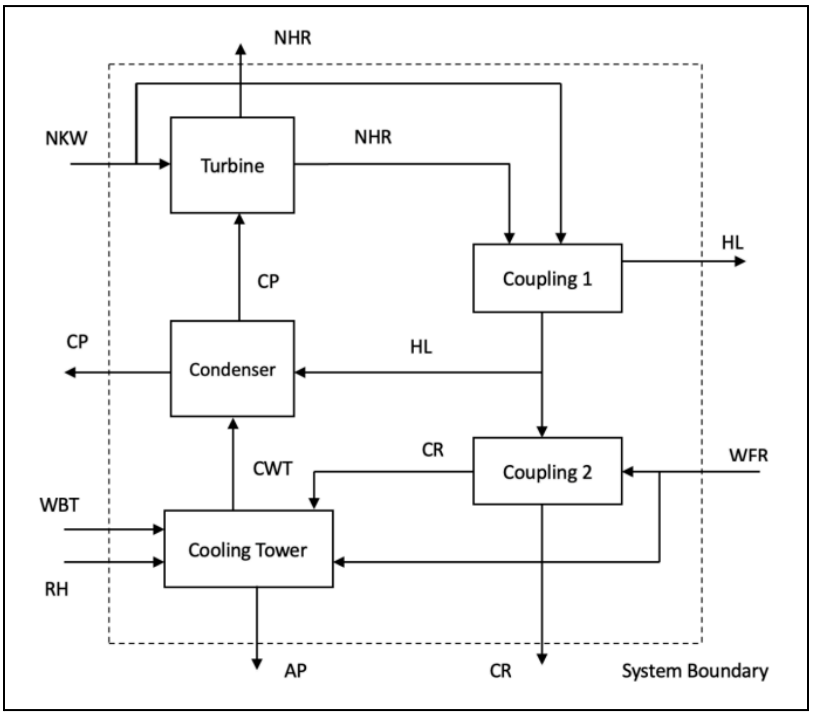

Figure 1. Turbine cold-end System Structure Diagram

NKW = turbine net output
NHR = turbine net heat rate
CP = condenser pressure
CWT = cold water temperature
HL = heat load
WBT = ambient wet-bulb temperature
RH = ambient relative humidity
CR = cooling range
WFR = cooling water flow rate

Figure 2. Mathematical Model Nomenclature

A turbine cold-end system is illustrated in [Figure 1] where the Turbine, Condenser, and Cooling Tower systems mathematical modelling blocks are shown in conjunction. Coupling 1 and Coupling 2 mathematical model blocks are also used in combination with variable transfer links to show the information flow paths between the systems and/or mathematical models. The nomenclature used in the mathematical model is shown in [Figure 2]. [Equations 1-6] illustrates the complete mathematical model for the turbine cold-end system. Utilizing the six equations, the definition of Tower approach, and the Newton-Raphson method—a root-finding approximation numerical method—the heat load, the Condenser pressure, the Turbine net heat rate, Turbine net output, the Tower approach and Tower cooling range under various ambient wet-bulb temperatures are approximated in this report.

The performance data assuming maximum steam throttle flow can be modelled by

$$NHR = -45.19(CP)^4 + 420(CP)^3 - 1442(CP)^2 + 2248(CP) + 6666 \quad (1)$$

and

$$NKW = 4883(CP)^4 - 44890(CP)^3 + 152600(CP)^2 + 231500(CP) + 383400. \quad (2)$$

The condenser and the mechanical-draft cooling tower performance can be modelled by

$$CP = 1.6302 - 0.50095 \times 10^{-1}(CWT) + 0.55796 \times 10^{-3}(CWT)^2 + 0.32946 \times 10^{-3}(HL) - 0.10229 \times 10^{-4}(HL)(CWT) + 0.16253 \times 10^{-6} \times (HL)(CWT)^2 + 0.42658 \times 10^{-6}(HL)^2 - 0.92331 \times 10^{-8}(HL)^2(CWT) + 0.71265 \times 10^{-10} \times (HL)^2(CWT)^2 \; for \; WFR = 145,000 \; GPM \tag{3}$$

and

$$CWT = -0.10046 \times 10^{2} + 0.22801 \times 10^{-3}(WFR) + 0.85396(CR) + 0.18617 \times 10^{-5}(CR)(WFR) + 0.10957 \times 10(WBT) - 0.22425 \times 10^{-5}(WBT)(WFR) \times -0.11978 \times 10^{-1}(WBT)(CR) + 0.14378 \times 10^{-7}(WBT)(CR)(WFR). \tag{4}$$

