A SIMPLIFIED NUCLEAR POWER MODULE DESIGN TOOL USING FORTRAN

Arlen B. Beers-Green, David Happe, Darcy Wagner, Teal Zega
Department of Mechanical Engineering, Gonzaga University, Spokane, WA 99258
Robert R. Stiger, Ph.D.
Department of Mechanical Engineering, Gonzaga University, Spokane, WA 99258

Abstract

This paper describes the design and development of a computer program in Fortran that facilitates steady-state system design optimization studies of a lunar- or Martian-based nuclear-electric power module. The simplified model used is a liquid metal reactor, regenerative Brayton cycle, with heat rejection through radiating panels. The program runs feed-forward, system-by-system heat transfer, thermodynamics, and fluid mechanics analysis according to user-defined thermo-fluid and system properties. The output includes functional system values such as maximum and minimum point temperatures, required reactor heat input, heat out to the environment, and total useable work provided by the power cycle. If the system properties supplied by the user result in an unstable configuration the program reports the problem, allowing the user first to determine stable configurations and then to optimize the design based on variable system criteria. This paper includes an overview of the Fortran code design as well as the results of verification and validation checks.

Introduction

In 2004, President George W. Bush and NASA revealed their plans for the near-term future of space exploration. They listed four goals and objectives:

- Implement a sustained and affordable human and robotic program to explore the solar system and beyond;
- Extend human presence across the solar system, starting with a human return to the Moon by the year 2020, in preparation for human exploration of Mars and other destinations;
- Develop the innovative technologies, knowledge, and infrastructures both to explore and to support decisions about the destinations for human exploration;
- Promote international and commercial participation in exploration to further U.S. scientific, security, and economic interests. [1]

With regard to the first three objectives, the development and implementation of a suitable power supply is vital. One of the most probable sources of power is a nuclear power module because of its potential to provide reliable power to a manned outpost. The high energy density in the fuel potentially allows for long term (greater than 5 years) operation of the power unit without the need for refueling or replacement parts. According to Dr. Steven Howe of the Idaho National Laboratory's Center for Space Nuclear Research, there is a need for a simple tool to analyze such nuclear electric power generation systems. A simplified yet expandable design tool, such as the one described in this paper, facilitates design testing of the many possible systems with limited or no monetary investment, as computer analysis is less expensive than physical design tests.

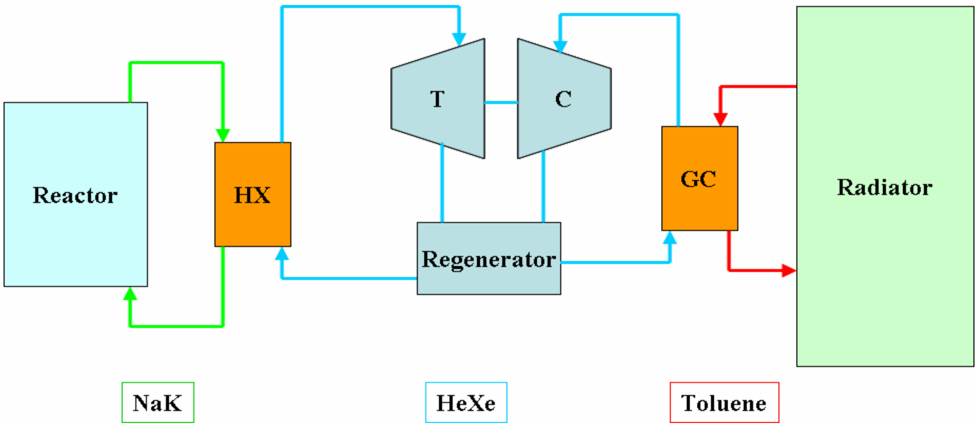


Figure 1 – Nuclear Power Module Model for Fortran Code

Figure 1 illustrates the basic nuclear power module that was chosen for the computer program. It is a liquid metal reactor, regenerative Brayton cycle, with heat rejection through radiating panels. There are two heat exchangers in the model: one that transfers heat from the reactor loop to the Brayton power conversion cycle, labeled “HX;” and one that transfers heat from the working fluid of the Brayton cycle to the working fluid of the radiator panel, labeled “GC” for “gas cooler.” Helium-Xenon and Toluene, respectively, are two fluids that are commonly used in this application, and are therefore used to identify their particular segment of the model, but the computer code allows the user to specify the thermodynamic properties of any material that is under consideration in the design process.

