References

Agudelo, A., Cortés, C. 2010. Thermal radiation and the second law. *Energy*, 35 (2), p. 679 - 691.

Bejan, A., 1982. *Entropy generation through heat and fluid flow*. Colorado: John Wiley.

Bejan, A. 1996. Method of entropy generation minimization, or modelling and optimization based on combined heat transfer and thermodynamics. *Rev Gén Therm*, 35, p. 637 - 646.

Bejan, A., Tsatsaronis, G. and Moran, M. 1996. *Thermal design and optimization*. New York: John Wiley.

Bejan, A. 1997. *Advanced engineering thermodynamics.* 2nd ed. Durham: John Wiley.

Bertocchi, R., Karni, J. and Kribus, A. 2004. Experimental evaluation of a non-isothermal high temperature solar particle receiver. *Energy*, 29, p. 687 – 700.

Burden, R.L. and Faires, J.D. 2005. *Numerical analysis*. 8th ed. Youngston State University: Thomson Brooks/Cole.

Çengel, Y.A. 2006. *Heat and mass transfer*. 3rd ed. Nevada, Reno: McGraw-Hill.

Chen, L., Zhang, W. and Sun, F. 2007. Power, efficiency, entropy-generation rate and ecological optimization for a class of generalized irreversible universal heat-engine cycles. *Applied Energy,* 84, p. 512 - 525.

Cheremisinoff, P.N. and Regino, T.C. 1978. *Principles and applications of solar energy*. Michigan: Ann Arbor Science Publishers.

Copper Development Association. 2006. *The Copper Tube Handbook*. New York: Copper Development Association.

Cornelissen, R.L. and Hirs, G.G. 1997. Exergetic optimization of a heat exchanger. *Energy Conversion and Management*, 1 (15-17), p. 1567 - 1576.

DME, Department of Minerals and Energy, Republic of South Africa. 2010. *Solar Energy*. [online]. Available at: http://www.dme.gov.za/energy/renew_solar.stm [Accessed: 20 July 2010].

Dittus, F.W. and Boelter, L.M.K. 1930. *University of California Publications on Engineering*, *2*, p. 433.

Dixon, S.L. 2005. *Fluid mechanics and thermodynamics of turbomachinery*. 5th ed. Liverpool: Elsevier Butterworth-Heinemann.

Duffie, J.A. and Beckman, W.A. 1991. *Solar engineering of thermal processes.* New York: John Wiley.

Fluri, T.P. 2009. The potential of concentrating solar power in South Africa. *Energy Policy*, 37, p. 5075 – 5080.

Garrett. 2009. *Garrett by Honeywell: Turbochargers, Intercoolers, Upgrades, Wastegates, Blow-Off Valves, Turbo-Tutorials.* Available at: http://www.TurboByGarrett.com [Accessed: 26 April 2010].

Gnielinski, V. 1976. New equations for heat and mass transfer in turbulent pipe and channel flow. *International Chemical Engineering*, 16, p. 359 - 368.

Heller, P., Pfänder, M., Denk, T., Tellez, F., Valverde, A., Fernandez, J. and Ring, A. 2006. Test and evaluation of a solar powered gas turbine system. *Solar Energy*, 80, p. 1225 - 1230.

Hesselgreaves, J.E. 2000. Rationalisation of second law analysis of heat exchangers. *International Journal of Heat and Mass Transfer*, 43 (22), p. 4189 - 4204.

Howell, J.R., Bannerot, R.B. and Vliet, G.C. 1982. *Solar-thermal energy systems.* New York: McGraw-Hill.

Ishikawa, H. and Hobson, P.A. 1996. Optimisation of heat exchanger design in a thermoacoustic engine using a second law analysis*. International Communications in Heat and Mass Transfer*, 23 (3), p. 325 - 334.

Johnson, G. 2009. Plugging into the sun. *National Geographic*, 216(3), p. 28 - 53.

Joshi, A.S., Dincer, I. and Reddy, B.V. 2009. Development of new solar exergy maps. *International Journal of Energy Research*, 33, p. 709 – 718.

Jubeh, N.M. 2005. Exergy analysis and second law efficiency of a regenerative Brayton cycle with isothermal heat addition. *Entropy*, 7 (3), p. 172 - 187.

Kreith, F. and Kreider, J.F. 1978. *Principles of solar engineering*. Colorado: Hemisphere.

Lerou, P.P.P.M., Veenstra, T.T., Burger, J.F., Ter Brake, H.J.M. and Rogalla, H. 2005. Optimization of counterflow heat exchanger geometry through minimization of entropy generation. *Cryogenics*, 45, p. 659 - 669.

Mills, D. 2004. Advances in solar thermal electricity technology. *Solar Energy*, 76, p. 19 - 31.

Narendra, S., Kaushik, S.C. and Misra, R.D. 2000. Exergetic analysis of a solar thermal power system. *Renewable Energy*, 19, p. 135 - 143.

Oğulata, R.T., Doba, F. and Yilmaz, T. 2000. Irreversibility analysis of cross flow heat exchangers. *Energy Conversion and Management*, 41 (15), p. 1585 - 1599.

Ordóñez, J.C. and Bejan, A. 2000. Entropy generation minimization in parallel-plates counterflow heat exchangers. *International Journal of Energy Research*, 24, p. 843 - 864.

Petela, R. 2010. *Engineering thermodynamics of thermal radiation*. New York: McGraw Hill.

Petukhov, B.S. 1970. Heat transfer and friction in turbulent pipe flow with variable physical properties. *Advances in Heat Transfer*, 6.

Pitz-Paal, R. 2007. High temperature solar concentrators. *Solar Energy Conversion and Photoenergy Systems*, [Ed. Galvez, J.B. and Rodriguez, S.M.], in *Encyclopedia of Life Support Systems (EOLSS),* Developed under the Auspices of UNESCO, Eolss Publishers, Oxford [http://www.eolss.net].

Prakash, M., Kedare, S.B. and Nayak, J.K. 2009. Investigations on heat losses from a solar cavity receiver. *Solar Energy*, 83, p. 157 - 170.

Ratts, B.E. and Raut, A.G. 2004. Entropy generation minimization of fully developed internal flow with constant heat flux. *Journal of Heat Transfer*, 126 (4), p. 656 - 659.

Reddy, K.S. and Sendhil Kumar, N. 2008. Combined laminar natural convection and surface radiation heat transfer in a modified cavity receiver of solar parabolic dish. *International Journal of Thermal Sciences*, 47, p. 1647 – 1657.

Reddy, K.S. and Sendhil Kumar, N. 2009. An improved model for natural convection heat loss from modified cavity receiver of solar dish concentrator. *Solar Energy*, 83, p. 1884 – 1892.

Sama, D.A. 1995. The use of the second law of thermodynamics in process design. *Journal of Energy Resources Technology*, 117, p. 179 - 185.

Sarangi, S. and Chowdhury, K. 1982. On the generation of entropy in a counterflow heat exchanger. *Cryogenics*, 22 (2), p. 63 - 65.

Schwarzbözl, P., Buck, R., Sugarmen, C., Ring, A., Jesús Marcos Crespo, M., Altwegg, P. and Enrile, J. 2006. Solar gas turbine systems: design, cost and perspectives. *Solar Energy*, 80, p. 1231 - 1240.

Sendhil Kumar, N. and Reddy, K.S. 2007. Numerical investigation of natural convection heat loss in modified cavity receiver for fuzzy focal solar dish concentrator. *Solar Energy*, 81, p. 846 – 855.

Sendhil Kumar, N. and Reddy, K.S. 2008. Comparison of receivers for solar dish collector system. *Energy Conversion and Management*, 49, p. 812 - 819.

Shah, R.K. 2005. Compact heat exchangers for micro-turbines. In *Micro Gas Turbines* (p. 2-1 – 2-18). Educational Notes RTO-EN-AVT-131, Paper 2. Neuilly-sur-Seine, France: RTO. Available from: http://www.rto.nato.int/abstracts.asp.

Shiba, T. and Bejan, A. 2001. Thermodynamic optimization of geometric structure in the counterflow heat exchanger for an environmental control system. *Energy*, 26, p. 493 - 511.

Shuai, Y., Xia, X. and Tan, H. 2008. Radiation performance of dish solar concentrator/cavity receiver systems. *Solar Energy*, 82, p. 13 - 21.

Snyman, J.A. 2000. The LFOPC leap-frog algorithm for constrained optimization. *Computers and Mathematics with Applications*, 40, p. 1085 - 1096.

Snyman, J.A. 2009. *Practical mathematical optimization*. Pretoria: University of Pretoria.

Steinfeld, A. and Schubnell, M. 1993. Optimum aperture size and operating temperature of a solar cavity-receiver. *Solar Energy*, 50, p. 19 – 25.

Stewart, J. 2003. *Calculus, early transcendentals.* McMaster: Brooks/Cole.

Stine, B.S. and Harrigan, R.W. 1985. *Solar energy fundamentals and design*. New York: John Wiley.

Sonntag, R.E., Borgnakke, C. and Van Wylen, G.J. 2003. *Fundamentals of thermodynamics.* New York: John Wiley.

STG International. 2010. *Specifications* [Online].
Available at: http://www.stginternational.org/specs.html
[Accessed: 2 August 2010].

Tsai, L. 2004. *Design and performance of a gas-turbine engine from an automobile turbocharger*. Unpublished. BSc Eng Mech., Massachusetts Institute of Technology.

Weston, K.C. 2000. *Energy Conversion*. [e-book]. Tulsa: Brooks/Cole.
Available at: http://www.personal.utulsa.edu/~kenneth-weston/
[Accessed: 16 February 2010]

Wilson, J.I.B. 1979. *Solar energy*. London: Wykeham.

Yilmaz, M., Sara, O.N. and Karsli, S. 2001. Performance evaluation criteria for heat exchangers based on second law analysis. *Exergy, an International Journal*, 1 (4), p. 278 - 294.

Zimparov, V. 2001. Extended performance evaluation criteria for enhanced heat transfer surfaces: heat transfer through ducts with constant heat flux. *International Journal of Heat and Mass Transfer*, 44 (1), p. 169 -180.

Zimparov, V.D., Da Silva, A.K. and Bejan, A. 2006a. Thermodynamic optimization of tree-shaped flow geometries with constant channel wall temperature. *International Journal of Heat and Mass Transfer*, 49, p. 4839 - 4849.

Zimparov, V.D., Da Silva, A.K. and Bejan, A. 2006b. Constructal tree-shaped parallel flow heat exchangers. *International Journal of Heat and Mass Transfer*, 49, p. 4558 - 4566.

Zimparov, V.D., Da Silva, A.K. and Bejan, A. 2006c. Thermodynamic optimization of tree-shaped flow geometries. *International Journal of Heat and Mass Transfer*, 49, p. 1619 - 1630.

Appendix A

COLLECTOR

This section describes the methodology behind the MATLAB function: '*collector*' (see Appendix C). This function follows the receiver-sizing algorithm of Stine and Harrigan (1985) shown in Figure A.1. A better understanding of the concentrator and its geometry is also given.