In addition to the performance equations, two coupling equations are needed to complete the mathematical model for the system. Assuming the heat load is the same for both the condenser and the cooling, the coupling between them is modelled by

$$CR = 2000\frac{(HL)}{(WFR)}. \tag{5}$$

If minor heat loss from the turbine system is neglected the waste heat rejection from the turbine must take place in entirety in the condenser. The waste heat rejection is the amount of heat removed from the condenser which is modelled by

$$HL = \frac{(NHR-3412)(NKW)}{10^6}. \tag{6}$$

Finally, the tower approach is defined as

$$AP = Tower\ Approach = CWT - WBT. \tag{7}$$

IV. Numerical Model

The Newton-Raphson method is defined as

$$x_{new} = x_{old} - \frac{f(x_{old})}{f'(x_{old})} \tag{8}$$

where initially x_{old} is the initial guess to begin the iteration and calculate x_{new}. Each iteration, x_{old} is replaced by x_{new} where the approximate relative error between x_{new} and x_{old} must decrease to indicate convergence. Approximate relative error is defined as

$$\varepsilon_{relative} = \left|\frac{x_{new} - x_{old}}{x_{new}}\right| \times 100\% \tag{9}$$

and the iterative approach of to perform the Newton-Raphson method is terminated when $\varepsilon_{relative}$ results in a desired accuracy.

However, to numerically approximate the unknown values intended utilizing [Equations 1-6] using Newton-Raphson method shown by [Equation 8] and the mathematical model must be translated into vector notation. [Equation 1-6] and the unknown quantities must be translated into the vector notation forms shown by [Figure 3].