It should be emphasized that the Fortran program described in this paper is a design tool that can be used to analyze and compare any number of nuclear power model systems that follow the basic configuration described in the previous paragraph. The selection of this model is restrictive in the sense that, without modification, it allows studies on only one type of module that might not correspond well with a specific system under consideration. However, the program surmounts this limitation in two ways to achieve great flexibility: 1) it allows the user to specify all material and physical properties, as well as the general physical parameters of the system; and 2) all of the program components were constructed as alterable subroutines within a module that is called by the main program. That means that the code is usable, as-is, for the defined module and is easily tailored to include any other module with the addition or modification of subroutines. For example, if the user had a system with two Brayton power conversion cycles instead of one, a simple statement to call the subroutine for the second cycle could be added to the main program.

The purpose of this paper is to describe the program and its potential as a design tool. Its purpose is not to present the results of a study of a particular system or set of parameters.

Analytical Model

Fortran was chosen because of its versatility with matrices and the fact that it is designed to run efficiently with engineering applications. The code is structured into three main subroutines: heat exchanger, Brayton cycle, and radiator. Each of the subroutines is described in brief at the end of this section, along with their respective methodologies. These subroutines are included in a module which is accessed in the main body of the program.

The program begins by requiring the rated power output of the reactor and a user supplied guess for the temperature of the fluid entering the reactor. The program then calls the heat exchanger subroutine and calculates the output values. These output values are then passed to the pre-Brayton cycle check subroutine. This subroutine ensures that the system values selected by the user are physically possible. The values from the initial pre-Brayton cycle are then passed to the Brayton cycle subroutine where initial values for turbine and compressor work are calculated. The heat rejected from the Brayton cycle is then passed across the gas coolers as a guess for the heat rejected by the radiator. An inlet temperature to the radiator is calculated from the heat exchanger subroutine and is passed to the radiator subroutine. A new value for heat rejected from the radiator is calculated by the radiator subroutine and is compared to the value for heat rejected by the Brayton cycle. If the point temperatures match, a flag is set to stop iterating. The Brayton cycle is then calculated backwards, using the heat rejected from the radiator as an input. The new point temperatures on the Brayton cycle are then calculated and passed across the front heat exchanger. A new value for the temperature out of the reactor is calculated. If the flag has been set to false, then the loop quits. If not, the loop continues until convergence is reached. In short, the program runs through the system forwards and backwards until convergence in the values for heat transferred across the gas cooler and heat rejected from the radiator match, since no heat can be stored in the radiators in the steady-state condition. This convergence indicates the convergence of all other point values in the system.

Several different checks are performed throughout the program in order to help the user to track possible system parameter errors. For instance, if the system pressure ratio is too large or the turbine efficiency is too low to result in a physically viable situation, then the program will stop and the user will be notified of optimal performance conditions. An additional check for negative enthalpies is performed during each iteration. If at any point negative enthalpy values are calculated, then the program will output an error that the system parameters do not work.