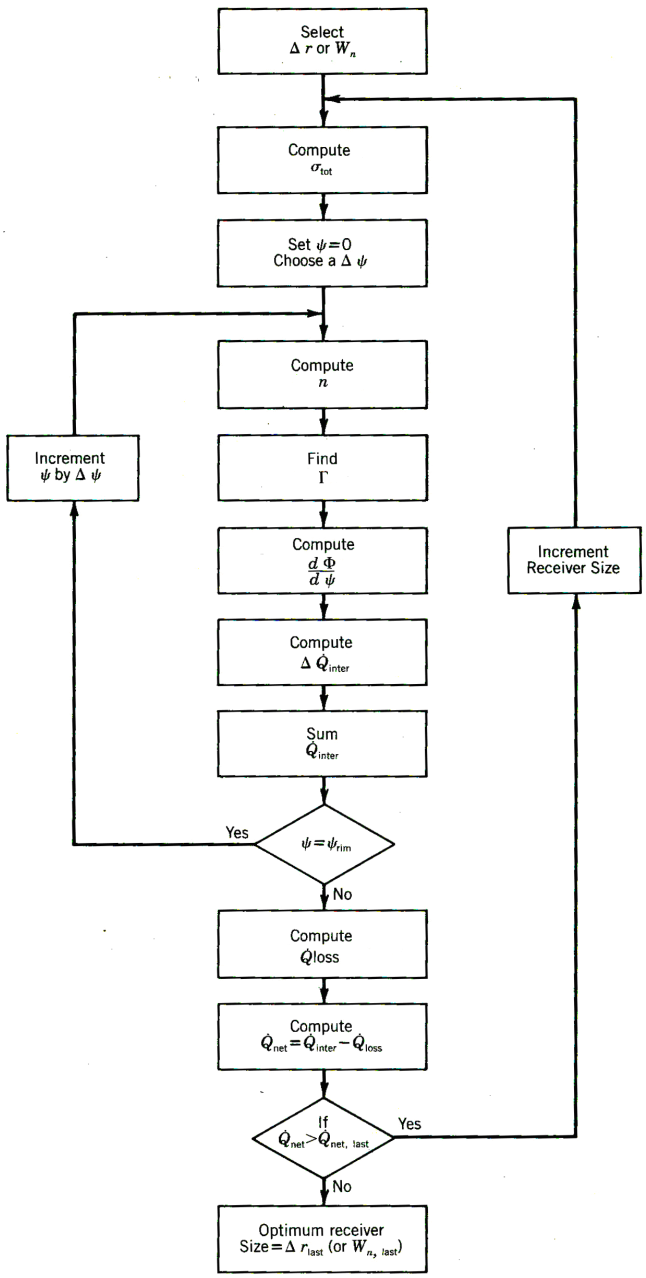

Figure A.1 Receiver-sizing algorithm (Stine and Harrigan, 1985).

The method of Figure A.1 is applied to establish the net absorbed heat rate of the cavity receiver as a function of the cavity receiver aperture, similar to Figure 2.27 in the literature. This is done in the function '*collector*' (Appendix C). The function starts of by asking the user to give the dish concentrator area and its rim angle. Figure A.2 shows the definition of the rim angle (Stine and Harrigan, 1985). The aperture area of a paraboloid (parabolic dish concentrator) is defined by

$$A_s = \pi R^2 \qquad \text{or} \qquad A_s = 4\pi f_c^{\ 2} \frac{\sin^2 \psi_{rim}}{(1+\cos\psi_{rim})^2} \tag{A.1}$$

in terms of the focal length (f_c) and the rim angle, ψ_{rim} (Stine and Harrigan, 1985).

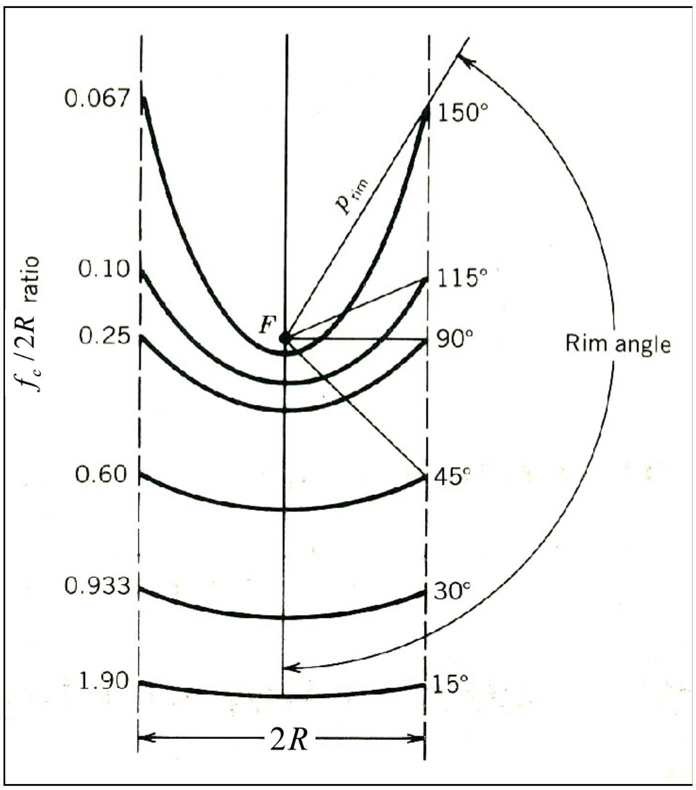

Figure A.2 Definition of the rim angle (Stine and Harrigan, 1985).

The focal length can be calculated when the rim angle and concentrator area are specified. Stine and Harrigan's algorithm requires one to compute the total parabolic concentrator error. According to Stine and Harrigan (1985), a typical parabolic concentrator error is 6.7 mrad. This error could be regarded as a user-specified constant since this error depends on the collector design, structure, tracking, alignment, mirror specular reflectance, etc. After these steps, the function goes to a while loop, starting at a rim angle of 0° through to an angle of ψ_{rim} in increments of 1° and computes the amount of intercepted solar energy per segment of concentrator area. The projection of the image width onto the focal plane (see Figure A.3) can be written as

$$d = \frac{\Delta r}{\cos \psi} \tag{A.2}$$

where ψ is the specific rim angle at the segment of the concentrator.

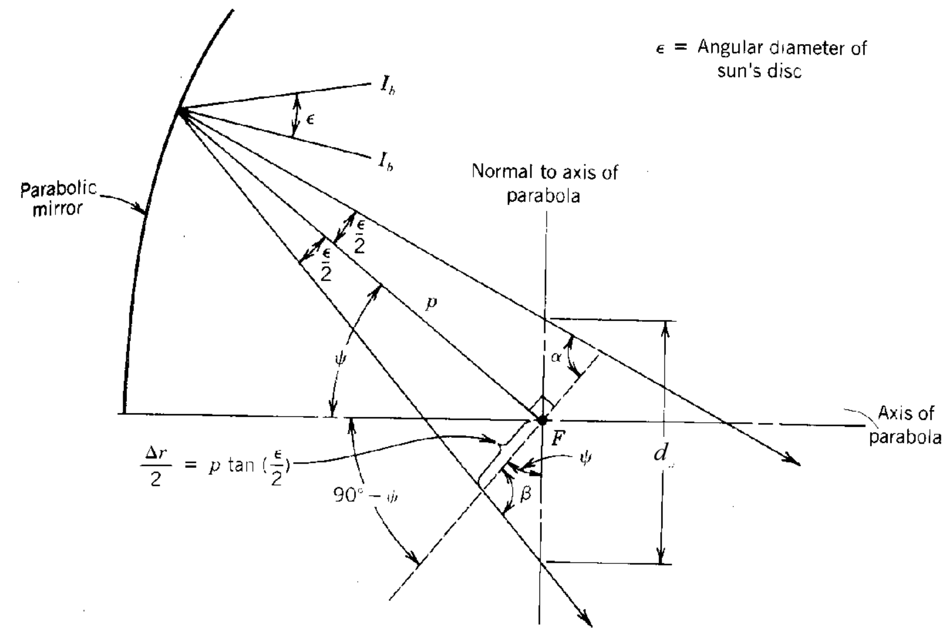

Figure A.3 Reflection of non-parallel rays from a parabolic mirror (Stine and Harrigan, 1985).

For a specific cavity receiver aperture diameter, d , Δr can be calculated. The parabolic radius at that segment can be calculated using equation A.3 (Stine and Harrigan, 1985).

$$p = \frac{2 f_c}{1 + \cos \psi} \tag{A.3}$$

The number of standard deviations, n , being considered can be calculated using equation A.4 (Stine and Harrigan, 1985),

$$\Delta r = 2p\tan\left(n\frac{\sigma_{tot}}{2}\right), \tag{A.4}$$

where σ_{tot} is the total parabolic concentrator error. According to Stine and Harrigan (1985), a typical parabolic concentrator error is 6.7 mrad. The next step is to find Γ. According to Stine and Harrigan (1985), the flux capture fraction is the ratio of the flux reflected from a parabolic surface in a shaft of light having width of n standard deviations of the total angular error. For the normally distributed reflected flux, the flux capture fraction is simply the area under the normal distribution function integrated from $-n/2$ to $+n/2$. A polynomial approximation to this normal integral, from Abramowitz and Stegun (1970, cited in Stine and Harrigan, 1985), can be written as:

$$\Gamma = 1 - 2^{*}Q \tag{A.5}$$

where:

r	= 0.2316419
b_1	= 0.319381530
b_2	= -0.356563782
b_3	= 1.781477937
b_4	= -1.821255978
b_5	= 1.330274429
y	= $n/2$
f_1	= $\frac{1}{\sqrt{2\pi}}e^{\frac{y^2}{2}}$
t_1	= $1/(1+r\,y)$
Q	= $f_1\,(b_1t_1 + b_2t_1^2 + b_3t_1^3 + b_4t_1^4 + b_5t_1^5)$

The next step in Stine and Harrigan's algorithm is to compute the slope: $\frac{d\Phi}{d\Psi}$ where, according to Stine and Harrigan (1985), for a parabolic dish:

$$d\Phi_{PD} = \frac{8\pi I_b f_c^{\,2}\sin\psi d\psi}{(1+\cos\psi)^2} \tag{A.6}$$

$d\Phi_{PD}$ is the total radiant flux reflected from the differential area (assuming no reflectance loss) to the point of focus. The following equation (Stine and Harrigan, 1985) is used to compute the rate of energy reflected from a strip (the parabolic mirror dish is divided into incremental rings) and intercepted by the receiver with aperture diameter, d.

$$\Delta \dot{Q}_i = \rho_s \alpha \Gamma \left(\frac{d\Phi}{d\psi} \right) \Delta \psi \tag{A.7}$$

All of these intercepted energy rates for all the rings are then added to give the total rate of intercepted energy, $\dot{Q}_i$, for the collector with d as receiver aperture diameter. The next step is to calculate $\dot{Q}_0$. According to Stine and Harrigan (1985), ideally, in a well-insulated cavity, the cavity temperature is reasonably uniform and heat loss occurs primarily by convection and radiation from the cavity aperture. The heat loss rate from the cavity is described in Section 3.5.1.1. Once the heat loss rate is available, $\dot{Q}_{net}$ can be calculated as:

$$\dot{Q}_{net} = \dot{Q}_i - \dot{Q}_0 \tag{A.8}$$

It is clear that the amount of absorbed heat rate, $\dot{Q}_{net}$, can be described in terms of the cavity aperture diameter, d. The function '*collector*' determines the net heat rate absorbed by the receiver for different cavity aperture sizes. The result is the curve shown in Figure A.4. This is for e_p = 0.0067 and ψ_{rim} = 45°, as suggested by Stine and Harrigan (1985). From these curves, one can see that there exists an aperture diameter that allows the maximum amount of solar power to be absorbed by the working fluid. Such a curve can be numerically approximated with the discrete least squares approximation method (Burden and Faires, 2005) or by using the function '*curvefit*' in MATLAB ($\dot{Q}_{net} = \sum_{i=0}^{10} x_i d^i$).