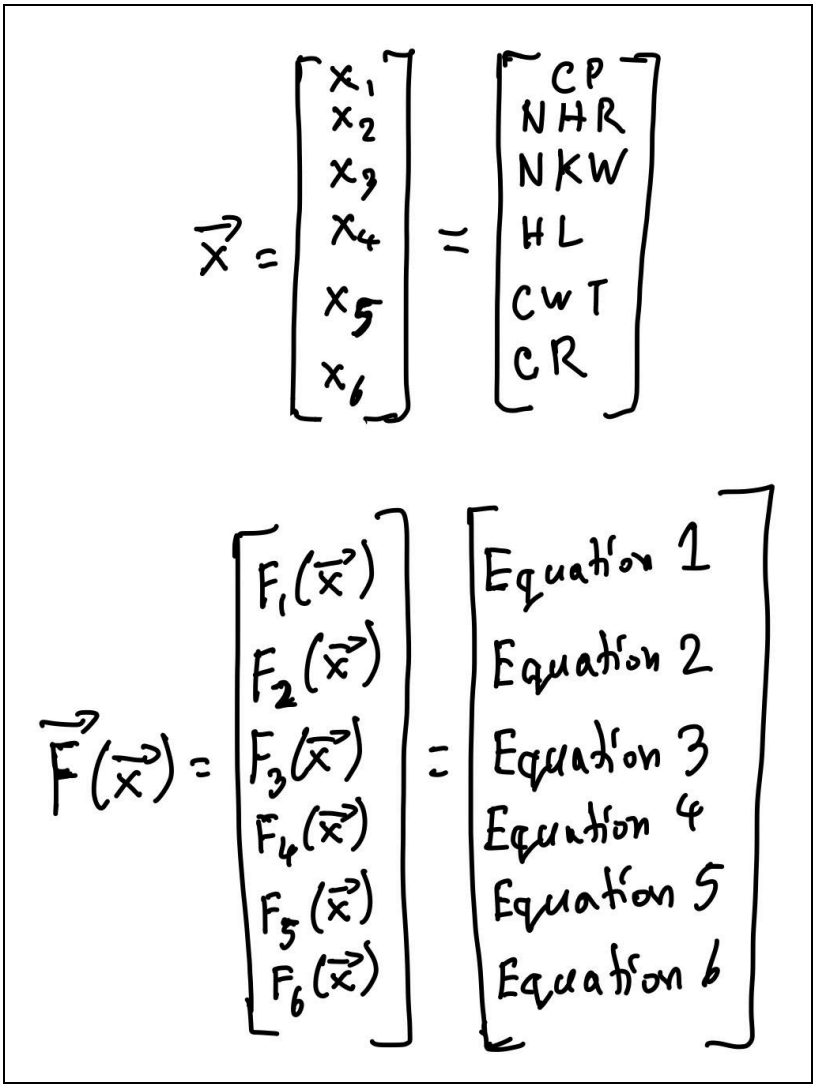

Figure 3. Mathematical model in vector forms

Next, a Jacobian matrix must be used to convert [Equation 8] into the form

$$\vec{x_{new}} = \vec{x_{old}} - J^{-1}(\vec{x_{old}}) \times \vec{F}(\vec{x_{old}}) \tag{10}$$

where J^{-1} is the inverse of the jacobian defined as

$$J_{ij} = \frac{\partial F_i}{\partial x_j} . \tag{11}$$

Note the Jacobian matrix is defined as

$$J(\vec{x}) = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \bullet & \bullet & \frac{\partial F_1}{\partial x_n} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \bullet & \bullet & \frac{\partial F_2}{\partial x_n} \\ \bullet & \bullet & \bullet & & \bullet \\ \bullet & \bullet & & \bullet & \bullet \\ \frac{\partial F_n}{\partial x_1} & \frac{\partial F_n}{\partial x_2} & \bullet & \bullet & \frac{\partial F_n}{\partial x_n} \end{bmatrix}$$

where F_n is the n_{th} function/equation and x_n is the n_{th} variable.

Translating [Equations 1-6] and the unknown quantities into vector notion forms allows to follow [Equation 10] form of the Newton-Raphson method. This is critical as the form shown by [Equation 10] allows to algorithmically implement the root finding numerical method to approximate the unknown quantities.

V. Algorithm

All equations are solved in 4 stages. In the first stage Equations [1 and 2] are solved to approximate unknown values CP, NHR, and NKW from which HL could directly be calculated using the approximated values using [Equation 6]. Similarly in stage 2, CR could directly be calculated through [Equation 5] since WFR is given to be 145,000 GPM and HL was calculated in stage 1. Next, in stage 3 equation 3 is rearranged to directly calculate CWT since all values that CWT rely on in equation 3 are by this stage known. Finally, in stage 4 equation 4 was also rearranged and similarly WBT is also calculated since by this stage all other values are known.

Pseudocode

1. Stage 1
    a. Set initial guess of CR, NHR, and NKW
    b. Set desired tolerance level of 10^{-12}% and max iteration of 1000 iterations
    c. Initialize variable for HL and set it equal to [Equation 6]
    d. WHILE relative error is less than set tolerance
        i. Form the function vector
        ii. Form the Jacobi Matrix
        iii. Perform Newton-Raphson on [Equations 1 and 2]
        iv. Update values for Newton-Raphson
    e. END WHILE
    f. Calculate HL from [Equation 6]
    g. Record and present approximated CP, NHR, NKW, and HL
2. Stage 2:
    a. Compute CR from [Equation 5] as HL and WFR are known
    b. Record and present calculated CR
3. Stage 3:
    a. Rearrange [Equation 3] to solve for CWT
    b. Plug in values as all values needed are known and compute CWT
    c. Record and present calculated CWT
4. Stage 4:
    a. Rearrange [Equation 4] to solve for WBT

b. Plug in all values as all values are known and compute WBT

c. Record and present calculated WBT

5. Compute AP using [Equation 7] as CWT and WBT are known.
6. Present a summary of all results

VI. Results

The set error tolerance level is 10^{-12}%.

Table 1. Table of Given Constant and Temperature Values

Given Value	**Numerical Value**
Circulating Water Flow Rate (WFR)	145000 GPM
Temperature Values	**Numerical Value**
Cold Water Temperature (CWT)	6.30000
Ambient Wet-Bulb Temperature (WBT)	17.1900

Table 2. Numerically Determined Values

Quantity Name	Numerical Value
Heat Load (HL)	1155.71
Condenser Pressure (CP)	1.44701
Turbine Net Heat Rate (NHR)	7973.97
Turbine Net Output (NKW)	253336
Tower Approach (AP)	-10.8900
Cooling Range (CR)	15.940

[Table 1] displays the given circulating water flow rate (WFR) value that was used to determine the other unknown values. Similarly, the table also displays the cold water and ambient wet-bulb temperatures that were used in numerically determining the desired unknown values. [Table 2] displays the numerically determined values of the desired initially unknown quantities.