Heat Exchanger

The heat exchanger subroutine utilizes basic heat transfer techniques for four different types of heat exchangers (user specified) making use of the NTU method to simplify the analysis. The applied equations for the heat exchanger subroutine are shown below.[2]

$$UA = \frac{1}{\frac{1}{h_i A_i} + \frac{1}{h_o A_o}}$$

$$q = \dot{m}_h\, c_{p,h}(T_{h,i} - T_{h,o}) \quad \rightarrow \quad T_{h,o} = T_{h,i} - \left(\frac{q}{\dot{m}_h\, c_{p,h}} \right)$$

$$q = \dot{m}_h\, c_{p,c}(T_{c,i} - T_{c,o}) \quad \rightarrow \quad T_{c,o} = T_{c,i} + \left(\frac{q}{\dot{m}_h\, c_{p,c}} \right)$$

$$q = \varepsilon C_{\min}(T_{h,i} - T_{c,i}) \qquad NTU = \frac{UA}{C_{\min}}$$

$C_{\min}$ is equal to C_c or C_h, whichever is smaller

The specific relations for each different type of heat exchanger are explicitly defined below. The user specifies which heat exchanger to use according to the number in front of each configuration.

1. Concentric Tube: Parallel Flow

$$\varepsilon = \frac{1-\exp\left[-NTU\,(1+C_r)\right]}{1+C_r}$$

2. Concentric Tube: Counter Flow

$$\varepsilon = \frac{1-\exp\left[-NTU\,(1-C_r)\right]}{1-C_r\exp\left[-NTU\,(1-C_r)\right]} \qquad (C_r < 1)$$

3. Shell-and-tube: One Shell Pass (2,4…tube passes)

$$\varepsilon_1 = 2\left\{1+C_r+(1+C_r^2)^{1/2}\times\frac{1+\exp\left[-(NTU)_1(1+C_r^2)^{1/2}\right]}{1-\exp\left[-(NTU)_1(1+C_r^2)^{1/2}\right]}\right\}^{-1}$$

4. Cross-flow (Single Pass): Both fluids unmixed

$$\varepsilon = 1-\exp\left[\left(\frac{1}{C_r}\right)(NTU)^{0.22}\left\{\exp\left[-C_r(NTU)^{0.78}\right]-1\right\}\right]$$

Brayton Cycle

The Brayton cycle subroutine employs fundamental thermodynamic principles to analyze the cycle point-by-point. The property look-up routine is utilized to determine point values in the appropriate form, either temperature, enthalpy or pressure ratios. Shown below are the applicable equations used in the Brayton cycle subroutine (see References for equation source). Shown below in Figure 2 is the Temperature versus Entropy (T vs. s) diagram for an ideal regenerative Brayton cycle, while our program uses an actual cycle with turbine and compressor efficiencies to account for non-isentropic conditions. The equations used correspond to the points defined in this diagram.

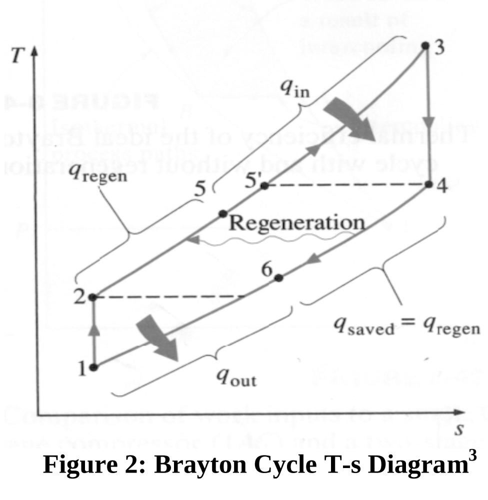


Figure 2: Brayton Cycle T-s Diagram[3]

$$(q_{in} - q_{out}) + (w_{in} - w_{out}) = h_{exit} - h_{inlet}$$

$$\eta_{th,Brayton} = 1 - \frac{1}{r_p^{(k-1)/k}} \qquad r_p = pressure\ ratio$$

$$q_{regen,act} = h_5 - h_2 \qquad q_{regen,\max} = h_{5'} - h_2 = h_4 - h_2$$

$$\in = \frac{q_{regen,act}}{q_{regen,\max}} = \frac{h_5 - h_2}{h_4 - h_2} = \frac{T_5 - T_2}{T_4 - T_2}$$

