Thermodynamic design optimisation of an open recuperative twin-shaft solar thermal Brayton cycle with combined or exclusive reheating and intercooling

Matthew Robert Meas

A thesis submitted in partial fulfilment of the requirements for the degree of Master of Science in Engineering (Mechanical Engineering) in the

Faculty of Engineering and the Built Environment
University of Cape Town

Supervised by
Prof. Tunde Bello-Ochende

March 2017

Declaration

I know the meaning of plagiarism and declare that all the work in this document, save for that which is properly acknowledged, is my own. This dissertation has been submitted to the Turnitin module and I confirm that my supervisor has seen my report and any concerns revealed by such have been resolved with my supervisor.

Signed by candidate

Signature Removed

Matthew R. Meas

Abstract

The Gouy-Stodola Theorem implies that the net power output of a system can be maximised by synchronously sizing the components, thus minimising the cumulative entropy generation rate. The resulting optimal design is related to, and therefore characteristic of, the cycle configuration, since the entropy generation rates in the individual components are interdependent.

In this work, optimal design of three common open solar thermal Brayton cycle variants is investigated and compared using principles of the second law of thermodynamics and the method of entropy generation minimisation. The basic cycle, modified accordingly to construct the reheated, intercooled and combined cases, comprises a modified cavity receiver, a counter-flow plate-type recuperator, and a pair of proprietary automotive turbochargers configured to operate as micro-turbines. An additional modified cavity receiver and cross-flow plate heat exchanger constitute the reheater and intercooler, respectively.

Net power output is expressed in terms of the temperature and pressure fields in each case, defined in terms of geometric variables characteristic of the components. Heat addition is calculated using the receiver sizing algorithm developed by Stine and Harrigan. Maximum constraints are applied to the recuperator and intercooler lengths and to the surface temperatures of the receiver and reheater absorber tubes. The dynamic-trajectory method is implemented to optimise the variables such that the net power output is maximised. An array of inputs are considered and compared, including 22 micro-turbine models, eight concentrator diameters ranging from six to 20 meters, and both circular and rectangular absorber tube profiles. The influence of receiver inclination, concentrator optics, environmental conditions and design constraints are investigated and the optimisation subroutines validated in the Flownex simulation environment.

Results show the optimised power output, operating conditions and design parameters. The intercooled case demonstrates both the highest ratio of total irreversibility to heat input and the highest power output per unit collector surface area. The combined and reheated cases follow. Temperature differences across the components are identified as the primary cause of entropy generation. The optimised heat exchanger lengths are shown to lie on their maximum constraints, and the channel cross-sections found to decrease in size with increasing mass flow rate such that the heat transfer area is maximised and the heat transfer effectiveness improved. As such, plate counts in the optimised heat exchangers are found to be relatively high, and investigation of various compact heat exchanger designs, and regenerative- as opposed to recuperative heat exchangers, is recommended for future work on this topic. The receiver and reheater geometric parameters are found to change such that the absorber tube surface temperatures are kept below the maximum constraint. Trends in the data obtained for circular section absorber tubes are found to be less smooth than the trends in the data obtained for absorber tubes of rectangular section, indicating that the geometric constraints required to maintain the receiver shape offer greater design flexibility for rectangular section absorber tubes than for absorber tubes of circular section.

It is concluded that the increases in the compressor and turbine outlet temperatures with mass flow rate and compressor pressure ratio drive the changes in the temperature differences across the heat exchangers, and thus the component entropy generation rates. The entropy generation rates must in turn be distributed during the optimisation procedure such that the cumulative rate is less than the power output, and all of the constraints are met.

Acknowledgements

I wish to express my sincere thanks to Professor Tunde Bello-Ochende for undertaking to facilitate this research as my supervisor.

I would also like to extend my appreciation to Professor Pieter Rousseau for his assistance in using the Flownex Simulation Environment, and to Mr Ernesto Ismail for his advice on approaching the numerical optimisation.

Thanks also go to the staff at the Energy Research Centre, whose work and teaching have shaped my approach to this research.

Finally, I wish to thank my friends and family for their continued support throughout my academic endeavours.

The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at are those of the author and are not necessarily to be attributed to the NRF.

Sola gratia

Contents

List of Figures

List of Tables

List of Symbols

Greek Symbols

Symbol	Description	Unit
α	Absorbtance	-
β	Coefficient of volume expansion	K^{-1}
β	Inclination angle	deg, rad
η	Efficiency	-
Γ	Capture fraction	-
λ	Wavelength	m
μ	Dynamic viscosity	$\mathrm{kg} \cdot \mathrm{m}^{-1} \cdot \mathrm{s}^{-1}$
Φ	Radiant flux	-
Φ	Receiver rim angle	deg, rad
Ψ	Flow exergy	$\mathrm{J} \cdot \mathrm{kg}^{-1}$
Ψ	Inclination angle	deg, rad
ρ	Density	$\mathrm{kg} \cdot \mathrm{m}^{-3}$
σ_B	Boltzmann constant	$\mathrm{W} \cdot \mathrm{m} \cdot \mathrm{K}^{-4}$
τ	Transmittance	-
ε	Effectiveness, exergetic efficiency, emissivity	-

Roman Symbols

Symbol	Description	Unit
$\dot{I}$	Irreversibility rate	W
$\dot{m}$	Mass flow rate	$\mathrm{kg} \cdot \mathrm{s}^{-1}$

$\dot{Q}^*$	Energy flux through receiver aperture	W
$\dot{Q}_0$	Rate of heat loss by convection and radiation	W
$\dot{Q}_{net}$	Net thermal energy absorbed by the working fluid	W
$\dot{S}$	Entropy rate	$\mathrm{W \cdot K^{-1}}$
$\dot{W}$	Net power output	W
$\dot{X}$	Exergy rate	W
A	Area	$\mathrm{m^2}$
a	Width of rectangular channel	m
b	Height of rectangular channel	m
C	Ratio of heat capacity rates	-
c	Specific heat (c_p or c_v)	$\mathrm{J \cdot kg^{-1} \cdot K^{-1}}$
CR	Concentration ratio	-
D	Receiver cavity diameter	m
d	Receiver aperture diameter	m
D_h	Hydraulic diameter	m
E	Exergy, energy	J
e_p	Parabolic concentrator error	rad
f	Darcy friction factor	-
g	Gravitational constant	$\mathrm{m \cdot s^{-2}}$
GC	Gradient vector of the constraint function	-
GF	Gradient vector of the objective function	-
Gr	Grashof number	-
H	Height	m
h	Convection heat transfer coefficient	$\mathrm{W \cdot m^{-2} \cdot K^{-1}}$
h	Incremental quantity	m
h	Specific enthalpy	$\mathrm{J \cdot kg^{-1}}$
I	Irradiance	$\mathrm{W \cdot m^{-2}}$
k	Heat capacity ratio (C_p/C_v)	-
k	Thermal conductivity	$\mathrm{W \cdot m^{-1} \cdot K^{-1}}$

L	Length	m
MT	Microturbine model number	-
n	Number of flow channels	-
N_c	Number of channels between aperture and receiver wall	-
NTU	Number of thermal transfer units	-
Nu	Nusselt number	-
P	Pressure	Pa
Q	Heat	J
R	Gas constant	$\mathrm{J \cdot kg^{-1} \cdot K^{-1}}$
r	Pressure ratio	-
R_f	Heat exchanger fouling factor	-
R_{dish}	Parabolic dish concentrator radius	m
$refl$	Mirror surface specular reflectivity	-
S	Entropy	$\mathrm{J \cdot K^{-1}}$
T	Temperature	K, °C
t	Thickness	m
t	Time	s
T^*	Apparent temperature of the sun as an exergy source	K, °C
U	Internal energy	J
U	Overall heat transfer coefficient	$\mathrm{W \cdot m^{-1} \cdot K^{-1}}$
V	Velocity	$\mathrm{m \cdot s^{-1}}$
V	Volume	$\mathrm{m^3}$
W	Width	m
W	Work	J
w	Wind factor	-
x	Dimension vector	-
z	Height	m
Pr	Prandtl number	-
Ra	Rayleigh number	-

Re	Reynolds number	-

Subscripts

Symbol	Description
0	Surrounding/environment
ΔP	Frictional pressure drop
ΔT	Temperature drop
∞	Surrounding/free-stream
a	Aperture
atm	Atmospheric
c	Cold stream, compressor
$conc$	Concentrator
$conv$	Convection
CV	Control volume
D	Diameter
e	Electrical, outlet
ext	External
gen	Generation
h	Hot stream
ic	Intercooler
in	Inlet
k	Control volume
m	Mean
p	Plate
rad	Radiation
rec	Receiver
reg	Recuperator
rev	Reversible
s	Surface

$surr$	Surrounding
sys	System
t	Turbine
th	Thermal
w	Inner wall of receiver cavity

Chapter 1

Introduction

1.1 Background

Widespread deployment of small-scale solar thermal power systems could conceivably become financially feasible in the near future. Commercial power generation is tending away from utility-scale plants towards smaller, interconnected, decentralised systems, and there is a global need for rural electrification (US EPA, 2016). Photovoltaic (PV) solar power systems have seen an increase in prevalence over the past decade and are already a viable option for these applications. Thermal solar power plants (SPPs) present further opportunity to diversify and improve the sustainability of energy supply in many regions if implemented in conjunction with PV technology. Large-scale concentrating solar power (CSP) systems based on the Rankine cycle are, however, the only such thermal SPPs to have gained any significant prevalence (IRENA, 2012).

The relatively high temperatures required for efficient operation have historically impeded the development of small-scale thermal solar power generation systems (Le Roux et al., 2013). Rankine cycle SPPs require relatively large condenser elements for heat rejection, since the systems need to be closed to the surroundings to preserve the working fluid. On the other hand, solar powered Stirling engines and Brayton converters require relatively large solar collectors due to the low heat transfer coefficients of the working fluids typically used. Brayton converters can operate on an open-cycle using atmospheric air as the working fluid, thus eliminating the need for a radiator. Accordingly, comparative analyses suggest that strategically

designed SPPs based on the Brayton cycle could offer high power densities and conversion efficiencies more cost-effectively than systems based on the Rankine and Stirling cycles (Mills, 2004).

Solar receivers are being researched to minimise heat loss and maximise heat transfer to the working fluid in thermal SPPs (Le Roux et al., 2014; Ngo, 2012), primarily using the method of entropy generation minimisation (EGM), a unified formulation of the concepts of heat transfer, fluid mechanics, and the second law of thermodynamics. Concomitant application of EGM and the first law of thermodynamics allows variations in the performance of a thermal system in response to changes in a given variable to be better understood (Jubeh, 2005). The synchronous optimisation of several components as part of an integrated solar thermal Brayton cycle (STBC) is also noted, such as the optimisation of the receiver and recuperator of a single-shaft open STBC by Le Roux et al. (2011).

The entropy generation rates in the individual components of a thermodynamic system are interdependent, and the optimal design yielded by the optimisation of a particular system is therefore related to the particular arrangement of the components in the system; that is to say, the optimal geometry for a particular arrangement of components in a thermodynamic system is not necessarily optimal for every arrangement possible. Recognising this, Jansen et al. (2015) investigated two STBC variants, building on the work of Le Roux et al. (2011) and applying EGM to geometrically optimise and compare the maximum net power output from single- and dual-recuperator twin-shaft open-air STBCs. Similar investigation of reheated or intercooled variants of the STBC is yet to be undertaken.

1.2 Research objectives

The aim of this investigation is to examine the effects of reheating and intercooling, exclusively and in combination, on the optimal design and maximum net power output of a small-scale open STBC. Continuing along the research path charted by Le Roux et al. (2011) and Jansen et al. (2015), a set of representative cases are formulated, subject to various simplifying assumptions for the cycle components, and optimised using EGM. The findings are intended to stimulate discussion around the ways in which the performance improvements theoretically attainable through reheating or intercooling might affect further study of small-scale STBCs.

1.3 Scope and limitations

The scope of the investigation is to be limited to the steady state thermodynamic analysis of systems with a maximum power rating of 500 kWe. Dry cooling and solar heat input without combustion are to be assumed. For the sake of simplicity, single objective optimisation is to be undertaken, treating selected component geometry parameters as the independent variables and net power output as the dependent variable to be optimised. The geometric optimisation is to be limited to the components designed for heat transfer to and from the systems, and geometric modelling of the turbomachinery and ducting is thus to be excluded from the scope of investigation.

1.4 Plan of development

The report commences with a review of central concepts from the second law of thermodynamics, and of previous studies on solar thermal power systems and their components. Analysis then proceeds with a problem formulation, in which the cases to be investigated are constructed and modelled using the principles of entropy generation minimisation and the second law of thermodynamics. The optimisation program and the algorithm behind it are briefly described, and the implementation verified through sensitivity analysis. The results follow, showing optimal system performance, operating conditions and component geometry, and salient observations are discussed. Validity and reproducibility of the results are confirmed by simulation in FLOWNEX SE, and the report concludes with a summary of key findings and recommendations for future work.

Chapter 2

Literature survey

Theoretical basis for subsequent development is established in this chapter, which commences with a review of central concepts from the second law of thermodynamics and their application to heat exchangers in power generation systems. Efficient use of solar radiation is subsequently considered, and literature on the optimisation of solar concentrators and receivers is assessed. An evaluation of various thermal solar energy conversion systems follows, prospects for future development are considered, and thermodynamic optimisation of the STBC is finally discussed.

2.1 Entropy and the quality of energy

The first and second law of thermodynamics govern the conservation and transformation of energy, respectively. The quality of an energy stream is termed *exergy*, and is defined as the maximum theoretical useful work obtainable as the system of interest and the environment interact towards equilibrium, with heat transfer occurring between the system and its environment only (Moran, 1999). Exergy is diminished through energy transformation, and according to the Gouy-Stodola Theorem, the lost available work is directly proportional to entropy production, that is to say, exergy is destroyed by irreversibility in the system (Bejan, 1997). Accordingly, the power output of a system can be maximised by minimising the entropy generation. This principle is the cornerstone of the entropy generation minimisation (EGM) method, which unifies the concepts of thermodynamics, fluid mechanics and heat transfer, enabling attributes of the design and performance of a system to be related.

Fundamental equations of the second law of thermodynamics are derived below for a general control volume.

2.1.1 The entropy balance of a control volume

Recall the Clausius inequality, which describes the entropy change associated with a process between two thermodynamic states as

$$\oint \frac{\delta Q}{T} \leq 0 \qquad \text{or} \qquad dS \geq \frac{\delta Q}{T} \tag{2.1}$$

from which it follows that the entropy change associated with an reversible process is equal to the entropy transfer

$$S_2 - S_1 = \int_1^2 \left(\frac{\delta Q}{T}\right)_{rev} \tag{2.2}$$

and that, for a closed system, the entropy change equals the entropy transfer plus the entropy generated by the irreversibilities, (Çengel and Boles, 2011), i.e.

$$S_2 - S_1 \;=\; \frac{\delta Q}{T} + \delta S_{gen} \tag{2.3}$$

A system and its surroundings may be considered two subsystems of an isolated system. Eq. 2.3 can therefore be rewritten as

$$\Delta S_{sys} = \int_1^2 \frac{\delta Q}{T} + S_{gen} \qquad \text{i.e.} \qquad S_{gen} = \Delta S_{total} = \Delta S_{sys} + \Delta S_{surr} \geq 0 \tag{2.4}$$

Now consider the equation for the conservation of energy in a fixed mass of a simple compressible substance subject to an internally reversible process, expressed in differential form as

$$\delta Q - \delta W = dU \tag{2.5}$$

where the differential heat and work quantities can be expressed as

$$\delta Q = T\,dS \qquad \text{and} \qquad \delta W = P\,dV$$

enabling the first Gibbs equation, relating the internal energy and work, to be obtained

$$T\,dS = dU + P\,dV \qquad (1^{\text{st}} \text{ Gibbs eqn.})$$

Note that the Gibbs equation is dependent on properties of state only, and may therefore be applied to an irreversible process between two states as well (Sonntag and Borgnakke, 2009).

The internal energy and the product of the pressure and volume in a thermodynamic system are both encapsulated in the definition of enthalpy

$$h = u + p\,v \qquad \text{i.e.} \qquad dh = du + p\,dv + v\,dp \tag{2.6}$$

which, with the first Gibbs equation, yields the seconds Gibbs equation

$$T\,ds = dh - v\,dp \qquad \text{i.e.} \qquad ds = \frac{dh}{T} - \frac{v\,dp}{T} \qquad (2^{\text{nd}} \text{ Gibbs eqn.})$$

Taking the change in enthalpy with temperature, and the equation of state for an ideal gas

$$dh = c_p\,dT \qquad \text{and} \qquad p\,dv = R\,T$$

and substituting it into the second Gibbs equation enables the entropy change associated with a process between two states to be expressed as

$$ds = \frac{c_p\,dT}{T} - \frac{R\,dp}{p} \qquad \text{i.e.} \qquad s_2 - s_1 = C_p \ln\frac{T_2}{T_1} - R\ln\frac{p_2}{p_1} \tag{2.7}$$

Invoking Eq. 2.4 and denoting the entropy transferred by mass flow and heat transfer as $\dot{m}s$ and $\dot{Q}/T$, then the general entropy balance for a control volume is

$$\frac{dS_{cv}}{dt} \;=\; \dot{m}C_p \ln\frac{T_2}{T_1} - \dot{m}R\ln\frac{P_2}{P_1} \;=\; \sum \dot{m}_i s_i - \sum \dot{m}_e s_e + \sum \frac{\dot{Q}_{CV}}{T} + \dot{S}_{gen} \tag{2.8}$$

Rearranging, and assuming steady flow, the rate of entropy generation in the control volume is

$$\dot{S}_{gen} \;=\; \dot{m}C_p \ln\frac{T_2}{T_1} - \dot{m}R\ln\frac{P_2}{P_1} - \sum \frac{\dot{Q}_{CV}}{T} \tag{2.9}$$

2.1.2 The exergy balance of a control volume

The exergy change of a system is the difference between the net exergy transfer and the exergy destroyed. Exergy can be classified into kinetic, potential, and chemical components, and can be transferred along with heat, mass, or mechanical work. The exergy associated with heat transfer is described by Çengel and Boles (2011) as equivalent to the work produced by a Carnot heat engine operating between the

source and environment temperatures, expressed as

$$\eta = 1 - \frac{T_0}{T_k}, \qquad \text{thus} \qquad X_{heat} = \left(1 - \frac{T_0}{T_k}\right) Q_k$$

and the exergy transferred by work is expressed as

$$X_{work} = \begin{cases} W - W_{surr}, & \text{for boundary work, and} \\ W & \text{for other forms of work} \end{cases}$$

Mass flow is accompanied by proportional exergy content, denoted by

$$X_{mass} = m\psi, \qquad \text{where} \qquad \psi = (h - h_0) - T_0 (s - s_0) + \frac{v^2}{2} + gz$$

Combining the components, the exergy balance of a control volume can be expressed as

$$\begin{aligned} \sum \left(1 - \frac{T_0}{T_k}\right) Q_k - [W - P_0 (V_2 - V_1)] + \sum_{in} m\psi \\ - \sum_{out} m\psi - X_{destroyed} = (X_2 - X_1)_{CV} \end{aligned} \tag{2.10}$$

For a steady flow system, the exergy balance, in rate form, is

$$\sum \left(1 - \frac{T_0}{T_k}\right) \dot{Q}_k - \dot{W} + \sum_{in} \dot{m}\psi - \sum_{out} \dot{m}\psi - \dot{X}_{destroyed} = 0 \tag{2.11}$$

The net work available from the system is therefore

$$\dot{W} = \dot{m} (\psi_{in} - \psi_{out}) + \sum \left(1 - \frac{T_0}{T_k}\right) \dot{Q}_k - \dot{X}_{destroyed} \tag{2.12}$$

where the exergy destroyed is proportional to entropy generated, i.e.

$$\dot{X}_{destroyed} = \dot{I} = T_0 \dot{S}_{gen} \geq 0 \qquad \text{(Guoy-Stodola Th.)}$$

The results of the foregoing derivations constitute a general set of equations that can be readily extended to systems of varying complexity. Said extension depends strongly on the chosen control volume boundary, which defines the system to be optimised, and may include a single component or multiple subsystems. With the choice of system boundary resolved, exergy transfers may be identified and combined into an expression for the performance of the system.

2.1.3 Exergy destruction in heat exchangers

Heat exchangers find widespread application in power generation systems as heater, intercooler, condenser and heat recovery units. As such, the optimisation of heat exchangers as standalone components and in integrated systems receives continuing scholarly consideration, as outlined in the reviews presented by Manjunath and Kaushik (2014).

Li et al. (2011) note the use of flow visualisation and numerical simulation techniques for analysing flow through compact heat exchangers (CHEs), which are characterised by large area densities and small flow passages. In so doing, the relationships between heat transfer coefficient and the high pressure drops and tendency towards laminar flow typical of CHEs can be better understood. Li et al. (2011) describe six major types of CHE and recommend that the all-welded plate, plate-fin, printed circuit, and ceramic heat exchanger CHEs be geometrically optimised under cost and space constraints, and considered for application as high-temperature solar receivers.

Bejan et al. (2013) present a method for designing regenerators based on the constructal law, a theory of design generation developed by Bejan (1997). According to the constructal law, all flow systems, animate and inanimate, have the natural tendency to evolve their configuration in time to promote easier flow. Bejan et al. (2013) extend the law to the design of regenerators by considering concepts based on porous flow architectures observed in nature, after which the constructal law is invoked to refine the design such that optimal heat transfer is achieved.

Guo et al. (2013) propose an alternative method for the optimisation of heat exchangers, based on the relatively new theory of entransy dissipation. However, Sekulic et al. (2015) present a review of literature on the theory, in which they advise against its use, deeming it controversial and indeed unnecessary for all but a few niche purposes.

Application of EGM to a parallel plate counter-flow heat exchanger is described by Ordóñez and Bejan (2000), based on the basic expression for the entropy balance of a control volume. The NTU method is used to determine the outlet temperatures of the fluid streams, and the expression evaluated using the log mean temperature differences to account for temperature variations along the length of the heat exchanger. Hesselgreaves (2000) consider parallel flow heat exchangers and flow

imbalance in counter-flow heat exchangers, and show that it is beneficial to have the cold stream be lower thermal in capacity.

Wang et al. (2005) describe the maximisation of net power output of an endoreversible closed regenerative Brayton cycle with intercooling. Choosing the overall pressure ratio, intercooling pressure ratio, and distribution of the total heat conductance across the heat exchangers as the independent variables, Wang et al. (2005) demonstrate the application of EGM to the analysis of an integrated system in non-dimensional terms.

2.2 Solar collectors

2.2.1 Irradiance

Electromagnetic radiation approximately equal to $6.33 \times 10^7\,\mathrm{W/m^2}$ is emitted at the surface of the sun, the apparent blackbody temperature of which is estimated to be approximately equal to 6 050 K. The radiation travels through space, spreading out at a rate proportional to the distance squared, and reaches the stratosphere having attenuated to a intensity approximately equal to $1\,367\,\mathrm{W/m^2}$ - the solar constant (Stine and Geyer, 2001). Solar irradiance is defined as the radiant solar energy flux incident upon a unit area (Winter et al., 1991) after further attenuation in the lower atmosphere, which can reach up to 30 percent of the nominal value. Global solar irradiance (GHI) spans the range of wavelengths from 0.3 to 3 µm on the electromagnetic spectrum (Çengel and Boles, 2011) and comprises a direct normal component (DNI) as well as a diffuse horizontal component (DHI) scattered into all directions by particles in the atmosphere (Winter et al., 1991). Factors such as location and landscape bring about variations in the solar resource above those associated with changes in weather, time of day and season of the year, and Kalogirou (2004) notes the existence of strategies for addressing variability, including machine learning forecasting methods and hybridisation with natural gas or biofuel, where possible.

Average daily insolation ranges from 4.5 to $6.5\,\mathrm{kWh/m^2}$ for most areas in South Africa (Department of Energy, 2015), with regions in the Northern Cape registering $6\,\mathrm{kWh/m^2}$ in June and up to $8\,\mathrm{kWh/m^2}$ in December (Fluri, 2013). An assumed irra-

diance approximately equal to $1\,000\,\mathrm{W/m^2}$ can therefore be considered appropriate for steady-state analysis of a system located in South Africa.

Goswami (2015), Winter et al. (1991) and Çengel and Boles (2011) treat the subject of solar irradiance from a heat transfer perspective. Degradation of the exergy content of solar irradiance is addressed by Bejan (1997), using the particle model of radiation to calculate the rate of entropy generation in an enclosed space. The irreversibilities associated with the transformation of solar radiation into mechanical power listed by Bejan (1997) are adiabatic free expansion, transformation of monochromatic radiation into blackbody radiation, scattering, and net radiative heat transfer, the upshot of which is a reduction in the temperature of the sun as an exergy source. According to Bejan (1997), entropy generated by adiabatic free expansion such as the throttling of an ideal gas is

$$S_{gen,1-2} = S_1 \left(\frac{T_1}{T_2} - 1 \right) > 0 \tag{2.13}$$

Net radiative heat transfer can be accounted for in the objective function by considering separately any significant radiative components identified when calculating the individual component heat loss rates. Taking into account the effects of atmospheric scattering and the transformation of monochromatic radiation into blackbody radiation, the exergy flux density of solar irradiance can then be expressed as

$$\dot{E} = \eta I \left(1 - \frac{T_0}{T^*} \right) \tag{2.14}$$

where I is the the measured DNI, η is the reflectivity of the reflector, and the apparent temperature of the sun as an exergy source, T^*, is approximately equal to 75 percent of the blackbody temperature of the sun (Farahat et al., 2009).

2.2.2 Collectors

The terms *concentrator*, *receiver* and *collector* are occasionally used interchangeably in the literature to refer to devices for harnessing solar radiation. Distinction between the terms is beneficial to the discussion that follows, hence the following disambiguation: the assembled concentrator and receiver are together referred to as the collector. The element in which radiant solar energy is absorbed and converted into another form is referred to as the receiver, which comprises an absorber, covers, and insulation. Incident radiation may be reflected onto the receiver by the internal

surfaces of the receiver cover (the case with non-concentrating collectors), or reflected and focussed by an external concentrator (Duffie and Beckman, 1976). Common collectors and their characteristics are listed in Table 2.1.

Table 2.1 Concentration ratio, maximum temperature and maximum power obtainable with methods of collecting and absorbing solar radiation in thermal systems (Winter et al., 1991).

	CR	T_{max} (°C)	P_{max}
Flat plate	1	200	1 MW_t
Vacuum tube	< 3	250	1 MW_t
Parabolic trough	40 - 80	400	100 MW_e
Parabolic dish	100 - 250	1 500	150 kW_t/unit
Heliostat	200 - 700	1 500	200 MW_e

Flat plate and vacuum tube collectors provide little or no concentration, and are therefore best suited for low temperature applications such as space- and service water heating. Heliostat and parabolic collectors (single and double curvature) are depicted in Fig. 2.1. Classified as point- or line-concentrating, these collectors are capable of the high temperatures needed to power a heat engine and are given particular consideration by Duffie and Beckman (1976), Goswami (2015) and Winter et al. (1991).

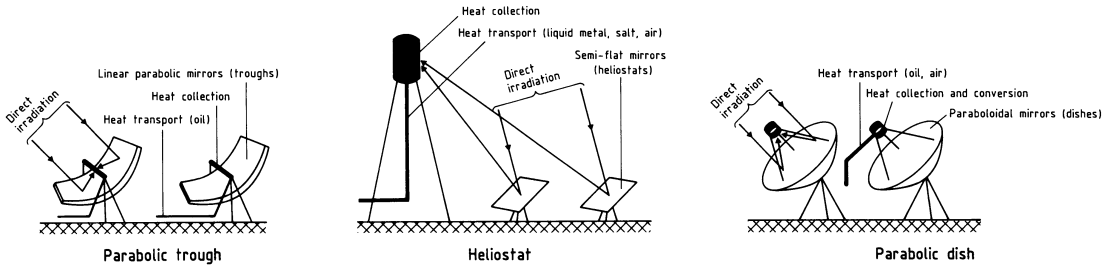


Figure 2.1 Common concentrating solar collectors (Winter et al., 1991).

Optics and efficiency

Duffie and Beckman (1976) define the geometric *concentration ratio* as the ratio of the net aperture or opening area of the collector to the area of the receiver or absorber. Assuming conservation of radiant flux, an increase in the geometric concentration ratio results in an equivalent increase in radiant flux density (Winter

et al., 1991). The angle of the collector rim affects the concentration ratio as shown in Fig. 2.2 (Stine and Geyer, 2001). Note that the curves represent general trends and are not accurately scaled with respect to one another. For a collector featuring a cavity receiver, the maximum concentration ratio corresponds to a rim angle of approximately 45°. Since the average temperature of the receiver fluid increases with increasing concentration ratio, it should be noted that absorber efficiency diminishes with increasing operating temperature, despite the fact that heat engine efficiency increases (Duffie and Beckman, 1976). Reducing thermal and optical losses is therefore critical to achieving an acceptable efficiency overall.

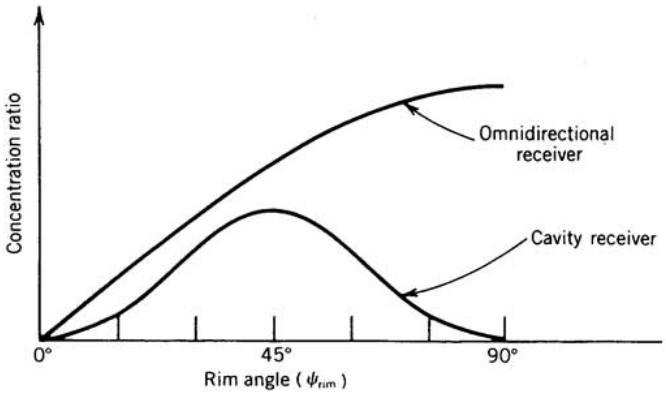


Figure 2.2 Relationship between concentration ratio and rim angle (Stine and Geyer, 2001).

Loss of the resource captured at the solar collector occurs by a variety of mechanisms related to the collector design. According to Stine and Geyer (2001), the optical energy flux at the absorber surface can be generally described with Eq. 2.15 as the product of the resource $I_a A_a$ and the constants Γ, ρ, τ and α representing the capture fraction, reflectance of intermediate surfaces, transmittance of windows and covers, and the absorptance of the absorber surface, respectively.

$$\dot{E}_{opt} = \Gamma \rho \tau \alpha I_a A_a \tag{2.15}$$

For concentration to be effective, the collector must remain oriented towards the sun such that the maximum DNI is captured. The type and extent of solar tracking are determined by the collector geometry, given that the acceptance angle decreases as the concentration ratio is increased. Single curvature collectors of moderate concentration ratio may therefore be intermittently turned about a single axis, provided that the Earth's axial tilt is accommodated (Goswami, 2015). High concentration ratios require dual-axis tracking on a constant basis. An evaluation of the IEA-SSPS project in Almería indicated that optical losses amounted to 42.9

percent of captured irradiance in the single-axis collectors as opposed to 24.3 percent for the dual-axis units (Winter et al., 1991).

Exacting material and optical requirements generally make tracking collectors the most costly components in solar thermal power systems (Murphy and Sallis, 1984). The corresponding pursuit of cost-performance balance calls for innovative design and economy of scale. To this end, the Solar Energy Research Institution proposed a heliostat design to offer low weight, simplified fabrication and high structural efficiency. Figure 2.3 shows the conceptual design in technological context. The heliostat comprises a thin polymeric, composite, or metal membrane stretched on a hollow toroidal frame, and uses a simple vacuum system to actively focus the image.

A cost reduction study by Sandia National Laboratories indicates that cost prices for stretched membrane heliostats amount to 180 $ per m^2 assuming 5 000 units per year and 143 $ per m^2 if 50 000 units are produced annually; approximately 9.8 and 13.5 percent more than conventional glass/metal heliostats, despite weighing approximately 830 kg less (Kolb et al., 2016). The additional costs were attributed to the fabrication of the membrane from one-meter-wide sections of stainless steel film. However, the stretched membrane collector demonstrated improved optical accuracy, which, if taken into account reduces the surpluses to approximately 3.7 and 5.6 percent of conventional heliostat costs, respectively.

Innovative use of modern control technologies can significantly reduce capital costs. A CSP pilot project at Stellenbosch University, uses local controllers to autonomously position the heliostat mirrors, mounted on modular triangular structures, or "pods" (Gauche, 2015). The wireless controllers cost an estimated 10 to 40 percent less than standard alternatives, and eliminate the need for cable trenches. The array, pictured in Fig. 2.4), requires no foundations and can be installed by two people. Helio100 is presently in the pre-commercial stages of development.

2.2.3 Receivers

The basic types of solar receivers are external, or exposed, and internal, or cavity receivers. Common designs are shown in Fig. 2.5. External receivers are typically omnidirectional, and are therefore common in central tower and parabolic trough systems. While mounting the absorber surface in a cavity reduces the acceptance angle, heat loss to the environment is also markedly reduced (Winter et al., 1991).

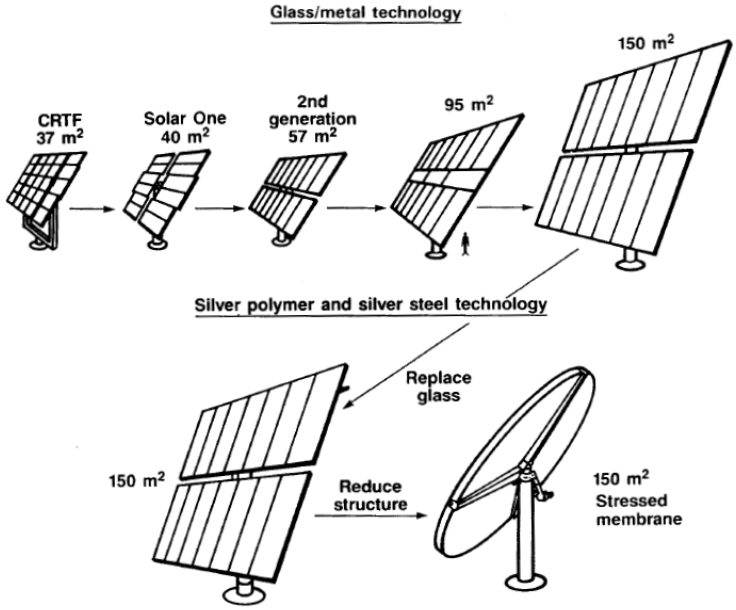


Figure 2.3 Mid-1980s development towards low-cost heliostats (Kolb et al., 2016).

Figure 2.4 Helio100 array at Mariendahl experimental farm (Photo: Helio100).

Although point-focus internal receivers are pictured in Fig. 2.5, linear cavity receivers exist for use with line-focussing collectors. Figure 2.5 also shows typical absorber elements.

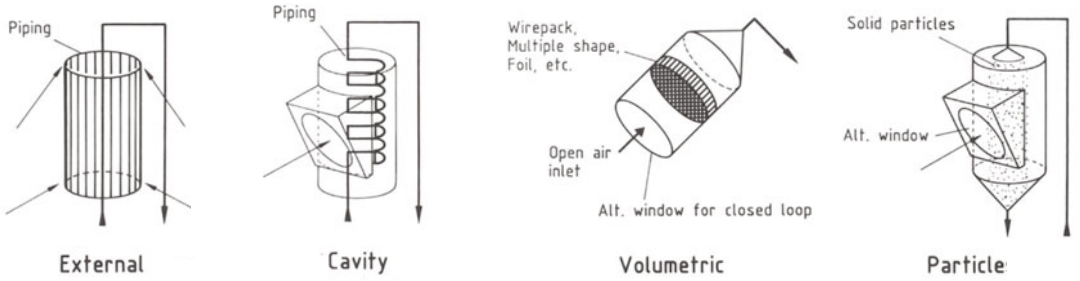


Figure 2.5 Fundamental solar receiver varieties (Winter et al., 1991).

Wire or ceramic heat transfer matrices offer high surface area to volume ratio, low frictional pressure drop, and improved contact with the fluid bulk - as do fluidised particle absorbers. The aforementioned properties promote rapid heat transfer to the thermal carrier fluid, thus enabling internal surface temperatures to be kept within safe limits. Volumetric and particle receivers can consequently withstand high flux densities and heat transfer loads (Winter et al., 1991). Furthermore, they may use air at ambient pressure by routing it through a heat exchanger connected to a separate circuit that contains the working medium of the power block.

The need for a secondary heat exchanger can be eliminated by circulating the conversion process fluid through a tubular absorber element, which may be covered in a selective coating to reduce radiative heat loss. Alternatively, a volumetric or particle receiver may be used, with the cavity sealed from the environment using a window. Internal pressures of several atmospheres can be attained if an inward dome is used instead of a flat window (Winter et al., 1991).

Heller et al. (2006) describe the testing of a volumetric receiver delivering air to a prototypical solar Brayton cycle engine at 1 000 °C. Delivery of air at 1 900 K and nitrogen at 2 100 K is described in an experimental study of a high temperature particle receiver with an 80 mm aperture and a concentration ratio of 2 500 (Bertocchi et al., 2004). Winter et al. (1991) note that high performance receivers generally call for greater cost and complexity. Solutions to material limitations are thus essential to the reduction of fabrication costs associated with high flux receivers (Lovegrove and Pye, 2012).

Losses and efficiency

Given a fixed concentrator area, the amount of thermal energy available for delivery to the conversion process equals the intercepted optical energy minus the heat loss to the surroundings. Cavity receivers can therefore be designed to intercept more optical energy by widening the receiver aperture. Note, however, that the surface between the aperture and the outer perimeter of the receiver housing reduces thermal losses by shielding the absorber from surrounding air. Consequently, a wider receiver aperture relative to a fixed external housing size results in an increased rate of heat loss by radiation and convection (Stine and Geyer, 2001). Larger dimensions overall can increase heat loss through the receiver insulation. These effects are represented graphically in Fig. 2.6 below. Friction loss in the absorber element can cause additional exergy destruction in the receiver, and geometry should consequently be optimised internally and externally to minimise the overall irreversibility.

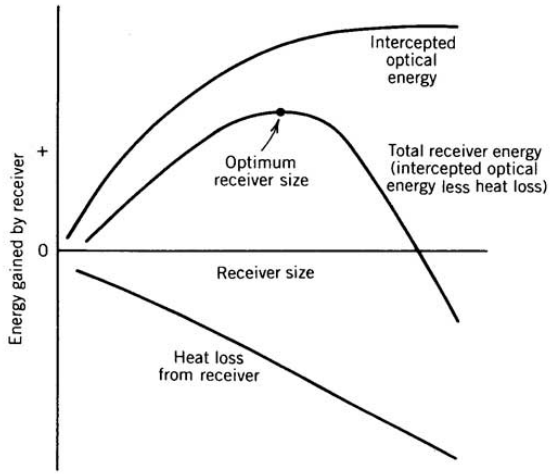


Figure 2.6 Change in energy intercepted and ambient heat loss with respect to receiver size (Stine and Geyer, 2001).

Le Roux et al. (2014) present an analysis using the method of total entropy generation minimisation to optimise the geometry of a cavity receiver for an open and direct STBC. The results show response of the maximum net power output to the effects of optical factors such as concentrator error and specular reflectivity, and variations in irradiance and wind speed. Sendhil Kumar and Reddy (2008), show that thermal losses from a hemispherical cavity receiver can be reduced by modifying the conventional design. Numerical comparison of the modified receiver against conventional designs shows that the design results in minimum heat loss. The reduction is shown to correspond to an area ratio of $A_w/A_a = 8$, where A_w is the inner surface area of the cavity and A_a is the aperture area.

In summary, calculation of the intercepted fraction of the available radiant energy entails the specification of an appropriate cavity geometry and dish concentrator aperture area, and the application of a suitable receiver sizing algorithm. The reader is referred to Appendix A for detail regarding application of the sizing algorithm developed by Stine and Harrigan.

2.3 Solar thermal energy conversion

The thermodynamic power cycles of greatest practical significance to solar thermal energy conversion are the Rankine, Brayton and Stirling cycles, depicted in Fig. 2.7 underneath the Carnot cycle, the theoretical model for describing completely reversible power cycles. The functional characteristics of converters based on the thermodynamic cycles are described below, and documented practical experience with proven conversion technologies is considered.

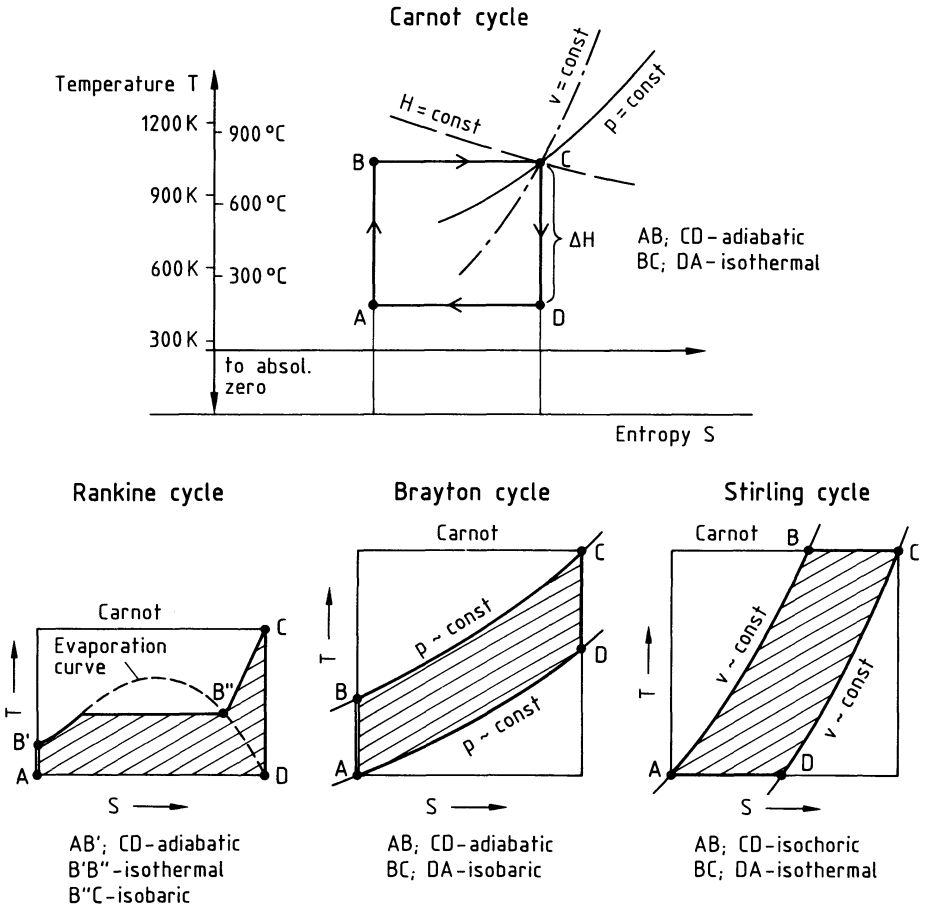


Figure 2.7 Process diagrams of common ideal power cycles (Winter et al., 1991).

2.3.1 Rankine converters

Reference to Fig. 2.7 shows that the Rankine cycle involves transition of the working fluid between liquid and vapour phase. High pressure vaporisation is thereby rendered practical, for a variety of substances, at relatively low, metallurgically manageable temperatures. The pervasiveness of the Rankine cycle in industry and in thermal power stations may be largely attributed to this advantage, as may the development of Rankine converters for each of the concentrators previously described (Lovegrove and Pye, 2012).

Standard Rankine converters use water as the working fluid, and upper cycle temperatures range between 600 °C and 1300 °C for central receiver systems; 200 °C and 600 °C for parabolic trough systems. Variants may use organic fluids such as hydrocarbons, fluorocarbons and silicone oils, or metals with low melting points, such as mercury, caesium, potassium, or lithium (Invernizzi, 2013). Rankine converters can attain gross conversion efficiencies around 40 percent when coupled to AC electrical generators (Lovegrove and Pye, 2012). Small scale reciprocating engines enable greater efficiency than steam turbines, but require maintenance at regular intervals. According to Stine and Geyer (2001), radial- or axial-flow turbines are therefore used, and may be staged, with reheating or bleeding between. Bled steam can be routed from the turbine to a feedwater heater, bypassing the condenser. Doing so can offer greater control over heat addition and rejection, enabling thermal efficiency to be improved.

Regarding condensers, which enable reuse of the working fluid, water-cooled shell-and-tube heat exchangers prevail, although air-cooled units may be used in water-scarce regions, where irradiance can often be particularly attractive (Lovegrove and Pye, 2012). Dry-cooled condensers require larger surface area and greater fan power to facilitate the same heat transfer, but do not require costly water treatment or cause any significant adverse environmental effects.

The Solar Energy Generating Systems (SEGS), in California, comprise nine parabolic trough Rankine converters commissioned between 1984 and 1990. Cumulative capacity is 361 MW, with an average capacity factor[1] of 21 percent (National Renewable Energy Laboratory, 2016). Records show increased annual generation

[1]Ratio of actual output over a time period to potential output assuming continuous operation at nameplate capacity over the same time period.

over time, and SEGS currently sells electricity at 5.57 US cents per kWh, as a result of upgrades implemented since recovery of the capital costs and conclusion of the initial power purchase agreements (CSP Today, 2013).

Ivanpah Solar Electricity Generating System, commissioned in 2014, has a net capacity of 377 MW, a gross annual conversion efficiency of 28.72 percent, and is currently the largest operational thermal SPP based on the Rankine cycle. It comprises three sub-critical, dry-cooled converters with central tower receivers for heating steam from 248.88 °C to 565.55 °C (National Renewable Energy Laboratory, 2016).

2.3.2 Stirling converters

The processes of the ideal Stirling cycle are predisposed to regenerative heat transfer. As Fig. 2.7 shows, gaseous working fluid undergoes isochoric heating and cooling between isothermal compression and expansion processes, theoretically enabling the rejected heat to be temporarily stored, and returned to the fluid in the following iteration. Accordingly, the maximum efficiency of an ideal Stirling engine is theoretically equivalent to that of a Carnot heat engine bounded by the same operating temperatures. Although perfect regeneration is impossible in practice, piston engines designed to emulate the ideal Stirling cycle attain thermal efficiencies between 30 and 40 percent (Mills, 2004). Power ratings are usually small, between 10 and 100 kW, as a Rankine or Brayton converter is generally more efficient or economically viable for large applications. Stirling engines are typically dish-mounted, as upper cycle temperatures range between 600 °C and 800 °C (Stine and Geyer, 2001). Working fluids include air, hydrogen and noble gases, with air somewhat less prevalent in modern engines. At 500 °C, the thermal conductivity of hydrogen is 40 percent greater than that of air (Tarlecki and Lior, 2007). Goswami (2015) notes that noble gases enable better regeneration due to higher heat capacity ratios, given that C_p and C_v apply to the isotherms and the regenerative processes, respectively, and recalling that $k = C_p/C_v$.

Stirling engines fall into two primary categories, namely kinematic, or piston-crankshaft, and free-piston, or linear-oscillating. Power losses are generally greater in the former, due to mechanical friction at the crossheads and the crankshaft. Certain linkages can introduce non-sinusoidal motion, which can exacerbate the frictional losses. The second arrangement requires little maintenance, as the piston

is bounded between the working fluid and a spring, usually a gas spring. The oscillating motion is transferred via magnetic coupling, and is typically used to drive a linear alternator (Lovegrove and Pye, 2012). Unswept cylinder volume can reduce thermal efficiency, and should be kept to the minimum required to enable adequate heat transfer.

The Solar Stirling Array at Tooele Army Depot is presently the largest of its kind in operation. The system comprises 429 dish-mounted Stirling engines using helium as the working fluid, with a cumulative power output of 1.5 MW (National Renewable Energy Laboratory, 2016).

2.3.3 Brayton converters

Brayton converters are essentially gas turbines in which expansion of the working medium is caused by heat derived from solar radiation as opposed to the combustion of hydrocarbon fuel. Open-cycle systems use atmospheric air as the working fluid, whereas closed-cycle systems typically use hydrogen or an inert gas (Lovegrove and Pye, 2012). Closed-cycle converters require a radiator to remove waste heat, and the thermal load that determines the necessary surface area is dependent on the properties of the working fluid.

Stine and Geyer (2001) note that most Brayton converters require an upper cycle temperature of at least 480 °C in order to reach continual operation, and that point-focussing collectors are consequently required (table 2.1). Nominal power ratings of commercial engines are, accordingly, around 100 kW_e (AORA, 2015; Stine and Geyer, 2001), and thermal efficiencies typically range between 23 and 35 percent (Mills, 2004). The H-1 gas turbine developed by Schelde Heron B.V. produces 1 360 kW at 39.5 percent conversion efficiency, using hot air as the working fluid in solar-only operation (Romero et al., 1999), and Turchi et al. (2013) suggest that Supercritical CO_2 Brayton cycles can attain more than 50 percent efficiency under dry-cooling conditions. Stine and Geyer (2001) note that power loss in Brayton engines is primarily due to turbomachinery inefficiencies, ineffective recuperation, and frictional pressure losses in the ducts.

Comparisons indicate that greatest efficiency is attained at a compressor pressure ratio around 2.5, and that a recuperative heat exchanger should be used for heat recovery from the exhaust gas. If heat recovery is not employed, the pressure ratio

must be considerably higher if comparable efficiency is to be attained (Stine and Geyer, 2001). Net power output can be improved by compressing and expanding the fluid in multiple stages, with intercooling and reheating processes between each of the relevant stages. Doing so raises the average temperature of heat rejection, and reduces the average temperature of heat rejection; consequently, recuperation is required to minimise the reduction in efficiency that results when reheating or intercooling is used(Çengel and Boles, 2011). Multi-staging is most advantageous if the turbine- and compressor pressure ratios are equal across each stage (Sonntag and Borgnakke, 2009). The marginal improvement decreases with each additional stage, and more than two or three stages cannot be justified economically (Çengel and Boles, 2011).

The Solar Compressed Air Turbine (SolarCAT) is a prototypical dish-mounted Brayton converter developed by Brayton Energy LLC. The system comprises a $320\,\mathrm{m}^2$ (20 m diameter) parabolic dish built by SouthWest Solar Technologies, and four serially-connected air-cooled, air-lubricated turbo-alternators rated at $53\,\mathrm{kW_e}$ each, which are driven by compressed air heated using a tubular cavity receiver (Brayton Energy LLC, 2011). Levelised electricity cost (LCOE) evaluations show several scenarios where LCOE less than or equal to $\$\,0.10$ per kWh might be possible for a 100 MWe solar farm comprising multiple SolarCAT units, keeping with the methodology and financial assumptions established in the System Advisor Model (Brayton Energy LLC, 2011).

Small-scale commercial Brayton converters are manufactured by AORA Solar (AORA, 2015). The product, based on an open-cycle micro-turbine, produces 100 kW of electrical power and 170 kW of thermal energy, using gas, diesel or bio-fuel to supplement or substitute thermal energy from a solar tower.