VII. Discussion

All the numerically determined values are reliable as they were under the set low error tolerance level of 10^{-12}%. These are significant;y promising results as numerical convergence was successful under the very low set error tolerance level. It can be considered that these approximated real values are critically reliable when compared to the exact real values.

Initially, the error tolerance level was set to be 10^{-4}% however it was impossible to accomplish convergence. After investigation it was determined that the output was never converging/diverging, but actually oscillating. The oscillation could indicate the existence of multiple roots. Moreover, it could also indicate severely undesired function behavior when considering the use of Newton-Raphson method for root finding, such as severe non-linearity or prone to overshooting. Nevertheless, after discovering an oscillating iteration even after increasing max allowed iteration to 100,000 iterations and attempting subjectively improved initial guesses it was found true that simply using Newton-Raphson method to solve all 6 equations simultaneously would never result in a promising result. Illustrating a weakness of the Newton-Raphson method.

Thus, it was decided to investigate the behavior of each of the equations by analyzing the behavior of values every iteration. It was found that [Equations 3 and 4] were the most problematic. [Equations 3 and 4] were the most responsible for the oscillating inoperable iterations. Thus, a divide and conquer approach was taken to solve the equations with suitable convergence behavior to approximate as many unknown values that could be found and then simply doing algebra or plugging in known values to the remaining equations to determine the remaining unknown quantities.

To further improve the reliability of the results, the set error tolerance level was increased in increments of 10^{-2}% while keeping the set max iteration of 100 iterations. The lowest error tolerance level that could be set was found to be 10^{-12}% and the investigation was terminated given accomplishment of severely reduced error tolerance value that the numerically determined results could almost be considered indistinguishable compared to the real exact values.

VIII. Conclusion

It was a daunting task to understand the turbine cold-end system and the given mathematical model. Nevertheless, it was understood well enough to have been able to translate it numerically to approximate the unknown desired quantities and devise an algorithm to achieve reliable numerical results. Understanding of the Newton-Raphson method and its relationship with the behavior was also critical in the success of producing reliable approximated numerical values. The analysis also illustrated the use and also the limitations of the Newton-Raphson method. Nevertheless, the limitation was overcome by decomposing the problem into stages and taking a step-wise approximation approach.