Radiator

The Radiator subroutine applies fluid mechanics and heat transfer fundamentals to calculate the required output.[4, 5, 6] The Newton-Rhapson iteration technique is used to find the friction factor and all calculations are completed under the assumption that a constant pressure header (manifold) is used at the top of the radiator and that the fluid flow will always be touching the channel walls.

$$\Delta \dot{E}_{mech} = \dot{m}\Delta e_{mech} = \dot{m}\left(\frac{P_2 - P_1}{\rho} + \frac{V_2^2 - V_1^2}{2} + g(z_2 - z_1)\right)$$

$$V_2 = \frac{\sqrt{\frac{2(\Delta P)}{\rho} + \alpha_1 V_1 + 2g(Z_1 - Z_2) + 2h_L}}{\alpha_2}$$

Velocity Profile: $V_2 = \left\{2gdz + V_1^2\right\}^{1/2}$

Heat Transfer through the radiator: $h_i(T_b - T_{s,i}) = -k\left(\frac{T_{s,i} - T_{s,o}}{\Delta x}\right) = \varepsilon\sigma(T_{s,o}^4 - T_\infty^4) + h_o(T_{s,o} - T_\infty)$

Each fluid channel in the radiator panel is discretized by a user-defined mesh. Essentially, the radiator panel is discretized into finitely small channels, each treated as a control volume. The program calculates a guess temperature at each control volume to be evaluated: T_b, $T_{s,i}$, $T_{s,o}$, T_{inf}. The program then calculates a guess for the work out of the radiator for that control volume. Based on this newly created work out, a new internal and external surface temperature is calculated. The routine continues guessing and checking these temperature values until convergence for that control volume is reached. The bulk outlet of the current control volume is then calculated and set as the bulk inlet of the next control volume. The program moves through each control volume in one channel and calculates the temperature profile. From this profile, the heat rejected from the radiator can be calculated for one channel. Depending on the number of operating channels, the total heat rejected from the radiators is then calculated.

Property Look-up and Linear Fit

The property look-up subroutine uses a simple check to confirm the property interval and assure the best possible linear fit. The subroutine requires a property array, such as enthalpy, relative pressure, or temperature and a point to evaluate at as inputs. The point to evaluate is compared to each value of the array line by line. If the point to evaluate at is less than the point in the array, the subroutine goes to the next line of the array. This process continues until the point to evaluate at exceeds a point in the array. The index, or line number, of the array is then the output of the subroutine.

In some instances, it is necessary to interpolate a value between points of the property arrays. A second subroutine is then invoked which takes two properties and creates a linear fit between two known points. The subroutine calculates the slope and y-intercept of the line, and then evaluates a point that is in between values on the property arrays.

Input and Output

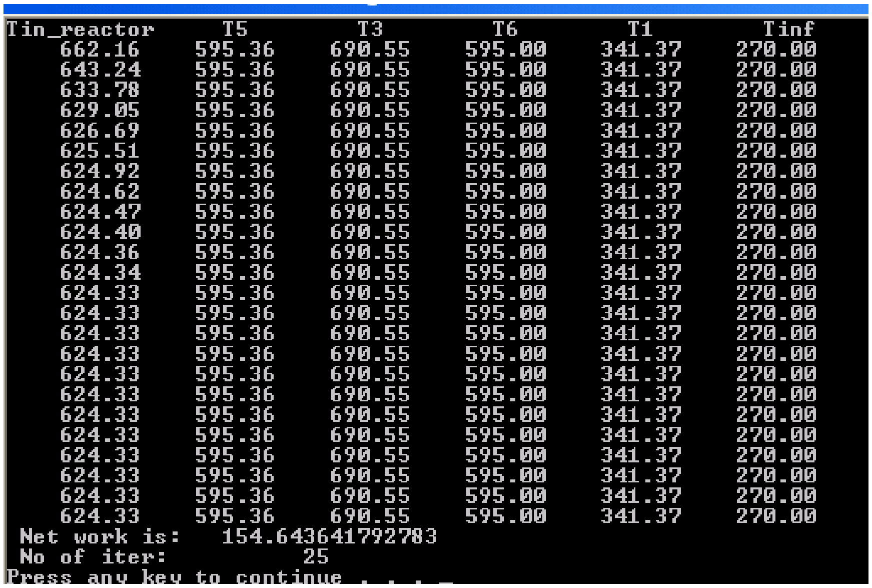
The program inputs are added by the user in columnar form in four different files, whose names and formats are described in the program user's/code developer's manual. Changing the variables in these four text files represents changes in the system model and will result in varying analyses of different nuclear power modules. These variables must be chosen in an educated way in order to obtain meaningful results. That is, though the program is designed to be simple and operable with minimal user training, inputs representing illogical or unrealistic systems will result in worthless outputs.