2.3.4 Emerging power cycles

Thermodynamic investigations indicate that greater power output and conversion efficiencies are attainable if the standard power cycles are modified or combined (Goswami, 2015). Gupta et al. (2015) present a review of investigations into the theoretical performance improvements attainable by solar-aided thermal power generation systems such as coal-fired power plants with solar feedwater heating. Entirely new cycles are also periodically proposed (Goswami, 1995; Kalina, 1984; Rogdakis and Antonopoulos, 1991).

A balanced Rankine-Brayton power cycle is described by Rovira et al. (2015), for low-to-moderate temperature applications. An indirect heat exchanger facilitates thermal communication between the fluid-loops of the constituent cycles. The authors show that the theoretical thermal efficiency of the configuration, assuming R121b and isobutene working fluids, is 11 percent greater than that of a transcritical R125 Rankine cycle, and 25 greater than that of a CO_2 Rankine cycle (Rovira et al., 2015).

Goodarzi et al. (2014) tender an analysis of a regenerative Brayton cycle with flow from the turbine outlet to the generator partially bypassed and routed to an inverse Brayton cycle. The arrangement is illustrated in (Figure 2.8).

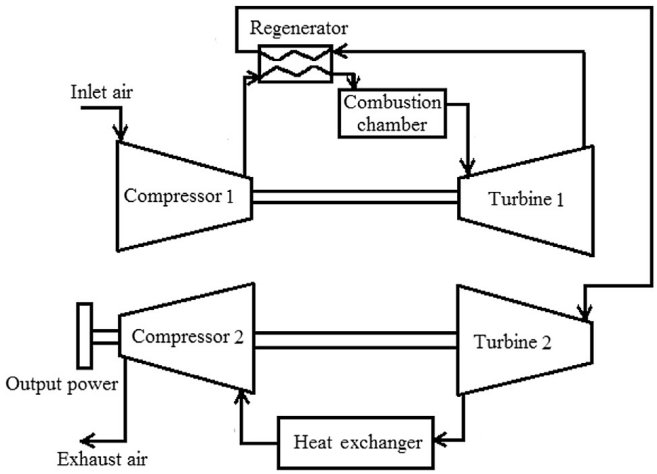


Figure 2.8 Combined regenerative- and inverse Brayton cycle (Goodarzi et al., 2014).

Results of the analysis show that a particular bypass mass flow rate exists for each pressure ratio of the direct cycle turbine at which the thermal efficiency and net output power are independent of the pressure ratio of the second turbine. It is further shown that a particular pressure ratio of the second turbine exists such that the net output power is allowed to vary without affecting the thermal efficiency, that is, power output from the system can be adjusted to follow varying load.

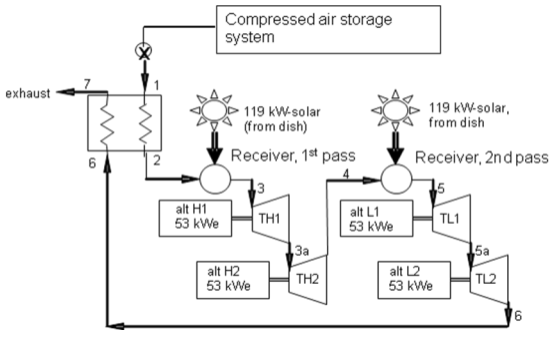
Technical refinement of commercial converters dominates the prototypical development space, and combination cycles such as these are mostly in the conceptual stages of development (Goswami, 2015). A noteworthy exception is the Kalina cycle (Kalina, 1984), which enables several working fluids to be mixed, in effect facilitating multiple power cycles in one hardware system. It is consequently being commercially pursued and optimised for geothermal application (Lovegrove and Pye, 2012).

Solar field	
HT fluid	Water
Number of heliostats	173 500
Heliostat aperture	15 m^2
Tower height	459 ft
Power block	
Technology	Steam Rankine
Turbine capacity (gross)	392 MW
Conversion efficiency	28.72 % (ave., p.a.)
Pressure	160 bar
Cooling method	Dry cooling
Fossil fuel backup	Natural gas
Storage	None

(a) Ivanpah Solar Electric Generating System, California (National Renewable Energy Laboratory, 2016). *Photo: Flickr Commons.*

Solar field	
HT fluid	Helium
Number of dishes	429
Dish aperture	35 m^2
Power block	
Technology	Stirling
Capacity (gross)	1.5 MW (3.5 kW ea.)
Storage	None

(b) Solar Stirling Array, Tooele Army Depot (National Renewable Energy Laboratory, 2016). *Photo: American Galvanizers Association.*



(c) Southwest Solar SolarCAT (Brayton Energy LLC, 2011). *Photo: Dakota Fabricating Inc.*

Figure 2.9 Commercial and prototypical solar thermal energy conversion systems based on the Rankine, Stirling and Brayton cycles.

2.3.5 Energy transport and storage

Energy transport is briefly alluded to in the foregoing discussions of flow in volumetric receivers and in combination systems. However, further discussion is warranted, as the type and design of the conversion system, as well as the power output, efficiency, and capacity factor, depend on the transmission of energy through the fluid-circuitry (Winter et al., 1991). The working medium of the power block is commonly circulated in a single loop, and directly heated by the absorber element, although multiple fluid-loops are sometimes used, linked by one or more indirect heat exchangers. The latter is observed, for instance, in large central tower systems, where HT fluid is heated at the receiver and transported to a heat exchanger to heat the working medium in the power block. Where applicable, thermal storage may be incorporated into the heat exchanger, which, depending on the technology, can require an intermediate fluid-loop and heat exchanger, as illustrated in Fig. 2.10.

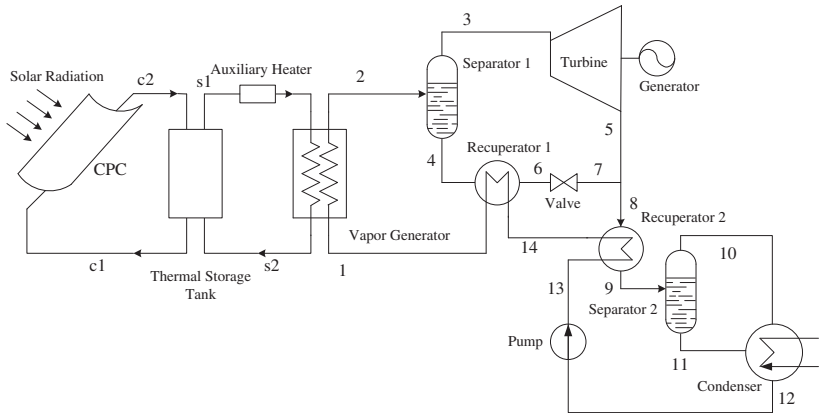


Figure 2.10 Kalina cycle with thermal storage and indirect heating (Wang et al., 2013).

The additional heat exchangers associated with intermediate fluid-circuits essentially increase the thermal inertia of the overall system, and indirect systems, particularly Rankine converters, are consequently less challenging to operate during periods of transient irradiance (Stine and Geyer, 2001). Smaller, lighter receivers can consequently be used, and central receiver towers more cost-effectively constructed, partially offsetting the costs of the additional heat exchangers and piping. Pipe diameters are also typically smaller, resulting in less surface area for heat loss. Direct, single-loop systems are inherently less complex, are capable of higher operating temperatures, and eliminate the heat loss associated with additional heat exchangers and piping (Stine and Geyer, 2001). However, direct parabolic trough CSP systems require high pressures in the collector fields, and scrupulous design to prevent tube burnout.

By enabling energy to be accumulated and retained for extended periods of time, thermal storage technologies enable smoother operation during insolation transients, but also enable higher capacity factors to be reached. The latter is of particular concern to operators and developers of plants needing to meet PPA quota and satisfy investors. Technologies under investigation include thermal storage in molten salt, ceramic bricks, and air-rock systems (Goswami, 2015; Lovegrove and Pye, 2012). Thermal storage has been realised in both large- and small-scale systems. The 250 MW Solana Generating Station (Phoenix, Arizona) uses steam in the Rankine converter and synthetic HT fluid in the parabolic trough collector field, and the 0.25 MW Augustine Fresnel 1 plant (France) uses water only, with a Ruths tank latent heat steam-accumlator for intermediate storage (National Renewable Energy Laboratory, 2016). The effectiveness of thermal storage is dependent on the HT fluids involved, and Stine and Geyer (2001) note that storage is impractical in systems using air as a working medium. The cost and thermal properties of various HT fluids are compared in Table 2.2. For the small-scale regenerative Brayton cycle, a direct system using air as the working fluid seems attractive from a cost perspective.

Table 2.2 Characteristics of working and intermediate fluids (Stine and Geyer, 2001).

	T_{max} (°C)	T_{freeze} (°C)	Cost	Positive	Negative
HT oils	425	-10	High	Low vapour pressure, Suitable for storage	Flammable
Steam	540	0	Low		Requires treatment
Nitrate oils	565	140-220	Medium low	Suited to storage	
Liquid sodium	600	98	Very high	Low vapour pressure, Suitable for storage	
Air or helium	850		Free or very low	Free	Large pipe diameters required

2.4 Thermodynamic optimisation of the solar thermal Brayton cycle

The research would suggest that dish-Brayton solar thermal energy converters have the potential to surpass Stirling engines in the market for modular dual-axis tracking (Mills, 2004). Although early sources are noted to indicate otherwise; to quote Winter et al. (1991) citing a 1984 Sandia report, "Brayton and Rankine cycle converters in dish applications have been assessed as alternatives to Stirling engines, but proved inferior in terms of performance", the outlook appears to have changed, as evidenced by the description by Tsai (2004) of the design and performance of a 22 kW gas turbine engine fashioned from an automotive turbocharger, and the commercialisation of open Brayton cycle microturbine converters by AORA Solar. Le Roux (2011) notes that the Garrett line of turbochargers, from by Honeywell International, is cost-effective, reliable, and readily available, and that performance maps and flow data for the devices are freely accessible from the company website, contrary to the practices of competing manufacturers. Turbocharger turbines are typically capable of withstanding inlet gas temperatures of 800 °C - 1 000 °C (Shah, 2005) - 1 050 °C intermittently, according to the flow data from Garret cited by Le Roux (2011) - limiting the maximum permissible temperature for flow from a solar receiver to the turbocharger turbine. In light of this and other information, thermodynamic optimisation of the STBC is discussed below, with particular consideration of small-scale systems.

2.4.1 Single objective optimisation

Le Roux et al. (2011) applied EGM to optimise the receiver and recuperator geometry of small-scale open-air single-shaft STBC, to maximise the net power output of the cycle. The optimisation was conducted for 21 Garrett micro-turbine models and eight parabolic dish diameters ranging between 6 and 16 m. The receiver was geometrically modelled as a hemispherical tubular cavity receiver, and the recuperator as a parallel plate counter-flow heat exchanger. Le Roux et al. (2011) then applied the Leap Frog Algorithm for Constrained Optimisation by Snyman (2000) to maximise the objective function. Results showed the large temperature difference across the receiver to be primary cause of entropy generation in the system.

Jansen et al. (2015) applied EGM to optimise and compare net power output from single- and dual-regenerator open-air twin-shaft STBCs, also using a parabolic dish collector. Like Le Roux et al. (2011), Jansen et al. (2015) also modelled the receiver as a hemispherical cavity receiver, the regenerators as parallel plate counter-flow heat exchangers, and the turbomachinery using data for a Garrett micro-turbine. The single regenerator cycle was shown to produce considerably more net power than the double regenerator cycle at all mass flow rates, as the optimum proportionality of the component entropy generation rates was such that smaller receiver geometry was necessary to compensate for the irreversibility associated with the second regenerator, allowing less thermal energy to be intercepted and converted.

2.4.2 Multi-objective optimisation

Sánchez-Orgaz et al. (2015) conducted a multi-objective, multi-parametric analysis to determine the optimum number of stages in a recuperative multi-step solar Brayton plant. Most multi-objective problems are such that a solution cannot be found which maximises each objective function precisely, requiring the problem to be solved by "satisficing," i.e. solving such that each objective is reasonably met. This often yields several unique solutions that each have to be measured against a set of criteria for a decision to be made.

Sánchez-Orgaz et al. (2015) used the NSGA-II[2] multi-objective evolutionary algorithm to determine the parameters with the greatest influence on the objective functions and optimise the functions accordingly, after which a distance-normalisation method was used to choose a unique solution. It was found that small variations in the overall pressure ratio, adiabatic coefficient, and ratio of the heat reservoir temperatures lead to rather different values of overall efficiency and maximum power output.

Multi-objective optimisation was also undertaken by Sadatsakkak et al. (2015), using NSGA-II to simultaneously optimise three objective functions describing the power output, ecological function and thermo-economic performance of an irreversible regenerative closed Brayton cycle. Likewise, Li et al. (2015), used NSGA-II to generate a front of Pareto solutions for thermo-economic objective functions describing the performance of a hybrid solar and fossil fuel Brayton system,

[2]Second generation Non-dominated Sorting Genetic Algorithm

from which a final solution was selected using the Shannon Entropy, LINMAP[3] and TOPSIS[4] decision making methods. The findings of Sánchez-Orgaz et al. (2015) demonstrate the utility of multi-objective and multi-parametric techniques for sensitivity analysis.

2.5 Summary

Application of EGM to the thermodynamic optimisation of components such as parallel plate heat exchangers and hemispherical cavity receivers is well documented in the literature (Ordóñez and Bejan, 2000; Sendhil Kumar and Reddy, 2008). Recent expansion of the method to the geometric optimisation of integrated systems comprising multiple components is also noted (Kalina, 1984; Rovira et al., 2015).

The research shows that small solar-powered Stirling engines are technologically mature and highly efficient, and that large scale converters based on the Rankine cycle dominate commercial development (Mills, 2004). Open cycle dish-Brayton converters appear to be a cost effective alternative for moderate (100 kW to 10 MW) power output applications (Romero et al., 1999).

Integrated geometric optimisations have been undertaken, using EGM, to determine maximum power output from a single-shaft open-air STBC (Le Roux et al., 2011), as well as from single- and dual-regenerator twin-shaft open cycles (Jansen et al., 2015). The method is yet to be applied to the integrated geometric optimisation to determine net power output from reheated or intercooled variants of the STBC.

Thermodynamic analyses of the STBC based on multi-criteria decision making are noted in the literature (Sadatsakkak et al., 2015; Sánchez-Orgaz et al., 2015), as are analyses of compact heat exchangers, volumetric receivers and pressurised particle receivers (Bertocchi et al., 2004; Li et al., 2011). The approaches outlined in these analyses, however, seem more appropriate for refinement of thermodynamic models than for a first-pass analysis of an integrated thermodynamic system, which is perhaps best done using simpler, conventional geometric models and single-objective mathematical optimisation.

---

[3]Linear programming techniques for Multidimensional Analysis of Preferences

[4]Technique for Order of Preference by Similarity to Ideal Solution

Chapter 3

Problem formulation

This chapter details the formulation of the optimisation problems related to the objective - investigating the effects of reheating and intercooling on the optimal design and maximum net power output of a small-scale open STBC. The cases to be optimised are formally described and schematically represented in section 3.1 below. Steady state operation is assumed, as per the scope and limitations outlined in the introductory chapter. Recall that the Guoy-Stodola Theorem, introduced in chapter 2, states that the power output can be maximised by minimising the entropy generation, and consequently the irreversibility, in each of the constituent components. Also recall that entropy generation in a control volume may occur as a result of mass flow, boundary work, frictional pressure losses and heat loss, and that the temperatures and pressures at the inlet and outlet are needed to evaluate the entropy balance equation. With each of the cases clearly defined, the flow, heat transfer and associated entropy generation in each of the components is modelled in terms of selected geometric variables, in section 3.2. The formulation of an objective function for each case follows in sections 3.3 and 3.4, after which the constraint functions describing the candidate solutions are derived, in section 3.5.

3.1 Case definitions

The specific cases to be investigated are presented below. All cases are based on the open and direct regenerative STBC with a parabolic dish collector and two-stage turbomachinery, which previous work reviewed in the literature survey indicates might be the most simple and cost-effective basis for small-scale Brayton converters.

Reference - Recuperation: Flow leaving the high-pressure (HP) compressor is routed through the recuperator to recover waste heat from the low-pressure (LP) turbine exhaust before being routed to the HP turbine via the receiver.

Case I - Reheating: The reference case is modified to incorporate an additional solar collector between the HP and LP turbines. Both identical and distinct receiver and reheater concentrator diameters are considered.

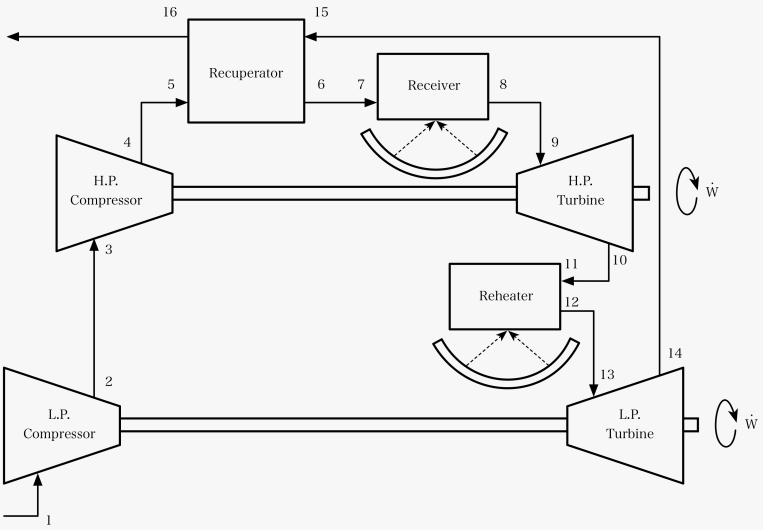


Figure 3.1 Case I: Recuperation and reheating only.

Case II - Intercooling: The reference case is modified to incorporate a natural draft air-to-air intercooler between the LP and HP compressors.

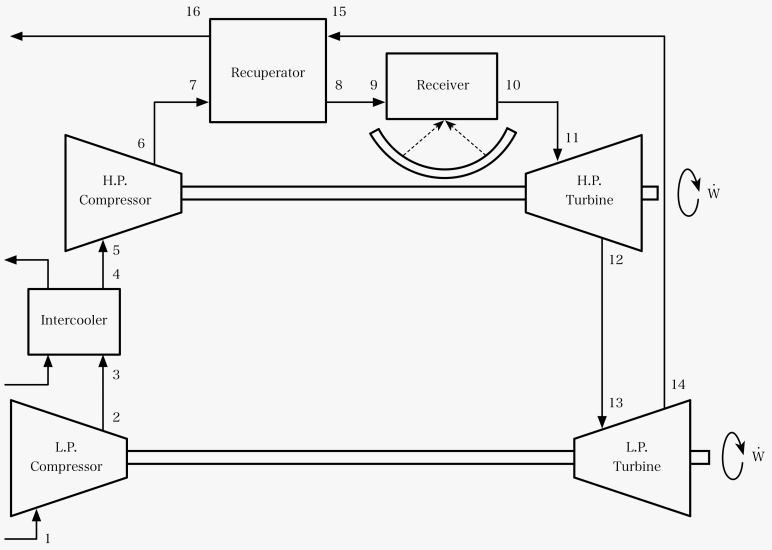


Figure 3.2 Case II: Recuperation and intercooling only.

Case III - Reheating and intercooling: The reference case is modified to incorporate a natural draft air-to-air intercooler and additional solar collector between the compressors and turbines respectively.

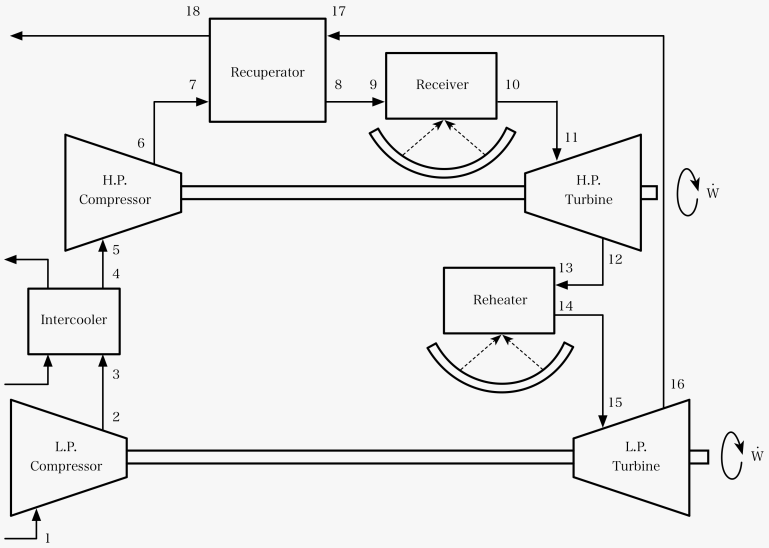


Figure 3.3 Case III: Recuperation, reheating and intercooling.

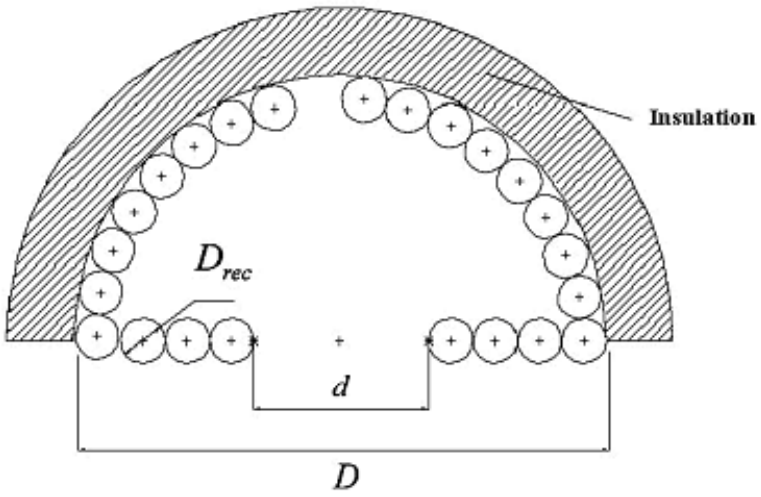
3.2 Component models

3.2.1 Receiver

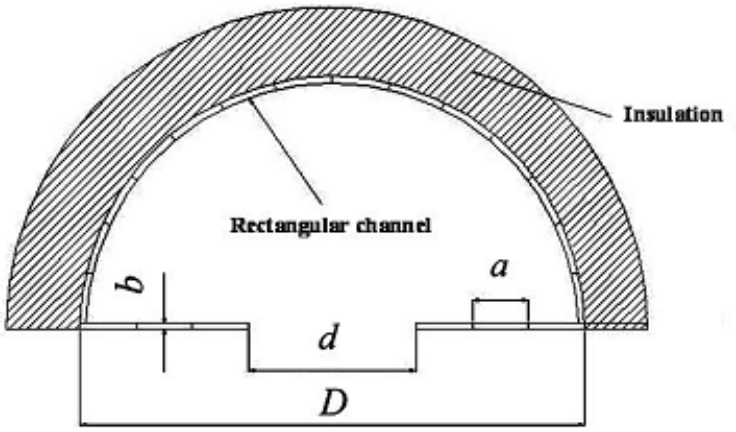
A modified cavity receiver design is considered in the analysis, on account of the observations drawn from the work of Sendhil Kumar and Reddy (2008). The design, shown schematically in Figs. 3.4a and 3.4b, comprises an absorber tube, of circular or rectangular cross-section, wound closely in an outward spiral around a central hole, and upwards in concentric rings to form a hemispherical coil. Both cross-sections are considered in the analysis. The working fluid circulates through the tube, gaining heat through the surfaces of the tube exposed to the hot air inside the cavity, and the external surfaces are insulated to minimise heat loss. The absorbed energy is the difference between the intercepted energy and the rate of heat loss, governed by the size of the aperture relative to the inner wall, i.e.

$$\dot{Q}_{net} \;=\; \dot{Q}^{*} - \dot{Q}_{0} \tag{3.1}$$

Sendhil Kumar and Reddy (2008) show that the rate of heat loss from the receiver is minimised when the area ratio is $A_w/A_a = 8$. With the area ratio fixed, the aperture diameter d and inner wall diameter D are expressed in terms of the length and diameter of the absorber tube, as described below.



(a) Circular cross-section absorber tube (Le Roux et al., 2011).



(b) Rectangular cross-section absorber tube (Le Roux et al., 2011).

Figure 3.4 Modified hemispherical cavity receiver design developed by Sendhil Kumar and Reddy (2008).

The net surface area of the inner receiver wall can be expressed both as the difference between the formulae for the surface area of a hemispherical shell and of a circle

$$\begin{aligned} A_w &= \frac{3\pi D^2}{4} - A_a \\ &= \frac{3\pi D^2}{4} - \frac{\pi d^2}{4} \end{aligned} \tag{3.2}$$

and as the product of the length and width of the absorber tube

$$A_w = aL_{rec} \tag{3.3}$$

where the width of the tube is expressed as

$$a = \frac{D_{h,rec}}{2}\left(\left(\frac{a}{b}\right)_{rec} + 1\right) \tag{3.4}$$

from the formula for the hydraulic diameter of a rectangular duct. Note that Eq. 3.4 can be used for circular cross-section tubes by setting the aspect ratio equal to $\left(\frac{a}{b}\right)_{rec} = 1$. The aperture diameter is then written in terms of the tube dimensions by solving Eqs. 3.2 and 3.3 simultaneously and substituting Eq. 3.4 into the result to obtain Eq. 3.5 (Le Roux et al., 2011).

$$d_{rec} = \sqrt{\frac{D_{h,rec}L_{rec}\left(\left(\frac{a}{b}\right)_{rec} + 1\right)}{4\pi}} \tag{3.5}$$

Rearranging Eq. 3.2 and setting $A_w = 8A_a$, the inner wall diameter is related to the aperture diameter and dimensions of the tube, with the use of Eq. 3.6 (Le Roux et al., 2011).

$$\begin{aligned} D &= 2\sqrt{\frac{A_w + A_a}{3\pi}} = 2\sqrt{\frac{9A_a}{3\pi}} \\ &= 2\sqrt{\frac{3d^2}{4}} = \sqrt{3}d \end{aligned} \tag{3.6}$$

Reddy and Kumar (2008) show that the ratio of radiative to convective heat loss from the receiver, at 400 °C, is 59:41 when $A_w/A_a = 8$, and the inclination of the receiver relative to the vertical plane is 90°, i.e. the cavity aperture is directed downwards. At 0° inclination, i.e. when the aperture is directed towards the horizon, air inside the cavity is more easily displaced by buoyancy-driven natural convection, changing the ratio to 71:52. The ratios for $A_w/A_a = 10$, equal to 47:20 and 57:43 at 90° and 0° inclination respectively, reflect the greater influence of convective heat loss for larger volumes of stagnant air in the receiver cavity (Reddy and Kumar, 2008). Le Roux (2011) notes that the calculations can be simplified, assuming similar heat loss proportions at the high receiver temperatures associated with the concentration ratios under consideration, as it follows that $\dot{Q}_{loss,rad} \approx \dot{Q}_{loss,conv}$, i.e.

$\dot{Q}_0 \approx 2\ \dot{Q}_{loss,conv}$, enabling the Nusselt number to be written as

$$Nu_D \;=\; 0.698 Gr_D^{0.209} \left(1 + \cos\beta\right)^{0.968} \left(T_w/T_\infty\right)^{-0.317} \left(d/D\right)^{0.425} \tag{3.7}$$

in terms of the dimensions, inclination and wall temperature of the receiver (Reddy and Kumar, 2009). As noted in the preceding chapters, the maximum temperature in the cycle is limited by material considerations. The wall temperature T_w is therefore treated together with the geometric constraints, in section 3.5. Substituting the convection heat transfer coefficient

$$h_{conv} \;=\; \frac{Nu_D k}{D} \tag{3.8}$$

into Eq. 3.7, the rate of radiative and convective heat loss from the receiver cavity is written as

$$\begin{aligned} \dot{Q}_0 \;=\; & 1.396 Gr_D^{0.209} \left(1 + \cos\beta\right)^{0.968} \left(T_w/T_\infty\right)^{-0.317} \\ & \left(d/D\right)^{0.425} \left(kA_a/D\right) \left(T_w - T_\infty\right) \end{aligned} \tag{3.9}$$

With the heat loss rate and aperture diameter known, the net absorbed heat rate is calculated using the equation

$$\begin{aligned} \dot{Q}_{net,rec} \;=\; & p_0 + p_1\, d_{rec}^{1/2} + p_2\, d_{rec} + p_3\, d_{rec}^{3/2} + p_4\, d_{rec}^2 + p_5\, d_{rec}^{5/2} \\ & + p_6\, d_{rec}^3 + p_7\, d_{rec}^{7/2} + p_8\, d_{rec}^4 + p_9\, d_{rec}^{9/2} + p_{10}\, d_{rec}^5 \end{aligned} \tag{3.10}$$

where the coefficients p_1 to p_{10} are calculated using the receiver sizing algorithm by Stine & Harrigan (Appendix A).

Finally, equation 2.9 is used to derive the expression for the rate of entropy generation in the receiver

$$\dot{S}_{gen,rec} \;=\; -\frac{\dot{Q}^*}{T^*} + \frac{\dot{Q}_0}{T_0} + \dot{m} C_{p0} \ln\frac{T_e}{T_i} - \dot{m} R \ln\frac{P_e}{P_i} \tag{3.11}$$

where the boundary of the control volume is drawn around the receiver. The term $\dot{Q}^*$ is the sum of the net absorbed heat transfer rate and the heat loss rate (Eq. 3.1) as opposed to the exergy rate of solar irradiance $\dot{E}$ (Eq. 2.14). The apparent sun temperature T^* is taken to be approximately equal to $0.75 \times 6\,050 = 4\,500\,\mathrm{K}$ (Farahat et al., 2009; Stine and Geyer, 2001).

3.2.2 Recuperator

A counter-flow parallel plate heat exchanger design is utilised for the recuperator, as advocated by (Shah, 2005). The design is shown schematically in Fig. 3.5. Denoting the number of flow channels by n, then the hot and cold fluid streams are each reticulated into $n/2$ channels. Fixing the total height of the recuperator enables the overall size to be kept within reasonable bounds and the remaining dimensions to be related, as described below.

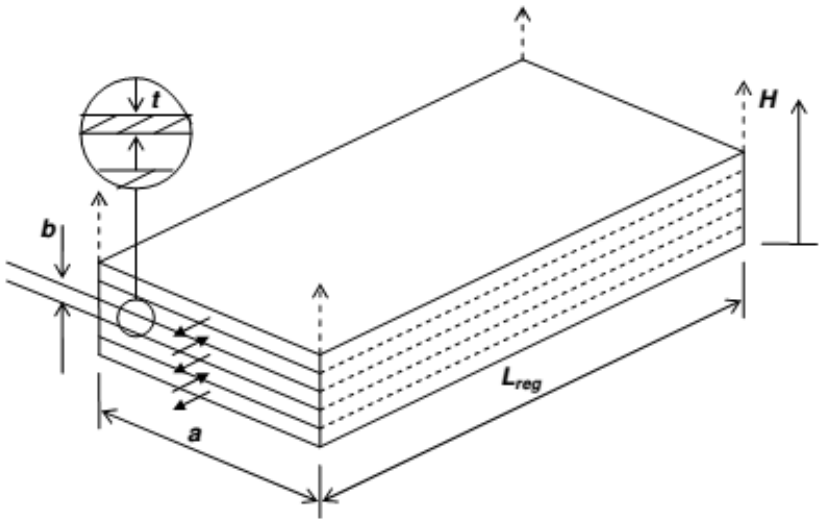


Figure 3.5 Counter-flow plate-type recuperator (Le Roux et al., 2011).

The number of channels that can be fit into the overall recuperator height H is

$$n = \frac{H}{t+b} \tag{3.12}$$

where t is the plate thickness, and b the channel height, written as

$$b = D_{h,reg}\frac{\left(\frac{a}{b}\right)_{reg}+1}{2\left(\frac{a}{b}\right)_{reg}} \tag{3.13}$$

by rearranging the formula for the hydraulic diameter of a rectangular duct. Accordingly, the mass flow rate through a single channel is

$$\dot{m}_c = \frac{2\dot{m}}{n} \tag{3.14}$$

and the channel surface area is

$$A_{s,reg} \;=\; 2\,(a+b)\,L_{reg} \;=\; D_{h,reg}L_{reg}\left(\left(\tfrac{a}{b}\right)_{reg}+1\right)\left(1+\left(\tfrac{a}{b}\right)^{-1}_{reg}\right) \tag{3.15}$$

The recuperator effectiveness is then determined, as subsequently described. For the sake of simplicity, heat loss from the outermost plate surfaces is expressed in terms of a small assumed temperature drop, and longitudinal conduction in the plates is not considered. Using the expression for the velocity of flow through the channels

$$V \;=\; \frac{\dot{m}}{\rho ab} \;=\; \frac{4\dot{m}\left(\frac{a}{b}\right)_{reg}}{\rho D^2_{h,reg}\left(\left(\frac{a}{b}\right)_{reg}+1\right)^2} \tag{3.16}$$

the formula for Reynolds number is written as

$$Re \;=\; \frac{\rho V D_{h,reg}}{\mu} \;=\; \frac{4\dot{m}_c\left(\frac{a}{b}\right)_{reg}}{\mu D_{h,reg}\left(\frac{a}{b}\right)^2_{reg}} \tag{3.17}$$

noting that the viscosity μ differs for the hot and cold fluid streams. The Darcy friction factors are then determined using the first Petukhov equation (Çengel, 2004)

$$f \;=\; \sqrt{0.79\ln\left(Re-1.64\right)} \tag{3.18}$$

the Nusselt numbers determined using the Gnielinski equation correlation

$$Nu_D \;=\; \frac{Pr\,(f/8)\,(Re-1000)}{1+12.7\,(f/8)^{1/2}\,(Pr^{2/3}-1)} \tag{3.19}$$

and the convective heat transfer coefficients determined using the relation

$$h \;=\; \frac{kNu_D}{D_{h,reg}} \tag{3.20}$$

Including perpendicular conduction through the plates, and the thermal resistance R_f due to fouling on either side, the overall heat transfer coefficient is

$$U \;=\; \left(\frac{1}{h_{cold}}+2R_f+\frac{t}{k}+\frac{1}{h_{hot}}\right)^{-1} \tag{3.21}$$

Noting the equivalence of the mass flow rate for the hot and cold streams, the heat capacity ratio is then written as

$$C = \frac{\dot{m}_c C_{p,cold}}{\dot{m}_c C_{p,hot}} = \frac{C_{p,cold}}{C_{p,hot}} \tag{3.22}$$

and the ε-NTU relations for a counter-flow heat exchanger (Kays and London, 1984) used to determine the number of thermal units

$$NTU = \frac{UA_s}{\dot{m}_c C_{p,cold}} \tag{3.23}$$

and the recuperator effectiveness

$$\varepsilon_{reg} = \frac{(1 - \exp(-NTU(1-C)))}{(1 - C\exp(-NTU(1-C)))} \tag{3.24}$$

Finally, Eq. 2.9 is applied separately to the hot and cold streams, shown and labelled in Fig. 3.6, and the resulting expressions summed to yield the expression for the rate of entropy generation in the recuperator

$$\dot{S}_{gen,reg} = \dot{m} C_{p0} \ln \left[\frac{T_{c,e} T_{h,e}}{T_{c,i} T_{h,i}} \left(\frac{P_{c,e} P_{h,e}}{P_{c,i} P_{h,i}} \right)^{\frac{(1-k)}{k}} \right] + \frac{\dot{m} C_{p0} (T_{m,reg} - T_0)}{T_0} \tag{3.25}$$

where the constant pressure heat capacity C_{p0} is determined for $T_{m,reg}$, the arithmetic mean temperature of the fluid streams.

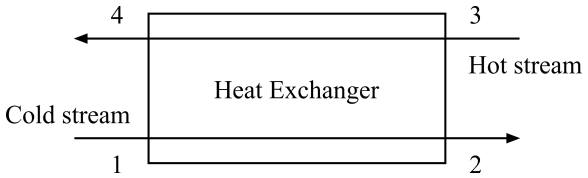


Figure 3.6 Heat transfer in a parallel plate counter-flow heat exchanger with unequal capacity rates.

3.2.3 Intercooler

A parallel plate heat exchanger design is utilised for the intercooler also. As shown in Fig. 3.7, a cross-flow arrangement is used, with the plates oriented vertically to promote movement of atmospheric air through the vertical channels

by buoyancy-driven natural convection. Heat transfer takes place by way of the following mechanisms, depicted and labelled in Fig. 3.8:

i. internal forced convection between the hot fluid and the internal plate surfaces
ii. perpendular conduction between the hot- and cold-side surfaces (again, longitudinal conduction is not considered), and
iii. natural convection and thermal radiation from the external plate surfaces to the ambient air

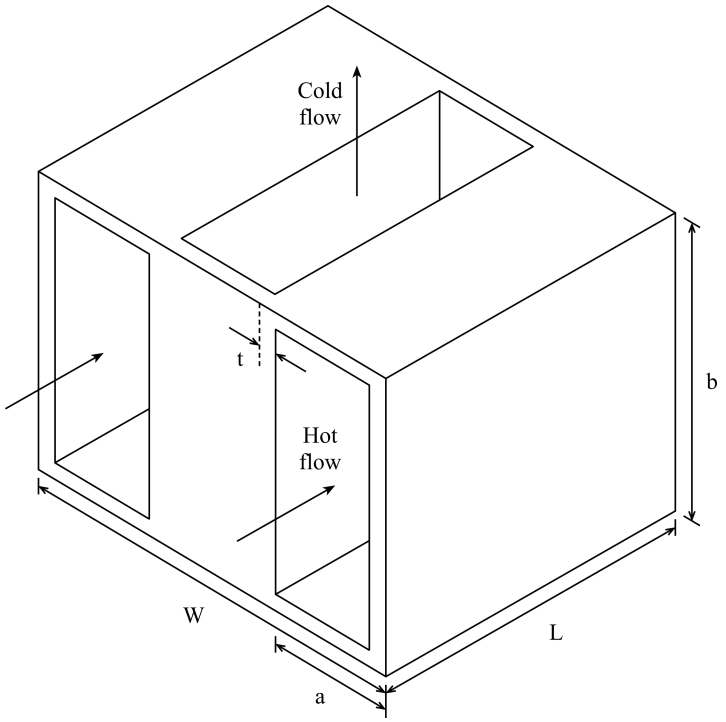


Figure 3.7 Schematic representation of the perpendicular-flow intercooler.

The intercooler geometry is modelled similarly to that of the recuperator, with the primary difference being that the width W_{ic} is fixed for the intercooler, and the number of plates determined using the channel width a, i.e.

$$n_{ic} = \frac{W_{ic}}{t_{ic} + a_{ic}} \tag{3.26}$$

where

$$a_{ic} = \frac{D_{h,ic}}{2}\left(\left(\frac{a}{b}\right)_{ic} + 1\right) \tag{3.27}$$

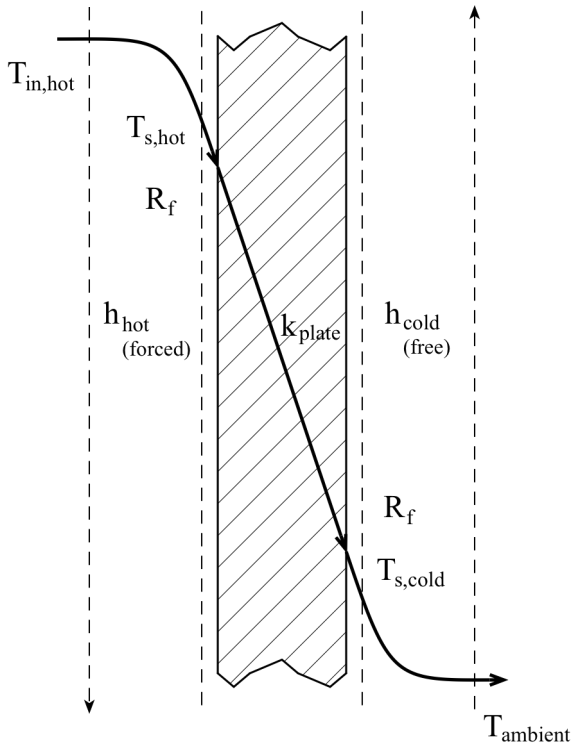


Figure 3.8 Mechanisms of heat transfer at the intercooler.

The mass flow rate of the hot stream through a single channel is

$$\dot{m}_{ic} = \frac{2\dot{m}}{n_{ic}} \tag{3.28}$$

and the surface area for heat transfer is

$$A_{s,ic} = b_{ic}L_{ic} = \frac{D_{h,ic}\left(\left(\frac{a}{b}\right)_{ic}+1\right)}{2\left(\frac{a}{b}\right)_{ic}}L_{ic} \tag{3.29}$$

The hot stream Reynolds number, friction factor, Nusselt number and convection heat transfer coefficient are determined using Eqs. 3.30 - 3.33, as done for the receiver and recuperator in the preceding sections

$$Re = \frac{4\dot{m}\left(\frac{a}{b}\right)_{ic}}{\mu_{hot}D_{h,ic}\left(\frac{a}{b}\right)_{ic}^{2}} \tag{3.30}$$

$$f = \left(0.79\ln\left(Re-1.64\right)\right)^{-2} \tag{3.31}$$

$$Nu_D = \frac{Pr\,(f/8)\,(Re-1000)}{1+12.7\,(f/8)^{1/2}\,(Pr^{2/3}-1)} \tag{3.32}$$

$$h = \frac{kNu_D}{D_h} \tag{3.33}$$

Thereafter, heat transfer by natural convection from the external plate surface to the cold fluid stream is considered. First, the Rayleigh number is determined using the equation

$$Ra = \frac{9.81\beta Pr_{cold}\,(T_{s,cold}-T_0)\,a_{ic}^3}{\left(\frac{\mu_{cold}}{\rho_{cold}}\right)^2} \tag{3.34}$$

where the coefficient of volume expansion β is defined as

$$\beta = \left(\frac{T_{s,cold}+T_0}{2}\right)^{-1} \tag{3.35}$$

The Nusselt number is then determined using the following relation (Bar-Cohen and Rohsenow, 1984)

$$Nu_{D,cold} = \left[\left(\frac{576}{Ra\left(\frac{a}{b}\right)_{ic}}\right)^2 + \left(\frac{2.873}{Ra\left(\frac{a}{b}\right)_{ic}}\right)^{\frac{1}{2}}\right]^{-\frac{1}{2}} \tag{3.36}$$

and used to determine the natural convection heat transfer coefficient

$$h = \frac{kNu_D}{D_h} \tag{3.37}$$

With both the internal and external heat transfer coefficients known, the intercooler effectiveness and heat rejection rate are evaluated next.

Heat removal from the hot fluid stream is described by Eq. 3.38. Equations 3.39 - 3.41 describe the convective heat transfer from the hot fluid stream to the intercooler plate, conductive heat transfer between the plate surfaces, and combined radiative and convective heat transfer from the external plate surface to the surrounding air, respectively.

$$\dot{Q} = \dot{m}_c C_{p,hot}\,(T_{in,hot}-T_{out,hot}) \tag{3.38}$$

$$\dot{Q} = \frac{h_{cold} A_s \left(T_{in,hot} - T_{out,hot}\right)}{\ln \dfrac{T_{s,hot} - T_{out,hot}}{T_{s,hot} - T_{in,hot}}} \tag{3.39}$$

$$\dot{Q} = \left(2R_f + \tfrac{t}{k}\right)^{-1} A_s \left(T_{s,hot} - T_{s,cold}\right) \tag{3.40}$$

$$\dot{Q} = h_{cold} A_s \left(T_{s,cold} - T_0\right) + \varepsilon \sigma_B A_s \left(T_{s,cold}^4 - T_0^4\right) \tag{3.41}$$

Equating Eqs. 3.38 - 3.41, the quartic equation

$$C_4 T_{s,cold}^4 + C_1 T_{s,cold} + C_0 = 0 \tag{3.42}$$

is obtained, in terms of the external surface temperature of the plates (see Fig. 3.8 for labels) and the coefficients C_4, C_1 and C_0, shown below.

$$C_4 = \left\{ \frac{\dot{m}_{ic} C_{p,hot}}{A_s} \left(2R_f + \frac{t}{k}\right)^2 \left[1 - \exp\left(\frac{-A_s h_{hot}}{\dot{m}_{ic} C_{p,hot}}\right)\right] + \frac{A_s}{\dot{m}_{ic} C_{p,hot} \left(2R_f + \frac{t}{k}\right)} \right\} \varepsilon \, \sigma_B \tag{3.43}$$

$$C_1 = \frac{\dot{m} C_{p,hot}}{A_s} \left(2R_f + \frac{t}{k}\right) \left(1 + 2R_f h_{hot} + \frac{t h_{cold}}{k}\right) \left[1 - \exp\left(\frac{-A_c h_{hot}}{\dot{m}_{ic} C_{p,hot}}\right) + \frac{A_s}{\dot{m}_{ic} C_{p,hot} \left(2R_f + \frac{t}{k}\right)}\right] - 1 \tag{3.44}$$

$$C_0 = \frac{\dot{m}_{ic} C_{p,hot}}{A_s} \left(2R_f + \frac{t}{k}\right) \left[\exp\left(\frac{-A_s h_{hot}}{\dot{m} C_{p,hot}}\right) - 1\right] T_{in,cold} - \frac{\dot{m}_{ic} C_{p,hot}}{A_s} \left(2R_f + \frac{t}{k}\right)^2 \left[1 - \exp\left(\frac{-A_s h_{hot}}{\dot{m}_{ic} C_{p,hot}}\right) + \frac{A_s}{\dot{m}_{ic} C_{p,hot} \left(2R_f + \frac{t}{k}\right)}\right] \left(\varepsilon \sigma_B T_0^4 + h_{cold} T_0\right) \tag{3.45}$$

Once known, the surface temperature $T_{s,cold}$ is substituted into the equation yielded by the combination Eqs. 3.38 and 3.40, to obtain the intercooler outlet temperature. Again, the expression for the rate of entropy generation in the intercooler is derived using Eq. 2.9.

$$\dot{S}_{gen,ic} = \dot{m}C_{p0}\ln\left(\frac{T_e}{T_i}\right) - \dot{m}R\ln\left(\frac{P_e}{P_i}\right) + \frac{\dot{m}C_{p0}\left(T_i - T_e\right)}{T_{s,cold}} \tag{3.46}$$

3.2.4 Turbomachinery

The turbine and compressor are not geometrically modelled or optimised in the present work. Rather, compression and expansion in conventional small-scale radial-flow micro-turbines is simulated using simple thermodynamic relations and operating characteristic data for proprietary automotive turbochargers.

Since the mass flow rate at which maximum net power output is produced is not known, the system must be optimised for several points of operation across the operating range of the turbocharger. The mass flow rate through the system is established by the compressor of the micro-turbine. A typical compressor performance map is shown in Fig. 3.9.

Figure 3.9 shows that the relationship between the system mass flow rate and the compressor pressure ratio is not an injective function, since the pressure ratio varies with shaft speed and compressor efficiency along lines of constant mass flow. The simplifying assumption is therefore made that the compressor and turbine both function at maximum efficiency, enabling the mass flow rate and compressor pressure ratio to be related by simple interpolation along the peak efficiency line through the centre of the maximum efficiency island on the compressor map (represented by the shaded region in Fig. 3.9), as suggested by Le Roux (2011).

Speed-specific turbine performance maps are not typically provided by the manufacturers of commercial automotive turbochargers. Instead, a single curve relating the mass flow rate and turbine pressure ratio is generally provided, obtained by consolidating the parts corresponding to maximum turbine efficiency of plots generated for different shaft speeds. In practice, however, the turbine functions much like a simple throttle, and is nearly independent of dimensionless rotational speed at pressure ratios above idle conditions (Crolla, 2015; Cumpsty, 2003). Therefore, only the mass flow rate through the system, and the pressure ratio of the compressors

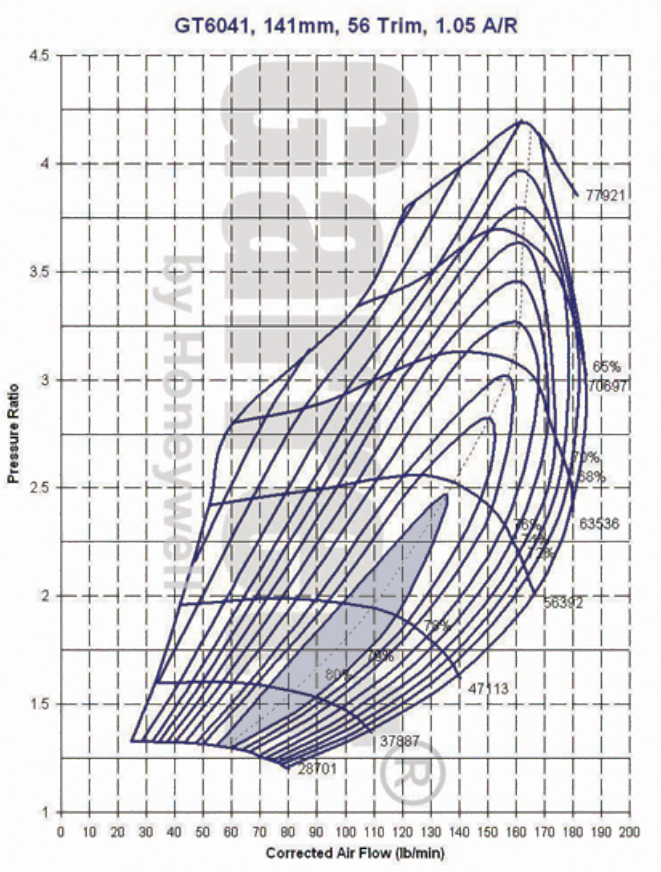


Figure 3.9 Garrett GT6041 turbocharger compressor performance map (Turbo by Garrett, 2016a).

are directly specified. Since the static pressure at the outlet of the LP turbine is equal to the atmospheric pressure, the turbine pressure ratio is solved for in the analysis, as described below in section 3.3.1.

The compressor and turbine entropy generation rates are determined using Eq. 3.47 derived using the ideal gas entropy generation equation (Eq. 2.9)

$$\dot{S}_{gen,comp/turb} = \dot{m} C_{p0} \ln \frac{T_e}{T_i} - \dot{m} R \ln \frac{P_e}{P_i} \tag{3.47}$$

The compression and expansion processes are assumed to be near-adiabatic, and the heat loss term is therefore omitted.