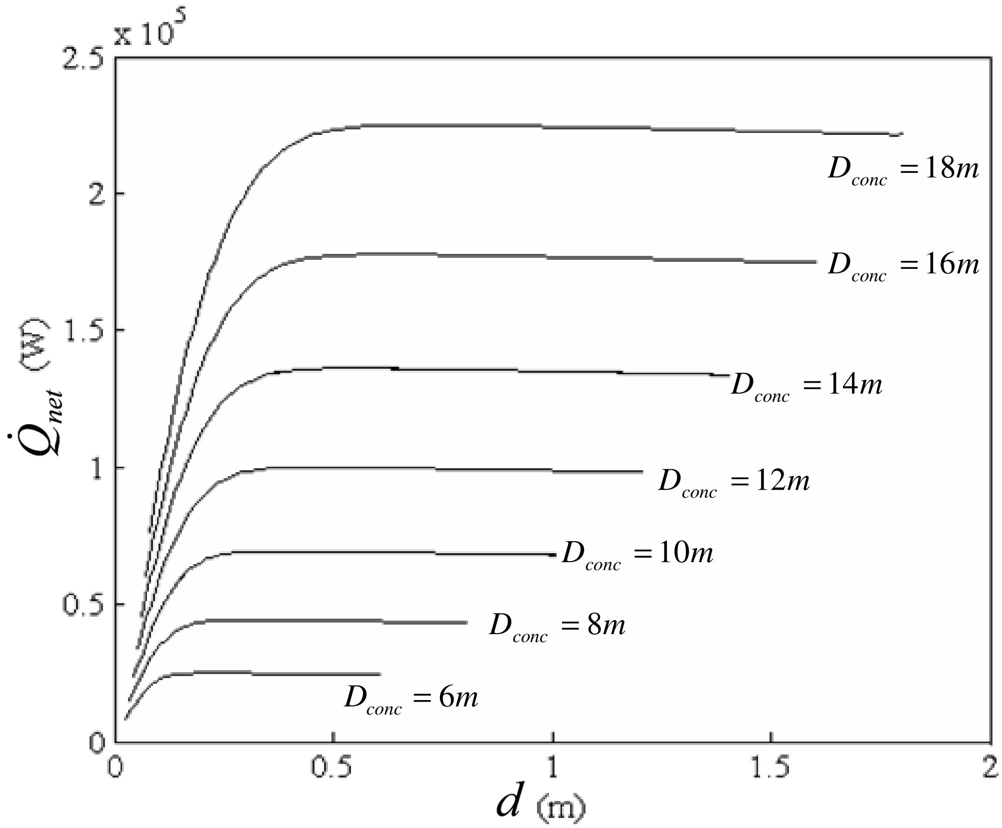

Figure A.4 Relation between net absorbed heat rate and the aperture diameter for a range of concentrator diameters according to the function '*collector*'.

The specific aperture diameter is coupled to the receiver's channel dimensions (its length, hydraulic diameter and aspect ratio – only for a plated receiver, see equation 3.20). The method of entropy generation minimisation can now be used to show whether or not it is better to have an aperture size at the optimum d, as suggested from the curve. The literature suggests that the optimum geometry for a component in a system is not necessarily the optimum geometry when considering the whole system. For this reason, d will not be chosen to be at its optimum, since this optimum is not necessarily the optimum for the whole system. Rather, the aperture diameter is written in terms of the geometry variables so that the net rate of heat absorbsion can be written as a function of the receiver geometry and can be included in the objective function (equation 3.61). The optimum aperture diameter can be found when the optimum geometry variables are found. In the function '*collector*', the shadow of the receiver and its insulation are also accounted for when calculating the available power at the receiver. Heat loss through conduction at the cavity receiver through the insulation is usually small and omitted. In the function '*collector*', however, it was assumed that the conduction heat loss rate is 10% of the sum of the radiation and convection heat loss rates.

References

Burden, R.L. and Faires, J.D. 2005. *Numerical analysis*. 8th ed. Youngston State University: Thomson Brooks/Cole.

Stine, B.S. and Harrigan, R.W. 1985. *Solar energy fundamentals and design*. New York: John Wiley.

Nomenclature

A	Area	m^2
d	Aperture diameter of cavity receiver (or W_n)	m
D	Diameter	m
e_p	Parabolic concentrator error	rad
f	Focal length	m
F	Focal point	-
I	Irradiance	W/m^2
n	Number of standard deviations in receiver-sizing algorithm	-
p	Parabolic radius	m
$\dot{Q}$	Heat transfer rate	W
Δr	Diameter of sun's disc at the focal point	m
R	Radius of parabolic dish concentrator	m
W_n	Aperture diameter of cavity receiver (or d)	m
x	Discrete least-squares approximation constant	-
α	Receiver absorptance	-
α	Defining angle at receiver aperture	rad
β	Defining angle at receiver aperture	rad
Γ	Flux capture fraction	-
$\Delta\psi$	Incremental parabola angle defining ring	-
ε	Angular diameter of sun's disc	-
ρ_s	Mirror surface specular reflectance	-
σ	Parabolic concentrator error	rad
Φ	Radiant flux	W

$d\Phi$	Total radiant flux reflected from differential concentrator area to focus point	-
ψ	Specific concentrator rim angle	-
ψ_{rim}	Concentrator rim angle	-

Subscripts:

0	Loss due to convection and radiation
b	Beam
c	Concentrator
$conc$	Concentrator
i	Intercepted total
inter	Intercepted
last	Last
loss	Loss
net , net	Net available for receiver fluid
PD	Parabolic dish
rim , rim	Rim / to the rim
s	Surface
tot	Total

Appendix B

ENTROPY GENERATION RATE TABLE

Table B.1 Entropy generation rate equations from the literature.

Eq.	Entropy generation research field		Entropy generation rate equation	Comments/ Symbols	References
1	**A. Internal flow**	Per unit tube length, constant heat flux, for all ducts (one-dimensional heat transfer duct)	$\dot{S}'_{gen} = \dfrac{d\dot{S}_{gen}}{dx} = \dfrac{q'\Delta T}{T^2(1+\tau)} + \dfrac{\dot{m}}{\rho T}\left(-\dfrac{dp}{dx}\right)$	$\tau = \Delta T / T$	Yilmaz et al. (2001); Bejan et al. (1996); Hesselgreaves (2000); Zimparov et al. (2006c); Bejan (1982)
2		Constant heat flux, per unit tube length, for all ducts	$\dot{S}'_{gen} = \dfrac{q'^2}{4T^2\dot{m}c_p}\dfrac{D_h}{St} + \dfrac{2\dot{m}^3}{\rho^2 T}\dfrac{f}{D_h A^2}$	$T =$ bulk temperature of the stream	Bejan (1982); Bejan et al. (1996)
3		Constant heat flux, per unit tube length, for a circular tube, single-phase fluid	$\dot{S}'_{gen} = \dfrac{q'^2}{\pi k T^2 Nu_D} + \dfrac{32\dot{m}^3 f}{\pi^2 \rho^2 T D^5}$	$T =$ bulk temperature of the stream	Bejan (1982); Bejan et al. (1996); Bejan (1996)

4		Constant heat flux, incompressible viscous fluid, laminar, fully developed	$\dot{S}_{gen} = \frac{q'}{\pi k Nu T_0^2} q' L_i + \frac{128\nu}{\rho \pi T_i} \dot{m}_i^2 \frac{L}{D_i^4}$	T_i = inlet temperature and assuming $\tau << 1$, where $T_i T_0 = T_i^2 = T_0^2$	Zimparov et al. (2006c)
5		Constant and uniform heat flux, per unit length of tube, for all tubes, single-phase, fully developed	$\dot{S}'_{gen} = \frac{\dot{q}'' P (T_w - T)}{T^2} + \frac{\dot{m}^3 f}{2\rho T D_h A^2}$	T = bulk fluid temperature, P = perimeter	Ratts and Raut (2004)
6		Constant and uniform heat flux, for all tubes with tube length L, fluid properties assumed to be constant, single-phase, fully developed	$\dot{S}_{gen} \cong \frac{(\dot{q}'')^2 P D_h L}{Nu T_1 T_2 k} + \frac{8 \dot{m}^3 f L}{\rho^2 T_{ave} D_h^3 P^2}$	T_1 and T_2 are the inlet and outlet fluid temperatures, P = perimeter	Ratts and Raut (2004)
7		Constant heat flux, including the fluid temperature variation along tube length of heat exchanger, for ideal gas or incompressible flow, circular	$\dot{S}_{gen} = \dot{m} c_p \Delta T \frac{4St(\Delta T / D)L}{T_i^2 [1 + 4St(\Delta T / T_i)(L/D)]} + \frac{\dot{m} u_m^2 f}{2St \Delta T} \ln\left(1 + 4St \frac{\Delta T L}{T_i D}\right)$	$u_m = 4\dot{m} / \rho / \pi / D^2$	Zimparov (2001)

8		Per unit tube length, constant channel wall temperature for all ducts (one-dimensional heat transfer duct)	$\dot{S}'_{gen} = \frac{d\dot{S}_{gen}}{dx} = \dot{m}c_p \frac{\Delta T dT}{T^2 dx} + \frac{\dot{m}}{\rho T}\left(-\frac{dp}{dx}\right)$	Assuming ideal gas or incompressible fluid and $\tau = \Delta T / T << 1$	Zimparov et al. (2006a)
9		Convective heat transfer in a duct with constant wall temperature, circular	$\dot{S}_{gen} = \dot{Q}_t \frac{\theta_o}{T_i T_o} + \frac{32\dot{m}^3 f}{\rho^2 \pi^2 D^5}\frac{L}{T_w}$	$\dot{Q}_t = \dot{m}c_p(T_o - T_i)$ $\theta_o = T_w - T_o$	Yilmaz et al. (2001)
10	**B. External flow**	Heat transfer and drag on an immersed body	$\dot{S}_{gen} = \frac{\dot{Q}_B(T_B - T_\infty)}{T_B T_\infty} + \frac{F_D U_\infty}{T_\infty}$		Bejan (1996)
11		Heat transfer and drag on an immersed body	$\dot{S}_{gen} = \frac{\dot{Q}\left(\overline{T}_w - T_\infty\right)}{T_\infty^2} + \frac{F_D U_\infty}{T_\infty}$		Bejan et al. (1996)
12	**C. Augmentation techniques**	For a single fin	$\dot{S}_{gen} = \frac{\dot{Q}_B \theta_B}{T_\infty{}^2} + \frac{F_D U_\infty}{T_\infty}$	$\theta_B = T_B - T_\infty$, $\theta_B << T_\infty$	Bejan (1982)

13	**D. Local entropy generation**		$\dot{S}'''_{gen} = \frac{1}{T}\left(\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y}\right) - \frac{1}{T^2}\left(q_x \frac{\partial T}{\partial x} + q_y \frac{\partial T}{\partial y}\right) +$ $\rho\left(\frac{\partial s}{\partial t} + v_x \frac{\partial s}{\partial x} + v_y \frac{\partial s}{\partial y}\right) + s\left[\frac{\partial \rho}{\partial t} + v_x \frac{\partial \rho}{\partial x} + v_y \frac{\partial \rho}{\partial y} + \rho\left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y}\right)\right]$		Bejan (1982)
14		The volumetric rate of entropy generation	$\dot{S}'''_{gen} = k\frac{(\nabla T)^2}{T^2} + \mu\frac{\phi}{T}$		Yilmaz et al. (2001); Bejan et al. (1996); Bejan (1982)
15	**E. Heat Exchangers**	For tubular, full size heat exchanger with constant heat flux, assume ideal gas or incompressible fluid	$\dot{S}_{gen} = \frac{\dot{Q}_t \Delta T}{T_i^2\left[1 + \left(\frac{\Delta T_m}{T_i}\right)\right]} + \frac{32\dot{m}^3 fL}{\rho^2\pi^2 T_i D^5}\frac{\ln\left[1+\left(\frac{\Delta T_m}{T_i}\right)\right]}{\left(\frac{\Delta T_m}{T_i}\right)}$	$u_m = 4\dot{m}/\rho/\pi/D^2$ and $\dot{Q}_t = \dot{m}c_p(T_o - T_i)$	Zimparov (2001); Yilmaz et al. (2001)
16		Counterflow and cross-flow heat exchangers, where the working fluid is an ideal gas with constant specific heat, for all tubes	$\dot{S}_{gen} = (\dot{m}c_p)_1 \ln\frac{T_{1,out}}{T_{1,in}} + (\dot{m}c_p)_2 \ln\frac{T_{2,out}}{T_{2,in}}$ $-(\dot{m}R)_1 \ln\frac{P_{1,out}}{P_{1,in}} - (\dot{m}R)_2 \ln\frac{P_{2,out}}{P_{2,in}}$	1 represents the cold stream and 2 represents the hot stream	Yilmaz et al. (2001); Hesselgreaves (2000); Bejan (1982); Bejan et al. (1996); Oğulata et al. (2000)