After the program has been run, an output screen appears with the calculated power output from the Brayton cycle, point temperatures around the system with the corresponding environmental temperature, and a value for the number of iterations that was needed to reach convergence. Additionally, an output file is created with these same values.

Results

Before integration of the entire program, the individual subroutines were checked against known problems. The heat exchanger subroutine and Brayton cycle subroutine both return accurate results as compared with Bergman and Cengel, respectively. The radiator panel subroutine also returns accurate results when compared with a known problem, as is indicated in graphical form in Figure 3. The blue line is the known bulk temperature profile as a function of location on the panel and the pink line is the bulk temperature profile returned by the program. It can be seen that 200 control volumes were deemed fine enough to achieve values within 0.1% accuracy.

After each subroutine was validated, the entire system was integrated and tested as a whole. As was previously described in the Analytical Model section, the system was closed around a point temperature value after a backwards and forward method of iteration. It is not clear whether a "zig-zag" technique such as this one has been used previously in a design tool application.



```
Tin_reactor      T5          T3          T6          T1          Tinf
    662.16      595.36      690.55      595.00      341.37      270.00
    643.24      595.36      690.55      595.00      341.37      270.00
    633.78      595.36      690.55      595.00      341.37      270.00
    629.05      595.36      690.55      595.00      341.37      270.00
    626.69      595.36      690.55      595.00      341.37      270.00
    625.51      595.36      690.55      595.00      341.37      270.00
    624.92      595.36      690.55      595.00      341.37      270.00
    624.62      595.36      690.55      595.00      341.37      270.00
    624.47      595.36      690.55      595.00      341.37      270.00
    624.40      595.36      690.55      595.00      341.37      270.00
    624.36      595.36      690.55      595.00      341.37      270.00
    624.34      595.36      690.55      595.00      341.37      270.00
    624.33      595.36      690.55      595.00      341.37      270.00
    624.33      595.36      690.55      595.00      341.37      270.00
    624.33      595.36      690.55      595.00      341.37      270.00
    624.33      595.36      690.55      595.00      341.37      270.00
    624.33      595.36      690.55      595.00      341.37      270.00
    624.33      595.36      690.55      595.00      341.37      270.00
    624.33      595.36      690.55      595.00      341.37      270.00
    624.33      595.36      690.55      595.00      341.37      270.00
    624.33      595.36      690.55      595.00      341.37      270.00
    624.33      595.36      690.55      595.00      341.37      270.00
    624.33      595.36      690.55      595.00      341.37      270.00
    624.33      595.36      690.55      595.00      341.37      270.00
 Net work is:    154.643641792783
 No of iter:             25
Press any key to continue . . . _
```

Figure 3- Sample output screen

A test case for the entire system was run using air as the working fluid. Variables such as heat exchanger configurations and radiator panel sizing were specified. The results for the different reactors used in the test case are included in Table 1. The inputted variables for the test case are shown in Table 2. A graph of rated power versus efficiency is included in Figure 4. Typical thermal efficiencies for Brayton cycles with a pressure ratio of around 1.25 are approximately 20%.[7,8,9] The values obtained for the program are included below and the system operates around 20%. It can be seen in Figure 4 that there is a maximum efficiency for the given system tested. The efficiency begins to drop after about 550kWt when the radiators fail to handle the load. Back heating can occur up to a point, around 550kWt, after which the program returns a negative enthalpy value. If the given system can not operate, the program recognizes it and exits with an error message.