3.2.5 Ducts

Geometric modelling and optimisation of the ducts are not considered in the analysis, as stated in the introductory chapter. The marginal heat loss is therefore approximated by assuming a small temperature drop across each duct, and each pressure drop expressed as a small percentage of the pressure in an adjacent component, as detailed below in section 3.3.1. Again, the expression for the entropy generation rate is derived using Eq. 2.9.

$$\dot{S}_{gen,duct} = \dot{m}C_{p0}\ln\left(\frac{T_e}{T_i}\right) - \dot{m}R\ln\left(\frac{P_e}{P_i}\right) + \frac{\dot{m}C_{p0}T_{drop}}{T_0} \tag{3.48}$$

3.3 Temperature and pressure relations

Since only the ambient temperature and pressure are known, the temperatures and pressures of the components between the LP compressor and recuperator outlets must be solved for. The temperature and pressure relations are shown here for Case III, though the solution process is similar for cases I and II.

3.3.1 Pressure field

The general pressure drop equation is obtained by rearranging the Darcy-Weisbach equation (Çengel, 2004) to obtain

$$\Delta p = f_D \cdot \frac{\rho}{2} \cdot \frac{V^2 L}{D} \tag{3.49}$$

Following the procedure for the derivation of Eqs. 3.17 and 3.16 describing the Reynolds number and the flow velocity, Eq. 3.49 is written in terms of the aspect ratio, hydraulic diameter and length of the flow passage as

$$\Delta P = \left(0.79\ln\frac{4\dot{m}\left(\frac{a}{b}\right)}{\mu D_h\left(\frac{a}{b}\right)} - 1.64\right)^{-2}\left[\frac{8\dot{m}^2\left(\frac{a}{b}\right)^2}{\rho\left(\left(\frac{a}{b}\right)+1\right)^4}\right]\left(\frac{L}{D_h^5}\right) \tag{3.50}$$

The pressure at each point in the system is then sequentially determined by the following procedure: The product of the compressor pressure ratio r, and the pressure

at the LP compressor inlet

$$P_1 = P_{atm} \tag{3.51}$$

gives the pressure at the LP compressor outlet

$$P_2 = rP_1 \tag{3.52}$$

The pressure at the intercooler inlet is then

$$P_3 = P_2\left(1 - \Delta P_{2-3}\right) \tag{3.53}$$

where ΔP_{2-3} denotes the pressure drop across the duct between points 2 and 3 in the system, expressed as a small percentage of the LP compressor outlet pressure. Using Eq. 3.50, the intercooler outlet pressure is written as

$$P_4 = P_3 - \left(0.79\ln\frac{4\dot{m_{c,ic}}\left(\frac{a}{b}\right)_{ic}}{\mu_{hot,ic}D_{h,ic}\left(\frac{a}{b}\right)_{ic}} - 1.64\right)^{-2} \left[\frac{8\dot{m_{c,ic}}^2\left(\frac{a}{b}\right)^2_{ic}}{\rho_{hot,ic}\left(\left(\frac{a}{b}\right)_{ic}+1\right)^4}\right]\left(\frac{L_{ic}}{{D_{h,ic}}^5}\right) \tag{3.54}$$

where the density and viscosity of the fluid are evaluated for the average fluid temperature. The inlet and outlet pressures of the HP compressor, and the cold stream inlet pressure of the recuperator are given by Eqs. 3.55, 3.56 and 3.57 respectively. Like the pressure drop ΔP_{2-3} across the duct between the LP compressor outlet and intercooler inlet, ΔP_{4-5} and ΔP_{6-7} are written in terms of the respective duct inlet pressures.

$$P_5 = P_4\left(1 - \Delta P_{4-5}\right) \tag{3.55}$$

$$P_6 = rP_5 \tag{3.56}$$

$$P_7 = P_6\left(1 - \Delta P_{6-7}\right) \tag{3.57}$$

With the use of Eq. 3.50, the cold stream outlet pressure of the recuperator is written as

$$P_8 = P_7 - \left(0.79 \ln \frac{4\dot{m_{c,reg}} \left(\frac{a}{b}\right)_{reg}}{\mu_{cold,reg} D_{h,reg} \left(\frac{a}{b}\right)_{reg}} - 1.64\right)^{-2} \left[\frac{8\dot{m_{c,reg}}^2 \left(\frac{a}{b}\right)^2_{reg}}{\rho_{cold,reg} \left(\left(\frac{a}{b}\right)_{reg} + 1\right)^4}\right] \left(\frac{L_{reg}}{{D_{h,reg}}^5}\right) \tag{3.58}$$

The receiver inlet pressure is then written as

$$P_9 = P_8 \left(1 - \Delta P_{8-9}\right) \tag{3.59}$$

and the pressure at the outlet of the receiver written as

$$P_{10} = P_9 - \left(0.79 \ln \frac{4\dot{m} \left(\frac{a}{b}\right)_{rec}}{\mu_{rec} D_{h,rec} \left(\frac{a}{b}\right)_{rec}} - 1.64\right)^{-2} \left[\frac{8\dot{m}^2 \left(\frac{a}{b}\right)^2_{rec}}{\rho_{rec} \left(\left(\frac{a}{b}\right)_{rec} + 1\right)^4}\right] \left(\frac{L_{rec}}{{D_{h,rec}}^5}\right) \tag{3.60}$$

for absorber tubes of rectangular cross-section, and as

$$P_{10,circ} = P_9 - \left(0.79 \ln \frac{4\dot{m}}{\mu_{rec} \pi D_{h,rec}} - 1.64\right)^{-2} \left(\frac{8\dot{m}^2}{\rho_{rec} \pi^2}\right) \left(\frac{L_{rec}}{{D_{h,rec}}^5}\right) \tag{3.61}$$

for circular tubes. The pressure at the HP turbine inlet is then

$$P_{11} = P_{10} \left(1 - \Delta P_{10-11}\right) \tag{3.62}$$

Since the pressure ratios across the turbines are at first unknown, the pressure P_{12} at the outlet of the HP turbine cannot be determined by dividing the inlet pressure P_{11} by the turbine pressure ratio. Noting, however, that the pressure at the hot stream outlet of the recuperator is

$$P_{18} = P_{atm} \tag{3.63}$$

the pressure drop across the recuperator hot stream channels is added back, thus yielding the inlet pressure

$$P_{17} = P_{18} + \left(0.79 \ln \frac{4\dot{m_{c,reg}} \left(\frac{a}{b}\right)_{reg}}{\mu_{hot,reg} D_{h,reg} \left(\frac{a}{b}\right)_{reg}} - 1.64 \right)^{-2} \left[\frac{8\dot{m_{c,reg}}^2 \left(\frac{a}{b}\right)^2_{reg}}{\rho_{hot,reg} \left(\left(\frac{a}{b}\right)_{reg} + 1 \right)^4} \right] \left(\frac{L_{reg}}{{D_{h,reg}}^5} \right) \tag{3.64}$$

Similarly, the pressure drop ΔP_{16-17} across the duct between the LP turbine and the recuperator is this time written in terms of the lower pressure, and added back to obtain the LP turbine outlet pressure

$$P_{16} = P_{17} \left(1 + \Delta P_{16-17}\right) \tag{3.65}$$

Since the pressure ratios across the two turbines are to be identical (see section 2.3.3), it follows that

$$r_{turb} = \frac{P_{11}}{P_{12}} = \frac{P_{15}}{P_{16}} \tag{3.66}$$

where the pressures P_{11} and P_{16} are known. By substituting Eq. 3.69/3.70 for the reheater outlet pressure P_{14} in Eq. 3.71, the LP turbine inlet pressure P_{15} is written in terms of the reheater inlet pressure P_{13} and pressure drop ΔP_{13-14}. Equation 3.68 is then substituted for P_{13} in the combination of Eqs. 3.69/3.70 and 3.71, yielding an equation for P_{15} in terms of P_{12}. Solving simultaneously and substituting the result for P_{15} into Eq. 3.66, the HP compressor outlet pressure is written as

$$P_{12} = \left[\frac{P_{16} P_{11}}{\left(1 - \Delta P_{14-15}\right)\left(1 - \Delta P_{12-13}\right)} + \frac{\Delta P_{13-14}^2}{4\left(1 - \Delta P_{12-13}\right)^2} \right]^{1/2} + \frac{\Delta P_{13-14}}{2\left(1 - \Delta P_{12-13}\right)} \tag{3.67}$$

The reheater inlet pressure is therefore written as

$$P_{13} = P_{12} \left(1 - \Delta P_{12-13}\right) \tag{3.68}$$

and the pressure at the reheater outlet written as

$$P_{14} = P_{13} - \left(0.79 \ln \frac{4\dot{m}\left(\frac{a}{b}\right)_{reh}}{\mu_{reh} D_{h,reh}\left(\frac{a}{b}\right)_{reh}} - 1.64\right)^{-2} \left[\frac{8\dot{m}^2\left(\frac{a}{b}\right)^2_{reh}}{\rho_{reh}\left(\left(\frac{a}{b}\right)_{reh}+1\right)^4}\right]\left(\frac{L_{reh}}{{D_{h,reh}}^5}\right) \tag{3.69}$$

for absorber tubes of rectangular cross-section, and as

$$P_{14,circ} = P_{13} - \left(0.79 \ln \frac{4\dot{m}}{\mu_{reh}\pi D_{h,reh}} - 1.64\right)^{-2} \left(\frac{8\dot{m}^2}{\rho_{reh}\pi^2}\right)\left(\frac{L_{reh}}{{D_{h,reh}}^5}\right) \tag{3.70}$$

for circular tubes. Accordingly, the pressure at the LP turbine inlet is then

$$P_{15} = P_{14}\left(1 - \Delta P_{14-15}\right) \tag{3.71}$$

For Case I, the HP turbine outlet pressure is determined using the procedure described above, as shown in Appendix B. On account of there being no reheater between the turbines in Case II, the arithmetic is simplified, and the relation for the turbine pressure ratios

$$r_{turb} = \frac{P_{11}}{P_{12}} = \frac{P_{13}}{P_{14}} \tag{3.72}$$

where the HP turbine inlet pressure P_{11}, LP turbine outlet pressure P_{14}, and pressure drop ΔP_{12-13} are known, may be manipulated as follows

$$r^2_{turb} = \frac{P_{11}}{P_{12}} \cdot \frac{P_{13}}{P_{14}} = \frac{P_{11}\cdot\left(1 - \Delta P_{12-13}\right)}{P_{12}P_{13}} \tag{3.73}$$

and the turbine pressure ratio written as

$$r_{turb} = \sqrt{\frac{P_{11}\left(1 - \Delta P_{12-13}\right)}{P_{14}}} \tag{3.74}$$

3.3.2 Temperature field

With the use of the temperature at the LP compressor inlet

$$T_1 = T_{atm} \tag{3.75}$$

the temperature at the LP compressor outlet is written as

$$T_2 = T_1 \left(1 + \frac{K_{comp} - 1}{\eta_{comp}}\right) \tag{3.76}$$

where η_{comp} is the compressor efficiency, and

$$K_{comp} = r_c^{\frac{k-1}{k}} \tag{3.77}$$

where k is the specific heat ratio of the working medium. The intercooler inlet temperature is written as

$$T_3 = T_2 - T_{drop} \tag{3.78}$$

where T_{drop} is the assumed temperature drop across the duct.

Equating and rearranging Eqs. 3.38 and 3.40, the intercooler outlet temperature is written as

$$T_4 = T_3 - \frac{A_s h_c}{\dot{m}_c C_{p,h}} \left(T_{s,c} - T_0\right) - \frac{A_s \varepsilon_p \sigma_B \left(T_{s,c}^4 - T_0^4\right)}{\dot{m}_c C_{p,h}} \tag{3.79}$$

where the temperature $T_{s,cold}$ is obtained by solving Eq. 3.42. The inlet and outlet temperatures of the HP compressor, and the cold stream inlet temperature of the recuperator are given by Eqs. 3.80, 3.81 and 3.82 respectively.

$$T_5 = T_4 - T_{drop} \tag{3.80}$$

$$T_6 = T_5 \left(1 + \frac{K_{comp} - 1}{\eta_{comp}}\right) \tag{3.81}$$

$$T_7 = T_6 - T_{drop} \tag{3.82}$$

Since the recuperator efficiency is at first unknown, the cold stream outlet temperature T_8 is initially assumed. The receiver inlet temperature is written as

$$T_9 = T_8 - T_{drop} \tag{3.83}$$

and the receiver outlet temperature as

$$T_{10} = T_9 + \frac{\dot{Q}_{net,rec}}{\dot{m} C_{p,rec}} \tag{3.84}$$

where the net rate of heat absorption $\dot{Q}_{net}$ is obtained by solving Eq. 3.10. The temperature at the HP turbine inlet is written as

$$T_{11} = T_{10} - T_{drop} \tag{3.85}$$

and the temperature at the HP turbine outlet as

$$T_{12} = T_{11} \left[1 - \eta_{turb} \left(1 - \frac{1}{K_{turb}} \right) \right] \tag{3.86}$$

$$K_{turb} = \left(\frac{P_{11}}{P_{12}} \right)^{\frac{k-1}{k}} = \left(\frac{P_{15}}{P_{16}} \right)^{\frac{k-1}{k}} \tag{3.87}$$

The reheater inlet and outlet temperatures are given by Eqs. 3.88 and 3.89 respectively, where the net rate of heat absorption is once again obtained by solving Eq. 3.10, using Eq 3.5 to determine the reheater aperture diameter from the hydraulic diameter, length, and aspect ratio.

$$T_{13} = T_{12} - T_{drop} \tag{3.88}$$

$$T_{14} = T_{13} + \frac{\dot{Q}_{net,reh}}{\dot{m} C_{p,reh}} \tag{3.89}$$

The inlet and outlet temperatures of the LP turbine are given by Eqs. 3.90 and 3.91 respectively

$$T_{15} = T_{14} - T_{drop} \tag{3.90}$$

$$T_{16} = T_{15}\left[1 - \eta_{turb}\left(1 - \frac{1}{K_{turb}}\right)\right] \tag{3.91}$$

and the recuperator hot stream inlet temperature by

$$T_{17} = T_{16} - T_{drop} \tag{3.92}$$

With the recuperator inlet temperatures known, the recuperator effectiveness is determined as described in section 3.2.2, and used to determine the outlet temperatures, written as

$$T_8 = T_7 + \varepsilon_{reg}\left(T_{17} - T_7\right) \tag{3.93}$$

$$T_{18} = T_{17} - \varepsilon_{reg}\left(\frac{C_{p,c}}{C_{p,h}}\right)\left(T_{17} - T_7\right) \tag{3.94}$$

The pressure field calculations are then repeated, replacing the assumed temperatures with the newly calculated values. Similarly, the temperature field calculations are updated using the newly calculated pressure values. The difference in recuperator hot stream outlet temperature is evaluated at the end of each iteration, and the procedure terminated once a value of 1e-3 is reached.

3.4 Objective function

Having derived the equations for the rate of entropy generation in the components, the objective functions to be optimised are constructed, as follows. Recall from section 2.1.2 that the exergy balance of a control volume is

$$\dot{W} = \dot{m}\left(\psi_{in} - \psi_{out}\right) + \sum\left(1 - \frac{T_0}{T_k}\right)\dot{Q}_k - \dot{X}_{destroyed}$$

Rearranging, and expanding the flow exergy terms denoted by $\dot{m}\psi$, the general equation for the irreversibility rate in an open STBC system is written as

$$\begin{aligned} \dot{X} &= T_0\dot{S}_{gen} = \dot{m}\psi_i - \dot{m}\psi_e + \left(1 - \frac{T_0}{T^*}\right)\dot{Q}^* - \dot{W}_t + \dot{W}_c \\ \therefore T_0\dot{S}_{gen} &= \dot{m}\left[h_i - h_e - T_0\left(s_i - s_e\right) + \frac{1}{2}\left({V_i}^2 - {V_e}^2\right) + g\left(z_i - z_e\right)\right] \\ &\qquad + \left(1 - \frac{T_0}{T^*}\right)\dot{Q}^* - \dot{W}_{net} \end{aligned} \tag{3.95}$$

where the negligible kinetic and potential energy terms fall away. The chemical energy terms associated with combustion do not apply and are therefore omitted. Rearranging equation 3.95, the net power output from the system is expressed as

$$\begin{aligned} \dot{W}_{net} &= \dot{W}_t - \dot{W}_c = -T_0\dot{S}_{gen} + \left(1 - \frac{T_0}{T^*}\right)\dot{Q}^* \\ &\qquad + \dot{m}C_{p0}\left(T_i - T_e\right) - \dot{m}T_0C_{p0}\ln\frac{T_i}{T_e} \end{aligned} \tag{3.96}$$

where $\dot{S}_{gen}$ denotes the sum of the individual component entropy generation rates. For demonstrative purposes, the complete objective function is constructed below for Case III. The objective functions for cases I and II are provided in Appendix B.

Proceeding with the formulation, the objective function is written as

$$\begin{aligned} \dot{W}_{net} &= \left(1 - \frac{T_0}{T^*}\right)\dot{Q}^* + \dot{m}C_{p0}\left(T_1 - T_{18}\right) + \dot{m}T_0C_{p0}\ln\frac{T_{18}}{T_1} \\ &- T_0\left\{\left[\dot{m}C_{p0}\ln\frac{T_2}{T_1} - \dot{m}R\ln\frac{P_2}{P_1}\right]_{LP\,compressor}\right. \\ &+ \left[\dot{m}C_{p0}\ln\left(\frac{T_3}{T_2}\right) - \dot{m}R\ln\left(\frac{P_3}{P_2}\right) + \frac{\dot{m}C_{p0}T_{drop}}{T_0}\right]_{duct\,2\text{-}3} \\ &+ \left[\dot{m}C_{p0}\ln\left(\frac{T_4}{T_3}\right) - \dot{m}R\ln\left(\frac{P_4}{P_3}\right) + \frac{\dot{m}C_{p0}\left(T_i - T_e\right)}{T_{s,cold}}\right]_{intercooler} \\ &+ \left[\dot{m}C_{p0}\ln\left(\frac{T_5}{T_4}\right) - \dot{m}R\ln\left(\frac{P_5}{P_4}\right) + \frac{\dot{m}C_{p0}T_{drop}}{T_0}\right]_{duct\,4\text{-}5} \\ &+ \left.\left[\dot{m}C_{p0}\ln\frac{T_6}{T_5} - \dot{m}R\ln\frac{P_6}{P_5}\right]_{HP\,compressor}\right. \end{aligned}$$

$$
\begin{aligned}
&+ \left[\dot{m}C_{p0}\ln\left(\frac{T_7}{T_6}\right) - \dot{m}R\ln\left(\frac{P_7}{P_6}\right) + \frac{\dot{m}C_{p0}T_{drop}}{T_0}\right]_{duct\,6\text{-}7} \\
&+ \left[\dot{m}C_{p0}\ln\left[\frac{T_8T_{18}}{T_7T_{17}}\left(\frac{P_8P_{18}}{P_7P_{17}}\right)^{\frac{(1-k)}{k}}\right] + \frac{\dot{m}C_{p0}T_{drop}}{T_0}\right]_{recuperator} \\
&+ \left[\dot{m}C_{p0}\ln\left(\frac{T_9}{T_8}\right) - \dot{m}R\ln\left(\frac{P_9}{P_8}\right) + \frac{\dot{m}C_{p0}T_{drop}}{T_0}\right]_{duct\,8\text{-}9} \\
&+ \left[-\frac{\dot{Q}^*}{T^*} + \frac{\dot{Q}_0}{T_0} + \dot{m}C_{p0}\ln\frac{T_{10}}{T_9} - \dot{m}R\ln\frac{P_{10}}{P_9}\right]_{receiver} \\
&+ \left[\dot{m}C_{p0}\ln\left(\frac{T_{11}}{T_{10}}\right) - \dot{m}R\ln\left(\frac{P_{11}}{P_{10}}\right) + \frac{\dot{m}C_{p0}T_{drop}}{T_0}\right]_{duct\,10\text{-}11} \\
&+ \left[\dot{m}C_{p0}\ln\frac{T_{12}}{T_{11}} - \dot{m}R\ln\frac{P_{12}}{P_{11}}\right]_{HP\,turbine} \\
&+ \left[\dot{m}C_{p0}\ln\left(\frac{T_{13}}{T_{12}}\right) - \dot{m}R\ln\left(\frac{P_{13}}{P_{12}}\right) + \frac{\dot{m}C_{p0}T_{drop}}{T_0}\right]_{duct\,12\text{-}13} \\
&+ \left[-\frac{\dot{Q}^*}{T^*} + \frac{\dot{Q}_0}{T_0} + \dot{m}C_{p0}\ln\frac{T_{14}}{T_{13}} - \dot{m}R\ln\frac{P_{14}}{P_{13}}\right]_{reheater} \\
&+ \left[\dot{m}C_{p0}\ln\left(\frac{T_{15}}{T_{14}}\right) - \dot{m}R\ln\left(\frac{P_{15}}{P_{14}}\right) + \frac{\dot{m}C_{p0}T_{drop}}{T_0}\right]_{duct\,14\text{-}15} \\
&+ \left[\dot{m}C_{p0}\ln\frac{T_{16}}{T_{15}} - \dot{m}R\ln\frac{P_{16}}{P_{15}}\right]_{LP\,turbine} \\
&+ \left[\dot{m}C_{p0}\ln\left(\frac{T_{17}}{T_{16}}\right) - \dot{m}R\ln\left(\frac{P_{17}}{P_{16}}\right) + \frac{\dot{m}C_{p0}T_{drop}}{T_0}\right]_{duct\,16\text{-}17}\Bigg\}
\end{aligned}
\tag{3.97}
$$

Note that a cancellation occurs between the receiver entropy generation rate, given by Eq. 3.11, and the term

$$\left(1 - \frac{T_0}{T^*}\right)\dot{Q}^* \tag{3.98}$$

from Eq. 3.96, such that

$$\begin{aligned}&\left(1-\frac{T_0}{T^*}\right)\dot{Q}^* - T_0\left(\frac{\dot{Q}_0}{T_0}-\frac{\dot{Q}^*}{T^*}\right)\\ &= \dot{Q}_{net}+\dot{Q}_0-\frac{\dot{Q}_{net}T_0}{T^*}-\frac{\dot{Q}_0T_0}{T^*} - \dot{Q}_0+\frac{\dot{Q}_{net}T_0}{T^*}+\frac{\dot{Q}_0T_0}{T^*} = \dot{Q}_{net}\end{aligned} \tag{3.99}$$

Also note that $\dot{Q}^* = \dot{Q}^*_{rec} + \dot{Q}^*_{reh}$ for Case III. Simplifying Eq. 3.97, the objective function for Case III ultimately becomes

$$\begin{aligned}\dot{W}_{net} &= \dot{Q}_{net} + \dot{m}C_{p,0}\left(T_1 - T_{18}\right) + \dot{m}T_0C_{p,0}\ln\frac{T_{18}}{T_1}\\ &- T_0\left\{\left[\dot{m}C_{p0}\ln\frac{T_2}{T_1} - \dot{m}R\ln\frac{P_2}{P_1}\right]_{LP\,compressor}\right.\\ &+ \left[\dot{m}C_{p0}\ln\left(\frac{T_3}{T_2}\right) - \dot{m}R\ln\left(\frac{P_3}{P_2}\right) + \frac{\dot{m}C_{p0}T_{drop}}{T_0}\right]_{duct\,2\text{-}3}\\ &+ \left[\dot{m}C_{p0}\ln\left(\frac{T_4}{T_3}\right) - \dot{m}R\ln\left(\frac{P_4}{P_3}\right) + \frac{\dot{m}C_{p0}\left(T_i - T_e\right)}{T_{s,cold}}\right]_{intercooler}\\ &+ \left[\dot{m}C_{p0}\ln\left(\frac{T_5}{T_4}\right) - \dot{m}R\ln\left(\frac{P_5}{P_4}\right) + \frac{\dot{m}C_{p0}T_{drop}}{T_0}\right]_{duct\,4\text{-}5}\\ &+ \left[\dot{m}C_{p0}\ln\frac{T_6}{T_5} - \dot{m}R\ln\frac{P_6}{P_5}\right]_{HP\,compressor}\\ &+ \left[\dot{m}C_{p0}\ln\left(\frac{T_7}{T_6}\right) - \dot{m}R\ln\left(\frac{P_7}{P_6}\right) + \frac{\dot{m}C_{p0}T_{drop}}{T_0}\right]_{duct\,6\text{-}7}\\ &+ \left[\dot{m}C_{p0}\ln\left[\frac{T_8T_{18}}{T_7T_{17}}\left(\frac{P_8P_{18}}{P_7P_{17}}\right)^{\frac{(1-k)}{k}}\right] + \frac{\dot{m}C_{p0}T_{drop}}{T_0}\right]_{recuperator}\\ &+ \left[\dot{m}C_{p0}\ln\left(\frac{T_9}{T_8}\right) - \dot{m}R\ln\left(\frac{P_9}{P_8}\right) + \frac{\dot{m}C_{p0}T_{drop}}{T_0}\right]_{duct\,8\text{-}9}\\ &+ \left[\dot{m}C_{p0}\ln\frac{T_{10}}{T_9} - \dot{m}R\ln\frac{P_{10}}{P_9}\right]_{receiver}\\ &+ \left[\dot{m}C_{p0}\ln\left(\frac{T_{11}}{T_{10}}\right) - \dot{m}R\ln\left(\frac{P_{11}}{P_{10}}\right) + \frac{\dot{m}C_{p0}T_{drop}}{T_0}\right]_{duct\,10\text{-}11}\end{aligned} \tag{3.100}$$

$$
\begin{aligned}
&+ \left[\dot{m}C_{p0}\ln\frac{T_{12}}{T_{11}} - \dot{m}R\ln\frac{P_{12}}{P_{11}}\right]_{HP\,turbine} \\
&+ \left[\dot{m}C_{p0}\ln\left(\frac{T_{13}}{T_{12}}\right) - \dot{m}R\ln\left(\frac{P_{13}}{P_{12}}\right) + \frac{\dot{m}C_{p0}T_{drop}}{T_0}\right]_{duct\,12\text{-}13} \\
&+ \left[\dot{m}C_{p0}\ln\frac{T_{14}}{T_{13}} - \dot{m}R\ln\frac{P_{14}}{P_{13}}\right]_{reheater} \\
&+ \left[\dot{m}C_{p0}\ln\left(\frac{T_{15}}{T_{14}}\right) - \dot{m}R\ln\left(\frac{P_{15}}{P_{14}}\right) + \frac{\dot{m}C_{p0}T_{drop}}{T_0}\right]_{duct\,14\text{-}15} \\
&+ \left[\dot{m}C_{p0}\ln\frac{T_{16}}{T_{15}} - \dot{m}R\ln\frac{P_{16}}{P_{15}}\right]_{LP\,turbine} \\
&+ \left[\dot{m}C_{p0}\ln\left(\frac{T_{17}}{T_{16}}\right) - \dot{m}R\ln\left(\frac{P_{17}}{P_{16}}\right) + \frac{\dot{m}C_{p0}T_{drop}}{T_0}\right]_{duct\,16\text{-}17}\Bigg\}
\end{aligned}
$$

3.5 Constraints

Receiver and reheater

The concentration ratio of a parabolic dish concentrator is typically between 100 and 250, as shown in Table 2.1. Since the concentrator dish radius is to be set as a parameter in the analysis, and the absorber geometry optimised, the cavity receiver aperture diameter is constrained, as follows (Le Roux et al., 2011):

$$\frac{A_{s,conc}}{A_a} = \frac{A_{s,conc}}{\frac{\pi}{4}d^2} > 100$$

$$\frac{\pi}{4}\left(\frac{D_h L\left(\frac{a}{b}+1\right)}{4\pi}\right) < \frac{A_{s,conc}}{100}$$

$$\frac{D_h L\left(\frac{a}{b}+1\right)}{16} - \frac{A_{s,conc}}{100} < 0 \qquad \text{(Cstr.1)}$$

A second constraint is applied to ensure that the aperture diameter corresponding to the calculated concentration ratio is greater than zero and large enough to admit sufficient radiation. It is written as

$$W_n\left(2\right) \; - \; \sqrt{\frac{D_h L\left(\frac{a}{b}+1\right)}{4\pi}} \; < \; 0 \tag{Cstr.2}$$

where $W_n\left(2\right)$ denotes the second entry in the vector of aperture diameters determined using the sizing algorithm (Appendix A) for the concentrator dish radius in question (Le Roux et al., 2011). The first entry in the vector is a minute initial estimate chosen to ensure that subsequent entries are larger, and is therefore not used. The aspect ratio is set to $a/b = 1$ for absorber tubes of circular cross-section, due to their symmetry about the centreline.

Constraint 3 is applied to ensure a hemispherical cavity shape as illustrated in Fig. 3.4. For a receiver comprising a rectangular section absorber tube, this requires the aspect ratio to be greater than, or equal to 2.5 (Le Roux et al., 2011), i.e.

$$2.5 - \left(\frac{a}{b}\right)_{rec} \; \leq \; 0 \tag{Cstr.3a}$$

However, the width of the absorber tube should not be allowed to exceed the length between the edge of the aperture and the inner wall of the cavity, on which Eqs. 3.4, 3.5 and 3.6 are established (Le Roux et al., 2011). Accordingly,

$$a \; \leq \; \tfrac{1}{2}\left(D-d\right) \; = \; \left(\frac{\sqrt{3}-1}{2}\right) d$$

$$\frac{D_h\left(\frac{a}{b}+1\right)}{2} - \left(\frac{\sqrt{3}-1}{2}\right)\sqrt{\frac{D_h L\left(\frac{a}{b}+1\right)}{4\pi}} \; \leq \; 0 \tag{Cstr.3b}$$

For a circular section absorber tube, a minimum of two spiral windings around the aperture is needed to ensure a hemispherical shape, viz.

$$2D_h \; \leq \; \tfrac{1}{2}\left(D-d\right)$$

$$2D_h - \left(\frac{\sqrt{3}-1}{2}\right)\sqrt{\frac{D_h L}{2\pi}} \; \leq \; 0 \tag{Cstr.3c}$$

The hydraulic diameter and length of the absorber tube cannot be negative, i.e.

$$-D_h \leq 0 \tag{Cstr.4a}$$

$$-L \leq 0 \tag{Cstr.4b}$$

The surface temperature of the absorber tube should be kept well below the melting temperatures of the tube- and turbocharger wheel materials. Le Roux et al. (2011) give the corresponding constraint as

$$T_s - T_{s,max} \leq 0 \tag{Cstr.5}$$

where the surface temperature is written as

$$T_s = T_i + \frac{\dot{Q}_{net}}{0.023 L_{rec}\left(\left(\frac{a}{b}\right)_{rec}+1\right)\left(1+\left(\frac{a}{b}\right)_{rec}^{-1}\right) kPr^{0.4}} \frac{1}{\left[\frac{4\dot{m}\left(\frac{a}{b}\right)_{rec}}{\mu D_{h,rec}\left(\left(\frac{a}{b}\right)_{rec}+1\right)^2}\right]^{0.8}} \tag{3.101}$$

for a rectangular section absorber, and as

$$T_s = T_i + \frac{\dot{Q}_{net}}{0.023\pi L k Pr^{0.4}\left(\frac{4\dot{m}}{\mu\pi D}\right)^{0.8}} \tag{3.102}$$

for a circular absorber tube, using the heat transfer relation $\dot{Q} = hA\Delta T$, the Dittus-Boelter relation for the Nusselt number in turbulent flow

$$Nu = 0.023 {Re_D}^{0.8} Pr^{0.4}$$

and the expression

$$A_s = D_{h,rec} L_{rec}\left(\left(\frac{a}{b}\right)+1\right)\left(1+\left(\frac{a}{b}\right)^{-1}\right)$$

for the total surface area of the absorber tube.

Recuperator

As noted by Le Roux et al. (2011), it may be desirable to place the recuperator underneath the concentrator dish in a SBTC system. Doing so would reduce the spatial footprint of the system without significant reducing the total surface area available for heat transfer. The corresponding constraint is written as

$$L_{reg} \leq \sqrt{\frac{A_{s,conc}}{\pi}} = \frac{D_{conc}}{2} \tag{Cstr.6}$$

Intercooler

Recall from section 3.2.3 that the intercooler plates are orientated vertically. To ensure that the channel height b_{ic} exceeds the width a_{ic}, the aspect ratio of the intercooler channel must be less than or equal to 1, viz.

$$\left(\frac{a}{b}\right)_{ic} - 1 \leq 0 \tag{Cstr.7}$$

Additionally, the motivation for Constraint 6 applies to the intercooler also, therefore

$$L_{ic} \leq \sqrt{\frac{A_{s,conc}}{\pi}} = \frac{D_{conc}}{2} \tag{Cstr.8}$$

3.6 Assumptions

The ambient conditions, material properties, and characteristic component metrics assumed in the present analysis are shown in Table 3.1. Also shown are the assumed temperature and pressure losses across the ducts.

Table 3.1 Assumptions used in the analysis (Çengel, 2004; Shah, 2005; Stine and Harrigan, 1985; Stine and Geyer, 2001; Winter et al., 1991).

	Description	Value	Unit
Working fluid (Air)			
R	Individual gas constant	287	J/kgK
k	Specific heat ratio (assumed constant)	1.4	-
Ambient conditions			
T_0	Ambient temperature	300	K

P_0	Ambient pressure	101.325	kPa
I	Irradiance	1 000	W/m^2
w	Non-dimensional wind factor	1	-
Concentrator and receiver			
CR_{min}	Minimum concentration ratio	100	-
β	Inclination angle	90	degrees
ψ_{rim}	Rim angle	45	degrees
e_p	Concentrator error	6.7	mrad
$refl$	Specular reflectivity of the concentrator dish	0.93	-
k_{ins}	Insulation thermal conductivity	0.05	W/mK
$T_{s,max}$	Maximum surface temperature	1 200 (Cu)	K
Recuperator			
k_{reg}	Plate thermal conductivity	401 (Cu)	W/mK
R_f	Fouling factor	0.001	-
t	Plate thickness	1	mm
H_{reg}	Overall height	1	m
$L_{reg,max}$	Maximum length	R_{dish}	m
Intercooler			
k_{ic}	Plate thermal conductivity	401 (Cu)	W/mK
R_f	Fouling factor	0.001	-
t	Plate thickness	1	mm
W_{ic}	Overall width	1	m
$L_{ic,max}$	Maximum length	R_{dish}	m
Ducts			
T_{drop}	Duct temperature loss	2	K
ΔP_{duct}	Duct pressure loss (decimal)	0.004	-

Chapter 4

Numerical method

The numerical method used to solve the optimisation problem is presented in this chapter. The optimisation algorithm is introduced, and the settings used in the present analysis listed. Lastly, the organisation and logic of the programs written for the execution of the optimisation are described.

4.1 Optimisation algorithm

The problem formulated in the preceding chapter is a single-objective multi-variable constrained optimisation problem, which can be solved using one of several available algorithms. A wide variety of algorithms have been disseminated in math and applied math literature, and computer scripts are available for many. Several numerical computing software packages such as MATLAB offer a compiled proprietary optimisation toolbox as an add-on application (The Mathworks, 2016). Optimisation algorithms are typically classified as either derivative-based, or evolutionary (also referred to as genetic) algorithms, and afford varying degrees of accuracy and computational efficiency depending on the smoothness and continuity of the objective function.

The gradients of the objective functions constituting the problem at hand cannot be analytically determined, since the temperature and pressure fields are calculated using an iterative process. An algorithm suitable for the present application must therefore accept numerically-determined gradients. Evolutionary algorithms do not require explicitly defined gradients. The scope of the search is therefore typically defined by specifying the number and coordinates of points in the initial population,

the settings to be applied, and the maximum number of generations. By default, the genetic algorithm included in the MATLAB optimisation toolbox generates starting coordinates by invoking a random number generator, although user-specification is supported (The Mathworks, 2016). Manual specification of this information for the present analysis would involve considerable trial and error, as the span of the solution space and approximate coordinates of the candidate solutions cannot be confidently estimated. On the other hand, there is a chance that the individuals in a random initial population will evolve towards a local minimum, failing to find the global minimum. Adjustments to the selection, genetic operator, and termination options may improve chances of finding a global solution, generally at the expense of computational efficiency.

The leapfrog algorithm for constrained optimisation (LFOPC) developed by Snyman (2000) meets the requirements set out above and has successfully undergone both standard analytical tests and application to a variety of practical engineering design problems. The method generates a dynamic trajectory path, using a penalty method to effect the constraints, that is relatively insensitive to local inaccuracies. As such, LFOPC is suitably robust for application to problems where the objective function contains experimental or numerical noise Snyman (2000). LFOPC is applied to the present analysis using subjectively determined values for the parameter settings shown in Table 4.1 below.

Table 4.1 Optimisation algorithm settings.

Parameter	Description	Default value	Chosen value
δ	Maximum step size	1	1
e_x	Convergence tolerance on the step size	1e-8	1e-11
e_g	Convergence tolerance on the norm of the gradient	1e-5	1e-5
μ_0	Initial penalty value	1e2	1e2
μ_{max}	Maximum penalty value	1e4	1e4
k_{max}	Maximum number of steps per phase	1e3	1e4

It is found that the large number of variables in the problem at hand necessitates a larger number of steps per phase and a smaller convergence tolerance on the step size to avoid premature termination at a suboptimal solution where the rate of change in the function value is small. As a result, increased RAM capacity and higher CPU speed are required for computation.[1]

[1]Results of the present work were obtained running MATLAB R2016a on Google Cloud Platform.

4.2 Programs

Optimisation of the objective functions using LFOPC is executed in three programs written in MATLAB. The constituent scripts of each are organised as indicated on the network diagram in Fig. 4.1. Each program comprises the main executing script, the function script in which the objective function is evaluated, auxiliary collector-sizing and property interpolation functions, and the LFOPC, gradient and constraints functions. The m-files are provided in Appendix B.

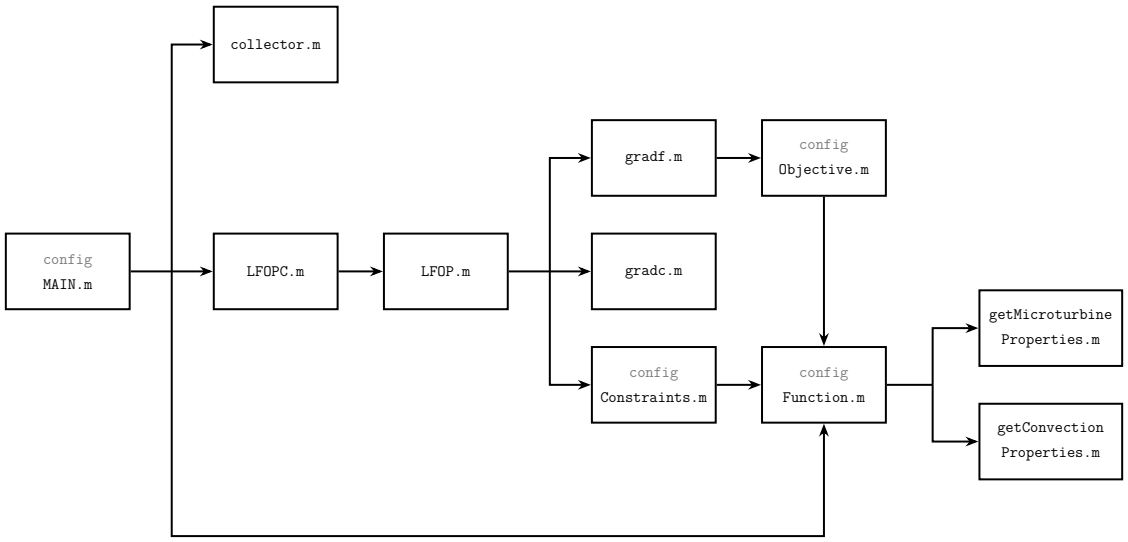


Figure 4.1 Network diagram of the optimisation routine, indicating function calls.

4.2.1 Optimisation variables and user-defined parameters

For each case under consideration, the net power output constitutes the dependent variable to be maximised, and the vector of geometric parameters, shown in Table 4.2, the independent variable to be optimised. The micro-turbine model, concentrator dish radii, and assumptions listed in Table 5.1 are not directly involved in the optimisation, but used in evaluation of the objective function, and thus defined by the user. As described in section 3.2.4, the geometric optimisation is repeated and the maximum net power output determined for several operating points across the mass flow range of each micro-turbine. Put differently, multiple power output and geometric optima are returned for each system analysed. The absolute maximum net power output may be considered the primary maximum, and the net power

output values at all other points considered secondary maxima. These distinctions should be kept in mind when interpreting the results of the optimisation.

Table 4.2 Optimisation variables.

Variable	Description	Numerical variable		
		Case 1: Reheating only	**Case 2: Intercooling only**	**Case 3: Combined reheating and intercooling**
$\left(\frac{a}{b}\right)_{ic}$	Aspect ratio of the intercooler channels		X(1)/100	X(1)/100
$D_{h,ic}$	Hydraulic diameter of the intercooler channels		X(2)/1000	X(2)/1000
L_{ic}	Length of the intercooler channels		X(3)	X(3)
$\left(\frac{a}{b}\right)_{reg}$	Aspect ratio of the recuperator channels	X(1)	X(4)	X(4)
$D_{h,reg}$	Hydraulic diameter of the recuperator channels	X(2)/100	X(5)/100	X(5)/100
L_{reg}	Length of the recuperator channels	X(3)	X(6)	X(6)
$\left(\frac{a}{b}\right)_{rec}$	Aspect ratio of the receiver tube	X(4)	X(7)	X(7)
$D_{h,rec}$	Hydraulic diameter of the receiver tube	X(5)/100	X(8)/100	X(8)/100
L_{rec}	Length of the receiver tube	X(6)	X(9)	X(9)
$\left(\frac{a}{b}\right)_{reh}$	Aspect ratio of the reheater tube	X(7)		X(10)
$D_{h,reh}$	Hydraulic diameter of the reheater tube	X(8)/100		X(11)/100
L_{reh}	Length of the reheater tube	X(9)		X(12)

In order to ascertain the effects of scale on the STBC variants, several concentrator diameter(s) and micro-turbine models are considered for each case. Accordingly, each specific arrangement or *configuration* analysed is labelled according to the convention (D_{rec}; D_{reh}; MT). For example, the combined reheating and intercooling cycle configuration with a receiver concentrator dish diameter of 16 m, reheater concentrator dish diameter of 10 m, and micro-turbine model number 8 is given the label (16; 10; 8), and the intercooled cycle configuration with a receiver concentrator dish diameter of 18 m and micro-turbine model number 12 is given the label (18; 12).

4.2.2 Logic

The basic sequence of operations executed by the main functions is outlined below. Actions pertaining to the reheater concentrator dish diameter are omitted for the optimisation of Case II. Although overall mass flow span differs significantly between

the 22 micro-turbine models listed in Appendix C, all values correspond to a compressor pressure ratio in the range $1.3 \leq r_c \leq 3$. The compressor pressure ratio is therefore used as the range parameter in the optimisation programs, and the mass flow rate determined by interpolation, as described in section 3.2.4, using Eq. 4.1 (Le Roux et al., 2011).

$$\dot{m}_i \;=\; \dot{m}_{min} \;+\; (\dot{m}_{max} - \dot{m}_{min})\,\frac{(r_i - r_{min})}{(r_{max} - r_{min})} \tag{4.1}$$

Starting with the smallest concentrator dish diameter(s) and the first micro-turbine model,

1. Call LFOPC starting at the lowest pressure ratio for the configuration.
2. Increment the pressure ratio, call LFOPC and repeat until the highest pressure ratio for the current micro-turbine model is reached.
3. Increment the micro-turbine model number, and repeat steps 1 to 3 until the last micro-turbine model is reached.
4. Increment the reheater concentrator dish diameter and repeat steps 1 to 4 until the reheater concentrator dish diameter equals that of the receiver concentrator.
5. Increment the receiver concentrator dish diameter, reset that of the reheater dish to the smallest value and repeat steps 1 to 5 until all configurations have been analysed.

Regarding the starting point X_0, it is found that small values of the initial coordinates are suitable for small configurations, and vice versa. Additionally, not all micro-turbines and concentrator diameters can be paired, due to disparities between the rate of entropy generation and the power that can be developed, for a given thermal input, at mass flow rates within the operating range of certain micro-turbine models. Ordinarily, this would require the user to find a suitable starting point, provided one exists for the configuration in question, by way of trial and error. To simplify matters, the starting point selection is automated, in congruence with the recommendation of Le Roux et al. (2011) that STBC optimisation code be progressively developed into compiled, user-friendly programs. Autonomous selection of valid starting points is effected using an iterative method, as described below.

For the first call to LFOPC in steps 1 and 2, a starting point of identical small values is used, i.e. $X_0 = c\vec{1}$, where $\vec{1}$ is an all-ones vector with length equal to the number

of geometry parameters as shown in Table 4.2, and c is a constant scaling factor. Typically, if the initial value for c is too small, the call to LFOPC will fail due to the calculation of one or more imaginary or undefined values in the temperature and pressure fields. In this event, an exit flag is parsed back to the main script, the factor c is incremented, and the call to LFOPC is reattempted. If the call then completes without calculating any imaginary or undefined values, the pressure ratio is increased, the constant c reset to the minimum value, and the procedure repeated. Furthermore, if step 2 runs to completion for all pressure ratios of the current micro-turbine model, the results are exported to a data file for subsequent analysis. If, however, a call to LFOPC fails to complete after the maximum value for c is reached, optimisation of the current micro-turbine model is terminated, and the procedure repeated for the optimisation of the next configuration. The consolidated, automated routine is represented diagrammatically in Fig. 4.2.

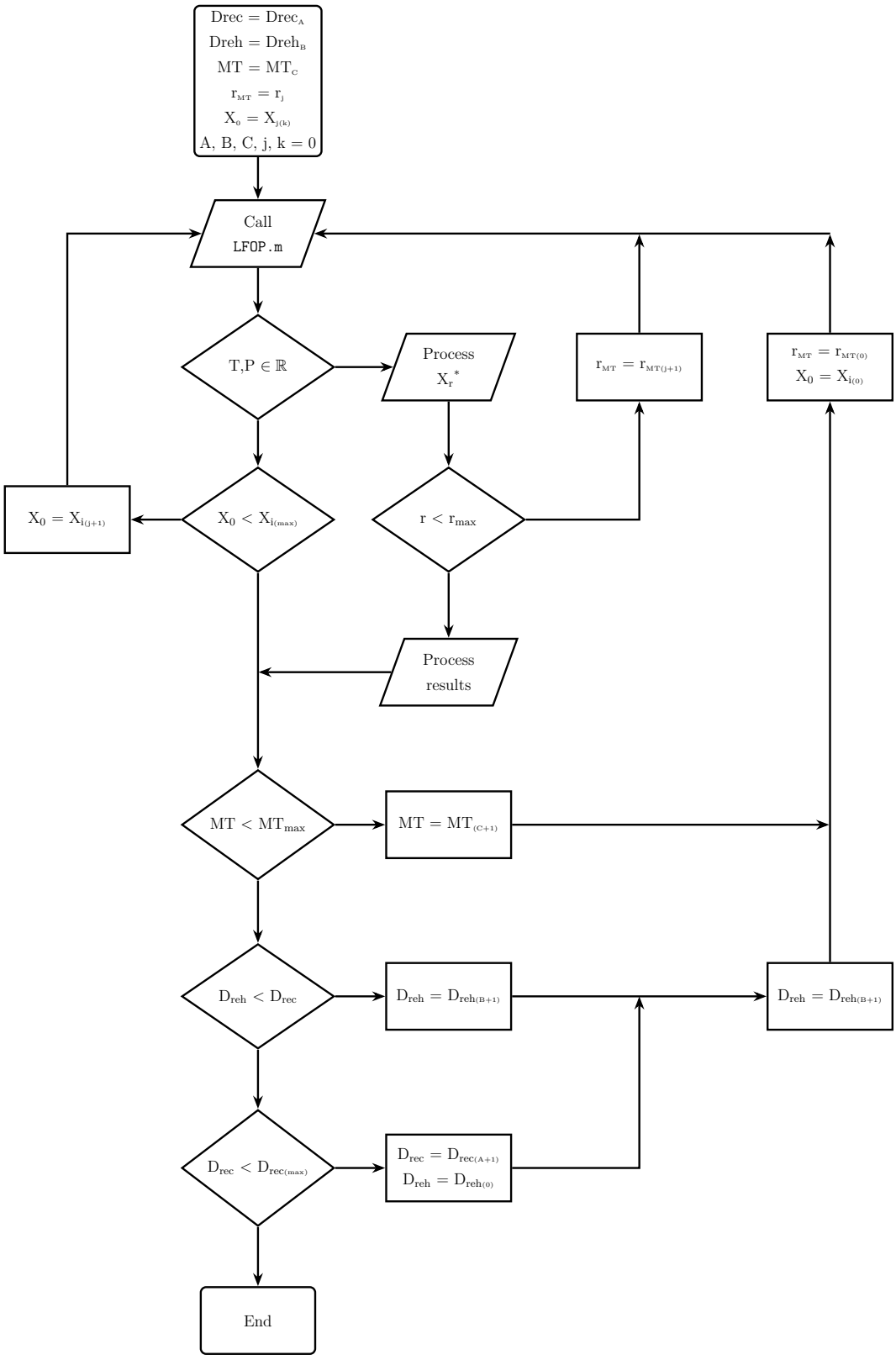


Figure 4.2 Flow chart indicating logic followed during execution of the optimisation routine.