17		Gas-to-gas application	$\dot{S}_{gen} = X\left\{\omega \ln\left[1+\varepsilon\left(\tau^{-1}-1\right)\right]+\ln\left[1+\omega\varepsilon\left(\tau-1\right)\right]+Y\right\}$ where $Y = -\omega\frac{R_1}{c_{p1}}\ln\left(1-\frac{\Delta p_1}{p_1^{\ in}}\right)-\frac{R_2}{c_{p2}}\ln\left(1-\frac{\Delta p_2}{p_2^{\ in}}\right)$	1 represents the cold stream and 2 represents the hot stream and $X = \left(\dot{m}c_p\right)_2$, $\omega = \left(\dot{m}c_p\right)_1 / \left(\dot{m}c_p\right)_2 \leq 1$ and $\tau = T_1/T_2$	Yilmaz et al. (2001)
18		Balanced counterflow heat exchanger, water, with perfect insulation	$\dot{S}_{gen} = \dot{I}/T_0$, where $\dot{I} = \dot{I}^{\Delta T} + \dot{I}^{\Delta P}$ and $\dot{I}^{\Delta T} = T_0\left[\dot{m}c_p \ln\frac{T_{1,out}}{T_{1,in}} + \dot{m}c_p \ln\frac{T_{2,out}}{T_{2,in}}\right]$ and $\dot{I}^{\Delta P} = \frac{\dot{m}}{\rho}\left(P_{1,in}-P_{1,out}\right)+\frac{\dot{m}}{\rho}\left(P_{2,in}-P_{2,out}\right)$	1 represents the cold stream and 2 represents the hot stream	Cornelissen and Hirs (1997)
19		Parallel plates counterflow, single-phase ideal gas fluids, fully developed, laminar or turbulent, adiabatic boundary	$\dot{S}_{gen} = (\dot{m}c_p)_a\left[\ln\frac{T_2}{T_1}-\left(\frac{R}{c_p}\right)_a \ln\frac{P_2}{P_1}\right] + (\dot{m}c_p)_e\left[\ln\frac{T_4}{T_3}-\left(\frac{R}{c_p}\right)_e \ln\frac{P_4}{P_3}\right]$	1 - cold stream in, 2 - cold stream out, 3 - hot stream in, 4 - hot stream out	Ordóñez and Bejan (2000)
20		Counterflow heat exchanger (with zero pressure drop)	$\dot{S}_{gen} = (\dot{m}c_p)_1 \ln\frac{T_{1,out}}{T_{1,in}} + (\dot{m}c_p)_2 \ln\frac{T_{2,out}}{T_{2,in}}$	1 represents the cold stream and 2 represents the hot stream	Bejan (1982); Hesselgreaves (2000)

21		Balanced counterflow with zero pressure drop	$\dot{S}_{gen} = \dot{m}c_p \ln\left[\frac{(1+T_1 NTU/T_2)(1+T_2 NTU/T_1)}{(1+NTU)^2}\right]$	1 represents the cold stream and 2 represents the hot stream and $\varepsilon = NTU/(NTU+1)$	Hesselgreaves (2000)
22		Balanced counterflow with finite pressure drop, constant heat flux, fully developed, perfect gas, 1 - cold stream	$\dot{S}_{gen} = -\dot{m}c_p\left[\frac{A}{2\Delta T}(T_1^{\ 2} - T_{1,out}^{\ \ 2}) - \Delta T\left[\frac{1}{T_{1,out}} - \frac{1}{T_1}\right]\right]$	$A = \frac{fG^2R^2}{2Stp^2c_p}$	Hesselgreaves (2000)
23	**F. Solar Receiver**	Isothermal collector	$\dot{S}_{gen} = \frac{\dot{Q}_0}{T_0} + \frac{\dot{Q}}{T_c} - \frac{\dot{Q}^*}{T^*}$ $= \frac{1}{T_0}\left[\dot{Q}^*\left(1-\frac{T_0}{T^*}\right) - \dot{Q}\left(1-\frac{T_0}{T_c}\right)\right]$		Bejan (1982); Bejan (1997)
24		Non-isothermal collector receiver - where a stream of single-phase fluid circulated through receiver	$\dot{S}_{gen} = \dot{m}c_p \ln\frac{T_{out}}{T_{in}} - \frac{\dot{Q}^*}{T^*} + \frac{\dot{Q}_0}{T_0}$	Pressure drop was neglected	Bejan (1982)

25		Entropy generation due to transformation of monochromatic radiation into blackbody radiation	$$S_{gen} = \frac{S\pi kT}{15^{0.25} h\nu}\left(\frac{\Delta \nu}{\nu}\right)^{-0.25}\left[\exp(h\nu / kT) - 1\right]^{0.25} - S$$	$S =$ original entropy inventory of system where $S = 4U/3T$ and $\Delta\nu/\nu =$ slenderness ratio of frequency band	Bejan (1997)
26		Entropy generation due to scattering	$$S_{gen} = \frac{4}{3} i_{\nu b}\left(\frac{1}{T_2} - \frac{1}{T_1}\right)$$	T_1 and T_2 represent monochromatic radiation temperatures before and after scattering	Bejan (1997)
27	**G. Components for whole system analysis**	Compressor or turbine for ideal gas	$$\dot{S}_{gen} = \dot{m}c_{p1-2}\ln(T_2/T_1) - \dot{m}R\ln(P_2/P_1)$$	1 - in, 2 - out	Jubeh (2005)
28		Ideal gas regenerator	$$\dot{S}_{gen} = \dot{m}c_{p2-6}\ln(T_6/T_2) - \dot{m}R\ln(P_6/P_2) + \dot{m}c_{p5-7}\ln(T_7/T_5) - \dot{m}R\ln(P_7/P_5)$$	2 - 6: cold stream, 5 - 7: hot stream	Jubeh (2005)
29		Open cycle ideal gas regenerator	$$\dot{S}_{gen} = \dot{m}c_{p0}\left[\ln\left[\frac{T_0 T_4}{T_1 T_3}\left(\frac{P_0 P_4}{P_1 P_3}\right)^{(1-k)/k}\right] + \frac{T_2 - T_0}{T_0}\right] + Y$$	1 - 2:cold stream, 3 - 4:hot stream, 0: atmospheric and $Y = \frac{\dot{Q}_{loss}}{T_0}$	Ordóñez and Bejan (2000)

References

Bejan, A., 1982. *Entropy generation through heat and fluid flow*. Colorado: John Wiley.

Bejan, A. 1996. Method of entropy generation minimization, or modelling and optimization based on combined heat transfer and thermodynamics. *Rev Gén Therm*, 35, p. 637 - 646.

Bejan, A., Tsatsaronis, G. and Moran, M. 1996. *Thermal design and optimization*. New York: John Wiley.

Bejan, A. 1997. *Advanced engineering thermodynamics.* 2nd ed. Durham: John Wiley.

Cornelissen, R.L. and Hirs, G.G. 1997. Exergetic optimization of a heat exchanger. *Energy Conversion and Management*, 1 (15-17), p. 1567 - 1576.

Hesselgreaves, J.E. 2000. Rationalisation of second law analysis of heat exchangers. *International Journal of Heat and Mass Transfer*, 43 (22), p. 4189 - 4204.

Jubeh, N.M. 2005. Exergy analysis and second law efficiency of a regenerative Brayton cycle with isothermal heat addition. *Entropy*, 7 (3), p. 172 - 187.

Oğulata, R.T., Doba, F. and Yilmaz, T. 2000. Irreversibility analysis of cross flow heat exchangers. *Energy Conversion and Management*, 41 (15), p. 1585 - 1599.

Ordóñez, J.C. and Bejan, A. 2000. Entropy generation minimization in parallel-plates counterflow heat exchangers. *International Journal of Energy Research*, 24, p. 843 - 864.

Ratts, B.E. and Raut, A.G. 2004. Entropy generation minimization of fully developed internal flow with constant heat flux. *Journal of Heat Transfer*, 126 (4), p. 656 - 659.

Yilmaz, M., Sara, O.N. and Karsli, S. 2001. Performance evaluation criteria for heat exchangers based on second law analysis. *Exergy, an International Journal*, 1 (4), p. 278 - 294.

Zimparov, V. 2001. Extended performance evaluation criteria for enhanced heat transfer surfaces: heat transfer through ducts with constant heat flux. *International Journal of Heat and Mass Transfer*, 44 (1), p. 169 -180.

Zimparov, V.D., Da Silva, A.K. and Bejan, A. 2006a. Thermodynamic optimization of tree-shaped flow geometries with constant channel wall temperature. *International Journal of Heat and Mass Transfer*, 49, p. 4839 - 4849.

Zimparov, V.D., Da Silva, A.K. and Bejan, A. 2006c. Thermodynamic optimization of tree-shaped flow geometries. *International Journal of Heat and Mass Transfer*, 49, p. 1619 - 1630.

Nomenclature

A	Cross-sectional area	m^2
c	Specific heat	J/kgK
D	Diameter	m
f	Friction factor	-
F_D	External drag force	N
G	Mass velocity	kg/sm^2
h	Planck's constant (solar radiation)	-
k	Boltzmann's constant (solar radiation)	-
k	Gas constant (c_p / c_v)	-
k	Thermal conductivity of a fluid	W/mK
i_{vb}	Number of watts arriving per unit area	W
$\dot{I}$	Irreversibility rate	W
L	Length	m
$\dot{m}$	Mass flow rate	kg/s
NTU	Number of transfer units	-
Nu	Nusselt number	-
P	Perimeter	m
P, p	Pressure	Pa
q	Heat transfer rate	W
$\dot{q}''$	Heat transfer flux	W/m^2
$\dot{Q}$	Heat transfer rate	W
R	Gas constant	J/kgK
s	Specific entropy	J/kgK
S	Entropy	J/K
$\dot{S}$	Entropy rate	W/K

$\dot{S}'''_{gen}$	Entropy generation rate per unit volume	W/m^{3}K
St	Stanton number	-
T	Temperature	K
T_0	Fluid flow temperature or environment temperature	K
T^*	Apparent temperature of the sun as an exergy source	K
U	Total energy inventory of the blackbody radiation	J
U_∞	Free-stream velocity	m/s
v	Frequency (solar radiation)	Hz
v	Kinematic viscosity	m^2/s
v	Velocity	m/s
x	Distance in x-direction	m
y	Distance in y-direction	m
ε	Heat transfer effectiveness	-
μ	Dynamic viscosity	kg/ms
ρ	Density	kg/m^3
ϕ	Dimensionless viscous dissipation	-

Subscripts:

0	Surrounding/Loss
0	Zero pressure (ideal gas) for c_p
a	Cold stream
ave	Average
B	Base
c	Collector
D	Based on diameter
e	Hot stream
gen	Generation
h	Hydraulic
i	Channel rank
i	Inlet
in	Inlet
$loss$	Loss

m	Mean
o	Out
out	Outlet
p	For constant pressure
v	For constant volume
w	Wall
x	In x-direction
y	In y-direction
∞	Surrounding area

Superscripts:

*	Solar
‘	Per unit length
.	Time rate of change
_	Average
in	Inlet
ΔP	Due to pressure difference
ΔT	Due to temperature difference