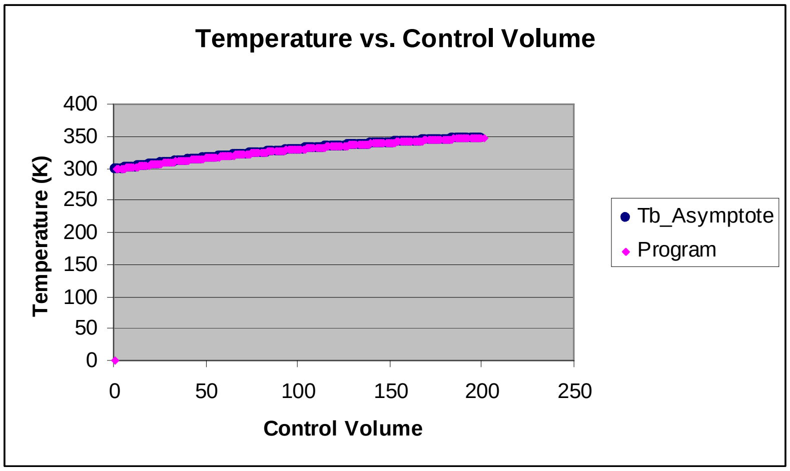


Figure 4 – Bulk Temperature vs. Control Volume for known and test cases[2]

A point that deserves comment is that there was not a test of the system using the materials properties of HeXe for the Brayton cycle, even though the program was originally intended for such a system. The reason for this is twofold—the team could not obtain meaningful property tables for HeXe and there were not published test cases similar enough to the basic test system under analysis with which to compare the analytical results. Fundamentally, however, this does not impugn the reliability of the program, because making the change from air as the working fluid to HeXe (or any other fluid, for that matter), involves no more than numerical changes in the input file. The program was purposefully designed to be flexible in this way and operates accurately with the materials properties values of air.

Conclusions

The original objective of this project—to design and develop a computer program in Fortran for simple steady-state system design optimization studies of a lunar- or Martian-based nuclear-electric power module—was met. The program is easy to use and highly flexible for future development and customization. The accuracy of the theoretical approach and mathematical calculations have been verified and validated through tests against known systems.

There are, however, several limitations to the current version of the program. It is not capable of performing transient analysis for the system, or of accounting for catastrophic events that occur during use. A catastrophic event, such as partial destruction of a radiator panel from a meteor, can not be handled in real time. An event such as this can be accounted for by reducing the initial input area on the radiator panel, but the program can not provide any information for what would happen to the system if parameters or components changed during actual, real-time use. All subroutines are based on the assumption that the program is to be used strictly for steady state analysis.

This program is a significant development in the field in that it can quickly achieve results suitable for general system performance, evaluation, and comparison without the need for extensive user training or capital investment. This contrasts sharply with the complex and unwieldy programs already in existence.

Reactor (kwt)	*Wnet (Kwe)*	*eff*
200	15.82306878	0.079115
250	32.54939123	0.130198
275	40.26557225	0.14642
300	48.79745145	0.162658
315	53.71175994	0.170514
325	57.24420415	0.176136
350	64.66663419	0.184762
375	75.39299187	0.201048
400	86.77064088	0.216927
450	111.2928975	0.247318
500	138.5665999	0.277133
525	152.5049756	0.290486
550	166.0213745	0.288733