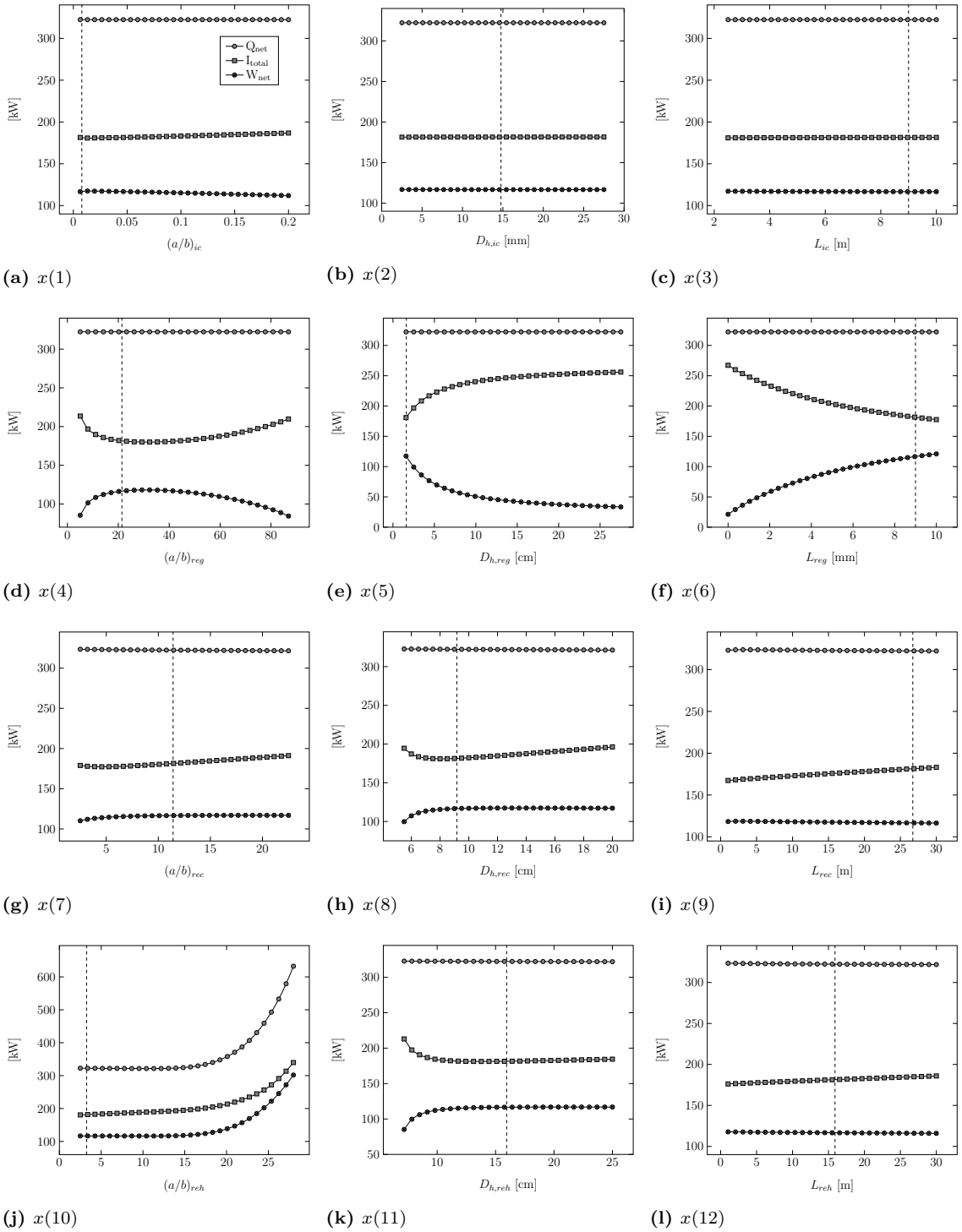
Chapter 5

Verification

The verification of the numerical method and problem formulation are presented in this chapter. Implementation of the numerical method is verified by independently varying the components of an optimised input argument vector to check that it is indeed the optimal combination. Implementation of the problem formulation is then verified by comparing the objective function and intermediate variables returned by the computer program against the theory and results of previous analyses, reviewed in chapter 2. Lastly, the assumed parameter and constraint values are altered, and verified by comparing the observed results and theoretically anticipated effects.

5.1 Optimisation

In order to test whether the actual points of maximum net power output correspond to the parameter values returned by the optimisation program, the net power output is independently plotted over a range of values for each geometry parameter, fixing all other geometry parameters at the values returned by the program. Figure 5.1 shows the plots for Case III, configuration (18; 12; 21). The system configuration labelling convention is described in section 5.2. Evidently, the parameter values returned by the program lie within reasonable proximity of the exact optima. The discrepancies are generally due to the small values of the partial derivatives, and enforcement of the constraints. The recuperator length, for instance, is limited to $L_{reg} = 9$ (Fig. 5.1f) and the reheater aspect ratio to $\left(\frac{a}{b}\right)_{reh} = 4.1$ (Fig. 5.1j). Although gradient discrepancies can be lessened by reducing the convergence tolerance on the norm of the gradient, doing so would increase computational requirements.



(a) $x(1)$ **(b)** $x(2)$ **(c)** $x(3)$

(d) $x(4)$ **(e)** $x(5)$ **(f)** $x(6)$

(g) $x(7)$ **(h)** $x(8)$ **(i)** $x(9)$

(j) $x(10)$ **(k)** $x(11)$ **(l)** $x(12)$

Figure 5.1 Partial derivatives of net heat input, total irreversibility, and maximum power output for Case III, configuration (18; 12; 21).

5.2 Modelling

Figure 5.2 shows the variation of net heat input, power output and irreversibility with mass flow rate, for Case I, configuration (16; 16; 21). As anticipated, the maxima of the downward concave net power output curves and the minima of the upward concave irreversibility curves lie roughly at the same mass flow rate, consistent with the Gouy-Stodola Theorem. There is fair consistency between the second-law power output curve and the curve generated according to the first law of thermodynamics using Eq. 5.1. The small difference is likely due to the assumptions made for temperature and pressure loss across the ducts (Le Roux et al., 2011).

$$\begin{aligned} W_{net,1} &= W_{turbines} - W_{compressors} \\ &= [\dot{m}C_{p0}(T_i - T_e)]_{HP\,turb.} + [\dot{m}C_{p0}(T_i - T_e)]_{LP\,turb.} \\ &\quad - [\dot{m}C_{p0}(T_e - T_i)]_{HP\,comp.} - [\dot{m}C_{p0}(T_e - T_i)]_{LP\,comp.} \end{aligned} \tag{5.1}$$

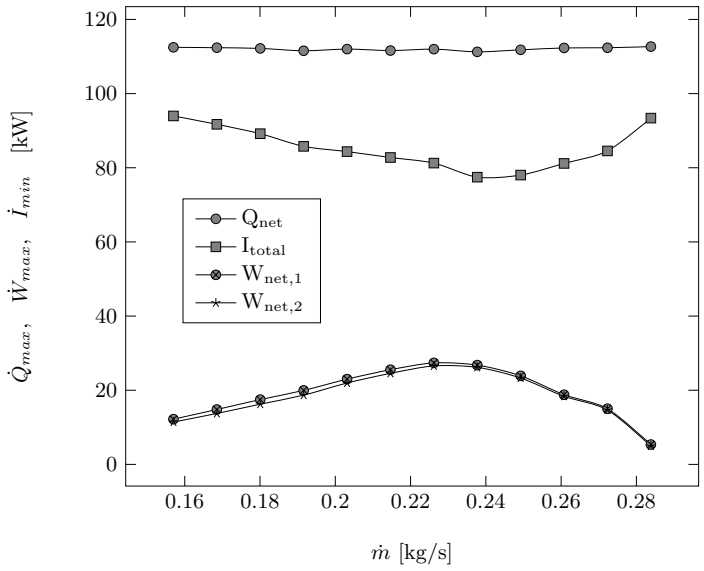


Figure 5.2 Variation in net heat input, power output, and irreversibility with mass flow rate; Case I, configuration (16; 16; 21).

Figure 5.3 shows net power output curves generated by considering various micro-turbine models for Case I, keeping the receiver and reheater concentrator dish diameters fixed at 6 m. The net power output curves generated considering various dish radii for Case I, with micro-turbine number 21, are shown in Fig. 5.4. Evidently, the results scale as anticipated when the configuration parameters are changed, with higher power output noted for larger dish radii and micro-turbines, and vice

versa. Note that no solutions are returned for configurations comprising very large parabolic dish collectors and small micro-turbines, due to mismatch between the low net power output and high receiver entropy generation rates. Likewise, no solutions are returned for large micro-turbines paired with small collectors, due to the high turbomachinery entropy generation rates compared to the heat input.

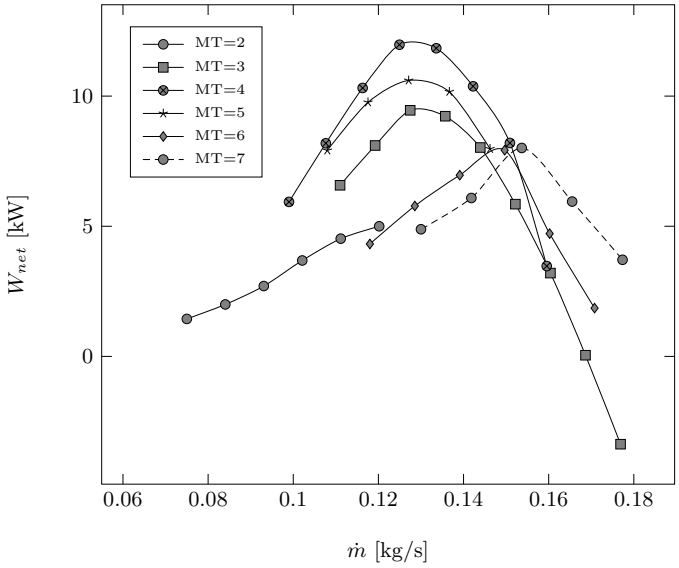


Figure 5.3 Variation in net power output with mass flow rate for Case I; $D_{h,rec} = 6\,m$, $D_{h,reh} = 6\,m$ and various micro-turbine models.

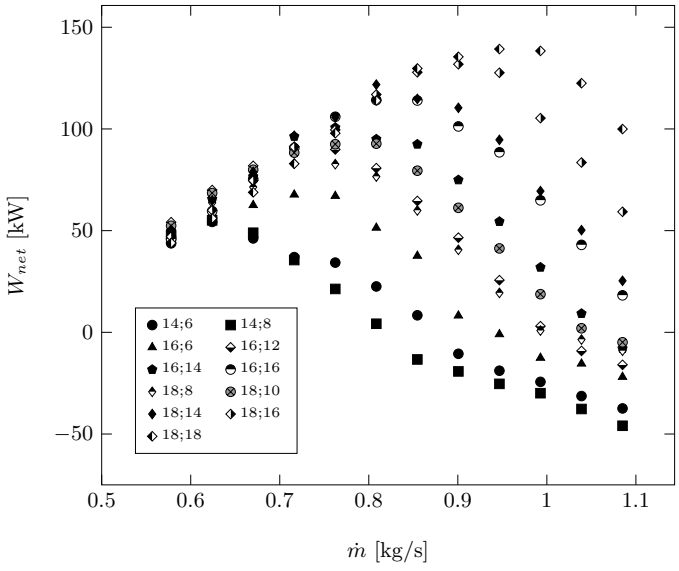


Figure 5.4 Variation in net power output with mass flow rate for Case I; various receiver and reheater concentrator diameters and MT = 21.

The differences in profile, most noticeable between the curves corresponding to the concentrator diameter pairs (14; 18) and (18; 18) in Fig. 5.4, indicate that the full net power output range is most accurately described with a near-symmetric curve. Any configurations for which the net power output curve is found to have more than one turning point, or values below zero across more than half of the operating range of the micro-turbine, are therefore excluded during sorting of the complete set of results. Thereafter, the configurations of each case found to have the most symmetrical net power output curves are selected as the representative configurations for the interpretation of the results in the following chapter, specifically:

- Case I: (10; 8; 11)
- Case II: (12; 13)
- Case III: (18; 12; 21)

5.3 Assumptions and constraints

Table 5.1 shows the default assumptions and constraint values applied in the present analysis, as well as the modified values used to assess the adequacy of the default values. Net heat input, power output, and irreversibility curves generated, for Case II, by assuming elevated values for each parameter, are shown in Fig. 5.5 - 5.7 against curves generated for default parameter values.

Table 5.1 Default and offset parameter values; Case II, configuration (16; 20).

	Default value	Increased value	Reduced value	Unit
Ambient conditions				
T_0	300	310	290	K
P_0	101.325	103	100	kPa
I	1 000	1 200	800	W/m^2
w	1	10	-	-
Concentrator and receiver				
CR_{min} (* elevated)	100	500	1 000*	-
β	90	30	60	degrees
ϕ_{rim}	45	60	30	degrees
e	6.7	35	-	mrad
refl	0.93	0.98	0.8	-
$T_{s,max}$	1 200	-	1 100	K
Recuperator				
k_{reg}	401	-	237	W/mK
H_{reg}	1	2	0.5	m
$L_{reg,max}$	R_{dish}	D_{dish}	-	m
Intercooler				
k_{ic}	401	-	237	W/mK
W_{ic}	1	2	0.5	m
$L_{ic,max}$	R_{dish}	D_{dish}	-	m

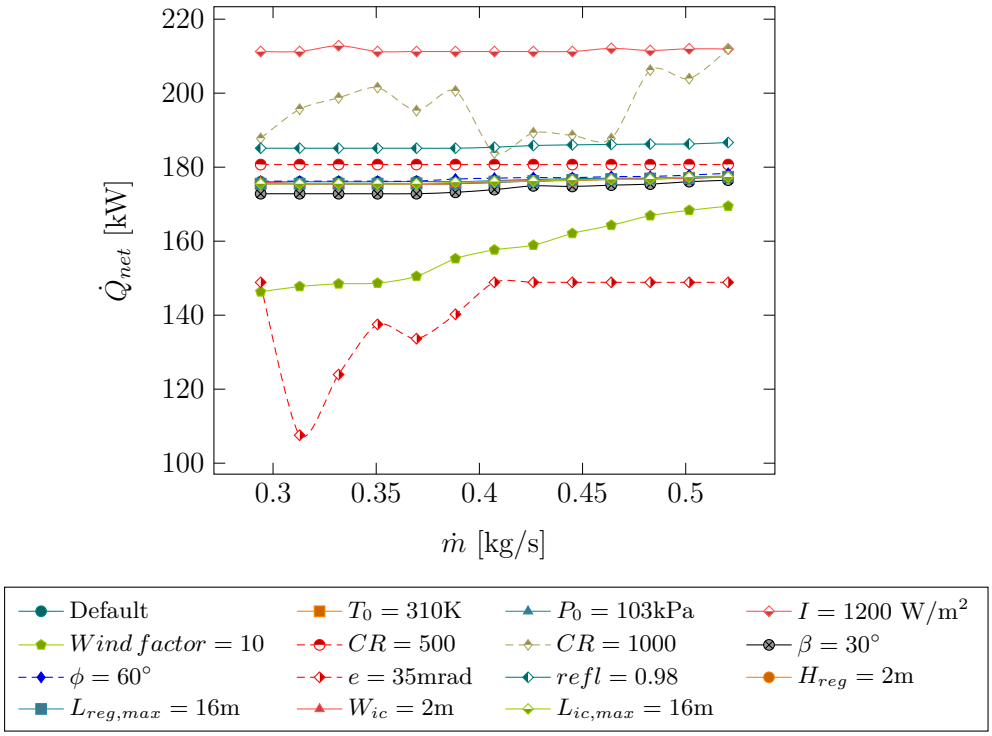


Figure 5.5 Net heat input - elevated parameter values; Case II (16; 20).

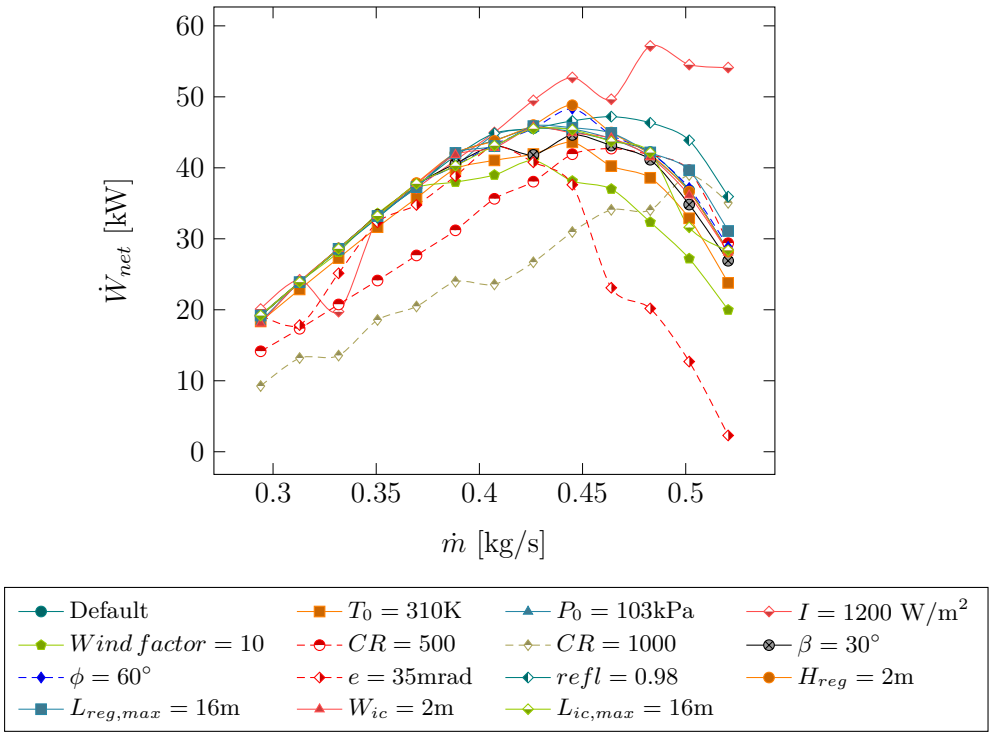


Figure 5.6 Net power output - elevated parameter values; Case II (16; 20).

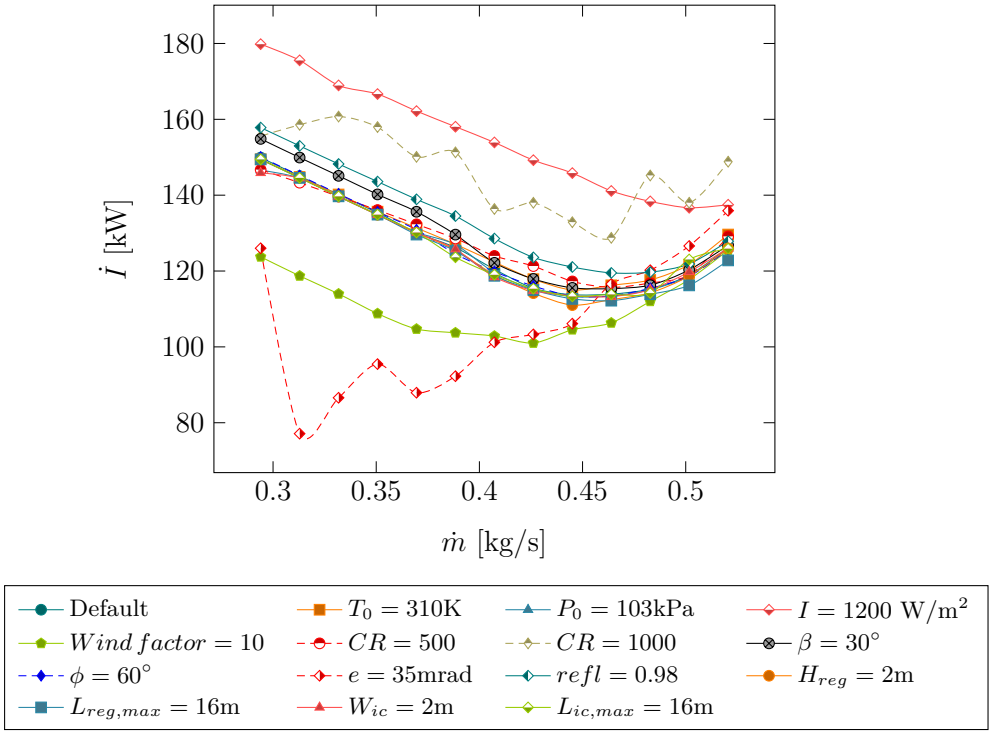


Figure 5.7 Irreversibility rate - elevated parameter values; Case II (16; 20).

As anticipated, increasing the irradiance and concentrator specular reflectivity values results in greater heat input and higher maximum net power output, although increased irreversibility is noted as a result.

Contrary effects are observed for decreased parameter values, as shown in Figs. 5.8 - 5.10. Lower ambient temperature and higher pressure result in slightly higher maximum net power output, and increasing the concentration ratio increases the net heat input. However, the maximum net power output decreases due to reduction in the receiver size in order to reduce thermal losses. Accordingly, the reductions are accompanied by an increase in the irreversibility rate.

Similar changes are noted when the maximum receiver surface temperature constraint is lowered, although no increase in the net heat input is observed. Strong wind and increased concentrator error also significantly reduce the net heat input and power output, as well as the irreversibility rate. By comparison, altering the concentrator rim angle and receiver inclination causes minimal deviation from the default curves shown in Figs. 5.5 - 5.10. Similarly, a reduction in the thermal conductivity of the recuperator and intercooler plate material has little effect on the variables shown.

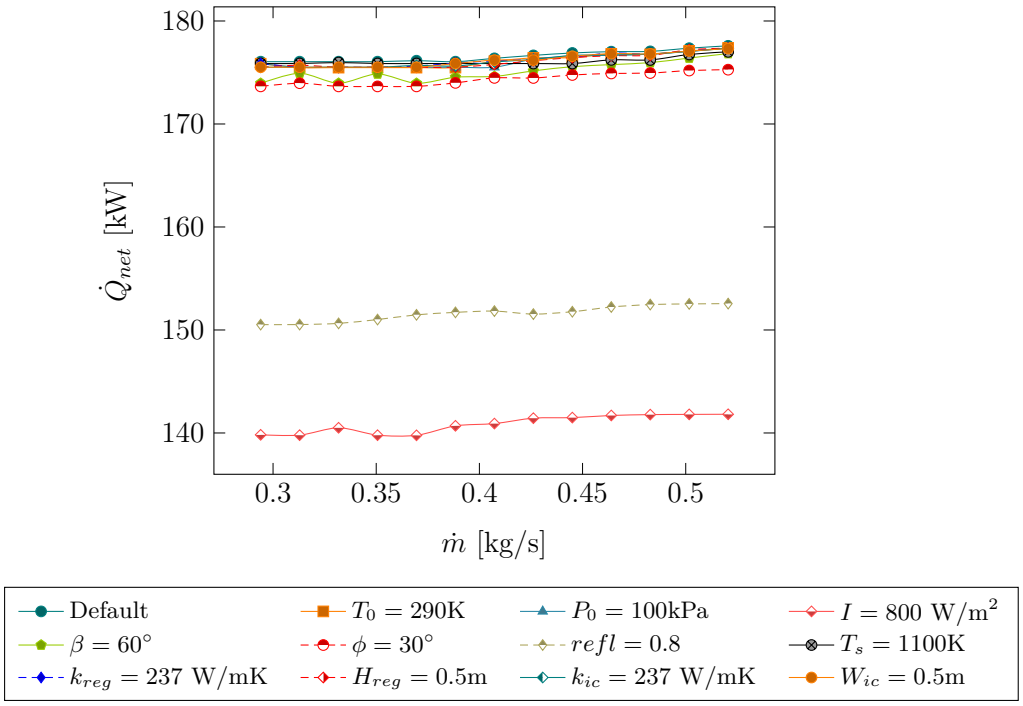


Figure 5.8 Net heat input - lower parameter values; Case II (16; 20).

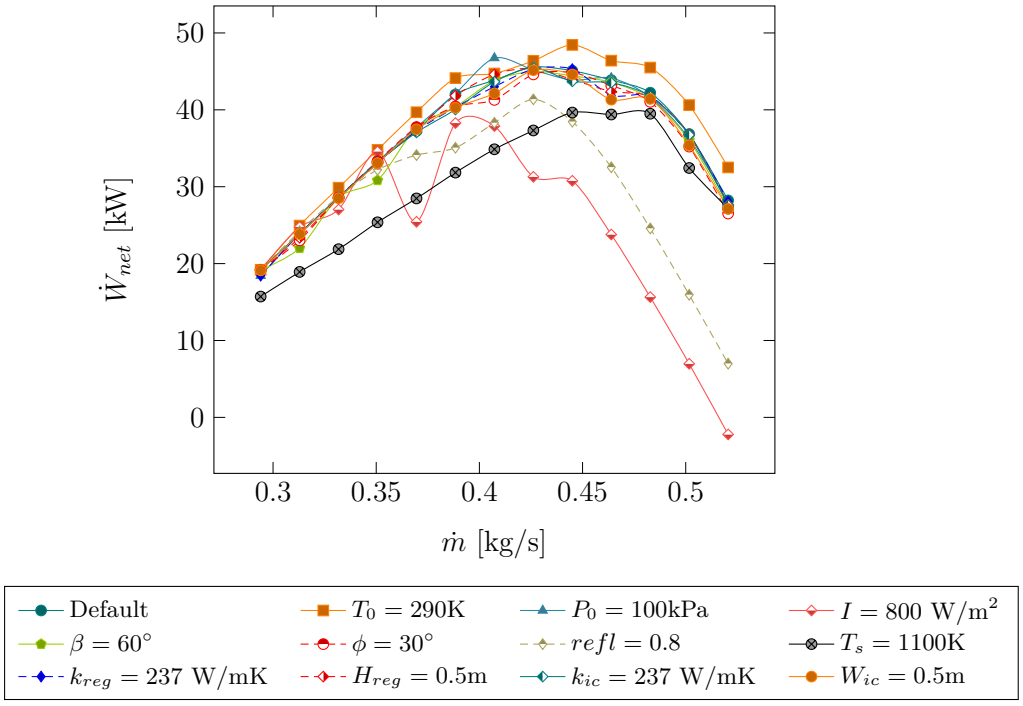


Figure 5.9 Net power output - lower parameter values; Case II (16; 20).

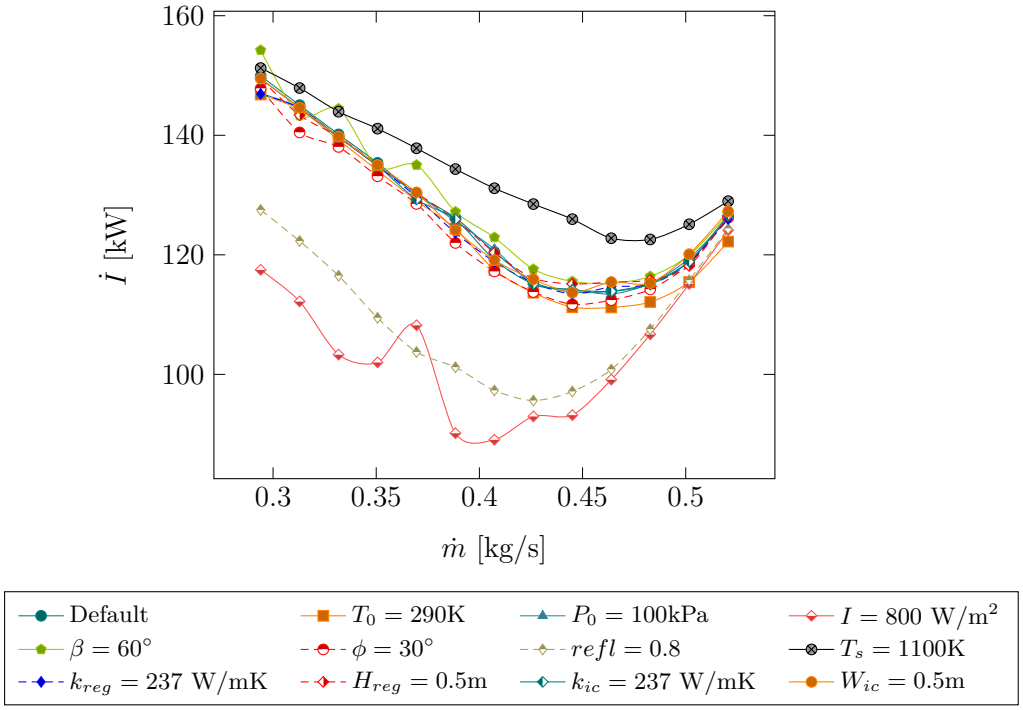


Figure 5.10 Irreversibility rate - lower parameter values; Case II (16; 20).

Changing the recuperator and intercooler constraint values has minimal effect on system performance, but does alter the optimal geometry parameter values returned (discussed in detail in chapter 6). For instance, doubling or halving the intercooler width changes the optimal receiver geometry as shown in Figs. 5.11 - 5.13.

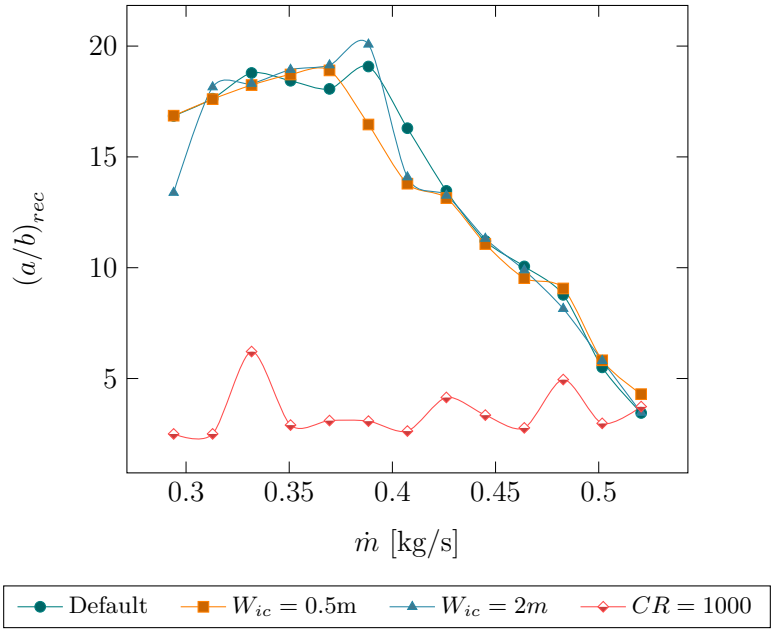


Figure 5.11 Effects of concentration ratio and intercooler width on optimal receiver aspect ratio.

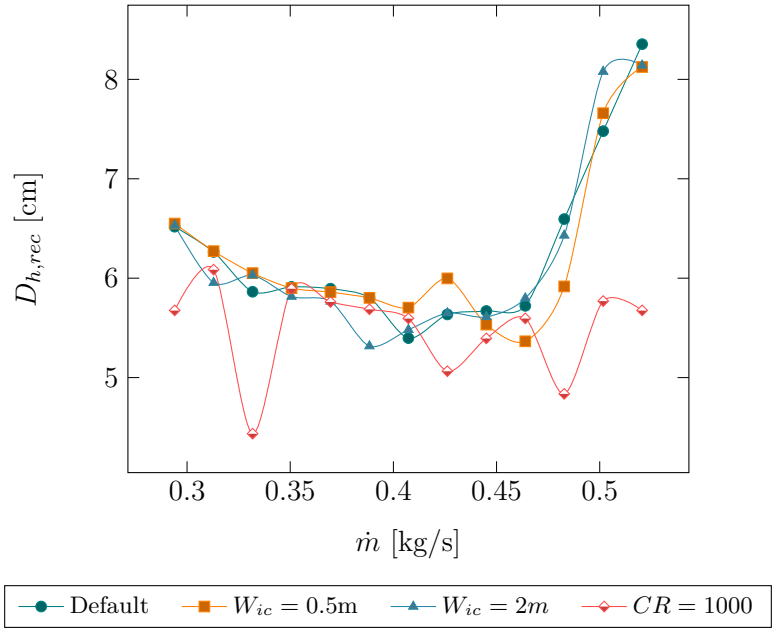


Figure 5.12 Effects of concentration ratio and intercooler width on optimal receiver hydraulic diameter.

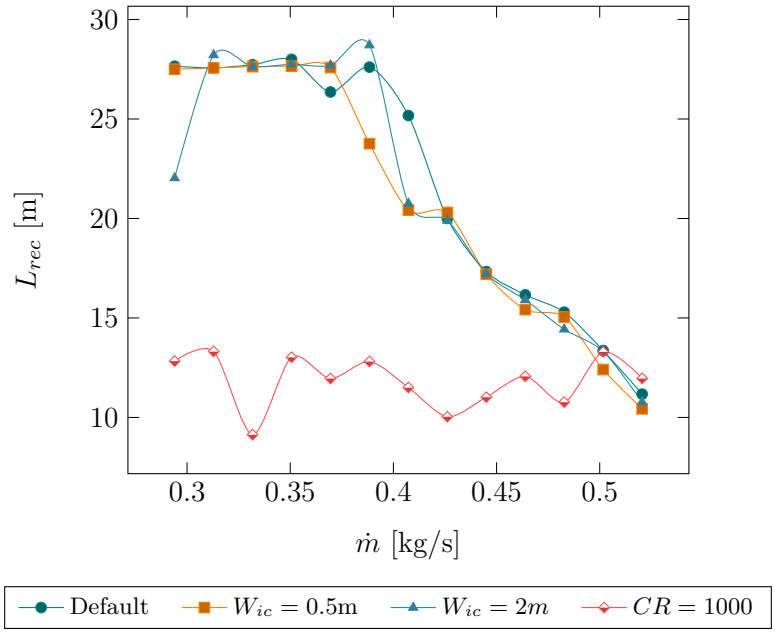


Figure 5.13 Effects of concentration ratio and intercooler width on optimal receiver length.

Though slight compared to the effects caused by changing parameters that alter the rate of heat absorption, such as the receiver concentration ratio, these effects support the findings noted in the literature regarding the interdependence between overall thermodynamic performance and optimal design of the individual components.

In summary, alteration of the default assumption- and constraint values within the control of the designer is found to facilitate no significant improvement in performance. These observations prevail for all cases and configurations, and the default assumptions and constraints are therefore concluded to be appropriate for the present analysis.

Chapter 6

Results

The results of the foregoing analysis are presented in this chapter. Optimal values of the objective function are shown for each of the three systems, and compared to the reference case. The relationships between the performance of the systems and the geometry of the components are then analysed by considering heat input, irreversibility and associated entropy generation, and operating conditions of the receivers, recuperators, intercoolers and reheaters.

6.1 Indicators of thermodynamic performance

6.1.1 Optimal availability

Figure 6.1 shows the optimal net power output relative to total concentrator surface area, for each case described in section 3.1. Optimal thermal efficiency relative to total concentrator surface area is shown for all feasible configurations in Fig. 6.2. Note that the data shown in Figs. 6.1 - 6.3 are primary optima, as defined in section 4.2.1. It can be seen from Figs. 6.1 and 6.2 that the intercooled cycle produces the greatest net power output, and has the highest thermal efficiency for a given concentrator surface area, followed by the combined, reheated, and reference cycles, respectively. The differences observed can be intuited by comparing the system irreversibilities, shown in Fig. 6.3, from which it can be seen that the percentage of net heat input lost to total irreversibility is greatest in the case of the reheated cycle, and least for the intercooled cycle.

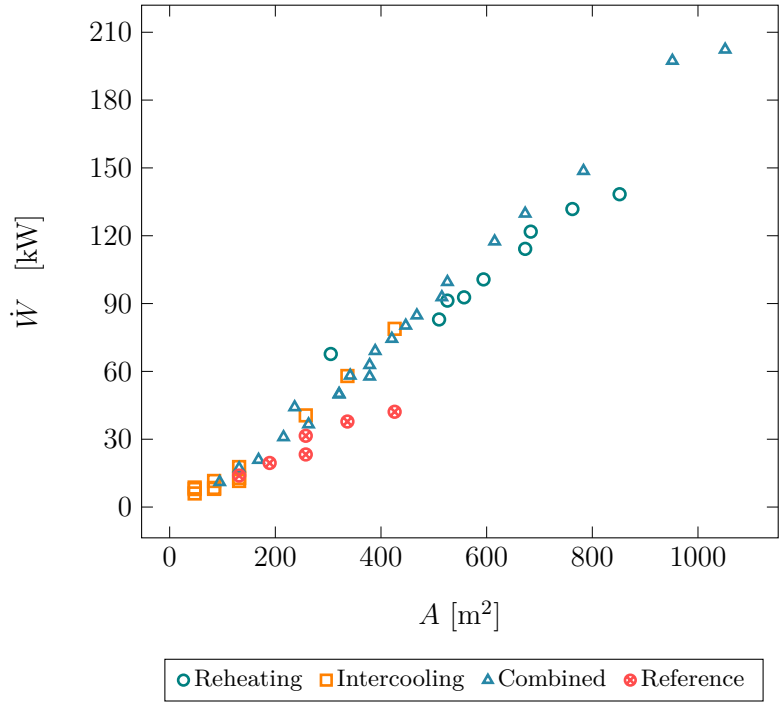


Figure 6.1 Maximum power output per unit total concentrator surface area.

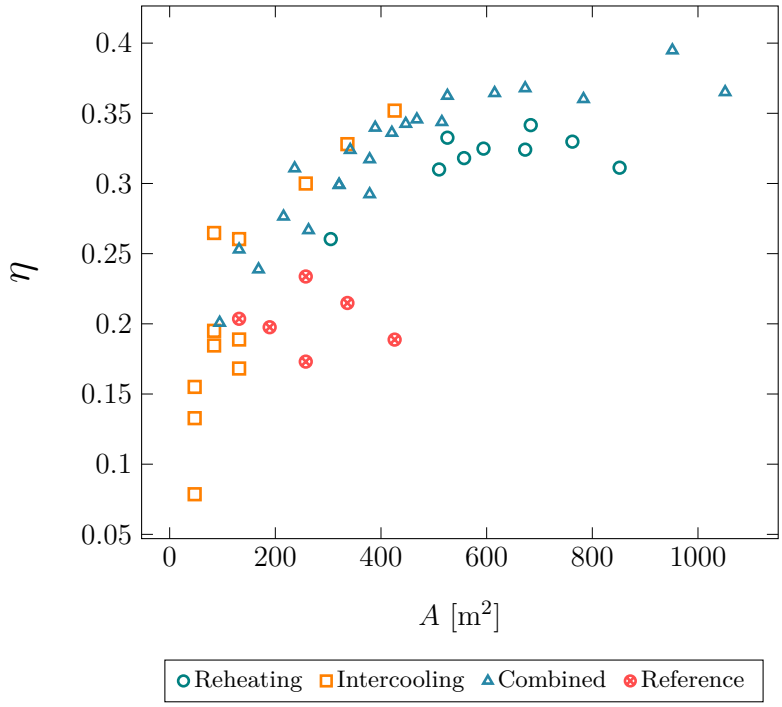


Figure 6.2 Maximum thermal efficiency per unit total concentrator surface area.

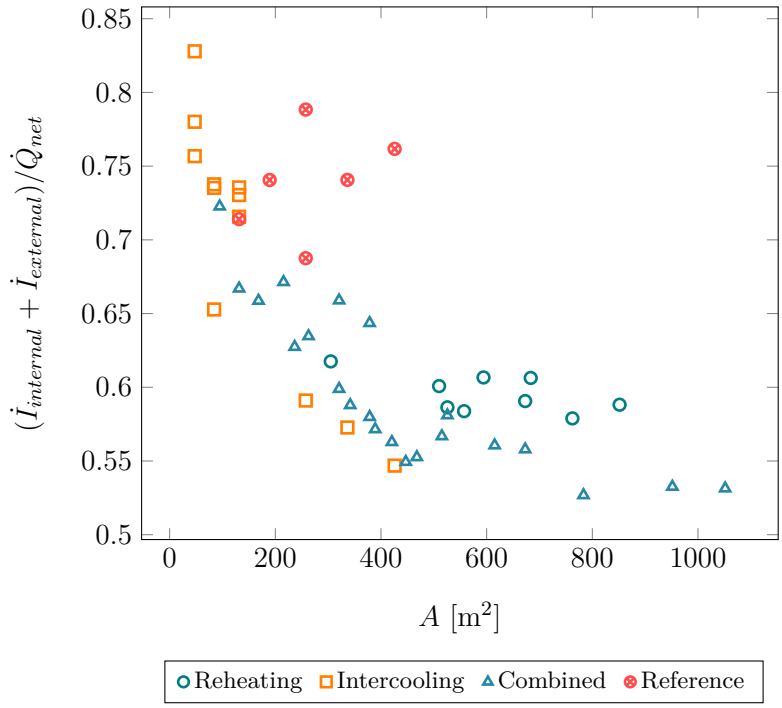


Figure 6.3 Minimum exergy lost to irreversibility per unit total concentrator surface area.

Note also that the ratio of internal to external irreversibility is typically greater in systems with intercoolers, as shown in Figs. 6.4 - 6.6. The increased irreversibility ratios are due to lower external irreversibilities, the causes of which are traced in the following section by examining the distribution of entropy generation in each case.

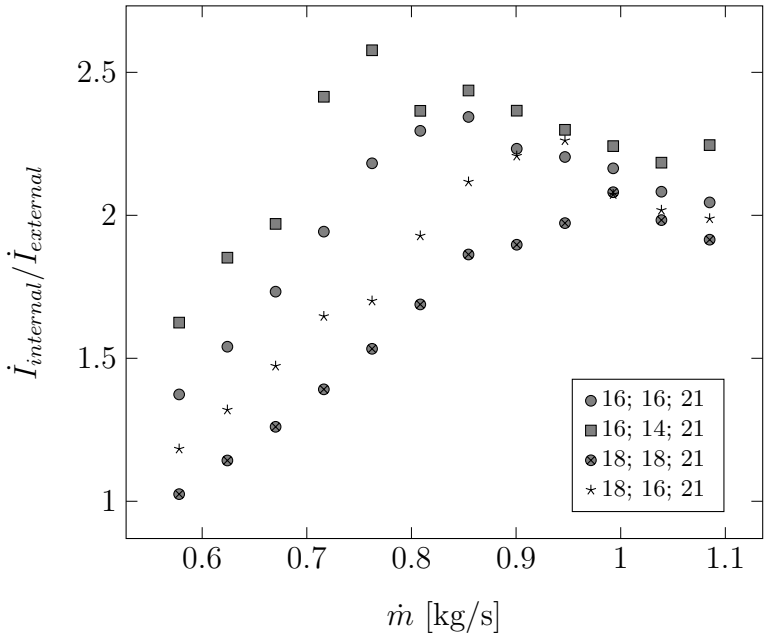


Figure 6.4 Variation in optimal system irreversibility ratio with mass flow rate; Case I.

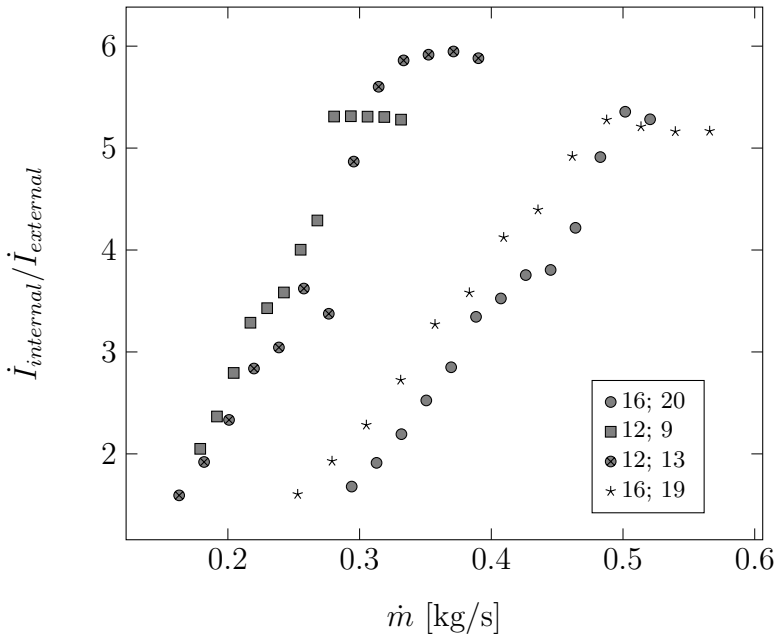


Figure 6.5 Variation in optimal system irreversibility ratio with mass flow rate; Case II.

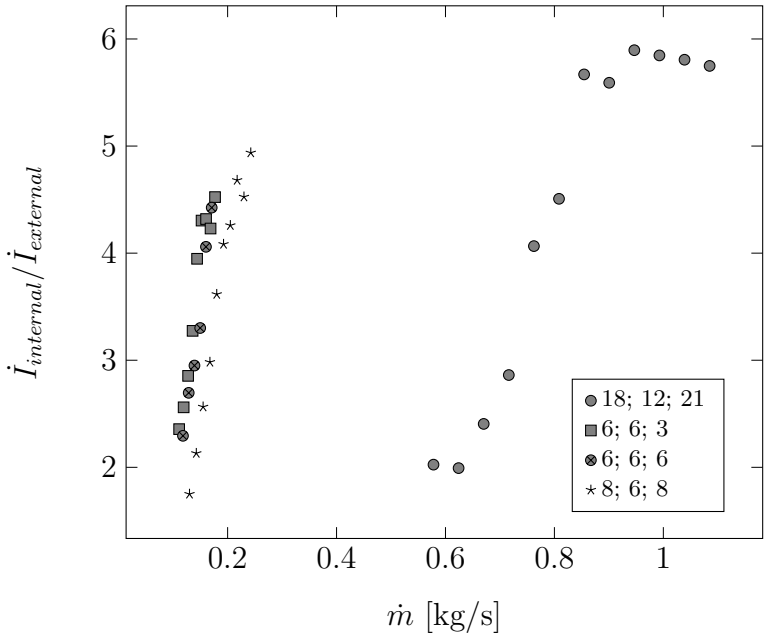


Figure 6.6 Variation in optimal system irreversibility ratio with mass flow rate; Case III.

6.1.2 Entropy generation

Figures 6.7 - 6.9 show the optimal distribution of entropy generation across the components in the systems with respect to mass flow rate. It can be seen that the receiver is a dominant source of entropy generation in all cases, and that contributions from turbines and compressors are initially small, increasing with mass flow rate. Conversely, the recuperator contributions are initially large, decreasing with mass flow rate. Intercooler entropy generation contributions are proportionally the smallest, increasing only slightly with mass flow rate. The relative proportion of the reheater contribution differs notably between cases I and III, like that of the recuperator in Case II relative to cases I and III.

Entropy generation in the receiver is notably higher in cases II and III. In Case II, this can be attributed to there being only one collector, and Figs. 6.7 - 6.9 suggest that entropy distribution in cases II and III is such that the increased entropy generation is justified by the associated increase in power output, shown in Tables 6.1 - 6.3.

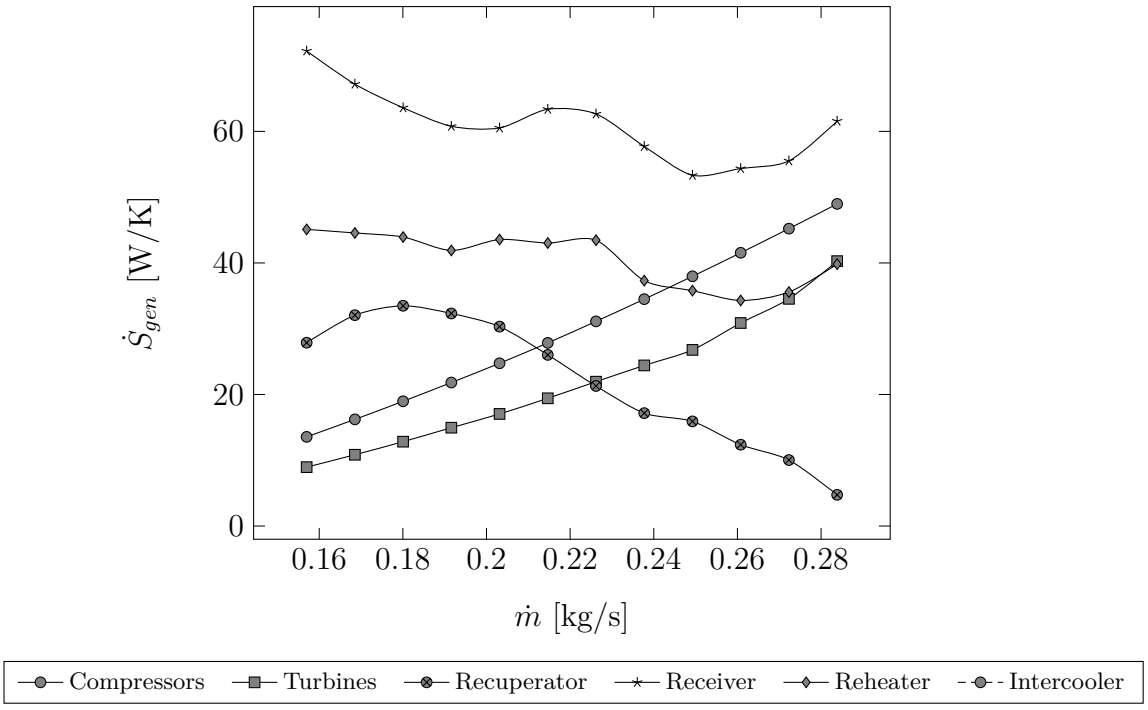


Figure 6.7 Change in optimal distribution of entropy generation with system mass flow rate; Case I (10; 8; 11).

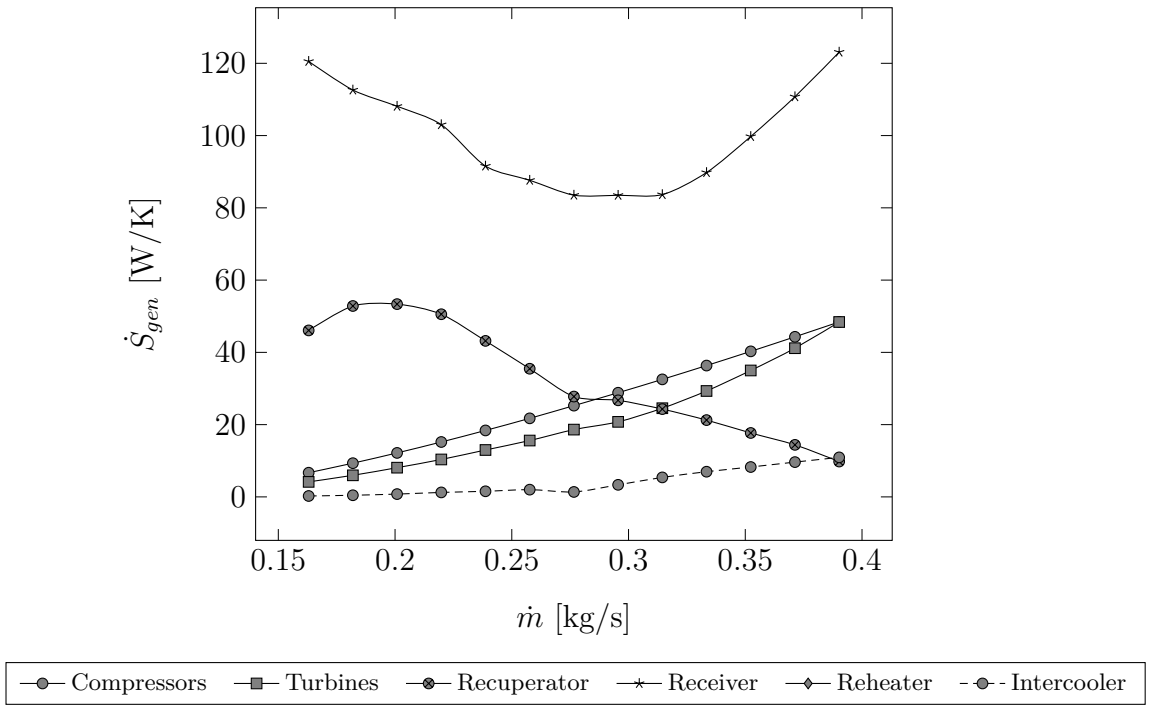


Figure 6.8 Change in optimal distribution of entropy generation with system mass flow rate; Case II (12; 13).