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Appendix C

MATLAB CODE

M-File - *collector*

```
clc;
format long;

global Gyes
global a0
global a1
global a2
global a3
global a4
global a5
global a6
global a7
global a8
global a9
global a10
global tel
global Wn
global As
global r
global choice
global change
global T0g
global Ig
global wg
global P1g
global hightg
global lengthg
global eg
global Tsg
global specreflg
global kg
global tg
global ksolidg
global betag
global alphag

%assumptions:
Ts          = 1050;


Tsurr       = 300;
if change ==1 & T0g>0
    Tsurr = T0g;
end


%Nu           = 14;
%air at 750K
k           = 0.05;


specrefl    = 0.93;
%Duffie & Beckman, 1991, p212
if change ==1 & specreflg>0
    specrefl = specreflg;
end


alpha       = 0.98;
if change ==1 & alphag>0
    alpha = alphag;
end
```

```
beta         = 90;
if change ==1 & betag>0
    beta = betag;
end


%average irradiance:
I            = 1000;
if change ==1 & Ig>0
    I = Ig;
end


%concentrator error:
e            = 0.0067;
if change==1 & eg~=[]
    e = eg;
end


%wind factor: 1-10 times natural
w            = 1;
if change==1 & wg>0
    w = wg;
end


t = 1;

disp('THE RELATIONSHIP BETWEEN THE INTERCEPTED ENERGY AND CAVITY APERTURE DIAMETER,')
disp('WILL NOW BE CALCULATED FOR THE CAVITY RECEIVER')
disp(' ')
As = input('Collector area (m^2): ')
%might want to include a optical efficiency here?


rim = input('Rim angle of collector (in degrees): ')

f = sqrt(As/(4*pi*(sind(rim))^2/(1+cosd(rim))^2))


%This is only a starting guess - a small guess such that it increases
Wn(t) = sqrt(4*(As/50000)/pi);

%According to Reddy & Senhil Kumar (2008 & 2009) for Aw/A1 = 8

WnAs(t) = 8*pi*Wn(t)^2/4;

rsphere(t) = sqrt((WnAs(t) + pi*Wn(t)^2/4)/3/pi);



Qnetfirst = 0.001;

increment = 0.01;

%first calculation


%assume total error



angle = 0;

dangle = 1;

sum = 0;
```

```
while angle < rim

    dr = Wn(t)*cosd(angle)
    p = 2*f/(1+cosd(angle))
    n = 2*atan(dr/2/p)/e

    %from appendix G
    r = 0.2316419;
    b1 = 0.319381530;
    b2 = -0.356563782;
    b3 = 1.781477937;
    b4 = -1.821255978;
    b5 = 1.330274429;

    x = n/2;
    f1 = 1/sqrt(2*pi)*exp(-(x^2)/2);
    t1 = 1/(1+r*x);
    Q = f1*(b1*t1 + b2*t1^2 + b3*t1^3 + b4*t1^4 + b5*t1^5);
    F = 1 - 2*Q;



    slope = 8*pi*I*f^2*sind(angle)/(1+cosd(angle))^2;

    %Global Pressure ratio should be global variable

    %insulation thickness
    Gr = 9.81*(Wn(t)*sqrt(3))^3/(4.765*10^-5/0.8)^2
    Nu = 0.698*Gr^0.209*(1+cosd(beta))^0.968*(3.5)^-0.317*(1/sqrt(3))^0.425

    h = w*Nu*k/2/rsphere(t);
    Qloss(t) = 2*h*pi/4*Wn(t)^2*(Ts-Tsurr);
    kins = 0.05;

    thick(t) = kins*4*pi*rsphere(t)^2*(Ts-Tsurr)/Qloss(t)*10;


    dQinter = specrefl*alpha*F*slope*pi/180*dangle;



    if pi*(rsphere(t)+thick(t))^2 < 4*pi*f^2*(sind(rim))^2/(1+cosd(rim))^2
    sum = sum + dQinter;
    end

    angle = angle + dangle;

end




%aperture used to determine h
Gr = 9.81*(Wn(t)*sqrt(3))^3/(4.765*10^-5/0.8)^2
    Nu = 0.698*Gr^0.209*(1+cosd(beta))^0.968*(3.5)^-0.317*(1/sqrt(3))^0.425
h = w*Nu*k/2/rsphere(t)



Qloss(t) = 2*h*pi/4*Wn(t)^2*(Ts-Tsurr);
%include conduction heat loss
Qloss(t) = Qloss(t) + Qloss(t)/10;



Qnet(t) = sum - Qloss(t)
```

```
iteration = 0;

while pi/4*Wn(t)^2 < As/100
    %this is to make sure the concentration ratio ratio is larger than 100
    %according to Solar (Robert Pitz-Paal)

    iteration = iteration +1;
    t = t+1;

Wn(t) = Wn(t-1) + increment



%According to Reddy & Senhil Kumar (2008 & 2009) for Aw/A1 = 8

WnAs(t) = 8*pi*Wn(t)^2/4;

rsphere(t) = sqrt((WnAs(t) + pi*Wn(t)^2/4)/3/pi);



angle = 0;

dangle = 1;

sum = 0;

while angle < rim

    dr = Wn(t)*cosd(angle);
    p = 2*f/(1+cosd(angle));
    n = 2*atan(dr/2/p)/e

    %from appendix G
    r = 0.2316419;
    b1 = 0.319381530;
    b2 = -0.356563782;
    b3 = 1.781477937;
    b4 = -1.821255978;
    b5 = 1.330274429;

    x = n/2;
    f1 = 1/sqrt(2*pi)*exp(-(x^2)/2);
    t1 = 1/(1+r*x);
    Q = f1*(b1*t1 + b2*t1^2 + b3*t1^3 + b4*t1^4 + b5*t1^5);
    F = 1 - 2*Q;



    slope = 8*pi*I*f^2*sind(angle)/(1+cosd(angle))^2;

    %insulation thickness
    Gr = 9.81*(Wn(t)*sqrt(3))^3/(4.765*10^-5/0.8)^2
    Nu = 0.698*Gr^0.209*(1+cosd(beta))^0.968*(3.5)^-0.317*(1/sqrt(3))^0.425
    h = w*Nu*k/2/rsphere(t);
    Qloss(t) = 2*h*pi/4*Wn(t)^2*(Ts-Tsurr);
    kins = 0.05;

    thick(t) = kins*4*pi*rsphere(t)^2*(Ts-Tsurr)/Qloss(t)*10;

    dQinter = specrefl*alpha*F*slope*pi/180*dangle;


    if pi*(rsphere(t)+thick(t))^2 < 4*pi*f^2*(sind(rim))^2/(1+cosd(rim))^2
```

```
    sum = sum + dQinter;
    end

    angle = angle + dangle;

end




Gr = 9.81*(Wn(t)*sqrt(3))^3/(4.765*10^-5/0.8)^2
    Nu = 0.698*Gr^0.209*(1+cosd(beta))^0.968*(3.5)^-0.317*(1/sqrt(3))^0.425

h = w*Nu*k/2/rsphere(t)


Qloss(t) = 2*h*pi/4*Wn(t)^2*(Ts-Tsurr);
%include conduction heat loss
Qloss(t) = Qloss(t) + Qloss(t)/10;




Qnet(t) = sum - Qloss(t);

end

plot(Wn,Qnet)


p = polyfit(Wn,Qnet,10)

for z = 1:length(Wn)

        PQnet(z) = p(11) + p(10)*Wn(z) + p(9)*Wn(z)^2 + p(8)*Wn(z)^3 + p(7)*Wn(z)^4 +
p(6)*Wn(z)^5 + p(5)*Wn(z)^6 + p(4)*Wn(z)^7 + p(3)*Wn(z)^8 + p(2)*Wn(z)^9 + p(1)*Wn(z)^10;

end

 fid = fopen('gauss.txt','w');

        fprintf(fid,'%6.10f   ',c);
        fclose(fid);

        %it seems that the ALG061 is very sensitive for digits, therefor
        %'%6.10f: meaning 10 digits.

        ALG061

        a0 = p(11);
        a1 = p(10);
        a2 = p(9);
        a3 = p(8);
        a4 = p(7);
        a5 = p(6);
        a6 = p(5);
        a7 = p(4);
        a8 = p(3);
        a9 = p(2);
        a10 = p(1);



        for z = 1:length(Wn)

        FQnet(z) = X(1) + X(2)*Wn(z) + X(3)*Wn(z)^2 + X(4)*Wn(z)^3 + X(5)*Wn(z)^4;

        end

        Figure(1)
```

```
        plot(Wn,FQnet)
        hold on
        plot(Wn,PQnet)
        hold off

        Figure(2)
        plot(Wn,PQnet)
        xlabel('Wn: Aperture diameter (m)');
        ylabel('Qnet intercepted (W)');

        clear X
        clear r
        clear k

%so now, FQnet is a function of Wn, which is a function of D and L of the
%receiver

% thus, if Wn = Dap = sqrt(D*L/2/pi)

%Q* - Qloss = a0 + a1*sqrt(D*L/2/pi) + a2*(D*L/2/pi) + a3*(sqrt(D*L/2/pi))^3 +
a4*(sqrt(D*L/2/pi))^4;
%also, accompanying this equation, are two constraints for the smallest and
%largest diameter
```