Table 1 – Results of Thermal Power and Electrical Power

Test Case

File Name: universal.txt

Variable Name	Description	Value	Unit
g	Gravity	9.81	$\mathrm{m/s^2}$
q_reactor	Q out of reactor	551000	kJ
Tin_reactor	Temp in to HX reactor side	700	K
cp_NaK	Specific heat of NaK	975	kJ/kg·K
cp_HeXe	Specific heat of HeXe	4186	kJ/kg·K
cp_toluene	Specific heat of Toluene	1000	kJ/kg·K
mdot_reactor	Mass flow rate of reactor fluid	1	kg/s
mdot_brayton	Mass flow rate of Brayton fluid	1	kg/s
mdot_radiator	Mass flow rate of radiator fluid	1	kg/s
reactor_eff	Efficiency of reactor	1	-

File Name: hx.txt

Variable Name	Description	Value	Unit
hx_config	Heat Exchanger Configuration	1	-
gas_config	Gas Cooler Configuration	1	-
ho	Heat Transfer Coefficient Outside	40	W/m^2
hi	Heat Transfer Coefficient Inside	40	W/m^2
A	Contact Area	200	m^2

File Name: brayton.txt

Variable Name	Description	Value	Unit
nt	Turbine Efficiency	0.8	-
nc	Compressor Efficiency	0.8	-
e	Recuperator Effectiveness	0.8	-
r	Pressure Ratio	1.25	-

File Name: radiator.txt

Variable Name	Description	Value	Unit
density	Density of the fluid	900	$\mathrm{kg/m^3}$
sigma	Boltzman’s Constant	5.67E-08	$\mathrm{W/m^2K^4}$
l	Length of the channel	11	m
dh	Height of control volume	0.025	m
roughness	Surface roughness of channel	0.0000002	m
cv	Specific heat of Radiator fluid	200	kJ/kg·K
no_channels	Number of Channels	20	-
radiator_area	Total radiating area	50	$\mathrm{m^2}$
kw	Thermal conductivity of wall	257	W/m·K
thickness	Wall thickness	0.001	m
P_wetted	Wetted perimeter of channel	0.078539816	m
emissivity	Emissivity of the surface	0.95	-
h_o	External convection coefficient	0	$\mathrm{W/m^2{\cdot}K}$
hl	Head loss per channel	0	m
in_area	Manifold, cross sectional area	0.025	$\mathrm{m^2}$
efficiency	Efficiency of radiator channel	1	-