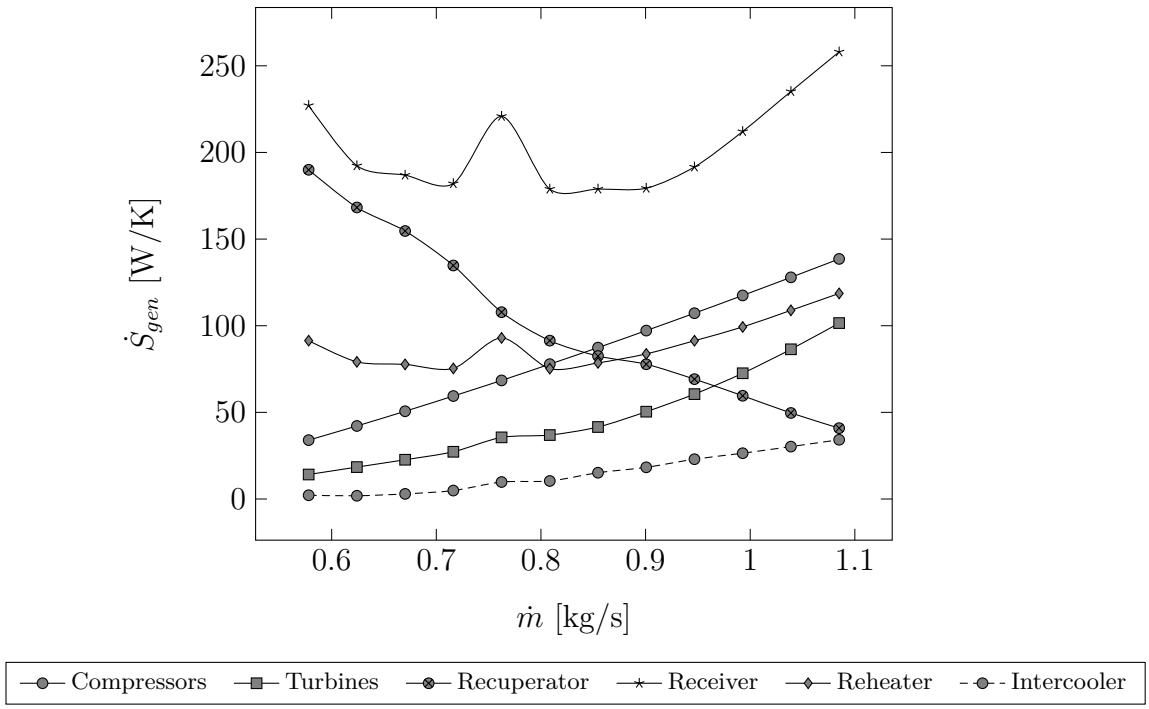
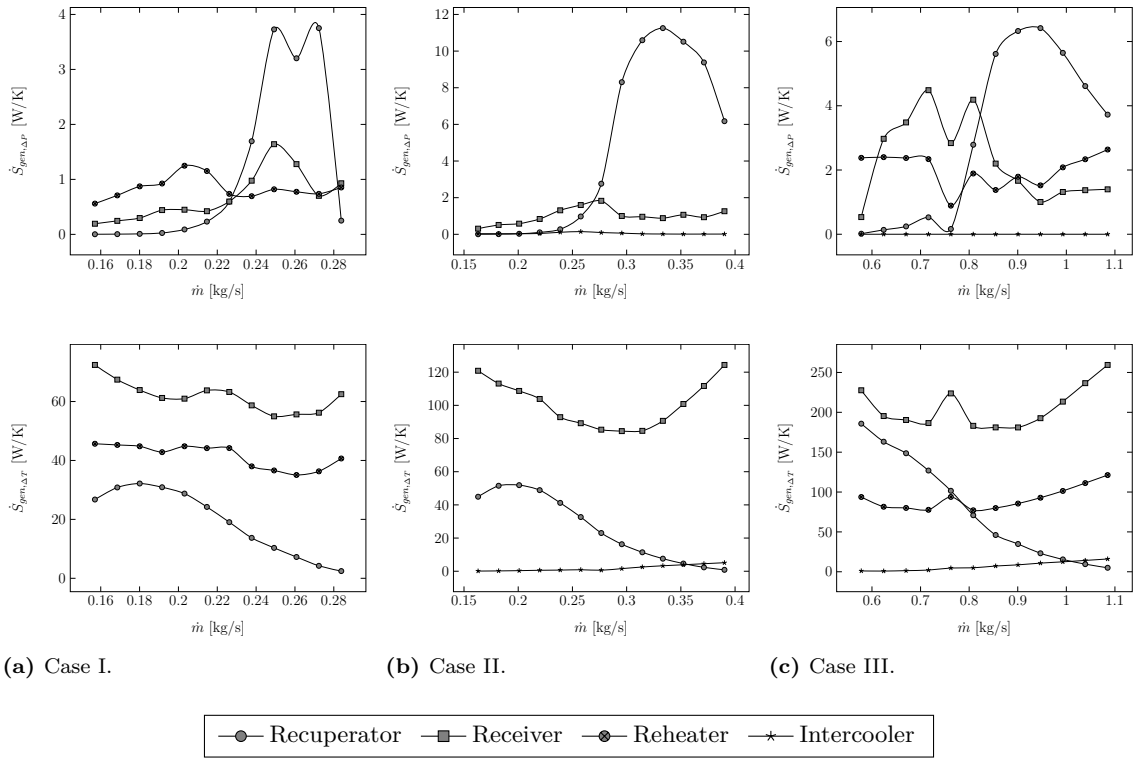


Figure 6.9 Change in optimal distribution of entropy generation with system mass flow rate; Case III (18; 12; 21).

The components of entropy generation due to temperature differences and frictional pressure drop across the system components are shown in Fig. 6.10, with respect to system mass flow rate. It can be seen that temperature differences are the primary cause of entropy generation in the system components.



(a) Case I. **(b)** Case II. **(c)** Case III.

Figure 6.10 Temperature and pressure components of entropy generation.

Figure 6.10 underscores the relatively small proportions of the entropy generation in the recuperator in the case of the intercooled cycle compared to the reheated and combined cycles, and of the entropy generation in the receiver in the case of the reheated cycle, compared to the intercooled and combined cycles.

This observation is indicative of interaction between the operating conditions of the system components, such that irreversibility is minimised, causing a greater proportion of exergy to become available in the form of net power output. These interactions are discussed in greater detail in section 6.2, below, where the relationships between the optimal geometry and operating conditions of the system components are considered.

Table 6.1 Optimal geometry variables and maximum net power output; Case I, configuration (10; 8; 11) - rectangular section absorber tube.

r	$\dot{m}$ (kg/s)	$(a/b)_{reg}$	$D_{h,reg}$ (cm)	L_{reg} (m)	$(a/b)_{rec}$	$D_{h,rec}$ (cm)	L_{rec} (m)	$(a/b)_{reh}$	$D_{h,reh}$ (cm)	L_{reh} (m)	$(\dot{W}_{net})_{max}$ (kW)
1.6	0.157	7.103	8.118	2.773	4.391	7.039	8.898	8.417	6.039	13.917	11.41631
1.7	0.169	5.490	7.189	3.656	5.017	6.743	9.513	8.476	5.928	14.047	13.73152
1.8	0.180	5.304	6.665	4.947	5.612	6.539	10.138	8.584	5.817	13.983	16.2394
1.9	0.192	5.202	4.782	4.996	6.473	5.957	10.439	8.019	5.940	13.121	18.67657
2.0	0.203	5.313	3.258	4.968	7.868	5.950	12.372	8.548	5.727	14.379	21.94851
2.1	0.215	6.452	2.271	4.978	9.391	6.055	14.753	7.992	6.061	14.224	24.5497
2.2	0.226	7.915	1.593	4.998	10.082	5.632	15.288	6.988	7.063	14.254	26.55777
2.3	0.238	8.905	1.122	5.000	8.752	5.214	13.381	4.976	7.511	10.962	26.17802
2.4	0.249	8.609	0.920	5.000	6.118	4.971	10.618	4.488	7.286	9.376	23.29149
2.5	0.261	8.967	0.969	4.977	4.755	5.329	8.605	2.677	7.652	6.605	18.36352
2.6	0.272	18.959	0.671	4.992	4.052	6.170	7.868	2.500	7.843	6.437	14.66956
2.7	0.284	18.160	1.600	4.935	2.541	5.962	5.990	2.504	7.576	6.226	4.935223

Table 6.2 Optimal geometry variables and maximum net power output; Case II, configuration (12; 13) - rectangular section absorber tube.

r	$\dot{m}$ (kg/s)	$(a/b)_{ic}$	$D_{h,ic}$ (cm)	L_{ic} (m)	$(a/b)_{reg}$	$D_{h,reg}$ (cm)	L_{reg} (m)	$(a/b)_{rec}$	$D_{h,rec}$ (cm)	L_{rec} (m)	$(\dot{W}_{net})_{max}$ (kW)
1.3	0.163	0.0357	9.346	4.922	6.708	7.182	3.449	12.741	6.380	20.642	8.903831
1.4	0.182	0.0277	9.441	4.861	5.358	6.297	4.986	14.123	5.741	20.841	12.89056
1.5	0.201	0.0296	9.315	5.202	5.381	5.127	5.975	13.683	5.819	20.087	17.0517
1.6	0.220	0.0305	9.479	5.615	5.756	3.486	5.996	13.166	5.532	19.577	21.27098
1.7	0.239	0.0361	8.687	5.562	5.820	2.581	5.998	10.447	5.380	15.641	24.34786
1.8	0.258	0.0398	7.865	5.376	6.253	1.771	5.998	8.579	5.014	13.491	27.21761
1.9	0.277	0.0189	7.361	4.643	8.107	1.110	6.000	8.174	5.560	13.246	29.79667
2.0	0.296	0.0216	11.714	5.959	19.262	0.565	6.000	6.594	7.035	15.518	31.38919
2.1	0.314	0.0236	16.146	6.000	27.970	0.457	6.000	4.293	7.194	9.777	27.25196
2.2	0.333	0.0227	16.502	6.000	27.501	0.471	6.000	2.500	7.378	6.988	21.75659
2.3	0.352	0.0225	18.398	5.998	34.715	0.435	5.995	2.501	7.180	5.896	15.47922
2.4	0.371	0.0213	17.957	6.000	34.373	0.465	6.000	2.500	6.814	5.593	8.60446
2.5	0.390	0.0200	17.283	6.000	39.419	0.489	6.000	2.500	6.630	5.442	1.243958

Table 6.3 Optimal geometry variables and maximum net power output; Case III, configuration (18; 12; 21) - rectangular section absorber tube

r	$\dot{m}$ (kg/s)	$(a/b)_{ic}$	$D_{h,ic}$ (mm)	L_{ic} (m)	$(a/b)_{reg}$	$D_{h,reg}$ (mm)	L_{reg} (m)	$(a/b)_{rec}$	$D_{h,rec}$ (m)	L_{rec} (m)	$(a/b)_{reh}$	$D_{h,reh}$ (cm)	L_{reh} (m)	$(\dot{W}_{net})_{max}$ (kW)
1.5	0.578	0.0109	10.933	5.468	6.500	5.818	7.244	7.689	10.170	20.772	5.618	12.142	18.843	54.80458
1.6	0.624	0.0090	11.702	6.569	8.066	5.061	8.990	9.154	9.804	23.345	5.377	12.856	19.225	71.17246
1.7	0.670	0.0085	13.049	8.041	11.812	3.458	9.000	11.722	9.504	28.352	5.204	14.153	20.608	90.06687
1.8	0.716	0.0083	14.789	8.774	15.617	2.412	8.998	12.291	9.093	28.346	4.105	14.953	17.899	104.9245
1.9	0.762	0.0079	14.763	8.999	21.412	1.612	8.999	11.418	9.173	26.729	3.253	15.948	15.906	117.4817
2.0	0.808	0.0074	14.285	8.999	22.664	1.191	8.924	8.677	9.115	20.743	2.591	14.829	12.532	117.2395
2.1	0.855	0.0073	15.183	9.000	27.840	0.894	9.000	5.854	10.330	16.602	2.505	14.585	11.986	112.6796
2.2	0.901	0.0088	26.848	9.000	71.000	0.679	8.997	3.006	13.375	15.975	2.500	16.956	13.916	106.5334
2.3	0.947	0.0081	23.961	9.000	55.222	0.606	9.000	2.500	14.349	11.781	2.500	15.693	12.885	91.18202
2.4	0.993	0.0078	24.053	9.000	50.169	0.657	9.000	2.500	14.397	11.822	2.500	15.559	12.778	72.71028
2.5	1.039	0.0076	25.757	9.000	76.279	0.563	9.000	2.500	13.566	11.160	2.500	14.692	12.058	55.75506
2.6	1.085	0.0073	25.399	9.000	81.157	0.581	9.000	2.500	13.185	10.822	2.500	14.044	11.527	35.71985

6.2 Operating conditions and optimal geometry

6.2.1 Recuperators

Considering Figs. 6.7 - 6.9 above, it can be seen that entropy generation in the recuperator decreases as the system mass flow rate increases. Conversely, the entropy generation rates in the compressors rise, due to the increase in the temperature differences across the compressors caused by the increasing pressure ratio. Evaluating the change in temperature field values with increasing mass flow rate, reproduced for Case III in Table 6.4, it can be seen that the temperature increase effected in the compressors increases, reducing the difference between the inlet temperatures of the recuperator.

Fig. 6.11 shows that the optimum number of recuperator channels increases as the system mass flow rate increases, increasing the total plate area and thus promoting heat transfer despite the reduction in the difference between the hot and cold stream temperatures. An increase in the number of thermal units transferred between the fluid streams is consequently noted. With the thermal potential difference reduced, the resulting heat transfer constitutes a greater percentage of the maximum attainable. Consequently, the recuperator effectiveness increases with increasing mass flow rate before reaching a plateau, as shown in Fig. 6.12.

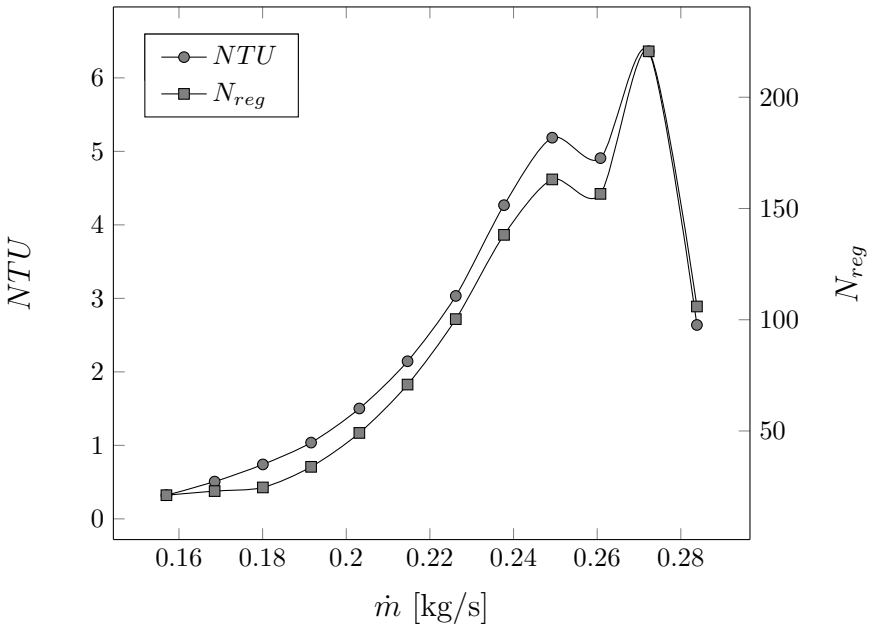


Figure 6.11 Variation in optimal recuperator NTU and number of recuperator channels with system mass flow rate; Case I (16; 16; 21).

Table 6.4 Change in temperature field values with system mass flow rate; Case III, configuration (18; 12; 21) - rectangular section absorber tube.

r	$\dot{m}$ (kg/s)	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	T_{12}	T_{13}	T_{14}	T_{15}	T_{16}	T_{17}	T_{18}
1.5	0.578	300	346.64	344.64	323.25	321.25	371.20	369.20	721.97	719.09	1067.47	1065.47	979.23	977.23	1126.67	1124.67	1033.64	1031.64	700.94
1.6	0.624	300	354.58	352.58	323.52	321.52	380.02	378.02	776.62	774.38	1094.57	1092.57	990.05	988.05	1126.26	1124.26	1018.77	1016.77	637.99
1.7	0.670	300	362.17	360.17	322.60	320.60	387.03	385.03	831.55	829.31	1124.97	1122.97	1004.22	1002.22	1130.60	1128.60	1009.25	1007.25	577.60
1.8	0.716	300	369.44	367.44	322.10	320.10	394.20	392.20	858.60	856.31	1132.21	1130.21	998.96	996.96	1117.53	1115.53	985.99	983.99	530.92
1.9	0.762	300	376.44	374.44	323.95	321.95	403.98	401.98	874.13	871.60	1130.68	1128.68	987.35	985.35	1099.12	1097.12	959.75	957.75	494.82
2.0	0.808	300	383.17	381.17	326.59	324.59	414.58	412.58	860.23	857.86	1103.46	1101.46	957.17	955.17	1063.34	1061.34	922.30	920.30	478.94
2.1	0.855	300	389.67	387.67	327.49	325.49	422.78	420.78	841.52	838.96	1072.69	1070.69	925.61	923.61	1026.55	1024.55	885.73	883.73	467.11
2.2	0.901	300	395.95	393.95	322.05	320.05	422.42	420.42	759.80	758.80	983.80	981.80	834.61	832.61	931.57	929.57	790.21	788.21	450.98
2.3	0.947	300	402.03	400.03	324.94	322.94	432.77	430.77	738.37	736.78	951.92	949.92	806.43	804.43	899.12	897.12	761.61	759.61	453.49
2.4	0.993	300	407.92	405.92	326.43	324.43	441.14	439.14	671.40	669.78	877.32	875.32	735.12	733.12	824.50	822.50	690.76	688.76	457.41
2.5	1.039	300	413.64	411.64	327.57	325.57	448.90	446.90	622.44	620.68	821.12	819.12	680.88	678.88	767.20	765.20	636.07	634.07	459.06
2.6	1.085	300	419.20	417.20	329.05	327.05	457.00	455.00	574.90	573.14	767.27	765.27	629.60	627.60	713.21	711.21	585.13	583.13	463.49

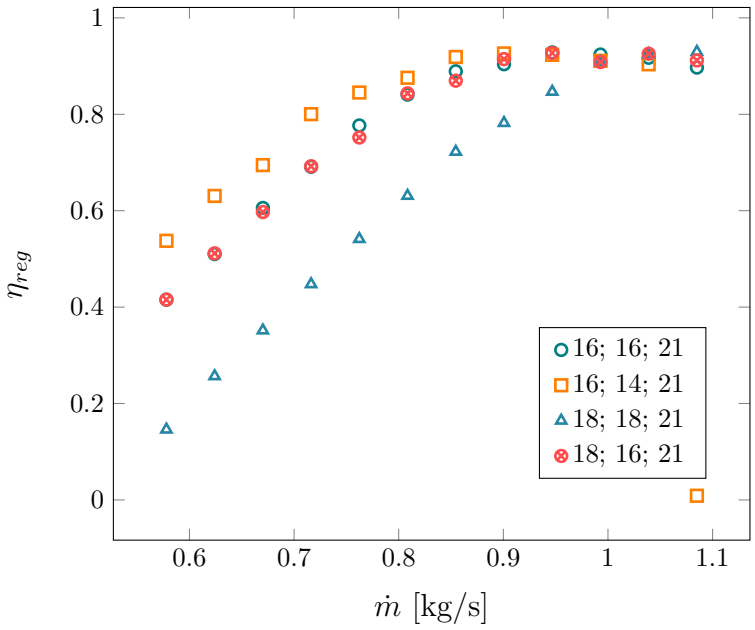


Figure 6.12 Relationship between recuperator effectiveness and system mass flow rate; Case I.

The channel length and aspect ratio increase with system mass flow rate as shown in Figs. 6.13 and 6.14, increasing the surface area of the recuperator plates. In each case, the channel length increases rapidly until the maximum constraint (Cstr.6) is reached.

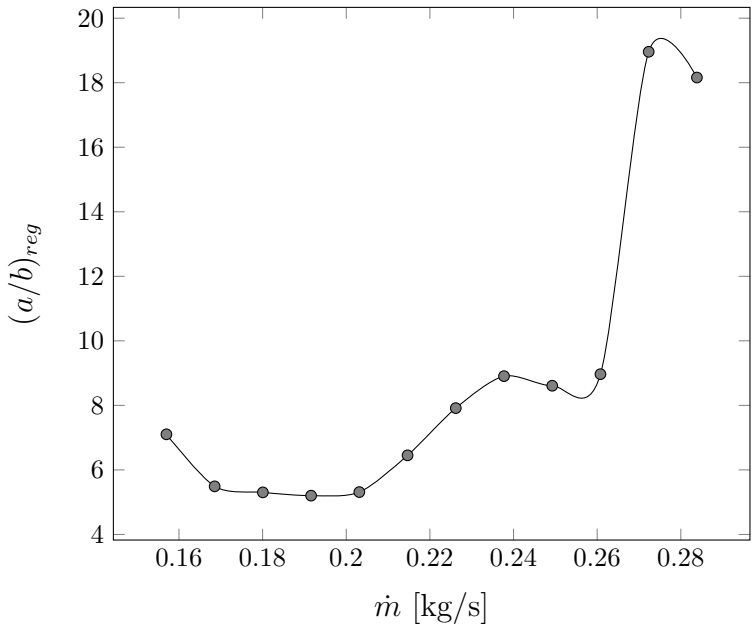


Figure 6.13 Relationship between recuperator channel aspect ratio and system mass flow rate, Case I (16; 16; 21).

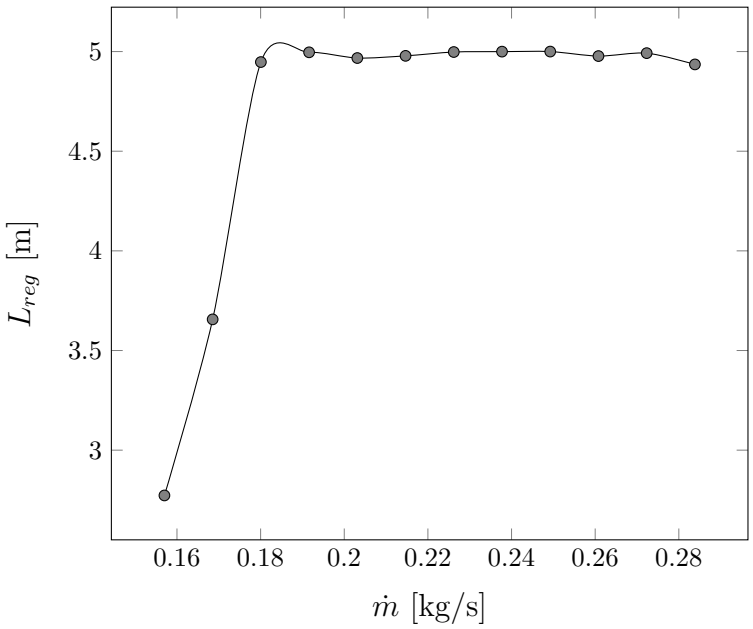


Figure 6.14 Relationship between recuperator length and system mass flow rate, Case I (16; 16; 21).

The channel hydraulic diameter decreases with increasing system mass flow rate (Fig. 6.15), on account of the increase in the number of plates brought about to increase the heat transfer effectiveness. The inverse relationship between the recuperator effectiveness and channel hydraulic diameter is shown in Figs. 6.16 - 6.18.

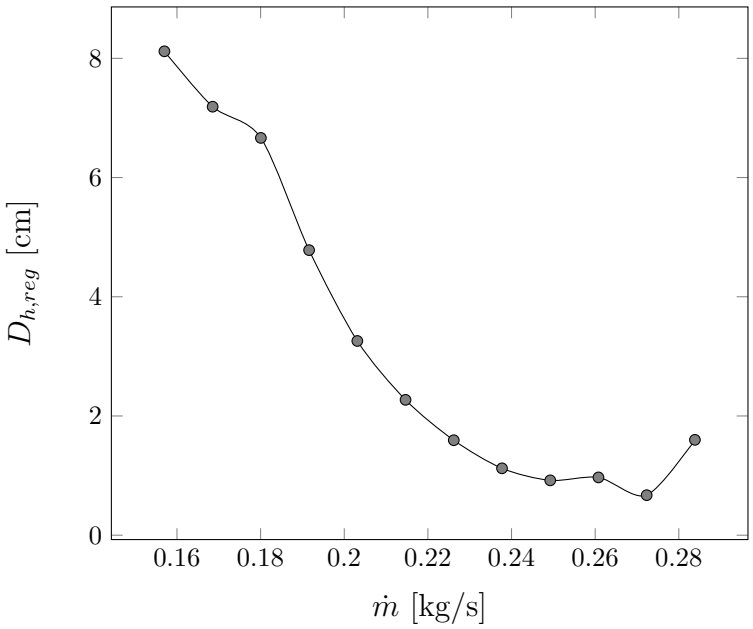


Figure 6.15 Relationship between recuperator channel hydraulic diameter and system mass flow rate, Case I (16; 16; 21).

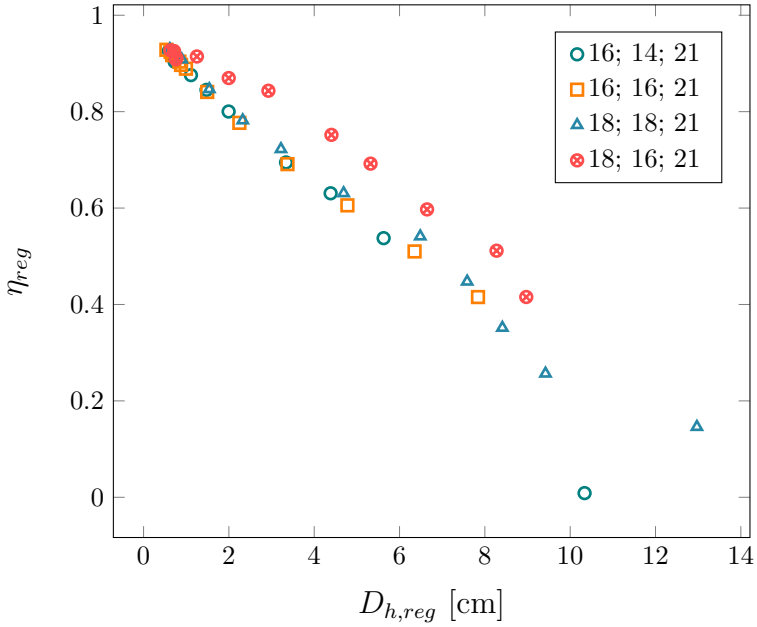


Figure 6.16 Relationship between recuperator heat transfer effectiveness and channel hydraulic diameter; Case I.

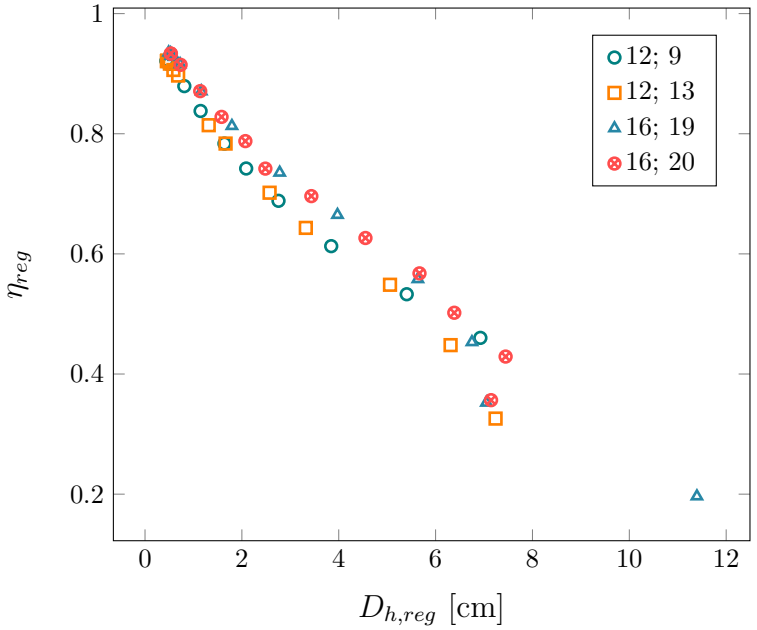


Figure 6.17 Relationship between recuperator heat transfer effectiveness and channel hydraulic diameter; Case II.

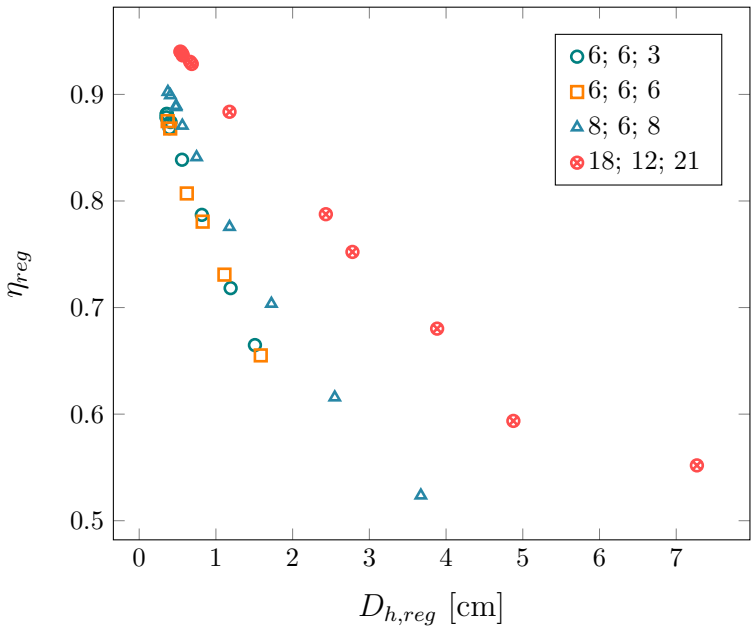


Figure 6.18 Relationship between recuperator heat transfer effectiveness and channel hydraulic diameter; Case III.

While the curves shown Figs. 6.13 - 6.15 pertain to Case I, identical trends are noted in the curves for cases II and III. Moreover, the absolute aspect ratio values are larger, and the hydraulic diameter values smaller for cases II and III compared to Case I, as shown in Tables 6.1 - 6.3. The differences are anticipated, given the relative entropy generation and net power output rates. Note also that the recuperator geometry shown pertains to configurations using rectangular cross-section receiver and reheater absorbers, and that similar behaviour is noted for circular tube absorbers.

6.2.2 Intercoolers

As previously mentioned, the increase in the compressor pressure ratio with mass flow rate causes an associated increase in the intercooler inlet temperature. Figure 6.19 shows that the inlet temperature increase is mediated by an increase in heat transfer effectiveness, defined in terms of the intercooler inlet and outlet temperatures by Eq. 6.1 (Turbo by Garrett, 2016b).

$$\eta_{ic} = \frac{T_i - T_e}{T_i} \tag{6.1}$$

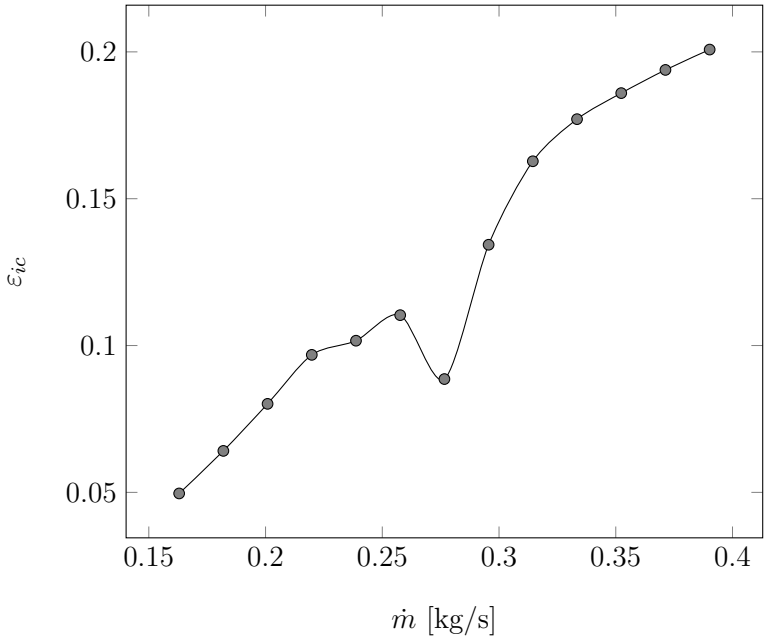


Figure 6.19 Relationship between optimal intercooler effectiveness and system mass flow rate; Case II.

Accordingly, the difference in temperature across the intercooler increases with mass flow rate. The difference between the temperatures at the LP compressor inlet (ambient temperature) and at the intercooler inlet is, however, relatively small in comparison to that across the recuperator fluid streams (Table 6.4), hence the small intercooler entropy production rates, as shown in Figs. 6.7 - 6.9. Note that the curves in question correspond to Case II, although similar behaviour is observed for Case III, and that no significant differences are observed in the intercooler parameters when comparing the results obtained considering rectangular and circular section absorber tubes.

Generally, the optimal number of intercooler channels, and the associated channel aspect ratio display a decreasing trend with increasing mass flow rate, as shown in Figs. 6.20 and 6.21. An increase in the number of flow channels is, however, noted in Case II, between $\dot{m} = 0.225\,\text{kg/s}$ and $\dot{m} = 0.275\,\text{kg/s}$, and a corresponding decrease in the channel aspect ratio, shown in Figs. 6.20 and 6.21, respectively. Figures 6.7 - 6.9 and the data in Table 6.4 suggest that the cumulative entropy production rate at mass flow rates in the middle of the operating range of the micro-turbines is such that the relatively low entropy production rates in the other components justifies the entropy generated in rejecting the relatively large quantity of heat by increasing the number of channels and reducing the channel aspect ratio.

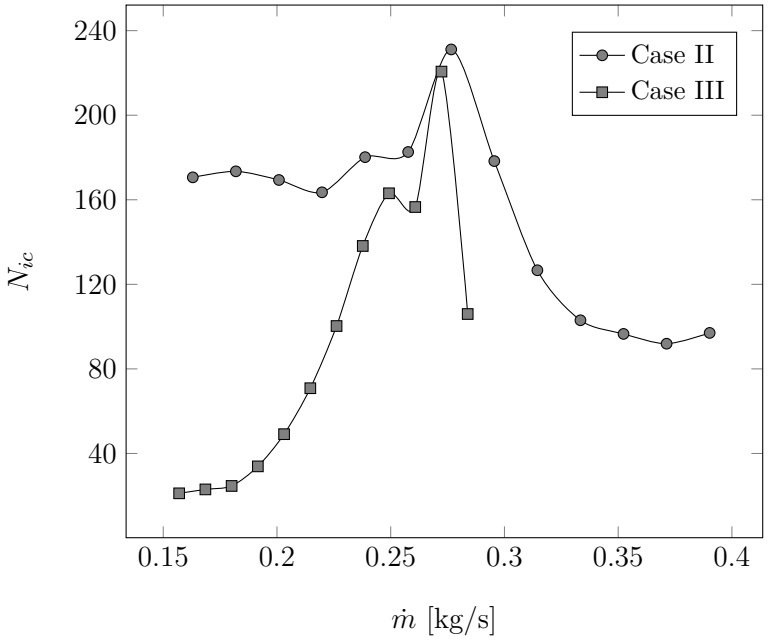


Figure 6.20 Relationship between optimal number of intercooler channels and system mass flow rate.

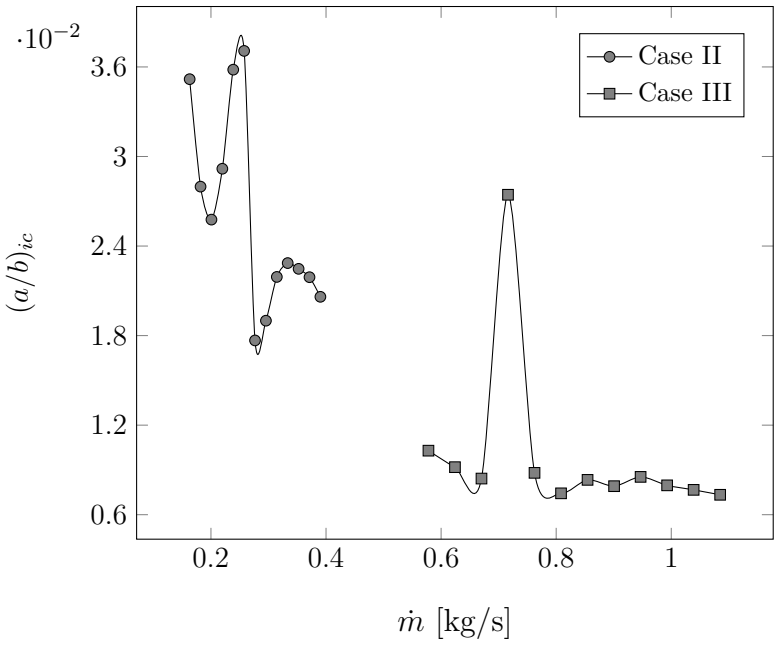


Figure 6.21 Relationship between optimal intercooler channel aspect ratio and system mass flow rate.

Figure 6.22 indicates that the channel hydraulic diameter increases as the mass flow rate through the system increases, causing the plate surface area to increase accordingly. Due to the small difference between the temperature of the hot fluid flowing internally and the temperature of the cooling ambient air, the values noted for the hydraulic diameter of the intercooler channels are an order of magnitude smaller than the values noted for the hydraulic diameter of the recuperator channels.

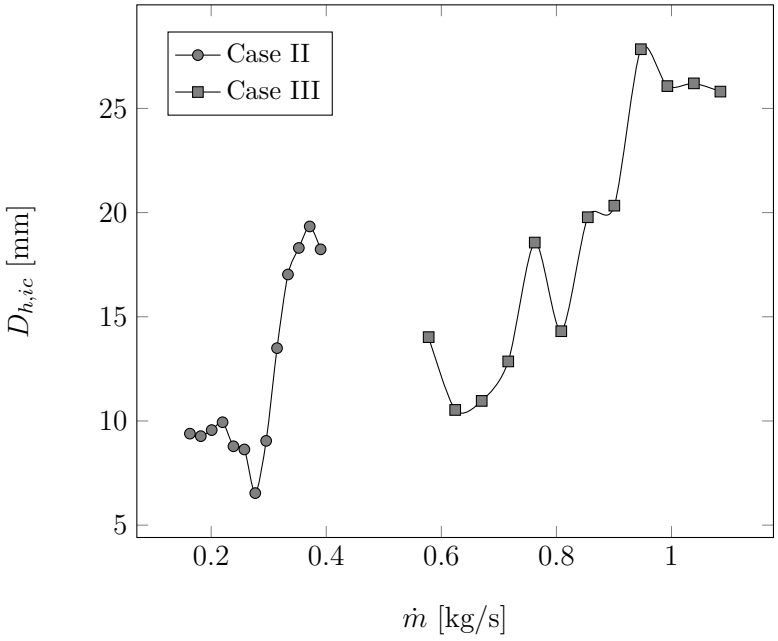


Figure 6.22 Relationship between optimal intercooler channel hydraulic diameter and system mass flow rate.

The length of the intercooler rises, like the length of the recuperator, before reaching a plateau at the maximum constraint, as shown in Fig. 6.23. The sharp decrease observed for Case II at $\dot{m} = 0.275\,\text{kg/s}$ corresponds to the increase in the number of channels and decrease in channel aspect ratio described above.

The mass flow rate through the intercooler channels is found to increase steadily with system mass flow rate, contrary to the mass flow rate through the recuperator channels, which decreases with increasing system mass flow rate, as shown in Fig. 6.24. Given that the recuperator entropy production rate, and especially the component due to frictional pressure loss, is significantly greater than that noted for the intercooler, as shown in Figs. 6.7 - 6.9 and Fig. 6.10, the difference observed between the trends in recuperator channel and intercooler channel mass flow rates with respect to system mass flow suggests that flow through the recuperator channels must be slowed down to control entropy generation, whilst flow through the intercooler channels may be allowed to increase to promote heat transfer.

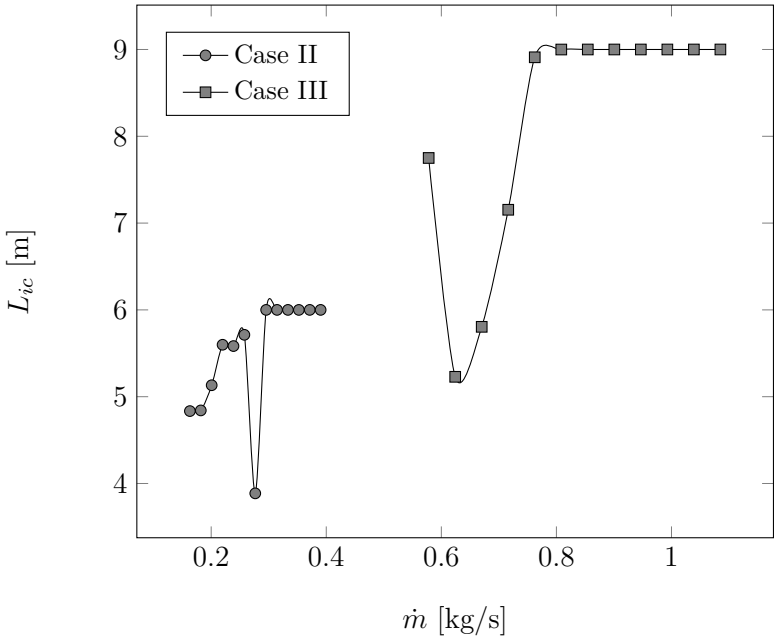


Figure 6.23 Relationship between optimal intercooler length and system mass flow rate.

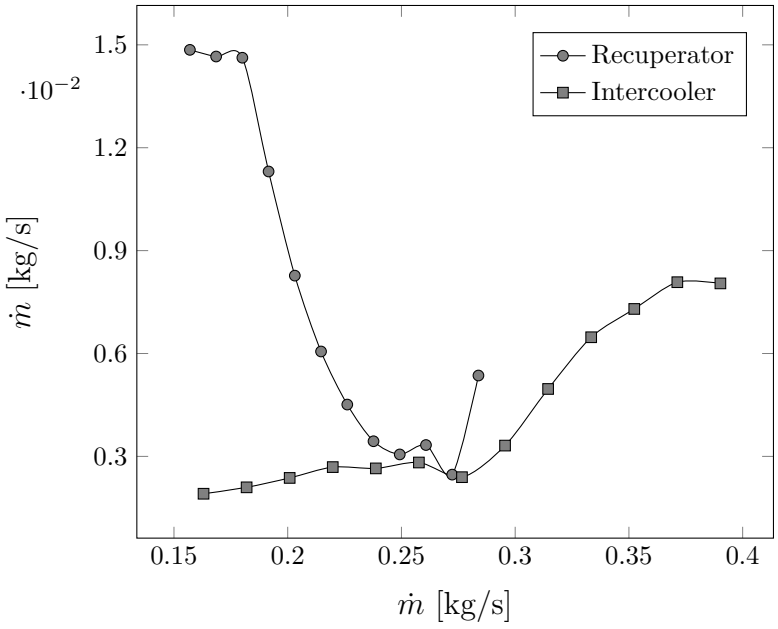


Figure 6.24 Variation in optimal heat exchanger channel mass flow rates with system mass flow rate.

A strong correlation between the optimal mass flow rate through the intercooler channels and the system mass flow rates is observed when the primary optima, i.e. the data corresponding to the operating points at which absolute maximum net power output is found, are considered, for the configurations of Case II. The relationship, shown in Fig. 6.25, can be approximated with the linear equation

$$\dot{m}_c \;=\; 0.0168\,\dot{m} - 0.0011 \tag{6.2}$$

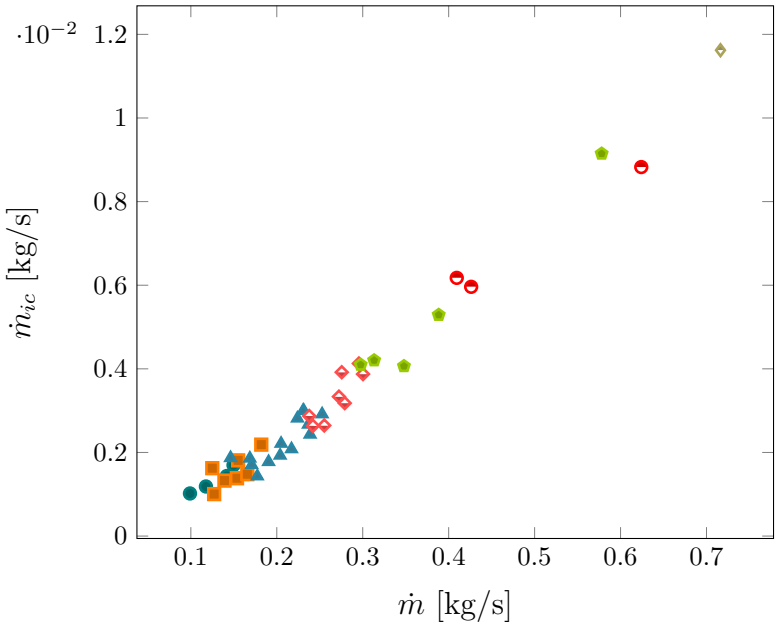


Figure 6.25 Relationship between intercooler channel mass flow rate at absolute maximum net power output (primary) and system mass flow rate; Case II.

6.2.3 Receivers

Rectangular section absorber tubes

Figures 6.26 - 6.28 show the change in total heat intercepted by the receivers with respect to mass flow rate. Downward concave trends are exhibited by the curves for each case; the curve for Case I being near-symmetric, and the curves for cases II and III showing truncated trends. Identical trends are observed for the receiver aperture diameter, shown in Fig. 6.29. A discussion of these trends follows, considering the constraints on the receiver geometry and surface temperatures, and distribution of entropy generation across the system components as described in the preceding discussion of the optimal heat exchanger geometry and operating conditions.

As previously mentioned in reference to Table 6.4, the recuperator inlet and outlet temperature, and consequently the receiver inlet temperature, rise as the system mass flow rate increases. Entropy generation in the recuperator, however, declines relatively rapidly as the difference between the average hot and cold stream temperatures is reduced. Conversely, entropy generation in the receiver increases, suggesting that an increase in the rate of entropy generation in the receiver, caused by the substantial difference in temperature across the absorber tube, is justified by the reduction in the entropy generation in the recuperator, and the increase in the net power output. The increased entropy generation in the receiver is effected by a widening of the receiver aperture and the resultant increase in the total heat intercepted. Note that since the the rate of heat loss through the receiver aperture also increases, the rate of heat absorption remains relatively constant.

The absolute receiver outlet temperature increases with increasing mass flow rate. Relative to the receiver inlet temperature, however, the increase in the outlet temperature diminishes, according to the heat transfer relation $\dot{Q} = \dot{m} C_p \Delta T$, where $\dot{Q}$ is constant. Beyond a certain point in the system operating range, only a small temperature increase is effected across the receiver, on account of $\dot{m}$ being very high in the aforementioned relation, causing the rate of heat loss to decrease. At these operating points, entropy generation in the receiver and turbomachinery, and the temperature at the recuperator inlet are very high. Consequently, the receiver aperture constricts to reduce the heat intercepted and limit total entropy production.

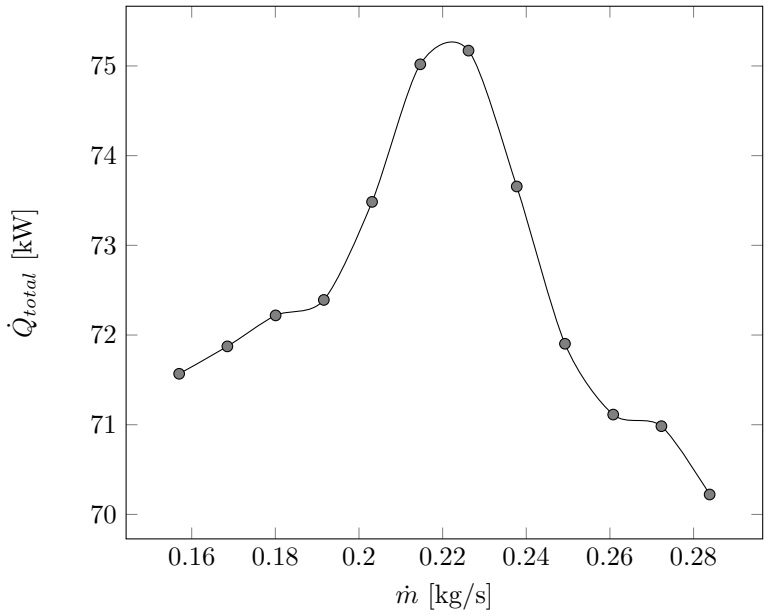


Figure 6.26 Change in heat intercepted by the receivers with mass flow rate; Case I.

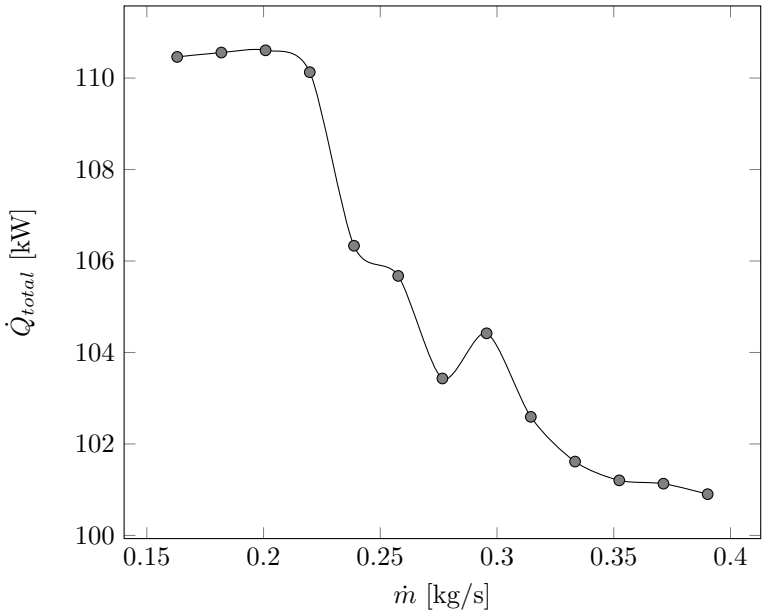


Figure 6.27 Change in heat intercepted by the receivers with mass flow rate; Case II.