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

```
function [F]=fun(X);

% fun
%
% Function evaluation for optimisation. This function should yield
% the objective function value.
%
%   synopsis:
%
%     [F] = fun(X)
%
%   where:
%
%       F = objective function value
%       X = variable vector
%
global Gyes
global a0
global a1
global a2
global a3
global a4
global a5
global a6
global a7
global a8
global a9
global a10
global tel
global Wn
global As
global r
global choice
global change
global T0g
global Ig
global wg
global P1g
global hightg
global lengthg
global eg
global Tsg
global specreflg
global kg
global tg
global ksolidg
global betag
global alphag


%parameters:
ec = Gyes(tel,2)/100;
et = Gyes(tel,5)/100;




rlow = Gyes(tel,3);
rhigh = Gyes(tel,4);
mlow = Gyes(tel,8);
mhigh = Gyes(tel,9);

%for r = rlow:0.05:rhigh


m = (mhigh - mlow)/(rhigh-rlow)*(r-rlow)+mlow;



%Recuperator:
hight = 1;
```

```
%if hightg>0
 %   hight=hightg;
%end


t = 0.001;
%if tg>0
 %  t = tg;
%end

ksolid = 401;
%if ksolidg>0
 %   ksolid=ksolidg;
%end


P1 = 80000;
%if P1g>0
 %   P1=P1g;
%end

T1 = 300;
%if T0g>0
 %   T1=T0g;
%end

T0=T1;


%X(4) = X4/1000 for scaling

%Cold side point3-4:
%assume mu,cold and Pr,c at +-350C
muc = 3.101*10^-5;
Prc = 0.6937;
kc = 0.04721;
cpc = 1056;
rhoc = 0.5664*(r*P1)/100000;

mplate = 2*m/hight*(t+X(4)/1000/X(3)/2*(X(3)+1));


Rec = 4*X(3)*mplate/muc/X(4)*1000/(X(3)+1)^2;

fc = (0.79*log(Rec)-1.64)^-2;

Nuc = fc/8*Prc*(Rec-1000)/(1+12.7*(fc/8)^0.5*(Prc^(2/3)-1));

hc = kc/X(4)*1000*Nuc;


%Hot side point9-10:
%assume mu,hot and Pr,h at +-450C
muh = 3.415*10^-5;
Prh = 0.6965;
kh = 0.05298;
cph = 1081;
rhoh = 0.488*(r*P1)/100000;


Reh = 4*X(3)*mplate/muh/X(4)*1000/(X(3)+1)^2;

fh = (0.79*log(Reh)-1.64)^-2;

Nuh = fh/8*Prh*(Reh-1000)/(1+12.7*(fh/8)^0.5*(Prh^(2/3)-1));

hh = kh/X(4)*1000*Nuh;


Rf = 0.0004;
Asplate = X(5)*X(4)/1000*(X(3)+1)*(1+1/X(3));
```

```
U = (1/hc + 2*Rf + X(5)/ksolid + 1/hh)^-1;

NTU = U*Asplate/mplate/cpc;

c = cpc/cph;

er = (1-exp(-NTU*(1-c)))/(1-c*exp(-NTU*(1-c)));


%constants


R = 287;
cp = 1004;
dT23 = 2;
dT45 = 2;
dT67 = 2;
dT89 = 2;



dP23 = 0.001;



dP34 = (fc*X(5)/(X(4)/1000)^5*8*mplate^2*X(3)^2/(X(3)+1)^4/rhoc)/(P1*r*(1-dP23))


%the pipes going up to the receiver should be smaller in diameter to
%maximize solar availability (according to Shah the pipe losses are 1%)
dP45 = 0.004;
dP56 = 0.04;
dP67 = 0.004;
dP89 = 0.001;



dP910 = (fh*X(5)/(X(4)/1000)^5*8*mplate^2*X(3)^2/(X(3)+1)^4/rhoh)/P1



Qloss23 = 2;
Qloss45 = 2;
Qloss67 = 2;
Qloss89 = 2;
Qlossr = 2;


k = 1.4;

clear i
clear D
clear E


%phase1:


    T2 = (T1*(1+(r^((k-1)/k)-1)/ec));
    T3 = (T2 - dT23);

%phase2:

    %initial guess

    T5 = 800;

     %choice 2 = pipe
    if choice == 2
```

```
        X(6)=1;
    end

    %round/plate - X(6) =1
    Q = a0 + a1*sqrt((X(1)/100)*X(2)/4/pi*(X(6)+1)) + a2*((X(1)/100)*X(2)/4/pi*(X(6)+1))
+ a3*((X(1)/100)*X(2)/4/pi*(X(6)+1))^(3/2) + a4*((X(1)/100)*X(2)/4/pi*(X(6)+1))^2 +
a5*((X(1)/100)*X(2)/4/pi*(X(6)+1))^(5/2) + a6*((X(1)/100)*X(2)/4/pi*(X(6)+1))^3 +
a7*((X(1)/100)*X(2)/4/pi*(X(6)+1))^(7/2) + a8*((X(1)/100)*X(2)/4/pi*(X(6)+1))^4 +
a9*((X(1)/100)*X(2)/4/pi*(X(6)+1))^(9/2) + a10*((X(1)/100)*X(2)/4/pi*(X(6)+1))^5;


    T6 = Q/m/1145 + T5;
    T7 = T6 - dT67;

%phase3:


    P2 = P1*r;
    P3 = P2*(1-dP23);


    P4 = P3*(1-dP34);
    P5 = P4*(1-dP45)

    mu = 4.2*10^-5;
    rho = 0.34*(r*P1)/100000;
    %round/plate:
    if choice==1
        P6 = P5 - ((0.79*log(4*m*X(6)/mu/(X(6)+1)^2/(X(1)/100))-1.64)^-
2)*(X(2)/(X(1)/100)^5)*(8*m^2*X(6)^2/rho/(X(6)+1)^4)
    end

    if choice == 2
        P6 = P5 - ((0.79*log(4*m/mu/pi/(X(1)/100))-1.64)^-
2)*(X(2)/(X(1)/100)^5)*(8*m^2/rho/pi^2);
    end

    P7 = P6*(1-dP67);
    P10 = P1;


    P9 = P10*(1+dP910);
    P8 = P9*(1+dP89);

%phase4:

    T8 = T7*(1-et*(1-1/((P7/P8)^((k-1)/k))));
    T9 = T8 - dT89;
    T10 = T9 - er*(T9-T2);

%phase5:

    T4 = er*(T9-T3)+T3;
    T5 = T4 - dT45;

 D(1) = 800;
 D(2) = T5;



 %____________________________________________________________________
    %repeat:

  i=2;
 while (abs(D(i)-D(i-1))  > 0.001) & (i < 100)
 i = i+1;

 %phase2:

    %initial guess
```

```
    %choice 2 = pipe
    if choice == 2
        X(6)=1;
    end

    %round/plate - X(6) =1
        Q = a0 + a1*sqrt((X(1)/100)*X(2)/4/pi*(X(6)+1)) +
a2*((X(1)/100)*X(2)/4/pi*(X(6)+1)) + a3*((X(1)/100)*X(2)/4/pi*(X(6)+1))^(3/2) +
a4*((X(1)/100)*X(2)/4/pi*(X(6)+1))^2 +  a5*((X(1)/100)*X(2)/4/pi*(X(6)+1))^(5/2) +
a6*((X(1)/100)*X(2)/4/pi*(X(6)+1))^3 + a7*((X(1)/100)*X(2)/4/pi*(X(6)+1))^(7/2) +
a8*((X(1)/100)*X(2)/4/pi*(X(6)+1))^4 + a9*((X(1)/100)*X(2)/4/pi*(X(6)+1))^(9/2) +
a10*((X(1)/100)*X(2)/4/pi*(X(6)+1))^5;


    T6 = Q/m/1145 + T5;
    T7 = T6 - dT67;

%phase3:


    P2 = P1*r;
    P3 = P2*(1-dP23);


    P4 = P3*(1-dP34);
    P5 = P4*(1-dP45);

    mu = 4.2*10^-5;
    rho = 0.34*(r*P1)/100000;
    if choice==1
        P6 = P5 - ((0.79*log(4*m*X(6)/mu/(X(6)+1)^2/(X(1)/100))-1.64)^-
2)*(X(2)/(X(1)/100)^5)*(8*m^2*X(6)^2/rho/(X(6)+1)^4);
    end

    if choice == 2
        P6 = P5 - ((0.79*log(4*m/mu/pi/(X(1)/100))-1.64)^-
2)*(X(2)/(X(1)/100)^5)*(8*m^2/rho/pi^2);
    end

    P7 = P6*(1-dP67);
    P10 = P1;


    P9 = P10*(1+dP910);
    P8 = P9*(1+dP89);

%phase4:

    T8 = T7*(1-et*(1-1/((P7/P8)^((k-1)/k))));
    T9 = T8 - dT89;
    T10 = T9 - er*(T9-T2);

%phase5:

  T4 = er*(T9-T3)+T3;
  T5 = T4 - dT45;

  D(i) = T5;

 end



Sgen1 = m*1007*log(T2/T1) - m*R*log(P2/P1)
Sgen2 = m*1007*log(T3/T2) - m*R*log(P3/P2) + Qloss23/T0
Sgen3 = m*1145*log(T5/T4) - m*R*log(P5/P4) + Qloss45/T0
Sgen4 = m*1070*(log((T4*T10/T3/T9)*(P4*P1/P3/P9)^((1-k)/k)) + (T10-T1)/T0) + Qlossr/T0
Sgen4a = m*1070*(log((T4*T10/T3/T9))+(T10-T1)/T0)-m*R*log((P4*P1/P3/P9))+Qlossr/T0
Sgen5 = m*1145*log(T6/T5) - m*R*log(P6/P5)
Sgen6 = m*1145*log(T7/T6) - m*R*log(P7/P6) + Qloss67/T0
Sgen7 = m*1070*log(T9/T8) - m*R*log(P9/P8) + Qloss89/T0
```

```
Sgen8 = m*1145*log(T8/T7) - m*R*log(P8/P7)

Sgen = Sgen1+Sgen2+Sgen3+Sgen4+Sgen5+Sgen6+Sgen7+Sgen8

%syms Wnet

-Q;
-T0*Sgen;
%objective function

%Wnet = -T0*Sgen + Q + m*1007*(T1-T10) - m*T1*1007*log(T1/T10);
Wnet = -T0*Sgen + Q - m*T1*1007*log(T1/T10);
%Note: The term m*1070*(T10-T1)/T0) at the recuperator entropy generation
% Sgen4, should actually be added here, but as it is it is fine

F1 = m*1145*(T7-T8)-m*1007*(T2-T1);
%T1
%T2
%T3
%T4
%T5
%T6
%T7
%T8
%T9
%T10
%m

%m*1007*(T1-T10)-m*T1*1007*log(T1/T10)

F=-Wnet
F1
```

```
function [GF]=gradf(X);
% gradf
%
% Gradient evaluation for optimisation. This function should yield
% the gradient vector of the objective function.
%
%   synopsis:
%
%    [GF] = gradf(X)
%
%   where:
%
%      GF = gradient vector of the objective function
%       X = variable vector
%

h = 0.00000001;

clear Xgradientp
clear Xgradientn

Xgradientp = X + [h 0 0 0 0 0];
Xgradientn = X - [h 0 0 0 0 0];

GF(1) = (fun(Xgradientp) - fun(Xgradientn))/(h*2);

clear Xgradientp
clear Xgradientn

Xgradientp = X + [0 h 0 0 0 0];
Xgradientn = X - [0 h 0 0 0 0];

GF(2) = (fun(Xgradientp) - fun(Xgradientn))/(h*2);

clear Xgradientp
clear Xgradientn

Xgradientp = X + [0 0 h 0 0 0];
Xgradientn = X - [0 0 h 0 0 0];

GF(3) = (fun(Xgradientp) - fun(Xgradientn))/(h*2);

clear Xgradientp
clear Xgradientn

Xgradientp = X + [0 0 0 h 0 0];
Xgradientn = X - [0 0 0 h 0 0];

GF(4) = (fun(Xgradientp) - fun(Xgradientn))/(h*2);

clear Xgradientp
clear Xgradientn

Xgradientp = X + [0 0 0 0 h 0];
Xgradientn = X - [0 0 0 0 h 0];

GF(5) = (fun(Xgradientp) - fun(Xgradientn))/(h*2);

clear Xgradientp
clear Xgradientn

Xgradientp = X + [0 0 0 0 0 h];
Xgradientn = X - [0 0 0 0 0 h];

GF(6) = (fun(Xgradientp) - fun(Xgradientn))/(h*2);
```

```
function [C]=conin(X);
% conin
%
% Inequality constraint function evaluation for optimisation. This
% function should yield the inequality constraint function values.
%
%   synopsis:
%     [C] = conin(X)
%
%   where:
%
%       C = inequality constraint function values
%       X = variable vector
global Gyes
global a0
global a1
global a2
global a3
global a4
global a5
global a6
global a7
global a8
global a9
global a10
global tel
global Wn
global As
global r
global choice
global change
global T0g
global Ig
global wg
global P1g
global hightg
global lengthg
global eg
global Tsg
global specreflg
global kg
global tg
global ksolidg
global betag
global alphag