Table 2 – Test Case Parameters

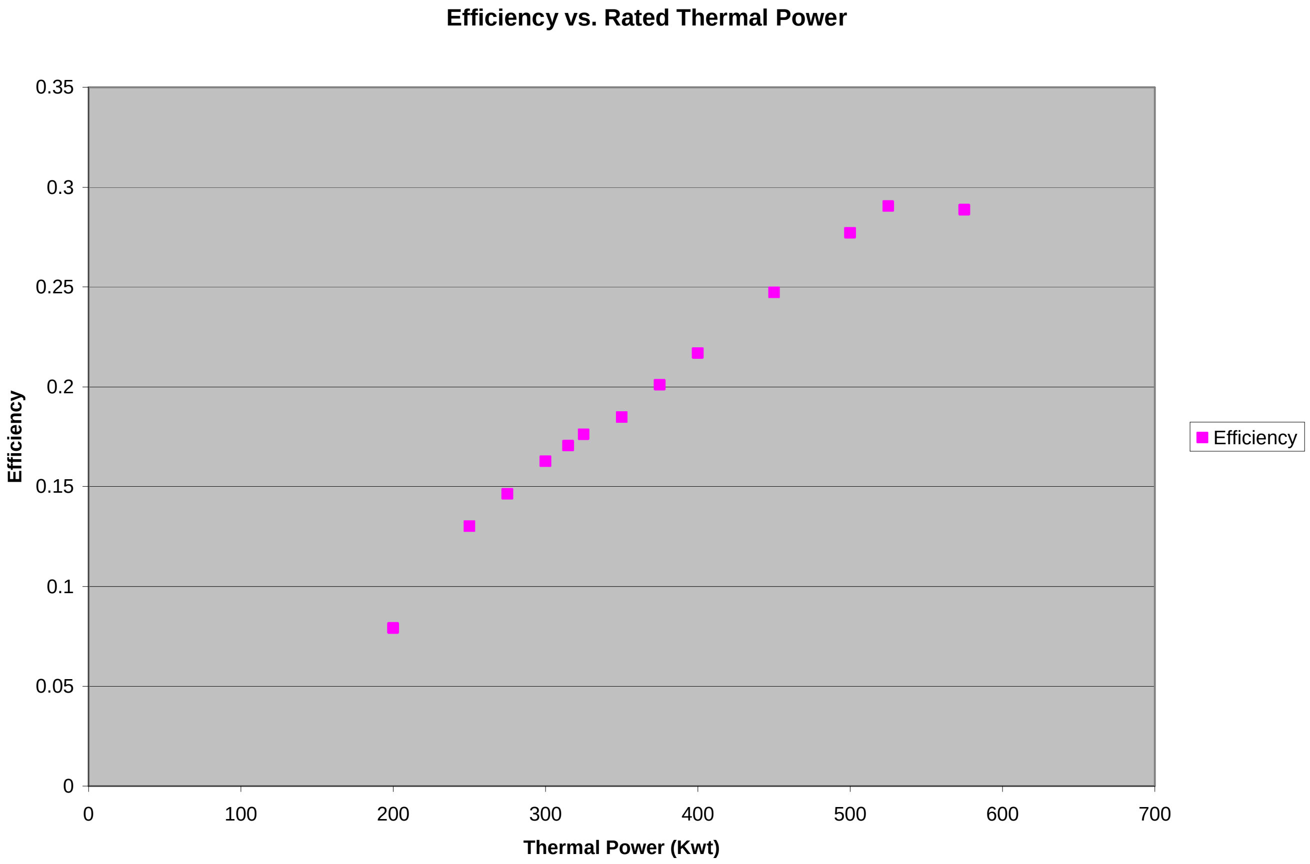


Figure 5 – Thermal Efficiency vs. Thermal Power

References

[1] Bush, George W. A Renewed Spirit of Discovery. United States Office of the Whitehouse. January 2004. September 24, 2006. <http://www.whitehouse.gov/news/releases/2004/01/20040114-1.html>

[2] Bergman, T. L., Dewitt, D. P., Incropera, F. P., & Lavine, A. S. (2007). Chapter 11 Heat Exchangers. In J. Hayton (Ed.), Fundamentals of Heat and Mass Transfer (pp.669-722). Hoboken: John Wiley & Sons, Inc.

[3] Cengel, Yunus A., Boles, Michael A. Thermodynamics: An Engineering Approach. Fifth Edition. New York: McGraw-Hill Higher Education, 2006.

[4] Siamidis, John; Mason, Lee; Beach, Duane; Yuko, James. “Heat Rejection Concepts for Brayton Power Conversion Systems.”NASA/TM—2005-213337.

[5] Siamidis, John; Mason, Lee. “A Comparison of Coolant Options for Brayton Power Conversion Heat Rejection Systems.” NASA/TM—2006-214121.

[6] Siamidis, John. “Heat Rejection Concepts for Lunar Fission Surface Power Applications.” AIAA 2006-4196

[7] Johnson, Paul K; Mason, Lee S. “Design and Off-Design Performance of 100 kWe-Class Brayton Power Conversion Systems.” NASA/TM—2005-213626

[8] Sridhar, K.R.. 24th International Conference on Environmental Systems : “Evaluation of a reverse Brayton Cycle Heat Pump for Lunar Base Cooling” Friedrichshafen, Germany. June 20-23, 1994.

[9] Diao, Zheng-Gang. “Simulation of Space Solar CBC Power Systems.” Chinese Academy of Sciences: Institute of Engineering Thermophysics”. New York: American Society of Mechanical Engineers. [1993: 7-7]