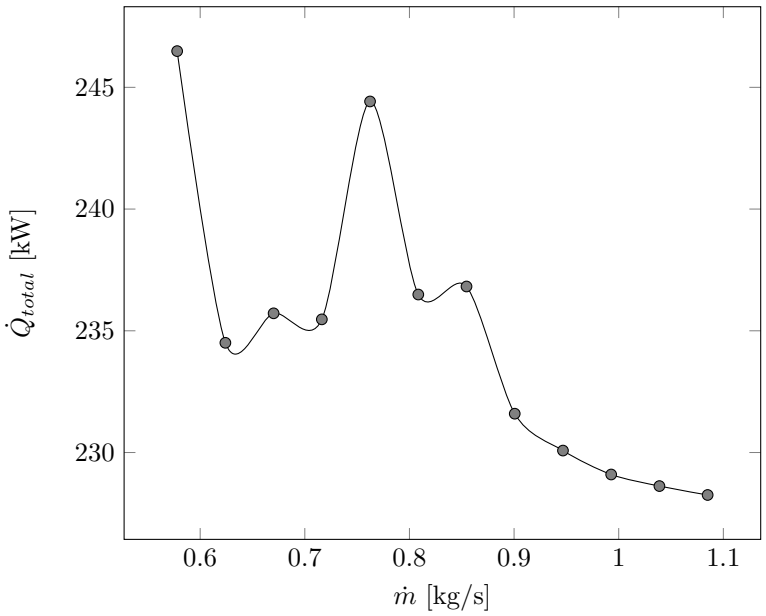


Figure 6.28 Change in heat intercepted by the receivers with mass flow rate; Case III.

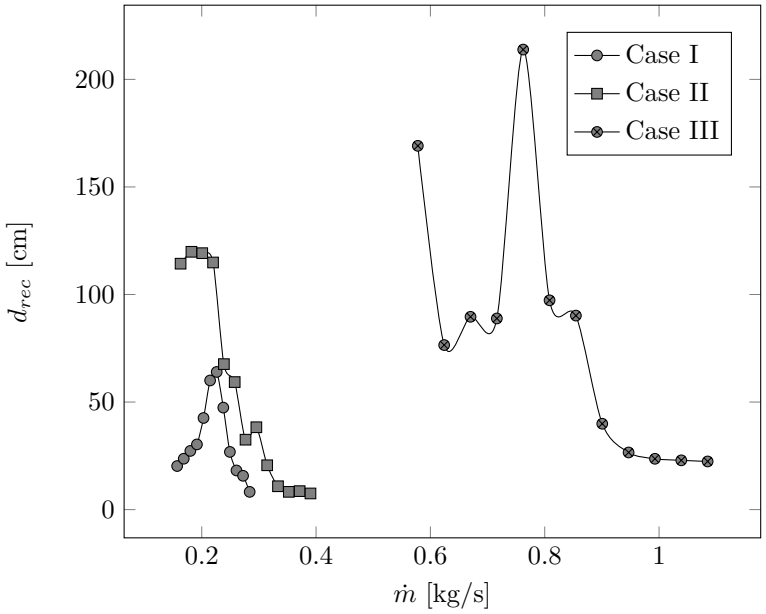


Figure 6.29 Variation in receiver aperture diameter with mass flow rate.

The optimal receiver aspect ratios are shown in Fig. 6.30, with respect to mass flow rate through the systems. Marked similarities are noted between the trends shown in Fig. 6.30 and those exhibited by the receiver aperture diameter and total heat intercepted, interpreted below by considering the relationships between the receiver geometry and operating conditions, and between the geometric parameters themselves, as described by Eq. 3.5.

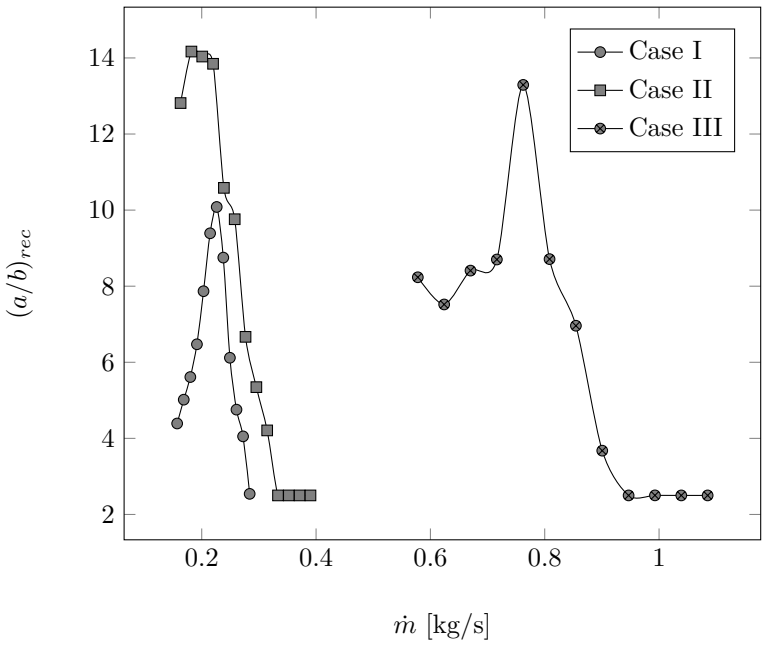


Figure 6.30 Change in optimal receiver aspect ratio with mass flow rate.

The receiver hydraulic diameter, shown in Figs. 6.31 - 6.33, decreases with increasing mass flow rate to ameliorate the increase in heat loss caused by the initial widening of the receiver aperture (Eq. 3.101). As the aperture constricts at higher mass flow rates, and the rate of heat loss reduces, the hydraulic diameter increases and a reduction in entropy generation due to pressure loss is noted, as shown in Fig. 6.10.

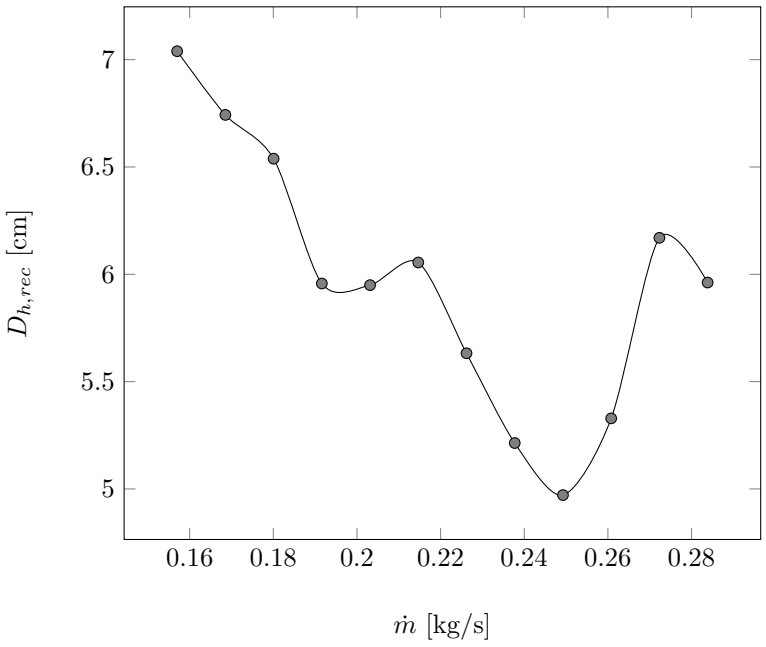


Figure 6.31 Change in optimal receiver hydraulic diameter with mass flow rate; Case I.

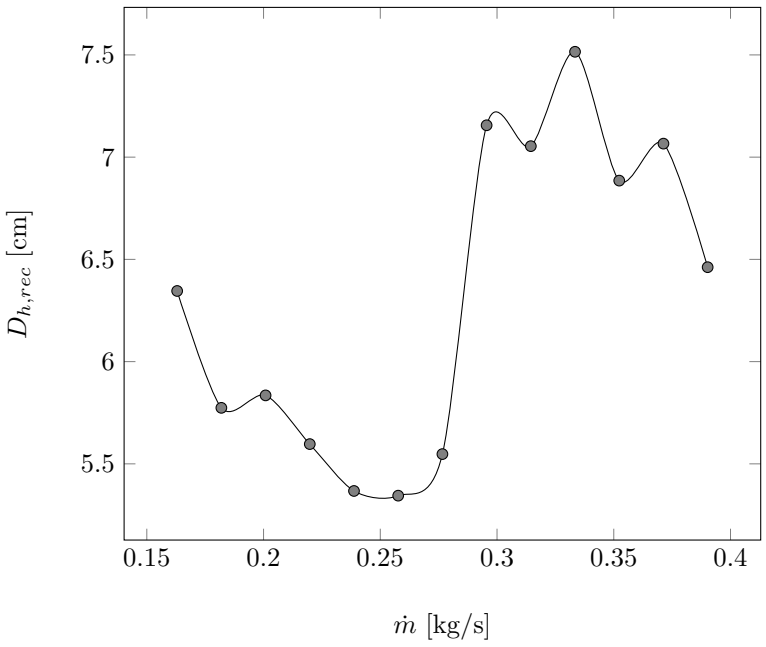


Figure 6.32 Change in optimal receiver hydraulic diameter with mass flow rate; Case II.

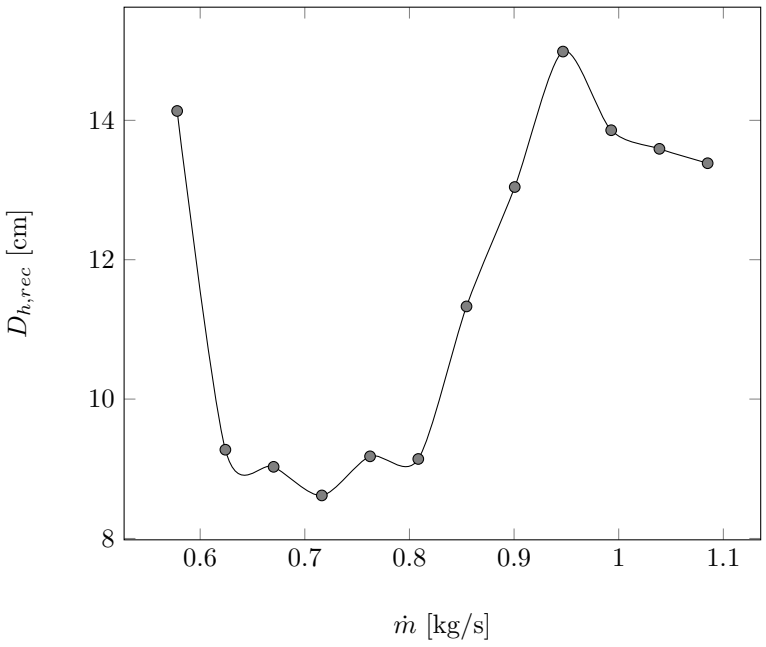


Figure 6.33 Change in optimal receiver hydraulic diameter with mass flow rate; Case III.

As with the aspect ratio, similarity is noted between the trends in the length of the absorber tubes (Fig. 6.34) and those exhibited by the receiver aperture diameter and rate of heat intercepted.

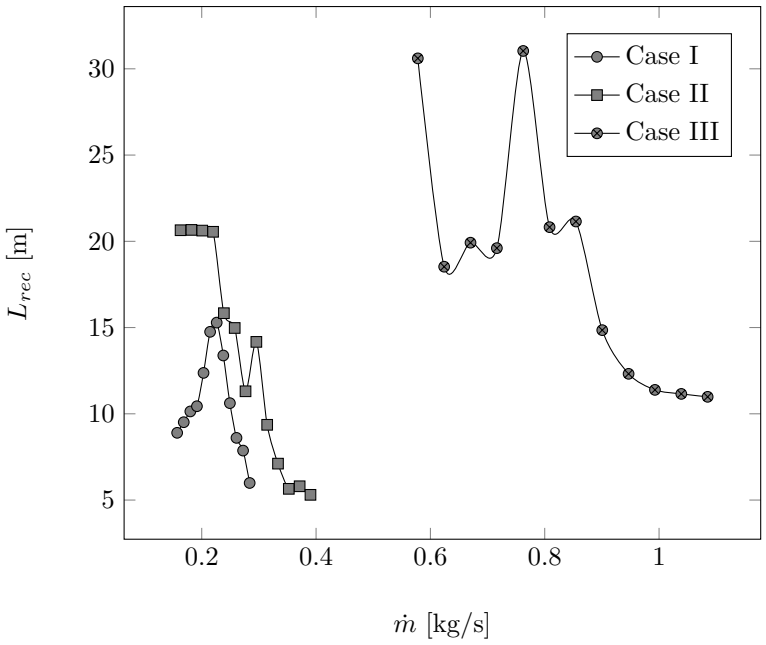


Figure 6.34 Variation in optimal receiver length with mass flow rate.

Equation 3.101 indicates that the marked increase in the receiver length and aspect ratio, and to a lesser extent, the decrease in the hydraulic diameter, facilitates a reduction in the surface temperature of the absorber tube, by increasing the total surface area of the tube. This is corroborated by the data, which indicate a rapid rise in the receiver surface temperature to the maximum constraint of 1 200 K, with lower surface temperature values recorded for larger values of the absorber length and aspect ratio.

Circular section absorber tubes

The variation in the heat intercepted and corresponding receiver aperture diameter are shown with respect to mass flow rate in Figs. 6.35 and 6.36 for receivers comprising absorber tubes of circular cross-section. While the curves shown pertain to Case I, similar trends are noted in the curves for cases II and III. Compared to the analogous curves for receivers comprising rectangular section tubes, the trends shown are less smooth, oscillating between high and low values, although the absolute values of both parameters are, on average, comparable between the geometry types.

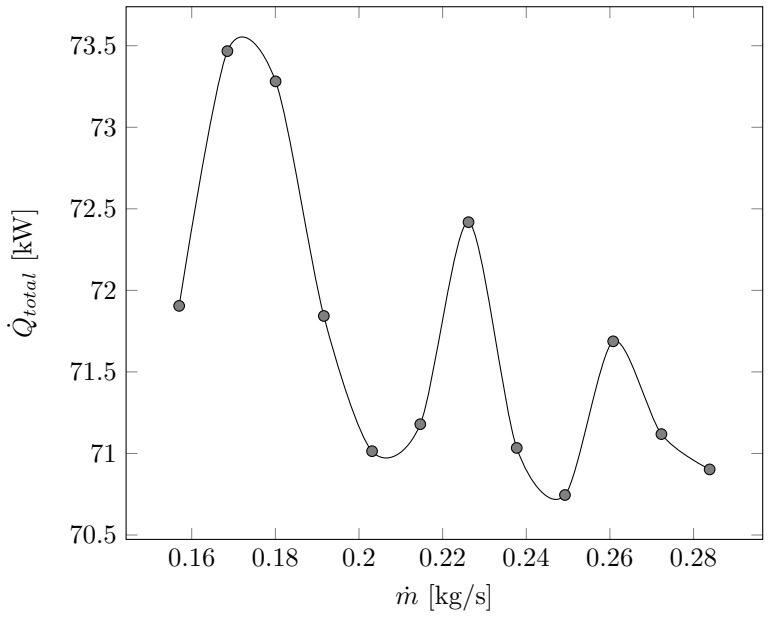


Figure 6.35 Variation in heat intercepted by the receivers with mass flow rate; Case I.

Fluctuation is also observed in the trends exhibited by the geometric parameters. The hydraulic diameter, shown in Fig. 6.37 should exhibit a decreasing trend with increasing mass flow rate in order to promote convective heat transfer, as previously stated. However, with the aspect ratio of the absorber tube fixed at $a/b = 1$,

variation in the receiver aperture diameter is determined primarily by change in the absorber tube length, according to Eq. 3.5. Similarly, adjustments to the surface area of the absorber tube to ensure the surface temperature is maintained below the maximum constraint of 1 200 K can be effected only through changes to the length and hydraulic diameter of the tube.

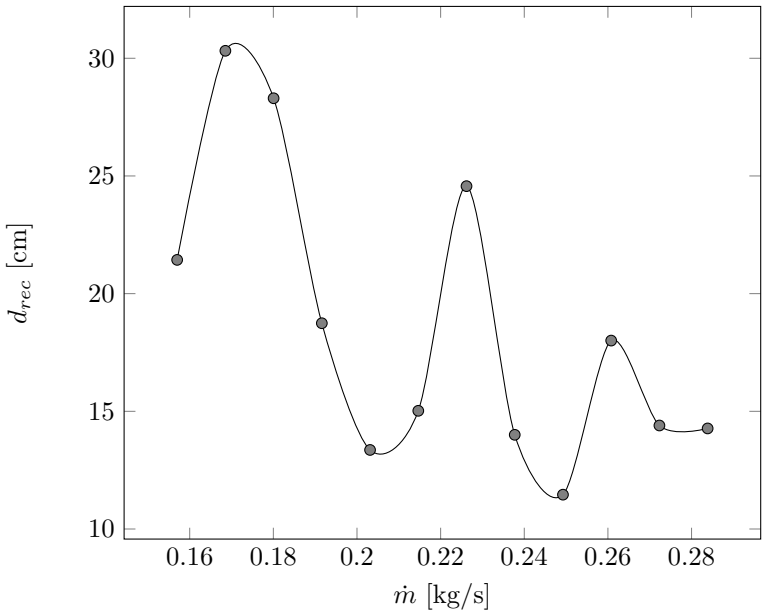


Figure 6.36 Variation in receiver aperture diameter with mass flow rate; Case I.

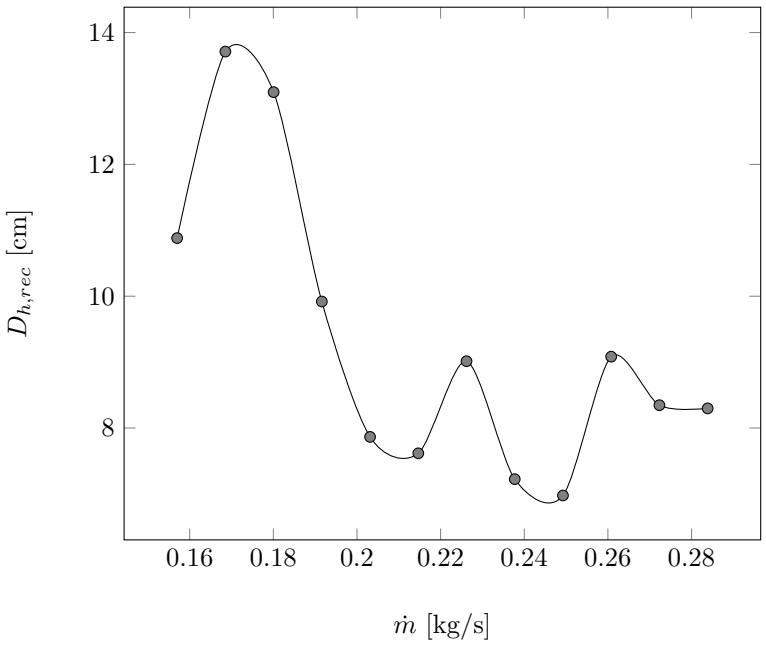


Figure 6.37 Relationship between optimal receiver hydraulic diameter and mass flow rate; Case I.

Recall from section 3.5 that absorber tubes of circular cross-section are constrained to a minimum of two revolutions around the aperture. As a result, circular section absorber tubes are inherently longer than tubes of rectangular section, which, due to the minimum constraint of 2.5 on the tube aspect ratio are generally wider, and, according to the results of the optimisation procedure, typically comprise only one revolution irrespective of mass flow rate. The data show that heat loss is consequently greater for circular section absorber tubes. Accordingly, the associated aperture diameters are smaller, constricting at lower mass flow rates than for rectangular section tubes in order to curb heat loss through the aperture.

The downward trends exhibited by the curves describing the absorber tube length and hydraulic diameter correspond to the decrease in the aperture diameter, and the sudden fluctuations in the geometric variables are effected in order to regulate the temperature of the tube surface at higher mass flow rates.

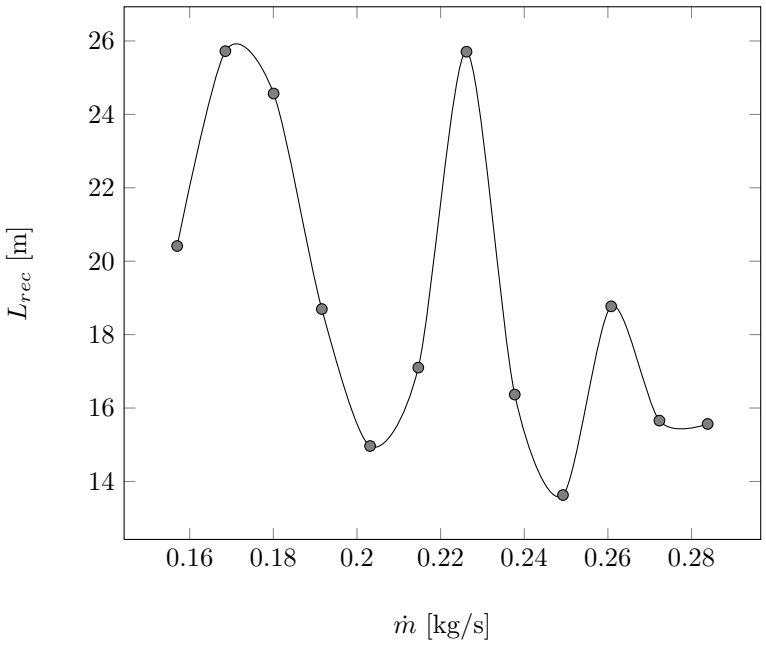


Figure 6.38 Relationship between optimal receiver length and mass flow rate; Case I.

Geometric optima that exhibit significant fluctuation may be suboptimal at adjacent operating points. Given the heavy fluctuation in the optima for absorber tubes of circular cross-section, the associated differences could be greater than for rectangular cross-section absorber tubes, for which the variations are more gradual. Accordingly, systems with receivers comprising circular section absorber tubes could face significant transient and off-design performance challenges. These findings suggest that the geometric constraints required to maintain the receiver shape offer greater design flexibility for rectangular section absorber tubes.

6.2.4 Reheaters

Figure 6.39 shows the variation in the optimal reheater aperture diameter and total heat intercepted with system mass flow rate. Note that the curves shown pertain to rectangular section absorber tubes. As shown in Table 6.4 the inlet and outlet temperatures are higher for the reheaters than for the receivers. The aperture diameter values are consequently smaller for the reheaters than for the receivers, and, like the values for receivers comprising circular absorber tubes, decrease with increasing mass flow rate, limiting the heat intercepted, associated entropy generation due to temperature difference, and heat loss through the aperture.

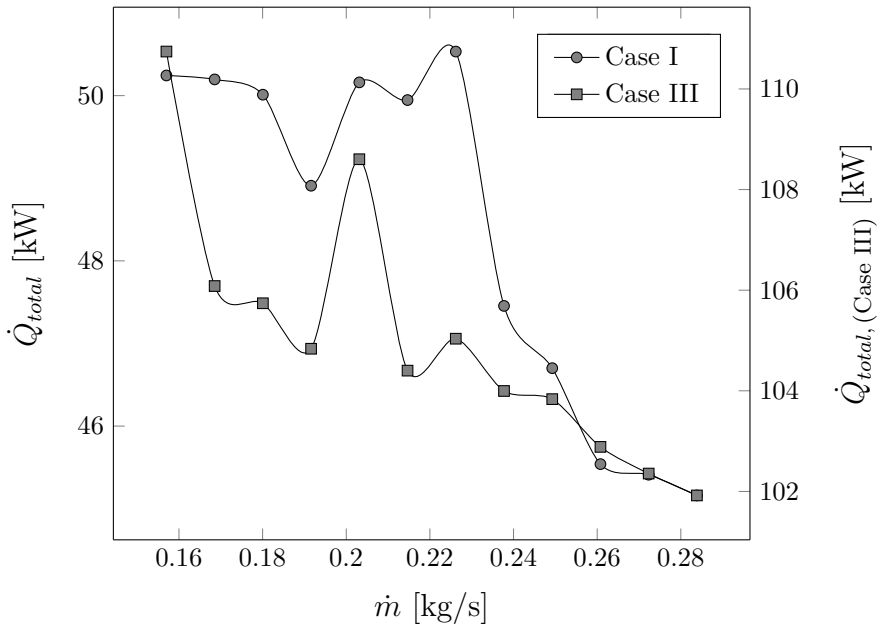


Figure 6.39 Change in heat intercepted by the reheaters with mass flow rate.

The trends exhibited by the associated geometric parameters are shown in Figs. 6.40 - 6.42. Compared to the receivers, the reheater length and aspect ratio values are smaller, since the aperture diameters are smaller. Conversely, the hydraulic diameter values are larger, due to greater frictional pressure losses caused by the higher ratio of viscous to inertial forces, i.e. lower Reynolds numbers, at the comparatively high reheater temperatures.

As previously noted for the receivers, the curves generated for reheaters comprising circular cross-section absorber tubes exhibit greater fluctuation than the analogous curves pertaining to rectangular section tubes, with smaller values noted for the aperture diameter and heat intercepted, and higher values for tube length.

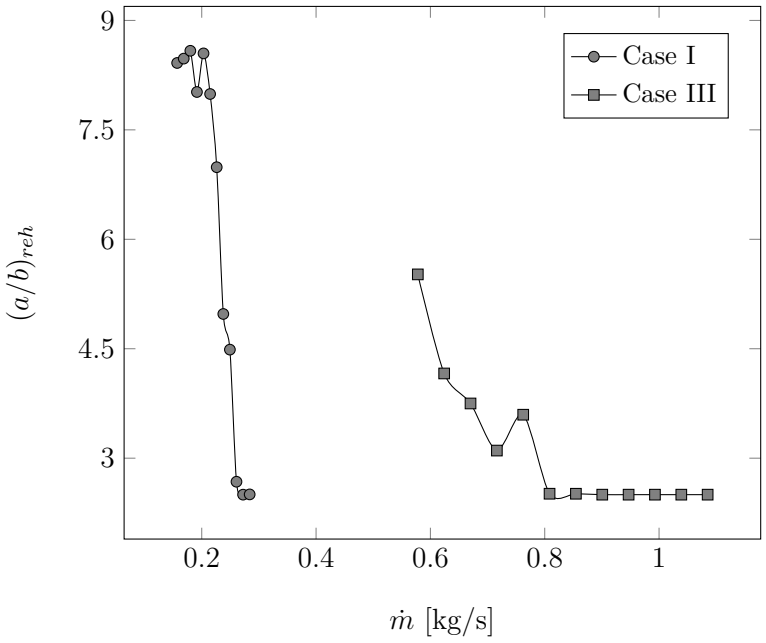


Figure 6.40 Relationship between optimal reheater aspect ratio and mass flow rate.

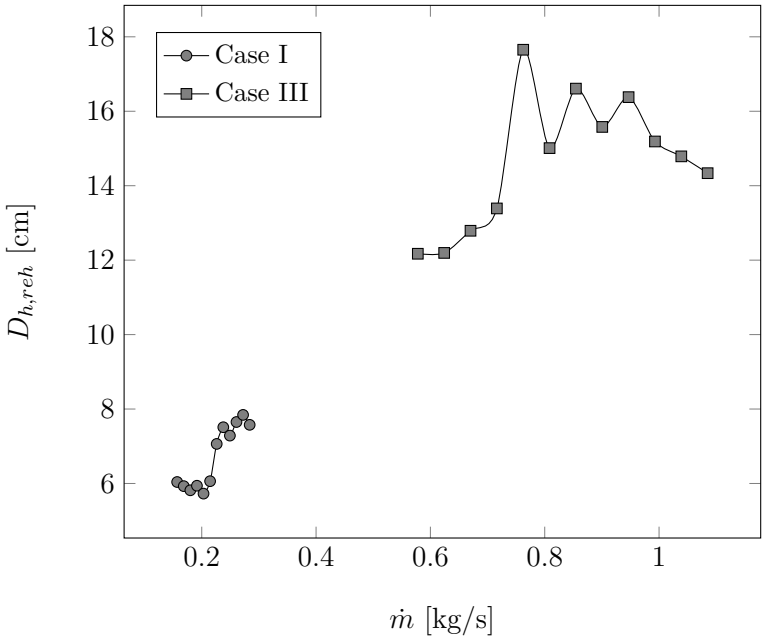


Figure 6.41 Relationship between optimal reheater hydraulic diameter and mass flow rate.

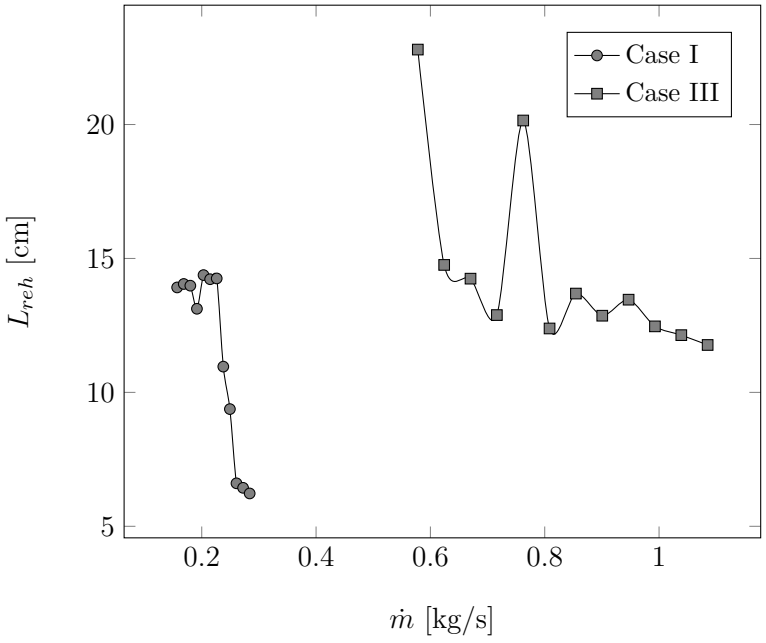


Figure 6.42 Relationship between optimal reheater length and mass flow rate.

Chapter 7

Validation

A validation of the foregoing results is presented in this chapter. The Flownex simulation environment is introduced, and simulation of the STBC variants according to the optimal geometry returned by the numerical optimisation in MATLAB, is described. Finally, the temperature and pressure field values determined using Flownex SE are compared with the results of the optimisation in MATLAB, to assess the reproducibility of the latter.

7.1 Flownex simulation environment

Flownex SE is a systems computational fluid dynamics code with an integrated graphical user interface, used for the design, simulation and optimisation of thermal fluid systems, as well as integrated simulation with other packages such as Mathcad and Engineering Equation Solver (M-Tech Industrial, 2015). Components are represented in Flownex as *elements* on a project canvas, each possessing one or more *nodes* depending on their intended function(s). Basic elements can be compounded into sub-assemblies, and custom components modelled using the scripting features.

Thermal-fluid networks are constructed in Flownex by establishing visual links between the nodes of various elements. The networks are discretised into conceptual control volumes, each then subjected to the application and implicit solution of a set of conservation equations, enabling transient, or steady state simulation (M-Tech Industrial, 2015). Outputs from the simulation may be charted, displayed on the project canvas, or exported in tabular format.

7.2 Simulation

The arrangement of the components constituting the Flownex network for Case III is shown in Fig. 7.1. Cases I and II are modelled similarly. The initialisation of the Flownex component input property pages using the optimal geometry returned by the optimisation in MATLAB is conducted as follows. The receiver and reheater are modelled using the pipe element, configured using the properties of commercial copper from the internal materials library. The hydraulic diameter, length, wall thickness, and net rate of heat absorption are specified as inputs. A composite heat transfer and recuperator element are used to model the intercooler and recuperator, respectively, with the cross-sectional area, plate thickness and material, and heat transfer surface area specified accordingly. In addition, the flow direction, number of flow channels, and overall heat transfer coefficient are specified for the recuperator, the plate capacitance and combined radiative and convective heat transfer coefficient for the intercooler, and the respective heat transfer and heat rejection rates determined upon steady-state solution of the network, using the $e-NTU$ method. Generic compressor and turbine elements are used, and the mass flow, pressure ratio, speed and efficiency scaling factors, as well as the shaft speeds, adjusted by trial and error to set the mass flow rate through the network and match the temperature and pressure ratios across the turbomachines returned by the optimisation in MATLAB. The ducts are modelled using the pipe element, with the length and hydraulic diameter of each set conservatively to 1 m and 0.5 m, respectively. The fluid properties of air are selected, and boundary conditions are applied at the peripheral nodes, as shown in Fig. 7.1.

7.3 Results

The nodal temperature and pressure values are shown in Tables 7.1 and 7.2, for cases I and II, respectively. Data for Case III are represented graphically, in Figs. 7.2 and 7.3. The values are found to differ by less than 4 percent, indicating good agreement between the results of the numerical optimisation in MATLAB and the simulation in Flownex SE, the only exception being the temperature at the recuperator hot stream outlet. Since heat loss to the surroundings from the outermost recuperator plate surfaces is taken into account in the numerical NTU calculation only, and Flownex SE employs a simplified $e-NTU$ method, the differences at the recuperator and intercooler are anticipated to be greater than at other nodes.

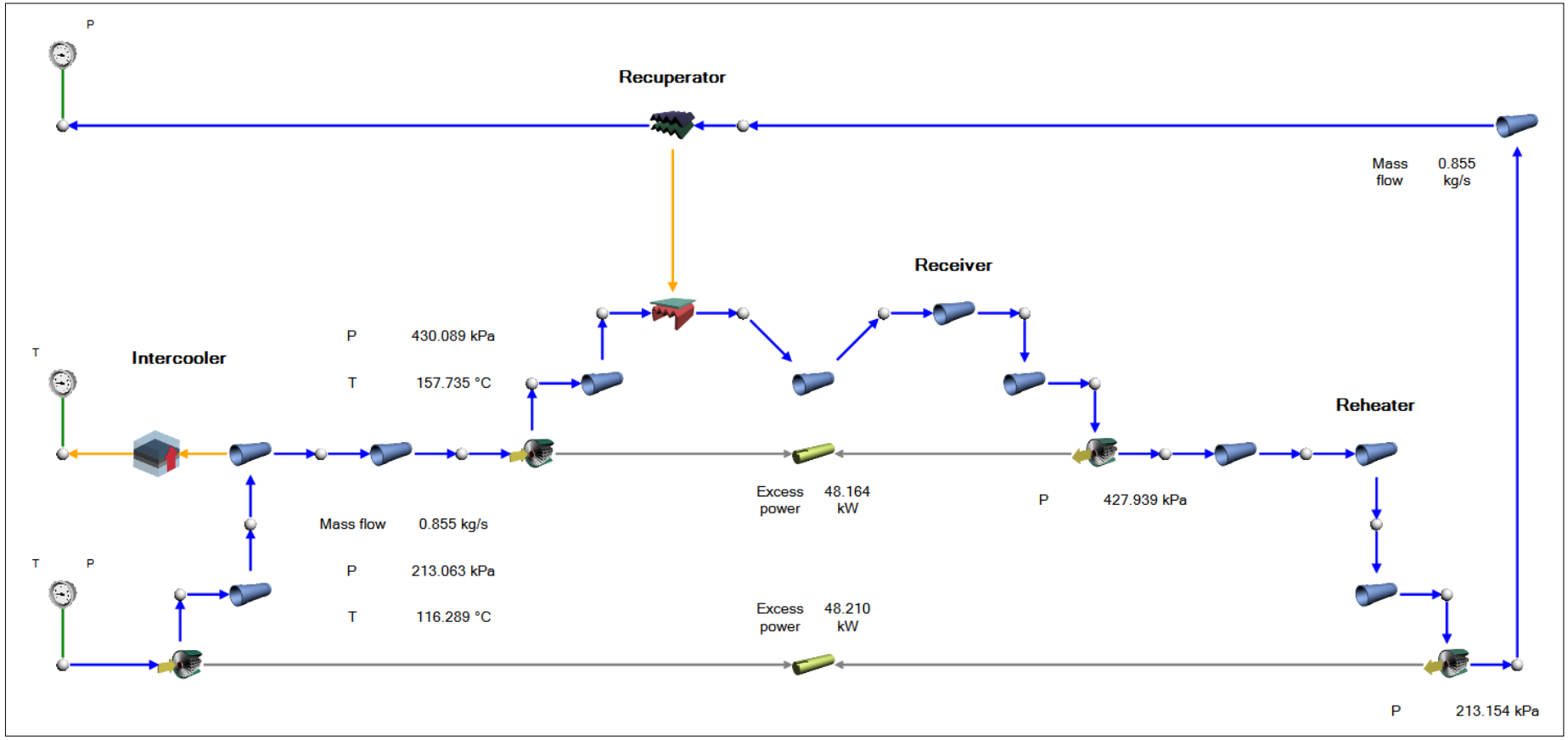


Figure 7.1 Flownex network for Case III - recuperation, intercooling and reheating.

Additional uncertainty is introduced through the scaling of the internal turbomachine performance charts, particularly the turbine efficiency, in the absence of speed-specific turbine data for the Garrett turbocharger range. Development of comprehensive turbomachine performance charts would aid detailed simulation of STBC variants, including investigation of transient operation, and is consequently recommended for future work.

Table 7.1 Nodal temperatures and pressures determined using MATLAB and Flownex SE; Case I, configuration (16; 16; 21) - rectangular section absorber tubes.

	Temperature (K)			Pressure (kPa)		
Node	MATLAB	FLOWNEX	Difference (%)	MATLAB	FLOWNEX	Difference (%)
1	300	299.85	0.05	101.325	101.325	0.00
2	383.1697	383.387	0.06	202.65	200.753	0.94
3	381.1697	383.387	0.58	201.8394	200.535	0.65
4	486.8424	485.86	0.20	403.6788	394.521	2.27
5	484.8424	485.86	0.21	402.0641	394.38	1.91
6	894.9569	880.127	1.66	399.4832	394.271	1.30
7	892.9569	880.128	1.44	397.8853	394.007	0.97
8	1085.41	1072.387	1.20	394.5691	393.005	0.40
9	1083.41	1072.388	1.02	392.9908	392.68	0.08
10	938.1831	936.363	0.19	204.9957	203.307	0.82
11	936.1831	936.363	0.02	204.1757	202.75	0.70
12	1127.429	1126.195	0.11	200.8583	200.817	0.02
13	1125.429	1126.196	0.07	200.0549	200.127	0.04
14	974.5696	996.34	2.23	104.3546	103.108	1.19
15	972.5696	996.34	2.44	104.2503	101.796	2.35
16	569.3126	611.69	7.44	101.325	101.437	0.11

Table 7.2 Nodal temperatures and pressures determined using MATLAB and Flownex SE; Case II, configuration (16; 21) - rectangular section absorber tube.

	Temperature (K)			Pressure (kPa)		
Node	MATLAB	FLOWNEX	Difference (%)	MATLAB	FLOWNEX	Difference (%)
1	300	299.85	0.05	101.325	101.325	0.00
2	354.5775	355.004	0.12	162.12	160.714	0.87
3	352.5775	355.0035	0.69	161.4715	160.56	0.56
4	323.0017	323.4185	0.13	161.4099	160.56	0.53
5	321.0017	323.4183	0.75	160.7643	160.421	0.21
6	379.3999	380.625	0.32	257.2228	258.267	0.41
7	377.3999	380.625	0.85	256.1939	258.165	0.77
8	842.8006	826.284	1.96	252.1279	257.94	2.31
9	840.3512	826.284	1.67	251.1194	257.707	2.62
10	1091.875	1074.928	1.55	247.9595	244.499	1.40
11	1089.875	1074.928	1.37	246.9677	246.914	0.02
12	992.3173	989.829	0.25	162.1237	162.523	0.25
13	990.3173	989.829	0.05	161.4752	162.066	0.37
14	901.6714	911.597	1.10	106.0015	102.593	3.22
15	899.6714	911.598	1.33	105.8956	101.9	3.77
16	440.3854	473.744	7.57	101.325	101.42	0.09

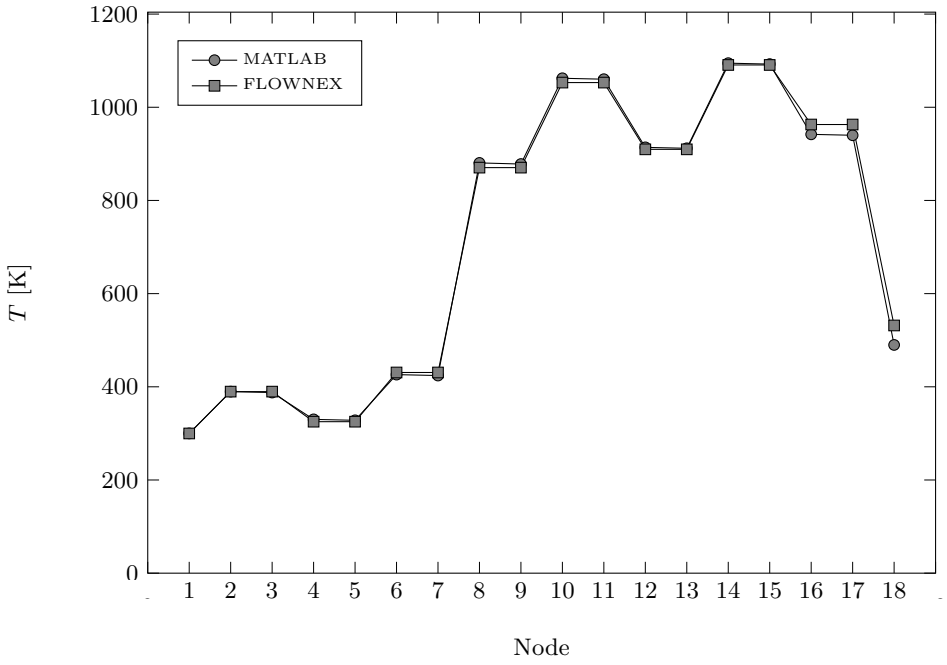


Figure 7.2 Nodal temperatures determined using MATLAB and Flownex SE; Case III, configuration: (16; 16; 21) - rectangular section absorber tube.

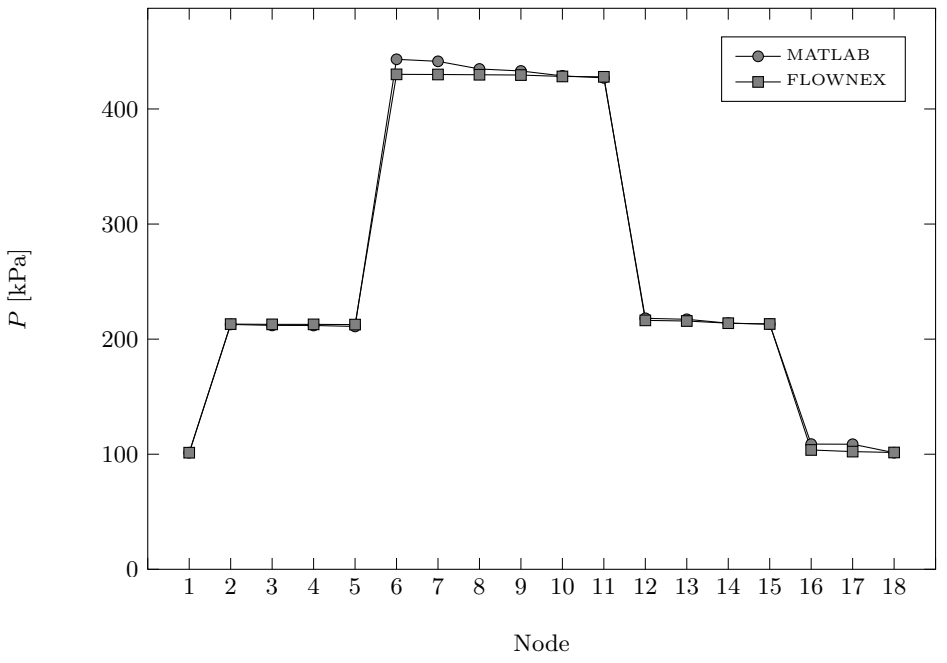


Figure 7.3 Nodal pressures determined using MATLAB and Flownex SE; Case III, configuration: (16; 16; 21) - rectangular section absorber tubes.

Chapter 8

Conclusions and recommendations

An investigation into the effects of reheating and intercooling on the maximum net power output that can be developed by an open air STBC is presented in the foregoing text. Three cases, based on a simple STBC comprising a modified cavity receiver, a counter-flow plate-type recuperator, and a pair of proprietary automotive turbochargers configured to operate as micro-turbines, are considered. The basic STBC is modified to incorporate additional modified cavity receivers and cross-flow plate heat exchangers which constitute reheaters and intercoolers, respectively, to construct the reheated, intercooled and combined cases.

Heat transfer and fluid flow through the system components are modelled such that the temperatures and pressures, and thus the entropy generation rate in each component, are expressed in terms of a set of geometric parameters, specifically the aspect ratio, hydraulic diameter and length of each component. An objective function, in terms of the geometric parameters, is then constructed for each case by performing an exergy balance for the system and summing the entropy generation rates for the individual components. The objective functions are subsequently maximised using the dynamic trajectory method. In accordance with the Gouy-Stodola Theorem, the cumulative rate of entropy generation is thereby minimised, such that the maximum net power output is maximised, thus yielding the optimal values for the geometric parameters, for each case considered. Multiple configurations of each case are assessed by considering various micro-turbine models, concentrator diameters, and profiles for the receiver and reheater absorber tubes.

The results indicate that the intercooled STBC produces the greatest specific net power output, and has the highest thermal efficiency per unit concentrator area,

followed by the combined and reheated case, respectively. The intercooled and combined cases are also shown to possess higher cumulative entropy generation rates than the reheated STBC. Entropy generation in the components is found to be primarily caused by temperature differences across the components, with entropy generation caused by frictional pressure drop found to account for a relatively small proportion of total entropy generation. The receivers are consequently found to be the primary site of entropy generation in all cases, followed by the reheaters and recuperators, respectively. Entropy generation in the turbomachinery and intercoolers is found to increase with system mass flow rate, with the latter constituting a relatively small proportion of the cumulative entropy generation rate.

Changes in the recuperator and intercooler geometric parameters are found to occur such that the total surface area available for heat transfer is maximised, causing the heat transfer effectiveness to increase with system mass flow rate. Entropy generation in the intercooler is found to increase to a maximum when the sum of the entropy generation rates for the other components is at a minimum. The heat exchanger parameters are unaffected by the receiver and reheater absorber tube cross-section. The receiver and reheater geometric parameters are found to change with system mass flow rate such that the absorber tube surface temperatures are maintained below the maximum constraint. Trends in the data obtained for circular section absorber tubes are found to be less smooth than the trends in the data obtained for absorber tubes of rectangular section. Since significant differences between the optimal and actual geometric parameters at a given operating point could occur as a result of the fluctuations, and affect system performance during transient or off-design operation, it is concluded that the geometric constraints required to maintain the receiver shape offer greater design flexibility for rectangular section absorber tubes than for absorber tubes of circular section.

Based on the findings of the investigation, the following conclusions are drawn. The temperature differences across the heat exchangers, and thus the component entropy generation rates, change as the compressor and turbine outlet temperatures increase with system mass flow rate and compressor pressure ratio. The entropy generation rates must in turn be distributed, during the optimisation procedure, such that the cumulative rate is less than the power output, and all of the constraints are met. Regarding the component geometry, the relatively high heat exchanger plate counts are concluded to be less than desirable in terms of practicality. While the results enable the cases to be ranked according to highest net power output, efficiency and entropy generation, relatively few empirical relations or precise guidelines for the

optimal sizing of the components are apparent. This observation emphasises the importance of integrated system optimisation as opposed to separate optimisations of the individual components, to ensure compatibility between the components in the specific system configuration.

The validation of these results in the Flownex simulation environment corroborates these conclusions. Based on the findings of the investigation and the outcomes of the validation procedure, it is recommended that alternative heat exchanger designs and flow arrangements be investigated in future. For instance, regeneration - involving intermittent thermal storage and alternating displacement of the hot and cold fluid streams through a common volume - may be considered instead of recuperation, as a means of reducing the recuperator size. Moreover, consideration of the compact heat exchanger designs reviewed by Li et al. (2011) as alternatives to the parallel plate heat exchanger design applied for the modelling of the recuperator and intercooler in the present work is recommended. The development of detailed models of the system components in the Flownex simulation environment is also recommended. Such models would enable more accurate simulation of STBC systems, and establish a framework that enables the simulation of transient operation, indirect heat input, and various thermal storage technologies. Detailed models would also facilitate multi-objective optimisation, which would allow for assessment of system costs.

The present work offers insight into the distribution of entropy generation in STBCs with reheating and intercooling, and indications of the size and attainable power output of such systems. Further investigation and experimental study based on the foregoing findings and recommendations is anticipated to provide clarity regarding the performance, cost and technical readiness of STBCs with reheating and intercooling, and advance the development of STBCs for small scale power generation.

Bibliography

AORA. Tulip DST Unit - Features Summary, 2015. URL `http://aora-solar.com/product/`.

Bar-Cohen, A. and Rohsenow, W. Thermally optimum spacing of vertical, natural convection cooled, parallel plates. *Journal of heat transfer*, 106(1):116–123, 1984.

Bejan, A. *Advanced Engineering Thermodynamics*. Wiley, New York, 1997.

Bejan, A., Lorente, S., and Kang, D.-H. Constructal design of regenerators. *International Journal of Energy Research*, 37:1509–1518, 2013.

Bertocchi, R., Karni, J., and Kribus, A. Experimental evaluation of a non-isothermal high temperature solar particle receiver. *Energy*, 29:687–700, 2004. doi: 10.1016/j.energy.2003.07.001.

Brayton Energy LLC. Brayton Power Conversion System. Technical report, Brayton Energy LLC, Hampton, 2011.

Çengel, Y. A. *Heat Transfer: A Practical Approach*. McGraw-Hill, New York, 2nd edition, 2004.

Çengel, Y. A. and Boles, M. A. *Thermodynamics: An Engineering Approach*. McGraw-Hill, New York, 7 edition, 2011.

Crolla, D. *Encyclopedia of Automotive Engineering*, volume 5. Wiley, 2015. ISBN 9780470974025.

CSP Today. SEGS experience suggests that CSP will age gracefully, 2013. URL `http://social.csptoday.com/markets/segs-experience-suggests-csp-will-age-gracefully`.

Cumpsty, N. *Jet Propulsion: A Simple Guide to the Aerodynamic and Thermodynamic Design and Performance of Jet Engines*. Cambridge engine technology series. Cambridge University Press, 2003. ISBN 9780521541442.

Department of Energy. State of Renewable Energy in South Africa. Technical report, DOE, 2015.

Duffie, J. A. and Beckman, W. A. *of Thermal Processes Solar Engineering*. Wiley, 1976. ISBN 9780470873663.

Farahat, S., Sarhaddi, F., and Ajam, H. Exergetic optimization of flat plate solar collectors. *Renewable Energy*, 34:1169–1174, 2009.

Fluri, T. P. The potential of concentrating solar power in South Africa. *Energy Policy*, 37(12):5075–5080, 2013. doi: 10.1016/j.enpol.2009.07.017.