IX. Appendix

```
clc; clear;
% ======= Stage 1: Solve Equations 1,2, and 6 to find CP, NHR, NKW =======
x = [1.6; 6800; 360000];% Initial Guess Vector: [CP, NHR, NKW]
% Convergence settings
tolerance = 1e-12;
max_iter = 1000;
WFR = 145000; %WFR is given
for iter = 1:max_iter
    CP = x(1);
    NHR = x(2);
    NKW = x(3);
    HL = ((NHR - 3412)*NKW)/1e6; % HL from Equation 6
    % Forming Function Vector
    F = zeros(3,1);
    F(1) = -45.19*CP^4 + 420*CP^3 - 1442*CP^2 + 2248*CP + 6666 - NHR;
    F(2) = 4883*CP^4 - 44890*CP^3 + 152600*CP^2 - 231500*CP + 383400 - NKW;
    F(3) = ((NHR - 3412)*NKW)/1e6 - HL;
    J = zeros(3,3); % --- Jacobian Matrix
    % Jacobi for Equation 1
    J(1,1) = -180.76*CP^3 + 1260*CP^2 - 2884*CP + 2248;
    J(1,2) = -1;
```

```
    J(1,3) = 0;
    % Jacobi for Equation 2
    J(2,1) = 19532*CP^3 - 134670*CP^2 + 305200*CP - 231500;
    J(2,2) = 0;
    J(2,3) = -1;
    % Jacobi for Equation 3
    J(3,1) = 0;
    J(3,2) = NKW / 1e6;
    J(3,3) = (NHR - 3412) / 1e6;
    % Newton-Raphson
    dx = -J \ F;
    x_new = x + dx;
    % Checking Relative error
    rel_error = abs((x_new - x) ./ x_new) * 100;
    % Display status
    %fprintf('Iter %3d | CP = %.4f | NHR = %.2f | NKW = %.2f | Max RelErr
= %.4e\n', ...
     %   iter, x(1), x(2), x(3), max(rel_error));
    if all(rel_error < tolerance)
        disp('Stage 1 Converged.');
        break;
    end
    x = x_new; % Setting new x for iteraton
end
% Final Outputs
CP = x(1);
NHR = x(2);
NKW = x(3);
HL = ((NHR - 3412)*NKW)/1e6;
fprintf('\nFinal Stage 1 Results:\n');
fprintf('CP  = %.6f\n', CP);
fprintf('NHR = %.2f\n', NHR);
fprintf('NKW = %.2f\n', NKW);
fprintf('HL  = %.2f\n', HL);
% ===== Stage 2: Direct Calculation of CR using Equation 5 =======
% HL from Stage 1
% WFR, given Water Flow Rate
CR = (2000 * HL) / WFR;  % Equation 5
fprintf('\nFinal Stage 2 Results (Updated):\n');
```

```
fprintf('CR  = %.2f\n', CR);
% ======= Stage 3: Solve Equation 3 for CWT  =========
% CP and HL known from Stage 1;
% WFR is given
CWT = 85; % Initial guess for CWT
% Convergence settings
tolerance = 1e-12;
max_iter = 1000;
for iter = 1:max_iter
    % Equation 3
    f = 1.6302 ...
        - 0.050095 * CWT ...
        + 0.00055796 * CWT^2 ...
        + 0.00032946 * HL ...
        - 0.000010229 * HL * CWT ...
        + 0.00000016253 * HL * CWT^2 ...
        + 0.00000042658 * HL^2 ...
        - 0.000000092331 * HL^2 * CWT ...
        + 0.000000000071265 * HL^2 * CWT^2 ...
        - CP;
    % Equation 3 Jacobian
    df = -0.050095 ...
         + 2 * 0.00055796 * CWT ...
         - 0.000010229 * HL ...
         + 2 * 0.00000016253 * HL * CWT ...
         - 0.000000092331 * HL^2 ...
         + 2 * 0.000000000071265 * HL^2 * CWT;
    % Newton-Raphson
    dCWT = -f / df;
    CWT_new = CWT + dCWT;
    rel_err = abs((CWT_new - CWT) / CWT_new) * 100; % Relative error check
    if rel_err < tolerance
        disp('Stage 3 Converged.');
        break;
    end
    CWT = CWT_new; % Newton-Raphson iteration
end
% Final Output
fprintf('\nFinal Stage 3 Result:\n');
```

```
fprintf('CWT = %.2f\n', CWT);
% ===== Stage 4: Direct Calculation of WBT from Equation 4 ======
% Using known values CR, CWT, WFR
% Rearrange Equation 4 to solve for WBT: CWT = A + B*WFR + C*CR + D*CR*WFR + E*WBT + F*WBT*WFR + G*WBT*CR + H*WBT*CR*WFR
%                                        WBT = (CWT - (A + B*WFR + C*CR + D*CR*WFR)) / (E + F*WFR + G*CR + H*CR*WFR)
% Coefficients of Equation 4
A = -10.046;
B = 0.00022801;
C = 0.85396;
D = 0.0000018617;
E = 1.0957;
F = -0.000022425;
G = -0.011978;
H = 0.00000014378;
% Calculate WBT
numerator = CWT - (A + B*WFR + C*CR + D*CR*WFR);
denominator = E + F*WFR + G*CR + H*CR*WFR;
WBT = numerator / denominator;
% Final Output
fprintf('\nStage 4 Result:\n');
fprintf('WBT = %.2f\n', WBT);
AP = CWT - WBT;
% Results Summary
fprintf('\n========== Final Results Summary ==========\n');
fprintf('CP  = %.6f\n', CP);
fprintf('NHR = %.2f\n', NHR);
fprintf('NKW = %.2f\n', NKW);
fprintf('HL  = %.2f\n', HL);
fprintf('WFR = %.0f GPM (Given)\n', WFR);
fprintf('CR  = %.2f\n', CR);
fprintf('CWT = %.2f\n', CWT);
fprintf('WBT = %.2f\n', WBT);
fprintf('Tower Approach (AP) = %.2f\n', AP)
fprintf('==========================================\n');
```