  if choice == 2
    X(6)=1;
end

C(1) = (X(1)/100)*X(2)*(X(6)+1)/16-As/100;
C(2) = Wn(2) -sqrt((X(1)/100)*X(2)*(X(6)+1)/4/pi);

%pipe/plate
if choice==1
    C3 = X(1)/100/2*(X(6)+1)-(sqrt(3)-1)/2*sqrt(X(1)/100*X(2)/4/pi*(X(6)+1));
end
if choice == 2
    C3 = 2*X(1)/100 - (sqrt(3)-1)/2*sqrt(X(1)/100*X(2)/2/pi);
end

C(3) = C3;
C(4) = Tsfunc(X) - 1200;
C(5) = -(X(1)/100);
C(6) = -X(2);
C7 = sqrt(As/pi)
C(7) = X(5) - C7;

if choice==1
C(8) = 2.5-X(6);
end
```

```
function [GC]=gradc(X);
% gradc
%
% Gradient evaluation for optimisation. This function should yield
% the gradient vectors of the inequality constraint functions.
%
%   synopsis:
%
%    [GC] = gradc(X)
%
%   where:
%
%      GC = gradient vectors for inequality constraints
%       X = variable vector
%
global Gyes
global a0
global a1
global a2
global a3
global a4
global a5
global a6
global a7
global a8
global a9
global a10
global tel
global Wn
global As
global r
global choice
global change
global T0g
global Ig
global wg
global P1g
global hightg
global lengthg
global eg
global Tsg
global specreflg
global kg
global tg
global ksolidg
global betag
global alphag


GC(1,1) = X(2)/100*(X(6)+1)/16;
GC(1,2) = (X(1)/100)*(X(6)+1)/16;
GC(1,3) = 0;
GC(1,4) = 0;
GC(1,5) = 0;
if choice==1
    GC(1,6) = (X(1)/100)*X(2)/16;
end

if choice==2
    GC(1,6) = 0;
end


GC(2,1) = -X(2)/100*(X(6)+1)/4/pi*0.5*((X(1)/100)*X(2)*(X(6)+1)/4/pi)^-0.5;
GC(2,2) = -(X(1)/100)*(X(6)+1)/4/pi*0.5*((X(1)/100)*X(2)*(X(6)+1)/4/pi)^-0.5;
GC(2,3) = 0;
GC(2,4) = 0;
GC(2,5) = 0;
if choice==1
     GC(2,6) = -(X(1)/100)*X(2)/4/pi*0.5*((X(1)/100)*X(2)*(X(6)+1)/4/pi)^-0.5;
end
```

```
if choice ==2
   GC(2,6)=0;
end



if choice ==1
    GC(3,1) = 1/200*X(6)+1/200-
1648431872091733/180143985094819840/(X(1)*X(2)/pi*(X(6)+1))^(1/2)*X(2)/pi*(X(6)+1);
    GC(3,2) = -
1648431872091733/180143985094819840/(X(1)*X(2)/pi*(X(6)+1))^(1/2)*X(1)/pi*(X(6)+1);
    GC(3,3) = 0;
    GC(3,4) = 0;
    GC(3,5) = 0;
    GC(3,6) = 1/200*X(1)-
1648431872091733/180143985094819840/(X(1)*X(2)/pi*(X(6)+1))^(1/2)*X(1)*X(2)/pi;
end

if choice == 2
    GC(3,1) = 1/50-
1648431872091733/180143985094819840*2^(1/2)/(X(1)*X(2)/pi)^(1/2)*X(2)/pi;
    GC(3,2) = -1648431872091733/180143985094819840*2^(1/2)/(X(1)*X(2)/pi)^(1/2)*X(1)/pi;
    GC(3,3) = 0;
    GC(3,4) = 0;
    GC(3,5) = 0;
    GC(3,6) = 0;
end



h = 0.00001;
GC(4,1) = (Tsfunc(X+[h 0 0 0 0 0])-Tsfunc(X-[h 0 0 0 0 0]))/(2*h);
GC(4,2) = (Tsfunc(X+[0 h 0 0 0 0])-Tsfunc(X-[0 h 0 0 0 0]))/(2*h);
GC(4,3) = (Tsfunc(X+[0 0 h 0 0 0])-Tsfunc(X-[0 0 h 0 0 0]))/(2*h);
GC(4,4) = (Tsfunc(X+[0 0 0 h 0 0])-Tsfunc(X-[0 0 0 h 0 0]))/(2*h);
GC(4,5) = (Tsfunc(X+[0 0 0 0 h 0])-Tsfunc(X-[0 0 0 0 h 0]))/(2*h);
GC(4,6) = (Tsfunc(X+[0 0 0 0 0 h])-Tsfunc(X-[0 0 0 0 0 h]))/(2*h);

GC(5,1) = -1/100;
GC(5,2) = 0;
GC(5,3) = 0;
GC(5,4) = 0;
GC(5,5) = 0;
GC(5,6) = 0;

GC(6,1) = 0;
GC(6,2) = -1;
GC(6,3) = 0;
GC(6,4) = 0;
GC(6,5) = 0;
GC(6,6) = 0;

GC(7,1) = 0;
GC(7,2) = 0;
GC(7,3) = 0;
GC(7,4) = 0;
GC(7,5) = 1;
GC(7,6) = 0;

if choice==1
    GC(8,1) = 0;
    GC(8,2) = 0;
    GC(8,3) = 0;
    GC(8,4) = 0;
    GC(8,5) = 0;

    GC(8,6) = -1;
end
```

```
function [Tsout]=Tsfunc(X);

global Gyes
global a0
global a1
global a2
global a3
global a4
global a5
global a6
global a7
global a8
global a9
global a10
global tel
global Wn
global As
global r
global choice
global change
global T0g
global Ig
global wg
global P1g
global hightg
global lengthg
global eg
global Tsg
global specreflg
global kg
global tg
global ksolidg
global betag
global alphag

%parameters:
ec = Gyes(tel,2)/100;
et = Gyes(tel,5)/100;


 rlow = Gyes(tel,3);
rhigh = Gyes(tel,4);
mlow = Gyes(tel,8);
mhigh = Gyes(tel,9);

%for r = rlow:0.05:rhigh


m = (mhigh - mlow)/(rhigh-rlow)*(r-rlow)+mlow;



%Recuperator:
hight = 1;
%if hightg>0
 %   hight=hightg;
%end


t = 0.001;
%if tg>0
 %  t = tg;
%end

ksolid = 401;
%if ksolidg>0
 %   ksolid=ksolidg;
%end


P1 = 80000;
```

```
%if P1g>0
 %   P1=P1g;
%end

T1 = 300;
%if T0g>0
 %   T1=T0g;
%end

T0=T1;


%X(4) = X4/1000 for scaling

%Cold side:
%assume mu,cold and Pr,c at +-350C
muc = 3.101*10^-5;
Prc = 0.6937;
kc = 0.04721;
cpc = 1056;
rhoc = 0.5664*(r*P1)/100000;

mplate = 2*m/hight*(t+X(4)/1000/X(3)/2*(X(3)+1));


Rec = 4*X(3)*mplate/muc/X(4)*1000/(X(3)+1)^2;

fc = (0.79*log(Rec)-1.64)^-2;

Nuc = fc/8*Prc*(Rec-1000)/(1+12.7*(fc/8)^0.5*(Prc^(2/3)-1));

hc = kc/X(4)*1000*Nuc;


%Hot side:
%assume mu,hot and Pr,h at +-450C
muh = 3.415*10^-5;
Prh = 0.6965;
kh = 0.05298;
cph = 1081;
rhoh = 0.488*(r*P1)/100000;


Reh = 4*X(3)*mplate/muh/X(4)*1000/(X(3)+1)^2;

fh = (0.79*log(Reh)-1.64)^-2;

Nuh = fh/8*Prh*(Reh-1000)/(1+12.7*(fh/8)^0.5*(Prh^(2/3)-1));

hh = kh/X(4)*1000*Nuh;


Rf = 0.0004;
Asplate = X(5)*X(4)/1000*(X(3)+1)*(1+1/X(3));

U = (1/hc + 2*Rf + X(5)/ksolid + 1/hh)^-1;

NTU = U*Asplate/mplate/cpc;

c = cpc/cph;

er = (1-exp(-NTU*(1-c)))/(1-c*exp(-NTU*(1-c)));


%constants


R = 287;
cp = 1004;
dT23 = 2;
```

```
dT45 = 2;
dT67 = 2;
dT89 = 2;

dP23 = 0.001;


 dP34 = (fc*X(5)/(X(4)/1000)^5*8*mplate^2*X(3)^2/(X(3)+1)^4/rhoc)/(P1*r*(1-dP23));



dP45 = 0.004;
dP56 = 0.04;
dP67 = 0.004;
dP89 = 0.001;



dP910 = (fh*X(5)/(X(4)/1000)^5*8*mplate^2*X(3)^2/(X(3)+1)^4/rhoh)/P1;


Qloss23 = 2;
Qloss45 = 2;
Qloss67 = 2;
Qloss89 = 2;
Qlossr = 2;


k = 1.4;

clear i
clear D
clear E


%phase1:


    T2 = (T1*(1+(r^((k-1)/k)-1)/ec));
    T3 = (T2 - dT23);

%phase2:

    %initial guess

    T5 = 800;

    if choice == 2
        X(6)=1;
    end
    %round/plate - X(6) =1

    Q = a0 + a1*sqrt((X(1)/100)*X(2)/4/pi*(X(6)+1)) + a2*((X(1)/100)*X(2)/4/pi*(X(6)+1))
+ a3*((X(1)/100)*X(2)/4/pi*(X(6)+1))^(3/2) + a4*((X(1)/100)*X(2)/4/pi*(X(6)+1))^2 +
a5*((X(1)/100)*X(2)/4/pi*(X(6)+1))^(5/2) + a6*((X(1)/100)*X(2)/4/pi*(X(6)+1))^3 +
a7*((X(1)/100)*X(2)/4/pi*(X(6)+1))^(7/2) + a8*((X(1)/100)*X(2)/4/pi*(X(6)+1))^4 +
a9*((X(1)/100)*X(2)/4/pi*(X(6)+1))^(9/2) + a10*((X(1)/100)*X(2)/4/pi*(X(6)+1))^5;


    T6 = Q/m/1145 + T5;
    T7 = T6 - dT67;

%phase3:

    P2 = P1*r;
    P3 = P2*(1-dP23);


    P4 = P3*(1-dP34);
    P5 = P4*(1-dP45);
```

```
    mu = 4.2*10^-5;
    rho = 0.34*(r*P1)/100000;
    if choice==1
        P6 = P5 - ((0.79*log(4*m*X(6)/mu/(X(6)+1)^2/(X(1)/100))-1.64)^-
2)*(X(2)/(X(1)/100)^5)*(8*m^2*X(6)^2/rho/(X(6)+1)^4);
    end

    if choice == 2
        P6 = P5 - ((0.79*log(4*m/mu/pi/(X(1)/100))-1.64)^-
2)*(X(2)/(X(1)/100)^5)*(8*m^2/rho/pi^2);
    end

    P7 = P6*(1-dP67);
    P10 = P1;


    P9 = P10*(1+dP910);
    P8 = P9*(1+dP89);

%phase4:

    T8 = T7*(1-et*(1-1/((P7/P8)^((k-1)/k))));
    T9 = T8 - dT89;
    T10 = T9 - er*(T9-T2);

%phase5:

    T4 = er*(T9-T3)+T3;
    T5 = T4 - dT45;

 D(1) = 800;
 D(2) = T5;



 %___________________________________________________________________
    %repeat:

  i=2;
 while (abs(D(i)-D(i-1))  > 0.001) & (i < 100)
 i = i+1;

 %phase2:

    %initial guess


    %round/plate - X(6) =1

    if choice == 2
        X(6)=1;
    end
    Q = a0 + a1*sqrt((X(1)/100)*X(2)/4/pi*(X(6)+1)) + a2*((X(1)/100)*X(2)/4/pi*(X(6)+1))
+ a3*((X(1)/100)*X(2)/4/pi*(X(6)+1))^(3/2) + a4*((X(1)/100)*X(2)/4/pi*(X(6)+1))^2 +
a5*((X(1)/100)*X(2)/4/pi*(X(6)+1))^(5/2) + a6*((X(1)/100)*X(2)/4/pi*(X(6)+1))^3 +
a7*((X(1)/100)*X(2)/4/pi*(X(6)+1))^(7/2) + a8*((X(1)/100)*X(2)/4/pi*(X(6)+1))^4 +
a9*((X(1)/100)*X(2)/4/pi*(X(6)+1))^(9/2) + a10*((X(1)/100)*X(2)/4/pi*(X(6)+1))^5;


    T6 = Q/m/1145 + T5;
    T7 = T6 - dT67;

%phase3:


    P2 = P1*r;
    P3 = P2*(1-dP23);


    P4 = P3*(1-dP34);
    P5 = P4*(1-dP45);
```

```
    mu = 4.2*10^-5;
    rho = 0.34*(r*P1)/100000;
    if choice==1
        P6 = P5 - ((0.79*log(4*m*X(6)/mu/(X(6)+1)^2/(X(1)/100))-1.64)^-
2)*(X(2)/(X(1)/100)^5)*(8*m^2*X(6)^2/rho/(X(6)+1)^4);
    end

   if choice == 2
        P6 = P5 - ((0.79*log(4*m/mu/pi/(X(1)/100))-1.64)^-
2)*(X(2)/(X(1)/100)^5)*(8*m^2/rho/pi^2);
    end

    P7 = P6*(1-dP67);
    P10 = P1;


    P9 = P10*(1+dP910);
    P8 = P9*(1+dP89);

%phase4:

    T8 = T7*(1-et*(1-1/((P7/P8)^((k-1)/k))));
    T9 = T8 - dT89;
    T10 = T9 - er*(T9-T2);

%phase5:

  T4 = er*(T9-T3)+T3;
  T5 = T4 - dT45;

  D(i) = T5;

 end

mu = 4.2*10^-5;



  if choice==1
      Tsout =
T6+Q/(X(1)/100)/X(2)/(X(6)+1)/(1+1/X(6))/(0.068/(X(1)/100))/0.023/0.71^0.4/(4*m*X(6)/mu/(
X(1)/100)/(X(6)+1)^2)^0.8;
  end

if choice == 2
    Tsout =
T6+(Q/(X(1)/100)/X(2)/pi)/(0.068/(X(1)/100)*0.023*0.71^0.4*(4*m/mu/pi/(X(1)/100))^0.8);
end
```