Gauche, P. TIA Helio100 technology development project Status and way forward, 2015.

Goodarzi, M., Kiasat, M., and Khalilidehkordi, E. Performance analysis of a modified regenerative Brayton and inverse Brayton cycle. *Energy*, 72:35–43, 2014. doi: 10.1016/j.energy.2014.04.072.

Goswami, D. Y. Solar thermal power: Status of technologies and opportunities for research. In *Proceedings of ASME Heat and Mass Transfer Conference, December*, Surathkal, India, 1995.

Goswami, D. Y. *Principles of Solar Engineering*. CRC Press, 3 edition, 2015. ISBN 9781466563797.

Guo, J., Huai, X., Li, X., Cai, J., and Wang, Y. Multi-objective optimization of heat exchanger based on entransy dissipation theory in an irreversible Brayton cycle system. *Energy*, 63:95–102, 2013.

Gupta, M. K., Kaushik, S. C., Ranjan, K. R., Panwar, N. L., Reddy, V. S., and Tyagi, S. K. Thermodynamic performance evaluation of solar and other thermal power generation systems: A review. *Renewable and Sustainable Energy Reviews*, 50:567–582, 2015. doi: 10.1016/j.rser.2015.05.034.

Heller, P., Pfänder, M., Denk, T., Tellez, F., Valverde, A., Fernandez, J., and Ring, A. Test and evaluation of a solar gas turbine system. *Solar Energy*, 80(10): 1225–1230, 2006. doi: 10.1016/j.solener.2005.04.020.

Hesselgreaves, J. E. Rationalisation of second law analysis of heat exchangers. *International Journal of Heat and Mass Transfer*, 43:4189–4204, 2000.

International Renewable Energy Agency . Renewable energy technologies: cost analysis series - Concentrating Solar Power. Technical Report 2, IRENA, 2012.

Invernizzi, C. *Closed Power Cycles: Thermodynamic Fundamentals and Applications*. Lecture Notes in Energy. Springer London, 2013. ISBN 9781447151401.

Jansen, E., Bello-Ochende, T., and Meyer, J. P. Integrated solar thermal Brayton cycles with either one or two regenerative heat exchangers for maximum power output. *Energy*, 86:737–748, 2015. doi: 10.1016/j.energy.2015.04.080.

Jubeh, N. M. Exergy Analysis and Second Law Efficiency of a Regenerative Brayton Cycle with Isothermal Heat Addition. *Entropy*, 7(3):172–187, 2005.

Kalina, A. I. Combined-cycle system with novel bottoming cycle. *Journal of engineering for gas turbines and power*, 106(4):737–742, 1984.

Kalogirou, S. A. Solar thermal collectors and applications. *Progress in Energy and Combustion Science*, 30:231–295, 2004. doi: 10.1016/j.pecs.2004.02.001.

Kays, W. and London, A. *Compact Heat Exchangers*. Krieger Publishing Company, 1984. ISBN 9781575240602.

Kolb, G. J., Jones, S. A., Donnelly, M. W., Gorman, D., Thomas, R., Davenport, R., and Lumia, R. Heliostat Cost Reduction Study. Technical Report November, Sandia National Laboratories, 2016.

Le Roux, W. G. Maximum net power output from an integrated design of a small-scale open and direct solar Thermal Brayton Cycle. Master's thesis, University of Pretoria, 2011.

Le Roux, W. G., Bello-Ochende, T., and Meyer, J. P. Thermodynamic optimisation of the integrated design of a small-scale solar thermal Brayton cycle. *International Journal of Energy Research*, 36:1088–1104, 2011. doi: 10.1002/er.

Le Roux, W. G., Bello-Ochende, T., and Meyer, J. P. A review on the thermodynamic optimisation and modelling of the solar thermal Brayton cycle. *Renewable and Sustainable Energy Reviews*, 28:677–690, 2013.

Le Roux, W. G., Bello-Ochende, T., and Meyer, J. P. The efficiency of an open-cavity tubular solar receiver for a small-scale solar thermal Brayton cycle. *Energy Conversion and Management*, 84:457–470, 2014. doi: 10.1016/j.enconman.2014.04.048.

Li, Q., Flamant, G., Yuan, X., Neveu, P., and Luo, L. Compact heat exchangers: A review and future applications for a new generation of high temperature solar receivers. *Renewable and Sustainable Energy Reviews*, 15(9):4855–4875, 2011. doi: 10.1016/j.rser.2011.07.066.

Li, Y., Liao, S., and Liu, G. Thermo-economic multi-objective optimization for a solar-dish Brayton system using NSGA-II and decision making. *International Journal of Electrical Power and Energy Systems*, 64:167–175, 2015. doi: 10.1016/j.ijepes.2014.07.027.

Lovegrove, K. and Pye, J. Fundamental principles of concentrating solar power (CSP) systems. In Lovegrove, K. and Stein, W., editors, *Concentrating Solar Power Technology*, Woodhead Publishing Series in Energy, chapter 2, pages 16–67. Woodhead Publishing, 2012. ISBN 978-1-84569-769-3. doi: http://dx.doi.org/10.1533/9780857096173.1.16.

M-Tech Industrial. *Flownex Library Theory Manual*. M-Tech Industrial, 2015.

Manjunath, K. and Kaushik, S. C. Second law thermodynamic study of heat exchangers: A review. *Renewable and Sustainable Energy Reviews*, 40:348–374, 2014. doi: 10.1016/j.rser.2014.07.186.

Mills, D. Advances in solar thermal electricity technology. *Solar Energy*, 76:19–31, 2004. doi: 10.1016/S0038-092X(03)00102-6.

Moran, M. J. Exergy analysis, costing, and assessment of environmental impact. In Bejan, A., Vadász, P., and Kröger, D. G., editors, *Energy and the Environment*, chapter 1, pages 1–10. Springer Netherlands, Dordrecht, 1999.

Murphy, L M (SERI) and Sallis D V (Dan-Ka Products) . Analytical Modeling and Structural Response of a Stretched-Membrane Reflective Module. Technical report, Solar Energy Research Institute, 1984.

National Renewable Energy Laboratory. Concentrating Solar Projects, 2016.

Ngo, L. C. Exergetic Analysis and Optimisation of a Parabolic Dish Collector for Low Power Application. In *3rd Stellenbosch University Centre for Renewable and Sustainable Energy Studies Student Symposium*, pages 1–15, 2012.

Ordóñez, J. C. and Bejan, A. Entropy generation minimization in parallel-plates counterflow heat exchangers. *International Journal of Energy Research*, 24(July 2016):843–864, 2000.

Reddy, K. S. and Kumar, N. S. Combined laminar natural convection and surface radiation heat transfer in a modified cavity receiver of solar parabolic dish. *International Journal of Thermal Sciences*, 47:1647–1657, 2008. doi: 10.1016/j.ijthermalsci.2007.12.001.

Reddy, K. S. and Kumar, N. S. An improved model for natural convection heat loss from modified cavity receiver of solar dish concentrator. *Solar Energy*, 83: 1884–1892, 2009. doi: 10.1016/j.solener.2009.07.001.

Rogdakis, E. and Antonopoulos, K. A high efficiency NH3/H2O absorption power cycle. *Heat Recovery Systems and CHP*, 11(4):263 – 275, 1991. doi: http://dx.doi.org/10.1016/0890-4332(91)90072-C.

Romero, M., Marcos, M. J., Blanco, M., and Baonza, F. Integration of Small Solar Tower Systems Into Distributed Power Islands. Technical report, Centre for Energy, Environment and Technology, Madrid, 1999.

Rovira, A., Muñoz, M., Sánchez, C., and Martínez-Val, J. M. Proposal and study of a balanced hybrid Rankine-Brayton cycle for low-to-moderate temperature solar power plants. *Energy*, 89:305–317, 2015. doi: 10.1016/j.energy.2015.05.128.

Sadatsakkak, S. A., Ahmadi, M. H., and Ahmadi, M. A. Thermodynamic and thermo-economic analysis and optimization of an irreversible regenerative closed Brayton cycle. *Energy Conversion and Management*, 94:124–129, 2015. doi: 10.1016/j.enconman.2015.01.040.

Sánchez-Orgaz, S., Pedemonte, M., Ezzatti, P., Curto-Risso, P. L., Medina, A., and Calvo Hernández, A. Multi-objective optimization of a multi-step solar-driven Brayton plant. *Energy Conversion and Management*, 99:346–358, 2015. doi: 10.1016/j.enconman.2015.04.077.

Sekulic, D. P., Sciubba, E., and Moran, M. J. Entransy: A misleading concept for the analysis and optimization of thermal systems. *Energy*, 80:251–253, 2015.

Sendhil Kumar, N. and Reddy, K. S. Comparison of receivers for solar dish collector system. *Energy Conversion and Management*, 49(4):812–819, 2008.

Shah, R. K. Compact Heat Exchangers for Microturbines. In Shah, R. K., Ishizuka, M., Rudy, T. M., and Wadekar, V. V., editors, *Proceedings of Fifth International Conference on Enhanced, Compact and Ultra-Compact Heat Exchangers: Science, Engineering and Technology*, pages 247 – 257, New Jersey, 2005. Engineering Conferences International.

Snyman, J. A. The LFOPC leap-frog algorithm for constrained optimization. *Computers and Mathematics with Applications*, 40(8):1085–1096, 2000.

Sonntag, R. E. and Borgnakke, C. *Fundamentals of Thermodynamics*. John Wiley, New York, 7 edition, 2009.

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

Stine, W. B. and Geyer, M. Power from the sun, 2001. URL `http://www.powerfromthesun.net/book.html`.

Tarlecki, J. and Lior, N. Analysis of thermal cycles and working fluids for power generation in space. *Energy Conversion and Management*, 48:2864–2878, 2007.

The Mathworks. *Optimization Toolbox User's Guide*. The Mathworks, 2016.

Tsai, L. Design and Performance of a Gas-Turbine Engine from an Automobile Turbocharger. Technical report, Massachussetts Institute of Technology, 2004.

Turbo by Garrett. Turbocharger Guide, 2016a. URL `http://www.turbobygarrett.com/turbobygarrett/intercooler-effectiveness`.

Turbo by Garrett. What is intercooler effectiveness and how do I measure it, 2016b. URL `http://www.turbobygarrett.com/turbobygarrett/intercooler-effectiveness`.

Turchi, S., Ma, Z., Neises, T., and Wagner, M. Thermodynamic Study of Advanced Supercritical Carbon Dioxide Power Cycles for Concentrating Solar Power Systems. *ASME Journal of Solar Engineering*, 135(4), 2013.

United States Energy Information Administration (Independent Statistics) . Electric Power Monthly, with Data for July 2016. Technical Report July, EPA, 2016.

Wang, J., Yan, Z., Zhou, E., and Dai, Y. Parametric analysis and optimization of a Kalina cycle driven by solar energy. *Applied Thermal Engineering*, 50:408–415, 2013.

Wang, W., Chen, L., Sun, F., and Wu, C. Power optimization of an endoreversible closed intercooled regenerated Brayton cycle. *International Journal of Thermal Sciences*, 44(1):89–94, 2005. doi: 10.1016/j.ijthermalsci.2004.06.002.

Winter, C. J., Sizmann, R. L., and Vant-Hull, L. L. *Solar Power Plants: Fundamentals, Technology, Systems, Economics*. Springer-Verlag Berlin, Heidelberg, 1 edition, 1991. ISBN 9783642647598.

Appendix A

Sizing algorithm

This section describes the implementation of the receiver sizing algorithm developed by Stine and Harrigan (1985) shown in Fig. A.1, in the MATLAB function *'collector.m'* (Appendix B.2.1). As described by the reference to Fig. 2.6 in Chapter 2, the algorithm is applied to relate the aperture diameter of the receiver and the net rate at which heat reflected from the concentrator is absorbed at the cavity receiver. The dish concentrator variables involved in the calculations to follow are shown and labelled in Fig. A.2.

An arbitrary, small receiver aperture diameter W_n is set to initialise the algorithm, after which the total parabolic concentrator error σ_{tot} must be computed. The concentrator error depends on the collector design, tracking and alignment, internal shading, and specular reflectivity, among other factors, and is consequently considered a user-specified constant for the purposes of the work at hand. A typical value of 6.7 mrad for parabolic dish concentrators is noted by Stine and Harrigan (1985). Thereafter, the intercepted solar flux, heat loss rate, and ultimately the net rate of heat absorbed are computed, and approximated with a mathematical function in terms of the receiver aperture diameter, as described below.

Starting at a concentrator rim angle of $\Psi = 0°$, Eq. A.1 (d denotes the receiver aperture diameter, W_n) is rearranged and used to determine the beam spread Δr of the image formed by projecting the reflected rays onto a plane normal to the parabolic radius p (see Fig. A.3), at the concentrator segment corresponding to Ψ.

$$d = \frac{\Delta r}{\cos \Psi} \tag{A.1}$$

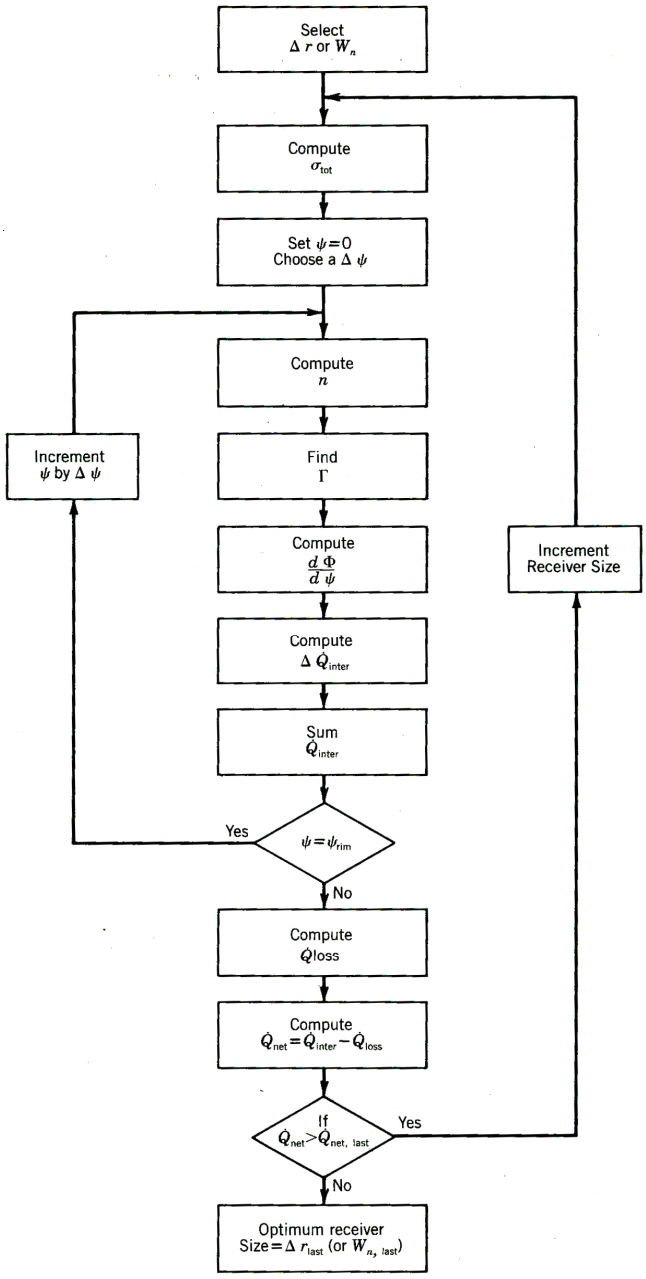


Figure A.1 Receiver sizing algorithm (Stine and Harrigan, 1985).

Equation A.2 is subsequently rearranged and used to determine the focal length f_c, given the aperture area of the paraboloidal concentrator dish.

$$A_s \;=\; \pi R^2 \;=\; 4\pi {f_c}^2 \frac{\sin^2 \Psi_{rim}}{\left(1+\cos \Psi_{rim}\right)^2} \tag{A.2}$$

To determine the flux capture fraction Γ, defined as the ratio of the flux reflected from a parabolic surface in a shaft of light having width equal to n standard deviations of the total angular error (Stine and Harrigan, 1985), the number of standard deviations, n, is calculated using the equation

$$\Delta r \;=\; 2p \tan\left(n \frac{\sigma_{tot}}{2}\right) \tag{A.3}$$

where

$$p \;=\; \frac{2f_c}{1+\cos \Psi} \tag{A.4}$$

Assuming the reflected flux is normally distributed, the flux capture fraction equals the area under the distribution curve, integrated from $-n/2$ to $n/2$, which can be approximated according to Abromowitz and Stegun (1970, cited in Stine and Harrigan, 1985) with the polynomial

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

where

$$Q \;=\; f_1 \left(b_1 t_1 + b_2 {t_1}^2 + b_3 {t_1}^3 + b_4 {t_1}^4 + b_5 {t_1}^5\right) \tag{A.6}$$

and

$$f_1 = \frac{1}{\sqrt{2\pi}} e^{\frac{y^2}{2}} \qquad\qquad y = n/2$$

$$t_1 = 1/\left(1+ry\right) \qquad\qquad r = 0.2316419$$

$$b_1 = 0.319381530$$

$$b_2 = -0.356563782$$

$$b_3 = 1.781477937$$

$$b_4 = -1.821255978$$

$$b_5 = 1.330274429$$

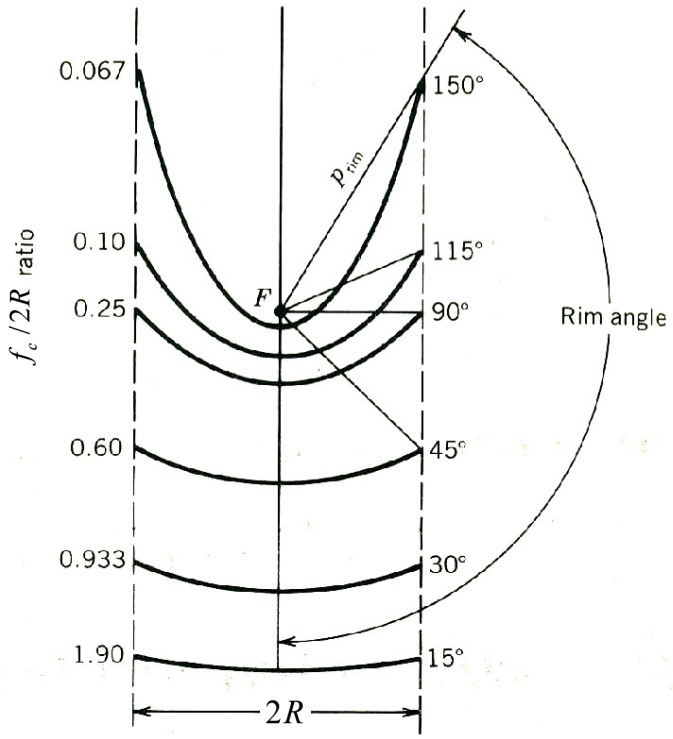


Figure A.2 Relationship between dish radius, focal length and rim angle (Stine and Harrigan, 1985).

The next step in the algorithm is to compute the slope $\frac{d\Phi}{d\Psi}$, which, for a parabolic dish, is written as

$$d\Phi_{PD} = \frac{8\pi I_b {f_c}^2 \sin\Psi d\Psi}{(1+\cos\Psi)^2} \tag{A.7}$$

where $d\Phi_{PD}$ denotes the total radiant flux reflected from the differential area of the parabolic concentrator to the focal plane of the receiver, assuming no reflectance loss (Stine and Harrigan, 1985).

Equation A.8 is then used to compute the rate of energy reflected by the incremental segment, or ring, of the parabolic mirror corresponding to the rim angle Ψ and intercepted by the receiver, with aperture diameter d.

$$\Delta\dot{Q}_i = \rho_s \alpha \Gamma \left(\frac{d\Phi}{d\Psi}\right) \Delta\Psi \tag{A.8}$$

Thereafter, the rim angle is incremented by $\Delta\Psi$ until $\Psi = \Psi_{max}$, and $\Delta\dot{Q}_i$ determined for each corresponding ring. The incremental intercepted energy rates are then summed to determine the total rate of energy $\dot{Q}_i$ intercepted by the receiver with aperture diameter d.

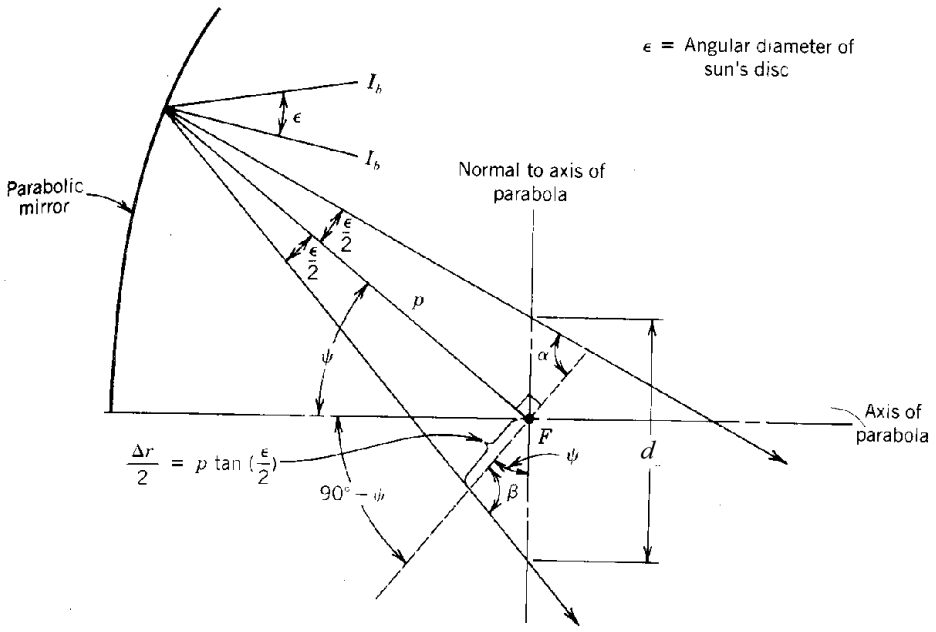


Figure A.3 Reflection of non-parallel rays from a parabolic mirror (Stine and Harrigan, 1985).

The rate of heat loss $\dot{Q}_0$ is subsequently determined. As described in sections 2.2.3 and 3.2.1, heat is primarily lost by convection and radiation from the cavity aperture, and is evaluated using Eqs. 3.7 - 3.9. Heat loss by conduction through the insulation is minimal, and consequently assumed to equal 10 percent of the summed radiative and convective losses. The net rate of heat absorption is then determined using the equation

$$\dot{Q}_{net} = \dot{Q} - \dot{Q}_0 \tag{A.9}$$

The algorithm is repeated, incrementing the overall size of the receiver until the radius of the hemispherical receiver exceeds the concentrator dish radius. The output, then is a curve, such as those shown in Fig. A.4, which describes the net rate at which heat from a concentrator with diameter D is absorbed by the receiver for a range of values of the cavity aperture diameter, d. Note that the curves shown in Fig. A.4 are for $e_p = 0.0067$ and $\Psi = 45°$, as suggested by Stine and Harrigan (1985).

Finally, the output of *collector.m* is numerically approximated using the *curvefit* function in MATLAB, to obtain Eq. 3.10 relating Q_{net} and d. Alternatively, the discrete least squares approximation method may be used.

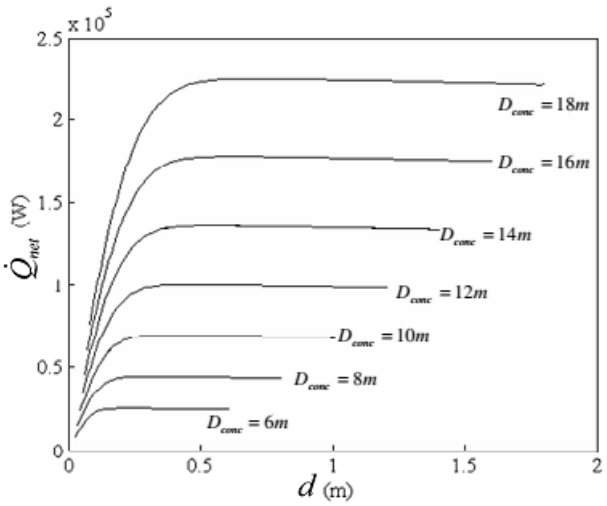


Figure A.4 Relation between net absorbed heat and the aperture diameter for a range of concentrator diameters (Le Roux, 2011).

It follows from the entropy minimisation principle that the aperture diameter corresponding to the maximum amount of solar energy available according to Fig. A.4 is not necessarily optimal for the performance of the integrated STBC. Consequently, Eq. 3.10 is substituted for Q_{net} in the objective function, and the heat absorbed consequently expressed in terms of the receiver geometry parameters to be optimised.

References

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

Le Roux, W. G. Maximum net power output from an integrated design of a small-scale open and direct solar Thermal Brayton Cycle. Master's thesis, University of Pretoria, 2011.

Appendix B

MATLAB code

B.1 Main script

```
clc
%reheatMAIN

                        %%%%%%%% INPUTS %%%%%%%%
% Concentrator(s) to analyse
minDishRadius1 = 3;
minDishRadius2 = 3;
maxDishRadius1  = 10;
maxDishRadius2  = 10;

% Micro-turbine(s) to analyse
firstTurbineNumber = 1;
lastTurbineNumber  = 22;

% Receiver construction
%construction = 1; %channel
construction = 2; %tube

% Step sizes between analyses
dishStepSize1 = 1;
dishStepSize2 = 1;
turbineStepSize = 1;

% Optimisation settings
% X  = [abreg Dhreg Lreg abrec Dhrec Lrec abreh Dhreh Lreh]
x0i  = [2.2 2.5 5 7.5 10 12.5 15 17.5 20 22.5 25]; % starting point...
    (s) for optimisation
eX   = 1e-11; % convergence tolerance xtol on the step movement (...
    default is 1e-8)
eg   = 1e-5; % convergence tolerance on the norm of the gradient (...
    default is 1e-5)
```

```
kmax = 1e4;  % maximum number of steps per phase (default is 1e3)

                        %%%%%%%%%%%%%%%%%%%%%%%%

%% ANALYSIS ROUTINE
disp([datestr(rem(now,1)), ': RUNNING OPTIMISATION ROUTINE FOR THE ...
   REGENERATIVE BRAYTON CYCLE WITH REHEATING'])
fprintf('\n')

dishRadiusList = minDishRadius1:dishStepSize1:maxDishRadius1;
spmd % runs (s)ame (p)rogram/routine in parallel for (m)ultiple (d)...
   atasets (in this case, dish radii associated with subroutines)
for distributionIndex = labindex:numlabs:(length(dishRadiusList))
    dishRadius1 = dishRadiusList(distributionIndex);

    for dishRadius2 = minDishRadius2:dishStepSize2:dishRadius1

        % Cycle ID
        cycleID = 'reheat';
        % Receiver ID
        receiverID = strcat('Drec', num2str(dishRadius1 * 2));
        % Reheater ID
        reheaterID = strcat('Dreh', num2str(dishRadius2 * 2));
        disp(strcat(['SUBROUTINE ', receiverID, reheaterID, ' ...
           ASSIGNED TO LAB ', num2str(labindex), '.']))

        nTurbine = firstTurbineNumber;
        runOnce = 0;
        while nTurbine <= lastTurbineNumber || runOnce == 0

            % Micro-turbine ID
            turbineID = strcat('MT', num2str(nTurbine));

            %% INITIALISATION
            % Microturbine properties
            microturbineArray = csvread('microturbineProperties.csv...
               ',1,1);
            propertiesArray = csvread('convectionProperties.csv'...
               ,1,0);
            minimumPressureRatio = microturbineArray(nTurbine,2);
            maximumPressureRatio = microturbineArray(nTurbine,3);
            pressureRatioRange = minimumPressureRatio:0.1:...
               maximumPressureRatio;
            n = length(pressureRatioRange);

            % Parameter arrays
            massFlowArray = zeros(n,1);
            performanceMeasureArray = zeros(n,4);
            dimensionArray = zeros(n,9);
            regPerformanceArray = zeros(n,7);
            recPerformanceArray = zeros(n,6);
            rehPerformanceArray = zeros(n,6);
            irreversibilityArray = zeros(n,7);
            entropyGenerationArray = zeros(n,8);
            temperatureArray = zeros(n,16);
```

```
pressureArray = zeros(n,16);

% Collector properties
rimAngle = 45;
Ts = 1200;
T0 = 300;
reflectivity = 0.93;
alpha = 0.98;
k = 0.05;
beta = 90;
irradiance = 1000;
concentratorError = 0.0067;
windFactor = 1;
[CF1, dishApertureArea1, dc1] = collector(dishRadius1, ...
    rimAngle, Ts, T0, reflectivity, alpha, k, beta, ...
    irradiance, concentratorError, windFactor);
[CF2, dishApertureArea2, dc2] = collector(dishRadius2, ...
    rimAngle, Ts, T0, reflectivity, alpha, k, beta, ...
    irradiance, concentratorError, windFactor);

%% OPTIMISATION
config = strcat(receiverID, reheaterID, turbineID);

xi = 1;
x0 = x0i(xi);
i = 1;
while i <= n

    pressureRatio = pressureRatioRange(i);

    % Optimal variables (X)
    if construction == 1
        X0 = [x0, x0, x0, x0, x0, x0, x0, x0, x0];
    else
        X0 = [x0, x0, x0, 1, x0, x0, 1, x0, x0];
    end
    [PC, x] = LFOPC(X0, eX, eg, kmax, nTurbine, ...
       pressureRatio, propertiesArray, ...
       microturbineArray, CF1, CF2, dishApertureArea1, ...
       dishApertureArea2, dc1, dc2, construction);
    % Move starting point if iteration limit is reached
    if PC == 1
        xi = xi + 1;
        if xi <= length(x0i)
            x0 = x0i(xi);
            continue
        elseif xi > length(x0i)
            % Check that themicroturbine is within the ...
               range applicable to the concentrator(s) ...
               analysed in the current subroutine
            nTurbine = nTurbine + turbineStepSize;
            disp(strcat(strcat(['    CONFIG ', config, ...
                ' ANALYSED.'])));
            runOnce = 1;
            break
```

```
                end
            end

            % Outputs
            [~, outputs, ~, I, S, T, P] = reheatFunction(x, ...
                nTurbine, pressureRatio, propertiesArray, ...
                microturbineArray, CF1, CF2, construction);
            massFlowArray(i,1) = outputs(1);
            performanceMeasureArray(i,:) = outputs(:,[4,3,2,6])...
                ;
            regPerformanceArray(i,:) = outputs...
                (:,[7,8,9,10,11,12,13]);
            recPerformanceArray(i,:) = outputs...
                (:,[14,15,16,17,18,19]);
            rehPerformanceArray(i,:) = outputs...
                (:,[20,21,22,23,24,25]);
            irreversibilityArray(i,:) = I(:,:);
            entropyGenerationArray(i,:) = S(:,:);
            temperatureArray(i,:) = T;
            pressureArray(i,:) = P;
            x = [x(1), x(2), x(3), x(4), x(5), x(6), x(7), x(8)...
                , x(9)];
            dimensionArray(i,:) = x;

        i = i + 1;
        xi = 1;
        x0 = x0i(xi);
        end

        if PC == 1
            continue
        end

    %% RESULTS PROCESSING
    % Display progress
    disp(strcat(['    CONFIG ', config, ' ANALYSED - SOLUTION ...
        FOUND.']));

    % Write data
    filename = strcat(cycleID, config, '.dat');
    fileID = fopen(filename,'w');
    Head = {'m','Wnet2','Wnet1','Qnet','Eff','abreg','Dhreg','...
        Lreg','abrec','Dhrec',...
         'Lrec','abreh','Dhreh','Lreh','mc_reg','fC_reg','...
            ReC_reg','fH_reg','ReH_reg','NTU',...
         'eff_reg','Qnet_rec','Qdot0_rec','Ts_rec','f_rec','...
            Re_rec','NR_rec','Qnet_reh','Qdot0_reh','Ts_reh',...
         'f_reh','Re_reh','NR_reh','Iinternal','Iexternal','...
            Icomp','Iturb','Irec','Ireh','Ireg',...
         'Sgen_TOTAL','Sgen_comp1','Sgen_comp2','Sgen_turb1','...
            Sgen_turb2','Sgen_rec','Sgen_reh','Sgen_reg','T1','...
            T2',...
         'T3','T4','T5','T6','T7','T8','T9','T10','T11','T12',...
            ...
```

```
            'T13','T14','T15','T16','P1','P2','P3','P4','P5','P6',...
                ...
            'P7','P8','P9','P10','P11','P12','P13','P14','P15','P16...
                '};
        formatSpec = ['%s %s %s %s %s %s %s %s %s %s %s %s %s %s %s...
            %s %s %s %s %s '...
            '%s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s ...
                %s %s '...
            '%s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s ...
                %s %s '...
            '%s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s ...
                %s %s \r\n'];
        fprintf(fileID,formatSpec,Head{1,:});
        ARR = horzcat(massFlowArray, performanceMeasureArray, ...
            dimensionArray, regPerformanceArray, recPerformanceArray...
            , rehPerformanceArray, irreversibilityArray, ...
            entropyGenerationArray, temperatureArray, pressureArray)...
            ;
        formatSpec = ['%f %f %f %f %f %f %f %f %f %f %f %f %f %f %f...
            %f %f %f %f %f '...
            '%f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f ...
                %f %f '...
            '%f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f ...
                %f %f '...
            '%f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f ...
                %f %f \r\n'];
        for row = 1:length(ARR(:,1))
            ARR2 = num2cell(ARR(row,:));
            fprintf(fileID,formatSpec,ARR2{1,:});
        end
        fclose(fileID);

        % Display progress
        if nTurbine == lastTurbineNumber
            disp(strcat([datestr(rem(now,1)), ': SUBROUTINE ', ...
                receiverID, reheaterID, ' COMPLETED.']))
        end

        runOnce = 1;
        nTurbine = nTurbine + turbineStepSize;
        end

    end

end
end

disp(strcat([datestr(rem(now,1)), ': OPTIMISATION ROUTINE ...
    COMPLETED.']))
beep
```

B.2 Auxiliary scripts

B.2.1 collector.m

```
function [CF1, dishApertureArea1, dc1] = collector(dishRadius, ...
    rimAngle, Ts, T0, reflectivity, alpha, k, beta, irradiance, ...
    concentratorError, windFactor)
format long

% Dish aperture area
  dishApertureArea1 = pi * dishRadius^2;
  focalLength = sqrt(dishApertureArea1 * (1 + cosd(rimAngle))^2 / ...
      ((4 * pi * sind(rimAngle)^2)));

%% First iteration
  j = 1;
  angle = 0;
  dAngle = 1;
  sum = 0;
  increment = 0.01;
  collectorApertureDiameter(j) = sqrt(4 * (dishApertureArea1 / 1e16...
      ) / pi);
  wallArea(j) = 8 * pi * collectorApertureDiameter(j)^2 / 4;
  collectorRadius(j) = sqrt((wallArea(j) + pi * ...
      collectorApertureDiameter(j)^2 / 4)/(3*pi));
  Gr = 9.81 * (collectorApertureDiameter(j) * 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 = windFactor * Nu * k / (2* collectorRadius(j));
  Qloss(j) = 2 * h * pi / 4 * collectorApertureDiameter(j)^2 * (Ts ...
      - T0);

  while angle < rimAngle
      deltaR = collectorApertureDiameter(j) * cosd(angle);
      parabolicRadius = 2 * focalLength / (1 + cosd(angle));
      nStandardDeviations = 2 * atan(deltaR / (2 * parabolicRadius)...
          ) / concentratorError;

      % from appendix G
      r = 0.2316419;
      b1 = 0.319381530;
      b2 = -0.356563782;
      b3 = 1.781477937;
      b4 = -1.821255978;
      b5 = 1.330274429;
      y = nStandardDeviations/2;
      f1 = 1 / sqrt(2 * pi) * exp(-(y^2) / 2);
      t1 = 1 / (1 + r * y);
      Q = f1 * (b1*t1 + b2*t1^2 + b3*t1^3 + b4*t1^4 + b5*t1^5);
      F = 1 - 2 * Q;
      slope = 8 * pi * irradiance * focalLength^2 * sind(angle) / ...
          (1 + cosd(angle))^2;
```

```
      % insulation thickness
      k_ins = 0.05;
      t(j) = k_ins * 4 * pi * collectorRadius(j)^2 * (Ts - T0) / ...
          Qloss(j) * 10;

      % integrate dQ
      dQ = reflectivity * alpha * F * slope * pi / 180 * dAngle;
          if pi * (collectorRadius(j) + t(j))^2 < 4 * pi * ...
              focalLength^2 * sind(rimAngle)^2 / (1 + cosd(rimAngle)...
              )^2
              sum = sum + dQ;
          end

      angle = angle + dAngle;
  end

  % include conduction loss
  Qloss(j) = Qloss(j) + Qloss(j) / 10;
  Qnet(j) = sum - Qloss(j);


%% Subsequent iterations
  while pi / 4 * collectorApertureDiameter(j)^2 < dishApertureArea1...
      /100
      j = j + 1;
      angle = 0;
      dAngle = 1;
      sum = 0;
      collectorApertureDiameter(j) = collectorApertureDiameter(j - ...
          1) + increment;
      wallArea(j) = 8 * pi * collectorApertureDiameter(j)^2 / 4;
      collectorRadius(j) = sqrt((wallArea(j) + pi * ...
          collectorApertureDiameter(j)^2 / 4)/(3*pi));
      Gr = 9.81 * (collectorApertureDiameter(j) * 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 = windFactor * Nu * k / (2* collectorRadius(j));
      Qloss(j) = 2 * h * pi / 4 * collectorApertureDiameter(j)^2 * ...
          (Ts - T0);

      while angle < rimAngle
        deltaR = collectorApertureDiameter(j) * cosd(angle);
        parabolicRadius = 2 * focalLength / (1 + cosd(angle));
        nStandardDeviations = 2 * atan(deltaR / (2 * ...
            parabolicRadius)) / concentratorError;

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

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

        slope = 8 * pi * irradiance * focalLength^2 * sind(angle) /...
             (1 + cosd(angle))^2;

        % insulation thickness
        k_ins = 0.05;
        t(j) = k_ins * 4 * pi * collectorRadius(j)^2 * (Ts - T0) / ...
           Qloss(j) * 10;

        dQ = reflectivity * alpha * F * slope * pi / 180 * dAngle;
        if pi * (collectorRadius(j) + t(j))^2 < 4 * pi * ...
           focalLength^2 * sind(rimAngle)^2 / (1 + cosd(rimAngle))...
           ^2
            sum = sum + dQ;
        end

        angle = angle + dAngle;
      end

      % include conduction loss
      Qloss(j) = Qloss(j) + Qloss(j) / 10;
      Qnet(j) = sum - Qloss(j);

  end

  dc1 = collectorApertureDiameter(2);

%% Approximate function with polynomial
   p = polyfit(collectorApertureDiameter, Qnet, 10);

   r0 = p(11);
   r1 = p(10);
   r2 = p(9);
   r3 = p(8);
   r4 = p(7);
   r5 = p(6);
   r6 = p(5);
   r7 = p(4);
   r8 = p(3);
   r9 = p(2);
   r10 = p(1);

    CF1 = [r0 r1 r2 r3 r4 r5 r6 r7 r8 r9 r10];
    format short
    clear X
end
```

B.2.2 getConvectionProperties.m

```
function [Cp, k, mu, Pr, rho] = getConvectionProperties(T, ...
   propertiesArray)

T_Array = propertiesArray(:,1);  mu_Array = propertiesArray(:,4);
Cp_Array = propertiesArray(:,2); Pr_Array = propertiesArray(:,5);
k_Array = propertiesArray(:,3);  rho_Array = propertiesArray(:,6);

%% Constants
    if T_Array(1) >= T
    Cp = Cp_Array(1);
    k = k_Array(1);
    mu = mu_Array(1);
    Pr = Pr_Array(1);
    rho = rho_Array(1);

    elseif T_Array(1) < T && T <= T_Array(2)
    Cp = Cp_Array(1) +...
        ((T-T_Array(1))/(T_Array(2)-T_Array(1)))*(Cp_Array(2)-...
           Cp_Array(1));
    k = k_Array(1) +...
        ((T-T_Array(1))/(T_Array(2)-T_Array(1)))*(k_Array(2)-...
           k_Array(1));
    mu = mu_Array(1) +...
        ((T-T_Array(1))/(T_Array(2)-T_Array(1)))*(mu_Array(2)-...
           mu_Array(1));
    Pr = Pr_Array(1) +...
        ((T-T_Array(1))/(T_Array(2)-T_Array(1)))*(Pr_Array(2)-...
           Pr_Array(1));
    rho = rho_Array(1) +...
      ((T-T_Array(1))/(T_Array(2)-T_Array(1)))*(rho_Array(2)-...
         rho_Array(1));

    elseif T_Array(2) < T && T <= T_Array(3)
    Cp = Cp_Array(2) +...
        ((T-T_Array(2))/(T_Array(3)-T_Array(2)))*(Cp_Array(3)-...
           Cp_Array(2));
    k = k_Array(2) +...
        ((T-T_Array(2))/(T_Array(3)-T_Array(2)))*(k_Array(3)-...
           k_Array(2));
    mu = mu_Array(2) +...
        ((T-T_Array(2))/(T_Array(3)-T_Array(2)))*(mu_Array(3)-...
           mu_Array(2));
    Pr = Pr_Array(2) +...
        ((T-T_Array(2))/(T_Array(3)-T_Array(2)))*(Pr_Array(3)-...
           Pr_Array(2));
    rho = rho_Array(2) +...
      ((T-T_Array(2))/(T_Array(3)-T_Array(2)))*(rho_Array(3)-...
         rho_Array(2));

    elseif T_Array(3) < T && T <= T_Array(4)
    Cp = Cp_Array(3) +...
```

```
    ((T-T_Array(3))/(T_Array(4)-T_Array(3)))*(Cp_Array(4)-...
       Cp_Array(3));
k = k_Array(3) +...
    ((T-T_Array(3))/(T_Array(4)-T_Array(3)))*(k_Array(4)-...
       k_Array(3));
mu = mu_Array(3) +...
    ((T-T_Array(3))/(T_Array(4)-T_Array(3)))*(mu_Array(4)-...
       mu_Array(3));
Pr = Pr_Array(3) +...
    ((T-T_Array(3))/(T_Array(4)-T_Array(3)))*(Pr_Array(4)-...
       Pr_Array(3));
rho = rho_Array(3) +...
  ((T-T_Array(3))/(T_Array(4)-T_Array(3)))*(rho_Array(4)-...
     rho_Array(3));

elseif T_Array(4) < T && T <= T_Array(5)
Cp = Cp_Array(4) +...
    ((T-T_Array(4))/(T_Array(5)-T_Array(4)))*(Cp_Array(5)-...
       Cp_Array(4));
k = k_Array(4) +...
    ((T-T_Array(4))/(T_Array(5)-T_Array(4)))*(k_Array(5)-...
       k_Array(4));
mu = mu_Array(4) +...
    ((T-T_Array(4))/(T_Array(5)-T_Array(4)))*(mu_Array(5)-...
       mu_Array(4));
Pr = Pr_Array(4) +...
    ((T-T_Array(4))/(T_Array(5)-T_Array(4)))*(Pr_Array(5)-...
       Pr_Array(4));
rho = rho_Array(4) +...
  ((T-T_Array(4))/(T_Array(5)-T_Array(4)))*(rho_Array(5)-...
     rho_Array(4));

elseif T_Array(5) < T && T <= T_Array(6)
Cp = Cp_Array(5) +...
    ((T-T_Array(5))/(T_Array(6)-T_Array(5)))*(Cp_Array(6)-...
       Cp_Array(5));
k = k_Array(5) +...
    ((T-T_Array(5))/(T_Array(6)-T_Array(5)))*(k_Array(6)-...
       k_Array(5));
mu = mu_Array(5) +...
    ((T-T_Array(5))/(T_Array(6)-T_Array(5)))*(mu_Array(6)-...
       mu_Array(5));
Pr = Pr_Array(5) +...
    ((T-T_Array(5))/(T_Array(6)-T_Array(5)))*(Pr_Array(6)-...
       Pr_Array(5));
rho = rho_Array(5) +...
  ((T-T_Array(5))/(T_Array(6)-T_Array(5)))*(rho_Array(6)-...
     rho_Array(5));

elseif T_Array(6) < T && T <= T_Array(7)
Cp = Cp_Array(6) +...
    ((T-T_Array(6))/(T_Array(7)-T_Array(6)))*(Cp_Array(7)-...
       Cp_Array(6));
k = k_Array(6) +...
```

```
        ((T-T_Array(6))/(T_Array(7)-T_Array(6)))*(k_Array(7)-...
           k_Array(6));
    mu = mu_Array(6) +...
        ((T-T_Array(6))/(T_Array(7)-T_Array(6)))*(mu_Array(7)-...
           mu_Array(6));
    Pr = Pr_Array(6) +...
        ((T-T_Array(6))/(T_Array(7)-T_Array(6)))*(Pr_Array(7)-...
           Pr_Array(6));
    rho = rho_Array(6) +...
      ((T-T_Array(6))/(T_Array(7)-T_Array(6)))*(rho_Array(7)-...
         rho_Array(6));

    elseif T_Array(7) < T
    Cp = Cp_Array(7);
    k = k_Array(7);
    mu = mu_Array(7);
    Pr = Pr_Array(7);
    rho = rho_Array(7);
    end

end
```

B.2.3 getMicroturbineProperties.m

```
function [massFlowRate, turbineEfficiency, compressorEfficiency] = ...
    getMicroturbineProperties(nTurbine, pressureRatio, ...
    microturbineArray)

  minimumPressureRatio = microturbineArray(:,2);
  maximumPressureRatio = microturbineArray(:,3);
  minimumMassFlowRate = microturbineArray(:,4);
  maximumMassFlowRate = microturbineArray(:,5);
  maximumTurbineEfficiency = microturbineArray(:,6);
  maximumCompressorEfficiency = microturbineArray(:,7);

  massFlowRate = minimumMassFlowRate(nTurbine) + ((...
      maximumMassFlowRate(nTurbine) - ...
      minimumMassFlowRate(nTurbine)) / (maximumPressureRatio(...
          nTurbine) - ...
      minimumPressureRatio(nTurbine))) * (pressureRatio-...
          minimumPressureRatio(nTurbine));

  turbineEfficiency = maximumTurbineEfficiency(nTurbine)/100;

  compressorEfficiency = maximumCompressorEfficiency(nTurbine)/100;

end
```