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

M-File - *getdata*

```
%uses 'Garrett.txt'

global Gyes
global a0
global a1
global a2
global a3
global a4
global tel
global choice

G = textread('Garrett.txt')


tel =0;

for i = 1:45

 %   if (G(i,10) < Qlow & G(i,11) > Qlow)|(G(i,10) < Qhigh & G(i,11) > Qhigh)|(G(i,10) <
Qmid & G(i,11) > Qmid)
        tel = tel+1;
        for j = 1:11
           Gyes(tel,j) = G(i,j);
       end
   end
%end


Gyes'


%number of turbochargers to be considered
tel

maxQ = max(Qnet)

%get the horsepower range

Qhigh = (maxQ + 0.3*maxQ)/750/0.3*0.4;
Qlow = (maxQ - 0.3*maxQ)/750/0.3*0.4;
Qmid = maxQ/750/0.3*0.4;

tel =0;

for i = 1:45

    if (G(i,10) < Qlow & G(i,11) > Qlow)|(G(i,10) < Qhigh & G(i,11) > Qhigh)|(G(i,10) <
Qmid & G(i,11) > Qmid)
        tel = tel+1;
        for j = 1:11
           Gyes2(tel,j) = G(i,j);
       end
   end
end

if tel>0
    Gyes2'
end
```

M-File - *once*

```
clear all;
clc;

disp('DEFAULT SETTINGS ');
disp(' ')
disp(' ')

disp('Surroundings:')
disp('I          =   1000')
disp('w          =   1')
disp('P1         =   80000')
disp('Tsurr      =   300')
disp(' ')

disp('Available space for recuperator:')
disp('height     =   1;')
disp('length     =   Diameter of dish')
disp(' ')

disp('Collector:')
disp('Ts         =   1050')
disp('e          =   0.0067')
disp('k          =   0.05')
disp('specrefl   =   0.93')
disp('alpha      =   0.98')
disp('beta       =   90')
disp(' ')

disp('Recuperator: ')
disp('t          =   0.001')
disp('ksolid     =   401')
disp(' ')


ask = input('Change default? Y/N ','s');

global change
global T0g
global Ig
global wg
global P1g
global hightg
global lengthg
global eg
global Tsg
global specreflg
global kg
global tg
global ksolidg
global betag
global alphag

if ask == 'Y' | ask == 'y'

    change = 1


    ask2 = input('Change default settings for surroundings? Y/N ','s');
    if ask2 == 'Y' | ask2 == 'y'
            T0g = input('Surrounding temperature (K): ');
            Ig = input('Average irradiance (W/m^2): ');
            wg = input('Wind factor (1-10): ');
            P1g = input('Atmospheric pressure (Pa): ');
    end

    ask2 = input('Change default settings for available space for recuperator? Y/N
','s');
    if ask2 == 'Y' | ask2 == 'y'
            hightg = input('Height of recuperator (m): ');
```

```
            lengthg = input('Maximum length of recuperator (m): ');

    end

    ask2 = input('Change default settings for the collector? Y/N ','s');
    if ask2 == 'Y' | ask2 == 'y'
            eg = input('Concentrator error (rad): ');
  %             Tsg = input('Maximum material surface temperature of receiver (K): (Note
that if this changes, the properties of air should be revised)' );
            specreflg =input('Reflectivity of the collector (<1): ');
  %         kg = input('Material conductivity of receiver: ');
            betag = input('Inclination of receiver in degrees (90degrees for system at
noon/horisontal receiver) ');
            alphag = input('Absorbtivity of receiver: ')

    end

       ask2 = input('Change default settings for the recuperator? Y/N ','s');
    if ask2 == 'Y' | ask2 == 'y'
            tg = input('Recuperator heat exchanger wall thickness between hot and cold
streams (m): ');
            ksolidg = input('Recuperator material conductivity: ');

    end

    end

collector;
ask = input('Run for all (type 52) or just 1 (type 1) ?','s')

    if ask=='1'
        getdata;


       tel=input('Choose number: (0 to stop)')
        while tel~=0


        rlow = Gyes(tel,3)
        rhigh = Gyes(tel,4)

        global r





        while r~=1
             r = input('Choose pressure ratio: ')
            choice = input('Should the receiver use plate or pipe: 1=plate, 2=pipe? ')
        %1=plate
        %2=pipe

            if tel>4 & tel<20
            lfopc([7 8 7 8 7 8],1,1e-7)
            end
            if tel<5
              lfopc([5 5 5 5 5 5],1,1e-7)
            end
            if tel>19
                lfopc([20 20 20 20 20 20],1,1e-7)
            end
              r=input('Select r (type 1 to end): ')


        end
         tel=input('Choose number: (0 to stop)')
        end
    end

    if ask =='52'
```

```
    ask2=input('What is the range of micro-turbines to be looked at? First type the
lowest number: ')
    ask3 = input('Now type the highest number: ')
     choice = input('Should the receiver use plate or pipe: 1=plate, 2=pipe? ')

    global r
    getdata

    clear Willem
    clear Willem2
    qqq=1


            if tel>4 & tel<20
            start = [7 8 7 8 7 8]
            end
            if tel<5
              start = [5 5 5 5 5 5]
            end
            if tel>19
                start = [20 20 20 20 20 20]
            end


        for tel=ask2:1:ask3

            if tel>4 & tel<20
            start = [7 8 7 8 7 8]
            end
            if tel<5
              start = [10 10 10 10 10 10]
            end
            if tel>19
                start = [20 20 20 20 20 20]
            end


            rlow = Gyes(tel,3)+0.001;
            rhigh = Gyes(tel,4)+0.001;

        for r = rlow:0.1:rhigh
            Willem(qqq,1)=r;
            lfopc(start,1,1e-7)
            result = ans;
            for www = 2:1:7
                Willem(qqq,www) = result(www-1);
                start(www-1) =result(www-1)
            end
        Willem2(qqq) = fun(result);
        qqq=qqq+1;
        end

            qqq=qqq+1;
            for www = 1:1:6
                Willem(qqq,www) = 0;
            end
            Willem2(qqq) = 0;

        end


    end
```

Appendix D

GARRETT MICRO-TURBINES

Table D.1 Data for the Garrett micro-turbines (Garrett, 2009).

		Compressor							Turbine
Micro-turbine description	**Micro-turbine model number (*MT*)**	Maximum compressor efficiency	Pressure ratio range for maximum compressor efficiency		Mass flow rate range for maximum compressor efficiency (lb/min)		Mass flow rate range for maximum compressor efficiency (kg/s)		Maximum turbine efficiency
GT1241	1	76	2	2.7	7.7	10.5	0.06087	0.08300	65
GT1544	2	75	1.7	2.25	8	11	0.06324	0.08696	62
GT1548	3	72	1.6	1.8	11	13	0.08696	0.1028	62
GT2052a	4	77	1.4	2.2	10.5	17	0.08300	0.1344	70
GT2052b	5	75	1.6	2	10	14	0.07905	0.1107	70
GT2052c	6	74	1.4	1.7	9	13	0.07115	0.1028	70
GT2056	7	78	1.65	2.5	16	22.5	0.1265	0.1779	65
GT2252	8	78	1.5	2.5	12.5	23	0.09881	0.1818	68
GT2259	9	76	1.65	2.1	13	18	0.1028	0.1423	70
GT2554R	10	71	1.4	2	12	21	0.0949	0.1660	65
GT2560R	11	78	1.3	2	13	26	0.1028	0.2055	65
GT2854R	12	71	1.4	2	12	21	0.0949	0.1660	76
GT2859R	13	75	1.5	2.6	13.5	26	0.1067	0.2055	75
GT2860Ra	14	71	1.5	2.1	12	18	0.0949	0.1423	75
GT2860Rb	15	77	1.5	2.4	16	30	0.1265	0.2372	68
GT2860Rc	16	76	1.5	2.3	16.5	28	0.1304	0.2213	72
GT2871R	17	76	1.6	2.6	20	36	0.1581	0.2846	60
	18	75	1.6	2.5	19	30	0.1502	0.2372	66
	19	75	1.6	2.6	20	35	0.1581	0.2767	66
GT2876R	20	75	1.5	2.25	20	36	0.1581	0.2846	62
GT3071R	21	77	1.7	2.8	23	37	0.1818	0.2925	72
	22	77	1.7	2.8	23	37	0.1818	0.2925	64
GT3076R	23	77	1.5	2.5	20	38	0.1581	0.3004	72
GT3271	24	77	1.9	2.75	20.5	30	0.1621	0.2372	64
GT3582R	25	79	1.5	2.5	26	43	0.2055	0.3399	70
GT3776	26	77	1.9	2.6	28	37	0.2213	0.2925	68
GT3782	27	76	1.75	2.75	27	42	0.2134	0.3320	68
GT3788R	28	78	1.8	2.4	35	48	0.2767	0.3795	71
GT4088	29	74	1.7	3	29	58	0.2292	0.4585	66
GT4088R	30	78	1.8	2.4	35	48	0.2767	0.3795	70
GT4094Ra	31	78	1.7	2.4	37	52	0.2925	0.4111	70
GT4094Rb	32	77	1.5	2.8	30	63	0.2846	0.5138	70
GT4294	33	78	1.8	3	36	65	0.2846	0.5138	74
GT4294R	34	78	1.8	3.1	36	64	0.2846	0.5059	74
GTX4294R	35	80	1.6	2.9	42	68	0.3320	0.5376	74
GT4202	36	77	1.5	2.75	30	70	0.2371	0.5534	74
GTX4202R	37	78	2	3.5	54	78	0.4269	0.6166	74
GT4508R	38	79	1.7	2.8	48	85	0.3794	0.6719	92
GT4708	49	79	1.5	2.7	40	85	0.3162	0.6719	69
GT4718	40	78	1.6	2.8	53	99	0.4190	0.7826	69
GT5533	41	77	1.4	3.3	40	110	0.3162	0.8696	80
	42	77	1.5	2.7	60	116	0.4743	0.9170	80
GT5541R	43	75	1.75	3.2	75	148	0.5929	1.170	80
GT6041a	44	80	1.3	2.5	55	136	0.4348	1.075	78
GT6041b	45	79	1.3	2.8	50	152	0.3953	1.202	78
					lb/min		kg/s		

Reference

Garrett. 2009. *Garrett by Honeywell: Turbochargers, Intercoolers, Upgrades, Wastegates, Blow-Off Valves, Turbo-Tutorials.* Available at: http://www.TurboByGarrett.com [Accessed: 26 April 2010].