B.3 Cycle scripts

B.3.1 reheatFunction.m

```
function [PC, outputs, Ts, I, S, T, P] = reheatFunction(x, nTurbine...
    , pressureRatio, propertiesArray, microturbineArray, CF1, CF2, ...
    construction)
x = [x(1) x(2)/100 x(3) x(4) x(5)/100 x(6) x(7) x(8)/100 x(9)];
if construction == 2
     x(4) = 1;
     x(7) = 1;
end
[mdot, eff_turb, eff_comp] = getMicroturbineProperties(nTurbine, ...
    pressureRatio, microturbineArray);

%% 1) CONSTANTS & ASSUMPTIONS
       % Ambient air
         R = 287;
         k = 1.4;
         T0 = 300;
         P0 = 101325;
       % Assumed temperatures
         T5 = 500;                          % recuperator cold inlet
         T6 = 750;                          % recuperator cold outlet
         T15 = 1000;                        % recuperator hot inlet
         T16 = 750;                         % recuperator hot outlet
         T7 = 750;                          % receiver inlet
         T8 = 1000;                         % receiver outlet
         T11 = 750;                         % reheater inlet
         T12 = 1000;                        % reheater outlet
         T_drop = 2;
         Tsun = 2470;

%% 2) THERMODYNAMIC ANALYSIS
error = 1;
convergenceCounter = 0;
while error > 1e-3
T7_0 = T7;
PC = 0;
convergenceCounter = convergenceCounter + 1;

% i)     RECUPERATOR
         H_reg = 1;
         t_plate = 0.001;
         k_plate = 401;
         b_Channel = x(2) * (x(1) + 1) / (2 * x(1));
         nChannels = H_reg / (t_plate + b_Channel);
         m_c = 2 * mdot / nChannels;
         % Cold side flow
         [CpC_reg, kC_reg, muC_reg, PrC_reg, rhoC_reg] = ...
            getConvectionProperties((T5 + T6) / 2, propertiesArray);
         ReC_reg = 4 * m_c * x(1) / (muC_reg * x(2) * (x(1) + 1)^2);
         fC_reg = (0.79 * log(ReC_reg) - 1.64)^-2;
```

```
NuC_reg = PrC_reg * (fC_reg / 8) * (ReC_reg - 1000) / (1 + ...
    12.7 * (fC_reg / 8)^0.5 * (PrC_reg^(2/3) - 1));
hC_reg = kC_reg * NuC_reg / x(2);
% Hot side flow
[CpH_reg, kH_reg, muH_reg, PrH_reg, rhoH_reg] = ...
    getConvectionProperties((T15 + T16) / 2, propertiesArray...
    );
ReH_reg = 4 * m_c * x(1) / (muH_reg * x(2) * (x(1) + 1)^2);
fH_reg = (0.79 * log(ReH_reg) - 1.64)^-2;
NuH_reg = PrH_reg * (fH_reg / 8) * (ReH_reg - 1000) / (1 + ...
    12.7 * (fH_reg / 8)^0.5 * (PrH_reg^(2/3) - 1));
hH_reg = kH_reg * NuH_reg / x(2);
% Effectiveness (e-NTU)
foulingFactor = 0.0004;
a_reg = x(2) * x(3) * (x(1) + 1) * (1 + 1 / (x(1)));
U = (1 / hC_reg + 2 * foulingFactor + t_plate / k_plate + 1...
     / hH_reg)^-1;
NTU = (U * a_reg / (m_c * CpC_reg));
C = CpC_reg / CpH_reg;
eff_reg = (1 - exp(-NTU * (1 - C))) / (1 - C * exp(-NTU * ...
    (1 - C)));
if eff_reg > 1
    PC = 1;
    outputs = 0;
    Ts = 0;
    I = 0;
    S = 0;
    T = 0;
    P = 0;
    return
end

% ii)   RECEIVER
[Cp_rec, k_rec, mu_rec, Pr_rec, rho_rec] = ...
    getConvectionProperties((T7 + T8) / 2, propertiesArray);
a_rec = x(5)*(x(4)+1)/2;
d_rec = sqrt(x(5) * x(6) * (x(4) + 1) / (4 * pi));
if construction == 1
    Nrec = ((sqrt(3)-1)*d_rec)/(2*a_rec);
else
    Nrec = (sqrt(3)-1)*d_rec/(2*x(5));
end
Re_rec = 4 * mdot * x(4) / (mu_rec * x(5) * (x(4) + 1)^2);
f_rec = (0.79 * log(Re_rec) - 1.64)^-2;
Qnet_rec = CF1(1) + CF1(2)*sqrt(d_rec) + CF1(3)*d_rec + CF1...
    (4)*d_rec^(3/2) + CF1(5)*d_rec^2 + CF1(6)*d_rec^(5/2) + ...
    CF1(7)*d_rec^3 + CF1(8)*d_rec^(7/2) + CF1(9)*d_rec^4 + ...
    CF1(10)*d_rec^(9/2) + CF1(11)*d_rec^5;
if construction == 1
    Ts_rec = T8 + Qnet_rec / (0.023*x(6)*(x(4)+1)*(1+1/x(4)...
        )*k_rec*Pr_rec^0.4*Re_rec^0.8);
else
    Ts_rec = T8 + Qnet_rec / (0.023*pi*x(6)*k_rec*Pr_rec^0...
        .4*(4*mdot/(mu_rec*pi*x(5)))^0.8);
end
```

```
% iii)  REHEATER
        [Cp_reh, k_reh, mu_reh, Pr_reh, rho_reh] = ...
           getConvectionProperties((T11 + T12) / 2, propertiesArray...
           );
        a_reh = x(8)*(x(7)+1)/2;
        d_reh = sqrt(x(8) * x(9) * (x(7) + 1) / (4 * pi));
        if construction == 1
            Nreh = ((sqrt(3)-1)*d_reh)/(2*a_reh);
        else
            Nreh = ((sqrt(3)-1)*d_reh)/(2*x(8));
        end
        Re_reh = 4 * mdot * x(7) / (mu_reh * x(8) * (x(7) + 1)^2);
        f_reh = (0.79 * log(Re_reh) - 1.64)^-2;
        Qnet_reh = CF2(1) + CF2(2)*sqrt(d_reh) + CF2(3)*d_reh + CF2...
           (4)*d_reh^(3/2) + CF2(5)*d_reh^2 + CF2(6)*d_reh^(5/2) + ...
           CF2(7)*d_reh^3 + CF2(8)*d_reh^(7/2) + CF2(9)*d_reh^4 + ...
           CF2(10)*d_reh^(9/2) + CF2(11)*d_reh^5;
        if construction == 1
            Ts_reh = T12 + Qnet_reh / (0.023*x(9)*(x(7)+1)*(1+1/x...
               (7))*k_reh*Pr_reh^0.4*Re_reh^0.8);
        else
            Ts_reh = T12 + Qnet_reh / (0.023*pi*x(9)*k_reh*Pr_reh^0...
               .4*(4*mdot/(mu_reh*pi*x(8)))^0.8);
        end
        Q_abs = Qnet_rec + Qnet_reh;

% iv)   PRESSURE DROPS
        Pdrop23 = 0.004;
        Pdrop45 = 0.004;
        Pdrop56 = (fC_reg * (8 * m_c^2 * x(1)^2 / (rhoC_reg * (x(1)...
            + 1)^4)) * (x(3) / x(2)^5));
        Pdrop67 = 0.004;
        if construction == 1
            Pdrop78 = f_rec * (8 * mdot^2 * x(4)^2 / (rho_rec * (x...
               (4) + 1)^4)) * (x(6) / x(5)^5);
        else
            Pdrop78 = ((0.79*log(4*mdot/(mu_rec*pi*x(5)))-1.64)^-2)...
               *(8*mdot^2/(rho_rec*pi^2))*(x(6)/(x(5)^5));
        end
        Pdrop89 = 0.004;
        Pdrop1011 = 0.004;
        if construction == 1
            Pdrop1112 = f_reh * (8 * mdot^2 * x(7)^2 / (rho_reh * (...
               x(7) + 1)^4)) * (x(9) / x(8)^5);
        else
            Pdrop1112 = ((0.79*log(4*mdot/(mu_reh*pi*x(8)))-1.64)...
               ^-2)*(8*mdot^2/(rho_reh*pi^2))*(x(9)/(x(8)^5));
        end
        Pdrop1213 = 0.004;
        Pdrop1415 = 0.001;
        Pdrop1516 = (fH_reg * (8 * m_c^2 * x(1)^2 / (rhoH_reg * (x...
           (1) + 1)^4)) * (x(3) / x(2)^5));

% v)    PRESSURE FIELD
```

```
        P1 = P0;                                    % Comp.1 inlet
        P2 = pressureRatio * P1;                    % Comp.1 outlet
        P3 = P2 * (1 - Pdrop23);                    % Comp.2 inlet
        P4 = pressureRatio * P3;                    % Comp.2 outlet
        P5 = P4 * (1- Pdrop45);                     % Reg. cold inlet
        P6 = P5 - Pdrop56;                          % Reg. cold outlet
        P7 = P6 * (1 - Pdrop67);                    % Receiver inlet
        P8 = P7 - Pdrop78;                          % Receiver outlet
        P9 = P8 * (1 - Pdrop89);                    % Turb.1 inlet
        P16 = P0;                                   % Reg. hot outlet
        P15 = P16 + Pdrop1516;                      % Reg. hot inlet
        P14 = P15 * (1 + Pdrop1415);                % Turb.2 outlet
        P10 = ((P14 * P9) / ((1 - Pdrop1213) * (1 - Pdrop1011)) + ...
            ...
             Pdrop1112^2/(4 * (1 - Pdrop1011)^2))^(1/2) + Pdrop1112 ...
                / ...
             (2 * (1 - Pdrop1011));                 % Turb. 1 outlet
        P11 = P10 * (1 - Pdrop1011);                % Reh. inlet
        P12 = P11 - Pdrop1112;                      % Reh. outlet
        P13 = P12 * (1 - Pdrop1213);                % Turb.2 inlet
        P = [P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 P12 P13 P14 P15 P16...
            ];

% vi)   TEMPERATURE FIELD
        T1 = T0;                                    % Comp.1 inlet
        Kcomp = pressureRatio^((k-1)/k);
        T2 = T1 * (1 + (Kcomp - 1) / eff_comp);     % Comp.1 outlet
        T3 = T2 - T_drop;                           % Comp.2 inlet
        T4 = T3 * (1 + (Kcomp - 1) / eff_comp);     % Comp.2 outlet
        T5 = T4 - T_drop;                           % Rec. cold inlet
        T8 = T7 + Qnet_rec / (mdot * Cp_rec);       % Rec. outlet
        T9 = T8 - T_drop;                           % Turb.1 inlet
        Kturb = (P9 / P10)^((k-1)/k);
        T10 = T9 * (1 - eff_turb * (1 - 1 / Kturb)); % Turb.1 ...
            outlet
        T11 = T10 - T_drop;                         % Reh. inlet
        T12 = T11 + Qnet_reh / (mdot * Cp_reh);     % Reh. outlet
        T13 = T12 - T_drop;                         % Turb.2 inlet
        T14 = T13 * (1 - eff_turb * (1 - 1 / Kturb)); % Turb.2 ...
            outlet
        T15 = T14 - T_drop;                         % Reg. hot inlet
        T6 = T5 + eff_reg * (T15 - T5);             % Reg. cold ...
            outlet
        T7 = T6 - T_drop;                           % Rec. inlet
        T16 = T15 - eff_reg * CpC_reg / CpH_reg * ...
             (T15 - T5);                            % Reg. hot oulet
        T = [T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 T13 T14 T15 T16...
            ];
        for i = 1:length(T)
            if isnan(T(i)) == 1 || isreal(T(i)) == 0 || ...
                convergenceCounter > 1e4;
                PC = 1;
                outputs = 0;
                Ts = 0;
                I = 0;
```

```
                S = 0;
                T = 0;
                P = 0;
                return
            end
        end

error = abs(T7 - T7_0) / T7_0;
end

% vii)  ENTROPY GENERATION
        [Cp_comp1, ~, ~, ~, ~] = getConvectionProperties((T1 + T2) ...
           / 2, propertiesArray);
        Sgen_comp1 = -mdot * Cp_comp1 * log(T1 / T2) + mdot * R * ...
           log(P1 / P2);

        [Cp_duct23, ~, ~, ~, ~] = getConvectionProperties((T2 + T3)...
            / 2, propertiesArray);
        Sgen_duct23 = mdot * Cp_duct23 * log(T3 / T2) - mdot * R * ...
           log(P3 / P2) + mdot * Cp_duct23 * T_drop / T0;

        [Cp_comp2, ~, ~, ~, ~] = getConvectionProperties((T3 + T4) ...
           / 2, propertiesArray);
        Sgen_comp2 = -mdot * Cp_comp2 * log(T3 / T4) + mdot * R * ...
           log(P3 / P4);

        [Cp_duct45, ~, ~, ~, ~] = getConvectionProperties((T4 + T5)...
            / 2, propertiesArray);
        Sgen_duct45 = mdot * Cp_duct45 * log(T5 / T4) - mdot * R * ...
           log(P5 / P4) + mdot * Cp_duct45 * T_drop / T0;

        T_reg = (T5 + T6 + T15 + T16)/4;
        [Cp_reg, ~, ~, ~, ~] = getConvectionProperties(T_reg, ...
           propertiesArray);
        Sgen_reg = mdot * Cp_reg * log((T6 * T16) / (T15 * T5) * ((...
           P6 * P16) / (P15 * P5))^((1-k)/k)) + mdot * Cp_reg * ...
           T_drop / T0;

        [Cp_duct67, ~, ~, ~, ~] = getConvectionProperties((T6 + T7)...
            / 2, propertiesArray);
        Sgen_duct67 = mdot * Cp_duct67 * log(T7 / T6) - mdot * R *...
           log(P7 / P6) + mdot * Cp_duct67 * T_drop / T0;

        Tstar = Tsun;
        beta = pi / 2;
        d_rec = sqrt(x(5) * x(6) * (x(4) + 1) / (4 * pi));
        D_rec = sqrt(3) * d_rec;
        v_rec = (mu_rec / rho_rec);
        Aa_rec = pi * d_rec^2 / 4;
        Gr_Drec = 9.81 * beta * (Ts_rec - T0) * D_rec^3 / v_rec^2;
        Qdot0_rec = 1.396 * Gr_Drec^0.209 * (1 + cos(beta))^0.968 *...
            (Ts_rec / T0)^(-0.317) * ...
            (d_rec / D_rec)^(0.425) * (k_rec * Aa_rec / D_rec) * (...
               Ts_rec - T0);
        Qrec_star = Qnet_rec + Qdot0_rec;
```

```
        Sgen_rec = -Qrec_star / Tstar + Qdot0_rec / T0 + mdot * ...
           Cp_rec * log(T8 / T7) - mdot * R * log(P8 / P7);

        [Cp_duct89, ~, ~, ~, ~] = getConvectionProperties((T8 + T9)...
            / 2, propertiesArray);
        Sgen_duct89 = mdot * Cp_duct89 * log(T9 / T8) - mdot * R * ...
           log(P9 / P8) + mdot * Cp_duct89 * T_drop / T0;

        [Cp_turb1, ~, ~, ~, ~] = getConvectionProperties((T9 + T10)...
            / 2, propertiesArray);
        Sgen_turb1 = -mdot * Cp_turb1 * log(T9 / T10) + mdot * R * ...
           log(P9 / P10);

        [Cp_duct1011, ~, ~, ~, ~] = getConvectionProperties((T10 + ...
           T11) / 2, propertiesArray);
        Sgen_duct1011 = mdot * Cp_duct1011 * log(T11 / T10) - mdot ...
           * R * log(P11 / P10) + mdot * Cp_duct1011 * T_drop / T0;

        d_reh = sqrt(x(8) * x(9) * (x(7) + 1) / (4 * pi));
        D_reh = sqrt(3) * d_reh;
        v_reh = (mu_reh/rho_reh);
        Aa_reh = pi * d_reh^2 / 4;
        Gr_Dreh = 9.81 * beta * (Ts_reh - T0) * D_reh^3 / v_reh^2;
        Qdot0_reh = 1.396 * Gr_Dreh^0.209 * (1 + cos(beta))^0.968 *...
            (Ts_reh / T0)^(-0.317) * ...
            (d_reh / D_reh)^(0.425) * (k_reh * Aa_reh / D_reh) * (...
               Ts_reh - T0);
        Qreh_star = Qnet_reh + Qdot0_reh;
        Sgen_reh = -Qreh_star / Tstar + Qdot0_reh / T0 + mdot * ...
           Cp_reh * log(T12 / T11) - mdot * R * log(P12 / P11);

        [Cp_duct1213, ~, ~, ~, ~] = getConvectionProperties((T12 + ...
           T13) / 2, propertiesArray);
        Sgen_duct1213 = mdot * Cp_duct1213 * log(T13 / T12) - mdot ...
           * R * log(P13 / P12) + mdot * Cp_duct1213 * T_drop / T0;

        [Cp_turb2, ~, ~, ~, ~] = getConvectionProperties((T13 + T14...
           ) / 2, propertiesArray);
        Sgen_turb2 = -mdot * Cp_turb2 * log(T13/T14) + mdot * R * ...
           log(P13 / P14);

        [Cp_duct1415, ~, ~, ~, ~] = getConvectionProperties((T14 + ...
           T15) / 2, propertiesArray);
        Sgenduct1415 = mdot * Cp_duct1415 * log(T15 / T14) - mdot *...
            R * log(P15 / P14) + mdot * Cp_duct1415 * T_drop / T0;

        Sgen_parts = Sgen_comp1 + Sgen_comp2 + Sgen_reg + Sgen_rec ...
           + Sgen_turb1 + Sgen_reh + Sgen_turb2;

        Sgen_ducts = Sgen_duct23 + Sgen_duct45 + Sgen_duct67 + ...
           Sgen_duct89 + Sgen_duct1011 + Sgen_duct1213 + ...
           Sgenduct1415;

% viii) IRREVERSIBILITIES
        I_comp = T0 * (Sgen_comp1 + Sgen_comp2) / 1000;
```

```
        I_reg = T0 * Sgen_reg / 1000;
        I_rec = T0 * Sgen_rec / 1000;
        I_reh = T0 * Sgen_reh / 1000;
        I_turb = T0 * (Sgen_turb1 + Sgen_turb2) / 1000;
        [Cp, ~, ~, ~, ~] = getConvectionProperties((T1 + T16) / 2, ...
            propertiesArray);
        I_internal = (I_comp + I_reg + I_rec + I_turb + I_reh);
        I_external = -(mdot * Cp * (T1 - T16) - mdot * T0 * Cp * ...
            log(T1 / T16)) / 1000;

% ix)   OUTPUT PARAMETERS
        Qnet = Q_abs / 1000;
        Wc = mdot * Cp_comp1 * (T2 - T1) + mdot * Cp_comp2 * (T4 - ...
            T3);
        Wt = mdot * Cp_turb1 * (T9 - T10) + mdot * Cp_turb2 * (T13 ...
            - T14);
        Wnet1 = (Wt - Wc) / 1000;
        eff_th = Wnet1 / Qnet;
        Sgen = Sgen_parts + Sgen_ducts;
        Qstar = Qrec_star + Qreh_star;
        W2 = -T0 * Sgen + (1 - (T0 / Tstar)) * Qstar + mdot * Cp * ...
            (T1 - T16) + mdot * T0 * Cp * log(T16 / T1);
        Wnet2 = W2 / 1000;
        if isreal(W2) == 0
            PC = 1;
            outputs = 0;
            Ts = 0;
            I = 0;
            S = 0;
            T = 0;
            P = 0;
            return
        end
        eff_ex = Wnet2 / Qnet;
        outputs = [mdot, Qnet, Wnet1, Wnet2, eff_th, eff_ex,...
            m_c, fC_reg, ReC_reg, fH_reg, ReH_reg, NTU, eff_reg,...
            Qnet_rec/1000, Qdot0_rec/1000, Ts_rec, f_rec, Re_rec, ...
                Nrec,...
            Qnet_reh/1000, Qdot0_reh/1000, Ts_reh, f_reh, Re_reh, ...
                Nreh];
        Ts      = [Ts_rec, Ts_reh];
        I       = [I_internal, I_external, I_comp, I_turb, I_rec, ...
            I_reh, I_reg];
        S       = [Sgen, Sgen_comp1, Sgen_comp2, Sgen_turb1, ...
            Sgen_turb2, Sgen_rec, Sgen_reh, Sgen_reg];
end
```

B.3.2 intercoolFunction.m

```
function [PC, outputs, Ts_rec, I, S, T, P] = intercoolFunction(x, ...
    nTurbine, pressureRatio, propertiesArray, microturbineArray, CF,...
     construction)
x = [x(1)/100 x(2)/1000 x(3) x(4) x(5)/100 x(6) x(7) x(8)/100 x(9)...
    ];
if construction == 2
     x(7) = 1;
end
[mdot, eff_turb, eff_comp] = getMicroturbineProperties(nTurbine, ...
    pressureRatio, microturbineArray);

%% 1) CONSTANTS & ASSUMPTIONS
      % Ambient air
        R = 287;
        k = 1.4;
        T0 = 300;
        P0 = 101325;
      % Assumed temperatures
        T3 = 375;                       % intercooler hot inlet
        T4 = 400;                       % intercooler hot outlet
        TsC_ic = 375;                   % intercooler surface
        T7 = 500;                       % recuperator cold inlet
        T8 = 750;                       % recuperator cold outlet
        T15 = 1000;                     % recuperator hot inlet
        T16 = 750;                      % recuperator hot outlet
        T9 = 750;                       % receiver inlet
        T10 = 1000;                     % receiver outlet
        T_drop = 2;
        Tsun = 2470;

%% 2) THERMODYNAMIC ANALYSIS
error = 1;
convergenceCounter = 0;
while error > 1e-3
T9_0 = T9;
PC = 0;
convergenceCounter = convergenceCounter + 1;

% i)    INTERCOOLER
        W_ic = 1;
        tPlate_ic = 0.001;
        kPlate_ic = 401;
        Eplate_ic = 0.03;
        sigmaB = 5.67e-8;
        % Cold side heat transfer
        [~, kC_ic, muC_ic, PrC_ic, rhoC_ic] = ...
           getConvectionProperties((TsC_ic + T0) / 2, ...
           propertiesArray);
        Beta_ic = ((TsC_ic + T0) / 2)^-1;
        aChannel_ic = x(2) * (x(1) + 1) / 2;
        bChannel_ic = x(2) * (x(1) + 1) / (2 * x(1));
        nChannels_ic = W_ic / (tPlate_ic + aChannel_ic);
```

```
As_ic = 2 * (aChannel_ic + bChannel_ic) * x(3);
mc_ic = 2 * mdot / nChannels_ic;
Ra_ic = 9.81 * Beta_ic * (TsC_ic - T0) * PrC_ic * ...
    aChannel_ic^3 / (muC_ic / rhoC_ic)^2;
NuC_ic = (576/(Ra_ic * x(1))^2 + 2.873/(Ra_ic * x(1))^0.5)...
    ^-0.5;
hC_ic = kC_ic * NuC_ic / aChannel_ic;
% Hot side flow (internal forced convection)
[CpH_ic, kH_ic, muH_ic, PrH_ic, rhoH_ic] = ...
    getConvectionProperties((T3 + T4) / 2, propertiesArray);
ReH_ic = 4 * mc_ic * x(1) / (muH_ic * x(2) * (x(1) + 1)^2);
fH_ic = (0.79 * log(ReH_ic) - 1.64)^-2;
NuH_ic = PrH_ic * (fH_ic / 8) * (ReH_ic - 1000) / (1 + 12.7...
     * (fH_ic / 8)^0.5 * (PrH_ic^(2/3) - 1));
hH_ic = kH_ic * NuH_ic / x(2);
% Surface temperature (quartic equation)
foulingFactor = 0.0004;
A1 = (mc_ic * CpH_ic / As_ic) * (2 * foulingFactor + ...
    tPlate_ic / kPlate_ic)^2 * (1 - exp(-As_ic * hH_ic / (...
    mc_ic * CpH_ic)) + ...
     As_ic / (mc_ic * CpH_ic * (2 * foulingFactor + ...
        tPlate_ic / kPlate_ic)));
A2 = (mc_ic * CpH_ic / As_ic) * (2 * foulingFactor + ...
    tPlate_ic / kPlate_ic) * (1 + 2 * foulingFactor * hC_ic ...
    + tPlate_ic * hC_ic / kPlate_ic) * ...
     (1 - exp(-As_ic * hH_ic / (mc_ic * CpH_ic)) + As_ic / (...
        mc_ic * CpH_ic * (2 * foulingFactor + tPlate_ic / ...
        kPlate_ic)));
A3 = (mc_ic * CpH_ic / As_ic) * (2 * foulingFactor + ...
    tPlate_ic / kPlate_ic) * (exp(-As_ic * hH_ic / (mc_ic * ...
    CpH_ic)) - 1);
C0 = A3 * T3 - A1 * (Eplate_ic * sigmaB * T0^4 + hC_ic * T0...
    );
C1 = A2 - 1;
C4 = A1 * Eplate_ic * sigmaB;
quarticEquation = [C4 0 0 C1 C0];
for qi = 1:5
    if isnan(quarticEquation(qi)) == 1 || isinf(...
        quarticEquation(qi)) == 1
         PC = 1;
         outputs = 0;
         Ts_rec = 0;
         I = 0;
         S = 0;
         T = 0;
         P = 0;
         return
    end
end
quarticEquationRoots = roots(quarticEquation);
quarticEquationRealRoots = quarticEquationRoots(imag(...
    quarticEquationRoots)==0);
positiveQuarticEquationRealRoots = quarticEquationRealRoots...
    (quarticEquationRealRoots>0);
if length(positiveQuarticEquationRealRoots) > 1
```

```
        if positiveQuarticEquationRealRoots(1) < ...
           positiveQuarticEquationRealRoots(2)
            TsC_ic = positiveQuarticEquationRealRoots(1);
        else
            TsC_ic = positiveQuarticEquationRealRoots(2);
        end
        else
            TsC_ic = positiveQuarticEquationRealRoots;
        end
        if isscalar(TsC_ic) == 0
            PC = 1;
            outputs = 0;
            Ts_rec = 0;
            I = 0;
            S = 0;
            T = 0;
            P = 0;
            return
        end

% ii)   RECUPERATOR
        H_reg = 1;
        t_plate_reg = 0.001;
        kplate_reg = 401;
        bChannel_reg = x(5) * (x(4) + 1) / (2 * x(4));
        nChannels_reg = H_reg / (t_plate_reg + bChannel_reg);
        mc_reg = 2 * mdot / nChannels_reg;
        % Cold side flow
        [CpC_reg, kC_reg, muC_reg, PrC_reg, rhoC_reg] = ...
           getConvectionProperties((T7 + T8) / 2, propertiesArray);
        ReC_reg = 4 * mc_reg * x(4) / (muC_reg * x(5) * (x(4) + 1)...
           ^2);
        fC_reg = (0.79 * log(ReC_reg) - 1.64)^-2;
        NuC_reg = PrC_reg * (fC_reg / 8) * (ReC_reg - 1000) / (1 + ...
           12.7 * (fC_reg / 8)^0.5 * (PrC_reg^(2/3) - 1));
        hC_reg = kC_reg * NuC_reg / x(5);
        % Hot side flow
        [CpH_reg, kH_reg, muH_reg, PrH_reg, rhoH_reg] = ...
           getConvectionProperties((T15 + T16) / 2, propertiesArray...
           );
        ReH_reg = 4 * mc_reg * x(4) / (muH_reg * x(5) * (x(4) + 1)...
           ^2);
        fH_reg = (0.79 * log(ReH_reg) - 1.64)^-2;
        NuH_reg = PrH_reg * (fH_reg / 8) * (ReH_reg - 1000) / (1 + ...
           12.7 * (fH_reg / 8)^0.5 * (PrH_reg^(2/3) - 1));
        hH_reg = kH_reg * NuH_reg / x(5);
        % Effectiveness (e-NTU)
        Aplate_reg = x(5) * x(6) * (x(4) + 1) * (1 + 1 / (x(4)));
        U_reg = (1 / hC_reg + 2 * foulingFactor + t_plate_reg / ...
           kplate_reg + 1 / hH_reg)^-1;
        NTU = (U_reg * Aplate_reg / (mc_reg * CpC_reg));
        C_reg = CpC_reg / CpH_reg;
        eff_reg = (1 - exp(-NTU * (1 - C_reg))) / (1 - C_reg * exp...
           (-NTU * (1 - C_reg)));
        if eff_reg > 1
```

```
            PC = 1;
            outputs = 0;
            Ts_rec = 0;
            I = 0;
            S = 0;
            T = 0;
            P = 0;
            return
        end

% iii)  RECEIVER
        [Cp_rec, k_rec, mu_rec, Pr_rec, rho_rec] = ...
            getConvectionProperties((T9 + T10) / 2, propertiesArray)...
            ;
        d_rec = sqrt(x(8) * x(9) * (x(7) + 1) / (4 * pi));
        a_rec = x(8)*(x(7)+1)/2;
        if construction == 1
            Nrec = ((sqrt(3)-1)*d_rec)/(2*a_rec);
        else
            Nrec = ((sqrt(3)-1)*d_rec)/(2*x(8));
        end
        Re_rec = 4 * mdot * x(7) / (mu_rec * x(8) * (x(7) + 1)^2);
        f_rec = (0.79 * log(Re_rec) - 1.64)^-2;
        Qnet_rec = CF(1) + CF(2)*sqrt(d_rec) + CF(3)*d_rec + CF(4)*...
            d_rec^(3/2) + CF(5)*d_rec^2 + CF(6)*d_rec^(5/2) + CF(7)*...
            d_rec^3 + CF(8)*d_rec^(7/2) + CF(9)*d_rec^4 + CF(10)*...
            d_rec^(9/2) + CF(11)*d_rec^5;
        if construction == 1
            Ts_rec = T10 + Qnet_rec / (0.023*x(9)*(x(7)+1)*(1+1/x...
                (7))*k_rec*Pr_rec^0.4*Re_rec^0.8);
        else
            Ts_rec = T10 + Qnet_rec / (0.023*pi*x(9)*k_rec*Pr_rec^0...
                .4*(4*mdot/(mu_rec*pi*x(8)))^0.8);
        end

% iv)   PRESSURE DROPS
        Pdrop23 = 0.004;
        Pdrop34 = fH_ic * (8 * mc_ic^2 * x(1)^2 / (rhoH_ic * (x(1) ...
            + 1)^4)) * (x(3) / x(2)^5);
        Pdrop45 = 0.004;
        Pdrop67 = 0.004;
        Pdrop78 = (fC_reg * (8 * mc_reg^2 * x(4)^2 / (rhoC_reg * (x...
            (4) + 1)^4)) * (x(6) / x(5)^5));
        Pdrop89 = 0.004;
        if construction == 1
            Pdrop910 = f_rec * (8 * mdot^2 * x(7)^2 / (rho_rec * (x...
                (7) + 1)^4)) * (x(9) / x(8)^5);
        else
            Pdrop910 = ((0.79*log(4*mdot/(mu_rec*pi*x(8)))-1.64)...
                ^-2)*(8*mdot^2/(rho_rec*pi^2))*(x(9)/(x(8)^5));
        end
        Pdrop1011 = 0.004;
        Pdrop1213 = 0.004;
        Pdrop1415 = 0.001;
```

```
        Pdrop1516 = fH_reg * (8 * mc_reg^2 * x(4)^2 / (rhoH_reg * (...
            x(4)+1)^4)) * (x(6) / x(5)^5);

% v)    PRESSURE FIELD
        P1 = P0;                            % Comp.1 inlet
        P2 = pressureRatio * P1;            % Comp.1 outlet
        P3 = P2 * (1 - Pdrop23);            % Intercooler hot inlet
        P4 = P3 - Pdrop34;                  % Intercooler hot outlet
        P5 = P4 * (1 - Pdrop45);            % Comp.2 inlet
        P6 = pressureRatio * P5;            % Comp.2 outlet
        P7 = P6 * (1 - Pdrop67);            % Reg. cold inlet
        P8 = P7 - Pdrop78;                  % Reg. cold outlet
        P9 = P8 * (1 - Pdrop89);            % Rec. inlet
        P10 = P9 - Pdrop910;                % Rec. outlet
        P11 = P10 * (1 - Pdrop1011);        % Turb.1 inlet
        P16 = P0;                           % Reg. hot outlet
        P15 = P16 + Pdrop1516;              % Reg. hot inlet
        P14 = P15 * (1 + Pdrop1415);        % Turb.2 outlet
        r = (P11 * (1 - Pdrop1213) / P14)^0.5;
        P12 = P11 / r;                      % Turb.1 outlet
        P13 = r * P14;                      % Turb.2 inlet
        P = [P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 P12 P13 P14 P15 P16...
            ];

% vi)   TEMPERATURE FIELD
        T1 = T0;                                    % Comp.1 inlet
        Kcomp = pressureRatio^((k-1)/k);
        T2 = T1 * (1 + (Kcomp - 1) / eff_comp); % Comp.1 outlet
        T3 = T2 - T_drop;                           % Intercooler hot ...
            inlet
        T4 = T3 - As_ic * hC_ic / (mc_ic * CpH_ic) * ...
             (TsC_ic - T0) - As_ic * Eplate_ic * ...
             sigmaB * (TsC_ic^4 - T0^4) / (mc_ic * CpH_ic); % ...
                Intercooler hot outlet
        Q_ic = mc_ic * CpH_ic * (T3-T4);
        T5 = T4 - T_drop;                           % Comp.2 inlet
        T6 = T5 * (1 + (Kcomp - 1) / eff_comp); % Comp.2 outlet
        T7 = T6 - T_drop;                           % Reg. cold inlet
        T9 = T8 - T_drop;                           % Rec. inlet
        T10 = T9 + Qnet_rec / (mdot * Cp_rec);  % Rec. outlet
        T11 = T10 - T_drop;                         % Turb.1 inlet
        Kturb = (P13 / P14)^((k-1)/k);
        T12 = T11 * (1 - eff_turb * (1 - 1 / Kturb)); % Turb.1 ...
            outlet
        T13 = T12 - T_drop;                         % Turb.2 inlet
        T14 = T13 * (1 - eff_turb * (1 - 1 / Kturb)); % Turb.2 ...
            outlet
        T15 = T14 - T_drop;                         % Reg. hot inlet
        T8 = T7 + eff_reg * (T15 - T7);             % Reg. cold outlet
        T16 = T15 - eff_reg * CpC_reg / CpH_reg * ...
             (T15 - T7);                            % Reg. hot oulet
        T = [T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 T13 T14 T15 T16...
            ];
        for i = 1:length(T)
```

```
            if isnan(T(i)) == 1 || isreal(T(i)) == 0 || ...
                convergenceCounter > 1e4;
                PC = 1;
                outputs = 0;
                Ts_rec = 0;
                I = 0;
                S = 0;
                T = 0;
                P = 0;
                return
            end
        end

error = abs(T9 - T9_0) / T9_0;
end

% vii)  ENTROPY GENERATION
        [Cp_comp1, ~, ~, ~, ~] = getConvectionProperties((T1 + T2) ...
            / 2, propertiesArray);
        Sgen_comp1 = -mdot * Cp_comp1 * log(T1 / T2) + mdot * R * ...
            log(P1 / P2);

        [Cp_duct23, ~, ~, ~, ~] = getConvectionProperties((T2 + T3)...
             / 2, propertiesArray);
        Sgen_duct23 = mdot * Cp_duct23 * log(T3 / T2) - mdot * R * ...
            log(P3 / P2) + mdot * Cp_duct23 * T_drop / T0;

        Sgen_ic = mdot * CpH_ic * log(T4 / T3) - mdot * R * log(P4 ...
            / P3) + mdot * CpH_ic * (T3 - T4) / TsC_ic;

        [Cp_duct45, ~, ~, ~, ~] = getConvectionProperties((T4 + T5)...
             / 2, propertiesArray);
        Sgen_duct45 = mdot * Cp_duct45 * log(T5 / T4) - mdot * R * ...
            log(P5 / P4) + mdot * Cp_duct45 * T_drop / T0;

        [Cp_comp2, ~, ~, ~, ~] = getConvectionProperties((T5 + T6) ...
            / 2, propertiesArray);
        Sgen_comp2 = -mdot * Cp_comp2 * log(T5 / T6) + mdot * R * ...
            log(P5 / P6);

        [Cp_duct67, ~, ~, ~, ~] = getConvectionProperties((T6 + T7)...
             / 2, propertiesArray);
        Sgen_duct67 = mdot * Cp_duct67 * log(T7 / T6) - mdot * R * ...
            log(P7 / P6) + mdot * Cp_duct67 * T_drop / T0;

        T_reg = (T7 + T8 + T15 + T16) / 4;
        [Cp_reg, ~, ~, ~, ~] = getConvectionProperties(T_reg, ...
            propertiesArray);
        Sgen_reg = mdot * Cp_reg * log((T8 * T16) / (T15 * T7) * ((...
            P8 * P16) / (P15 * P7))^((1-k)/k)) + mdot * Cp_reg * ...
            T_drop / T0;

        [Cp_duct89, ~, ~, ~, ~] = getConvectionProperties((T8 + T9)...
             / 2, propertiesArray);
```

```
        Sgen_duct89 = mdot * Cp_duct89 * log(T9 / T8) - mdot * R * ...
           log(P9 / P8) + mdot * Cp_duct89 * T_drop / T0;

        Tstar = Tsun;
        beta = pi / 2;
        D_rec = sqrt(3) * d_rec;
        v_rec = (mu_rec / rho_rec);
        Aa_rec = pi * d_rec^2 / 4;
        Gr_D = 9.81 * beta * (Ts_rec - T0) * D_rec^3 / v_rec^2;
        Qdot0_rec = 1.396 * Gr_D^0.209 * (1 + cos(beta))^0.968 * (...
           Ts_rec / T0)^(-0.317) * (d_rec / ...
            D_rec)^(0.425) * (k_rec * Aa_rec / D_rec) * (Ts_rec - ...
               T0);
        Qstar_rec = Qnet_rec + Qdot0_rec;
        Sgen_rec = -Qstar_rec / Tstar + Qdot0_rec / T0 + mdot * ...
           Cp_rec * log(T10 / T9) - mdot * R * log(P10 / P9);

        [Cp_duct1011, ~, ~, ~, ~] = getConvectionProperties((T10 + ...
           T11) / 2, propertiesArray);
        Sgen_duct1011 = mdot * Cp_duct1011 * log(T11 / T10) - mdot ...
           * R * log(P11 / P10) + mdot * Cp_duct1011 * T_drop / T0;

        [Cp_turb1, ~, ~, ~, ~] = getConvectionProperties((T11 + T12...
           ) / 2, propertiesArray);
        Sgen_turb1 = -mdot * Cp_turb1 * log(T11 / T12) + mdot * R *...
            log(P11 / P12);

        [Cp_duct1213, ~, ~, ~, ~] = getConvectionProperties((T12 + ...
           T13) / 2, propertiesArray);
        Sgen_duct1213 = mdot * Cp_duct1213 * log(T13 / T12) - mdot ...
           * R * log(P13 / P12) + mdot * Cp_duct1213 * T_drop / T0;

        [Cp_turb2, ~, ~, ~, ~] = getConvectionProperties((T13 + T14...
           ) / 2, propertiesArray);
        Sgen_turb2 = -mdot * Cp_turb2 * log(T13 / T14) + mdot * R *...
            log(P13 / P14);

        [Cp_duct1415, ~, ~, ~, ~] = getConvectionProperties((T14 + ...
           T15) / 2, propertiesArray);
        Sgen_duct1415 = mdot * Cp_duct1415 * log(T15 / T14) - mdot ...
           * R * log(P15 / P14) + mdot * Cp_duct1415 * T_drop / T0;

        Sgen_parts = Sgen_comp1 + Sgen_ic + Sgen_comp2 + Sgen_reg +...
            Sgen_rec + Sgen_turb1 + Sgen_turb2;

        Sgen_ducts = Sgen_duct23 + Sgen_duct45 + Sgen_duct67 + ...
           Sgen_duct89 + Sgen_duct1011 + Sgen_duct1213 + ...
           Sgen_duct1415;

% viii) IRREVERSIBILITIES
        I_comp = T0 * (Sgen_comp1 + Sgen_comp2);
        I_ic = T0 * Sgen_ic;
        I_reg = T0 * Sgen_reg;
        I_rec = T0 * Sgen_rec;
        I_turb = T0 * (Sgen_turb1 + Sgen_turb2);
```

```
        [Cp, ~, ~, ~, ~] = getConvectionProperties((T1 + T16) / 2, ...
            propertiesArray);
        I_internal = (I_comp + I_ic + I_reg + I_rec + I_turb) / ...
            1000;
        I_external = -(mdot * Cp * (T1 - T16) - mdot * T0 * Cp * ...
            log(T1 / T16)) / 1000;

% ix)   OUTPUT PARAMETERS
        Qnet = Qnet_rec / 1000;
        Wc = mdot * Cp_comp1 * (T2 - T1) + mdot * Cp_comp2 * (T6 - ...
            T5);
        Wt = mdot * Cp_turb1 * (T11 - T12) + mdot * Cp_turb2 * (T13...
             - T14);
        Wnet1 = (Wt - Wc) / 1000;
        eff_th = Wnet1 / Qnet;
        Sgen = Sgen_parts + Sgen_ducts;
        W2 = -T0 * Sgen + (1 - (T0 / Tstar)) * Qstar_rec + mdot * ...
            Cp * (T1 - T16) + mdot * T0 * Cp * log(T16 / T1);
        Wnet2 = W2 / 1000;
        if isreal(W2) == 0
            PC = 1;
            outputs = 0;
            Ts_rec = 0;
            I = 0;
            S = 0;
            T = 0;
            P = 0;
            return
        end
        eff_ex = Wnet2 / Qnet;
        outputs = [mdot, Qnet, Wnet1, Wnet2, eff_th, eff_ex,...
            mc_ic, Ra_ic, fH_ic, ReH_ic, Q_ic, TsC_ic,...
            mc_reg, fC_reg, ReC_reg, fH_reg, ReH_reg, NTU, eff_reg,...
                ...
            Qnet_rec/1000, Qdot0_rec/1000, Ts_rec, f_rec, Re_rec, ...
                Nrec];
        I       = [I_internal, I_external, I_comp, I_turb, I_rec, ...
            I_reg, I_ic];
        S       = [Sgen, Sgen_comp1, Sgen_comp2, Sgen_turb1, ...
            Sgen_turb2, Sgen_rec, Sgen_reg, Sgen_ic];
end
```

B.3.3 combinedFunction.m

```
function [PC, outputs, Ts, I, S, T, P] = combinedFunction(x, ...
    nTurbine, pressureRatio, propertiesArray, microturbineArray, CF1...
    , CF2, construction)
x = [x(1)/100 x(2)/1000 x(3) x(4) x(5)/100 x(6) x(7) x(8)/100 x(9) ...
    x(10) x(11)/100 x(12)];
if construction == 2
     x(7) = 1;
     x(10) = 1;
end
[mdot, eff_turb, eff_comp] = getMicroturbineProperties(nTurbine, ...
    pressureRatio, microturbineArray);

%% 1) CONSTANTS & ASSUMPTIONS
      % Ambient air
        R = 287;
        k = 1.4;
        T0 = 300;
        P0 = 101325;
      % Assumed temperatures
        T3 = 375;                       % intercooler hot inlet
        T4 = 400;                       % intercooler hot outlet
        TsC_ic = 375;                   % intercooler surface
        T7 = 500;                       % recuperator cold inlet
        T8 = 750;                       % recuperator cold outlet
        T17 = 1000;                     % recuperator hot inlet
        T18 = 750;                      % recuperator hot outlet
        T9 = 750;                       % receiver inlet
        T10 = 1000;                     % receiver outlet
        T13 = 750;                      % reheater inlet
        T14 = 1000;                     % reheater outlet
        T_drop = 2;
        Tsun = 2470;

%% 2) THERMODYNAMIC ANALYSIS
error = 1;
convergenceCounter = 0;
while error > 1e-3
T9_0 = T9;
PC = 0;
convergenceCounter = convergenceCounter + 1;

% i)    INTERCOOLER
        W_ic = 1;
        tPlate_ic = 0.001;
        kPlate_ic = 401;
        Eplate_ic = 0.03;
        sigmaB = 5.67e-8;
        % Cold side heat transfer
        [~, kC_ic, muC_ic, PrC_ic, rhoC_ic] = ...
           getConvectionProperties((TsC_ic + T0) / 2, ...
           propertiesArray);
        Beta_ic = ((TsC_ic + T0) / 2)^-1;
```

```
aChannel_ic = x(2) * (x(1) + 1) / 2;
bChannel_ic = x(2) * (x(1) + 1) / (2 * x(1));
nChannels_ic = W_ic / (tPlate_ic + aChannel_ic);
As_ic = 2 * (aChannel_ic + bChannel_ic) * x(3);
mc_ic = 2 * mdot / nChannels_ic;
Ra_ic = 9.81 * Beta_ic * (TsC_ic - T0) * PrC_ic * ...
    aChannel_ic^3 / (muC_ic / rhoC_ic)^2;
NuC_ic = (576/(Ra_ic * x(1))^2 + 2.873/(Ra_ic * x(1))^0.5)...
    ^-0.5;
hC_ic = kC_ic * NuC_ic / aChannel_ic;
% Hot side flow (internal forced convection)
[CpH_ic, kH_ic, muH_ic, PrH_ic, rhoH_ic] = ...
    getConvectionProperties((T3 + T4) / 2, propertiesArray);
ReH_ic = 4 * mc_ic * x(1) / (muH_ic * x(2) * (x(1) + 1)^2);
fH_ic = (0.79 * log(ReH_ic) - 1.64)^-2;
NuH_ic = PrH_ic * (fH_ic / 8) * (ReH_ic - 1000) / (1 + 12.7...
     * (fH_ic / 8)^0.5 * (PrH_ic^(2/3) - 1));
hH_ic = kH_ic * NuH_ic / x(2);
% Surface temperature (quartic equation)
foulingFactor = 0.0004;
A1 = (mc_ic * CpH_ic / As_ic) * (2 * foulingFactor + ...
    tPlate_ic / kPlate_ic)^2 * (1 - exp(-As_ic * hH_ic / (...
    mc_ic * CpH_ic)) + ...
     As_ic / (mc_ic * CpH_ic * (2 * foulingFactor + ...
         tPlate_ic / kPlate_ic)));
A2 = (mc_ic * CpH_ic / As_ic) * (2 * foulingFactor + ...
    tPlate_ic / kPlate_ic) * (1 + 2 * foulingFactor * hC_ic ...
    + tPlate_ic * hC_ic / kPlate_ic) * ...
     (1 - exp(-As_ic * hH_ic / (mc_ic * CpH_ic)) + As_ic / (...
         mc_ic * CpH_ic * (2 * foulingFactor + tPlate_ic / ...
         kPlate_ic)));
A3 = (mc_ic * CpH_ic / As_ic) * (2 * foulingFactor + ...
    tPlate_ic / kPlate_ic) * (exp(-As_ic * hH_ic / (mc_ic * ...
    CpH_ic)) - 1);
C0 = A3 * T3 - A1 * (Eplate_ic * sigmaB * T0^4 + hC_ic * T0...
    );
C1 = A2 - 1;
C4 = A1 * Eplate_ic * sigmaB;
quarticEquation = [C4 0 0 C1 C0];
for qi = 1:5
    if isnan(quarticEquation(qi)) == 1 || isinf(...
        quarticEquation(qi)) == 1
        PC = 1;
        outputs = 0;
        Ts = 0;
        I = 0;
        S = 0;
        T = 0;
        P = 0;
        return
    end
end
quarticEquationRoots = roots(quarticEquation);
quarticEquationRealRoots = quarticEquationRoots(imag(...
    quarticEquationRoots)==0);
```

```
        positiveQuarticEquationRealRoots = quarticEquationRealRoots...
            (quarticEquationRealRoots>0);
        if length(positiveQuarticEquationRealRoots) > 1
        if positiveQuarticEquationRealRoots(1) < ...
            positiveQuarticEquationRealRoots(2)
            TsC_ic = positiveQuarticEquationRealRoots(1);
        else
            TsC_ic = positiveQuarticEquationRealRoots(2);
        end
        else
            TsC_ic = positiveQuarticEquationRealRoots;
        end
        if isscalar(TsC_ic) == 0
            PC = 1;
            outputs = 0;
            Ts = 0;
            I = 0;
            S = 0;
            T = 0;
            P = 0;
            return
        end

% ii)   RECUPERATOR
        H_reg = 1;
        t_plate_reg = 0.001;
        kplate_reg = 401;
        bChannel_reg = x(5) * (x(4) + 1) / (2 * x(4));
        nChannels_reg = H_reg / (t_plate_reg + bChannel_reg);
        mc_reg = 2 * mdot / nChannels_reg;
        % Cold side flow
        [CpC_reg, kC_reg, muC_reg, PrC_reg, rhoC_reg] = ...
            getConvectionProperties((T7 + T8) / 2, propertiesArray);
        ReC_reg = 4 * mc_reg * x(4) / (muC_reg * x(5) * (x(4) + 1)...
            ^2);
        fC_reg = (0.79 * log(ReC_reg) - 1.64)^-2;
        NuC_reg = PrC_reg * (fC_reg / 8) * (ReC_reg - 1000) / (1 + ...
            12.7 * (fC_reg / 8)^0.5 * (PrC_reg^(2/3) - 1));
        hC_reg = kC_reg * NuC_reg / x(5);
        % Hot side flow
        [CpH_reg, kH_reg, muH_reg, PrH_reg, rhoH_reg] = ...
            getConvectionProperties((T17 + T18) / 2, propertiesArray...
            );
        ReH_reg = 4 * mc_reg * x(4) / (muH_reg * x(5) * (x(4) + 1)...
            ^2);
        fH_reg = (0.79 * log(ReH_reg) - 1.64)^-2;
        NuH_reg = PrH_reg * (fH_reg / 8) * (ReH_reg - 1000) / (1 + ...
            12.7 * (fH_reg / 8)^0.5 * (PrH_reg^(2/3) - 1));
        hH_reg = kH_reg * NuH_reg / x(5);
        % Effectiveness (e-NTU)
        Aplate_reg = x(5) * x(6) * (x(4) + 1) * (1 + 1 / (x(4)));
        U_reg = (1 / hC_reg + 2 * foulingFactor + t_plate_reg / ...
            kplate_reg + 1 / hH_reg)^-1;
        NTU = (U_reg * Aplate_reg / (mc_reg * CpC_reg));
        C_reg = CpC_reg / CpH_reg;
```

```
        eff_reg = (1 - exp(-NTU * (1 - C_reg))) / (1 - C_reg * exp...
            (-NTU * (1 - C_reg)));
        if eff_reg > 1
            PC = 1;
            outputs = 0;
            Ts = 0;
            I = 0;
            S = 0;
            T = 0;
            P = 0;
            return
        end

% iii)  RECEIVER
        [Cp_rec, k_rec, mu_rec, Pr_rec, rho_rec] = ...
            getConvectionProperties((T9 + T10) / 2, propertiesArray)...
            ;
        a_rec = x(8)*(x(7)+1)/2;
        d_rec = sqrt(x(8) * x(9) * (x(7) + 1) / (4 * pi));
        if construction == 1
            Nrec = ((sqrt(3)-1)*d_rec)/(2*a_rec);
        else
            Nrec = (sqrt(3)-1)*d_rec/(2*x(8));
        end
        Re_rec = 4 * mdot * x(7) / (mu_rec * x(8) * (x(7) + 1)^2);
        f_rec = (0.79 * log(Re_rec) - 1.64)^-2;
        Qnet_rec = CF1(1) + CF1(2)*sqrt(d_rec) + CF1(3)*d_rec + CF1...
            (4)*d_rec^(3/2) + CF1(5)*d_rec^2 + CF1(6)*d_rec^(5/2) + ...
            CF1(7)*d_rec^3 + CF1(8)*d_rec^(7/2) + CF1(9)*d_rec^4 + ...
            CF1(10)*d_rec^(9/2) + CF1(11)*d_rec^5;
        if construction == 1
            Ts_rec = T10 + Qnet_rec / (0.023*x(9)*(x(7)+1)*(1+1/x...
                (7))*k_rec*Pr_rec^0.4*Re_rec^0.8);
        else
            Ts_rec = T10 + Qnet_rec / (0.023*pi*x(9)*k_rec*Pr_rec^0...
                .4*(4*mdot/(mu_rec*pi*x(8)))^0.8);
        end

% iv)   REHEATER
        [Cp_reh, k_reh, mu_reh, Pr_reh, rho_reh] = ...
            getConvectionProperties((T13 + T14) / 2, propertiesArray...
            );
        a_reh = x(11)*(x(10)+1)/2;
        d_reh = sqrt(x(11) * x(12) * (x(10) + 1) / (4 * pi));
        if construction == 1
            Nreh = ((sqrt(3)-1)*d_reh)/(2*a_reh);
        else
            Nreh = ((sqrt(3)-1)*d_reh)/(2*x(11));
        end
        Re_reh = 4 * mdot * x(10) / (mu_reh * x(11) * (x(10) + 1)...
            ^2);
        f_reh = (0.79 * log(Re_reh) - 1.64)^-2;
        Qnet_reh = CF2(1) + CF2(2)*sqrt(d_reh) + CF2(3)*d_reh + CF2...
            (4)*d_reh^(3/2) + CF2(5)*d_reh^2 + CF2(6)*d_reh^(5/2) + ...
```

```
            CF2(7)*d_reh^3 + CF2(8)*d_reh^(7/2) + CF2(9)*d_reh^4 + ...
            CF2(10)*d_reh^(9/2) + CF2(11)*d_reh^5;
        if construction == 1
            Ts_reh = T14 + Qnet_reh / (0.023*x(12)*(x(10)+1)*(1+1/x...
                (10))*k_reh*Pr_reh^0.4*Re_reh^0.8);
        else
            Ts_reh = T14 + Qnet_reh / (0.023*pi*x(12)*k_reh*Pr_reh...
                ^0.4*(4*mdot/(mu_reh*pi*x(11)))^0.8);
        end
        Q_abs = Qnet_rec + Qnet_reh;

% v)    PRESSURE DROPS
        Pdrop23 = 0.004;
        Pdrop34 = fH_ic * (8 * mc_ic^2 * x(1)^2 / (rhoH_ic * (x(1) ...
            + 1)^4)) * (x(3) / x(2)^5);
        Pdrop45 = 0.004;
        Pdrop67 = 0.004;
        Pdrop78 = (fC_reg * (8 * mc_reg^2 * x(4)^2 / (rhoC_reg * (x...
            (4) + 1)^4)) * (x(6) / x(5)^5));
        Pdrop89 = 0.004;
        if construction == 1
            Pdrop910 = f_rec * (8 * mdot^2 * x(7)^2 / (rho_rec * (x...
                (7) + 1)^4)) * (x(9) / x(8)^5);
        else
            Pdrop910 = ((0.79*log(4*mdot/(mu_rec*pi*x(8)))-1.64)...
                ^-2)*(8*mdot^2/(rho_rec*pi^2))*(x(9)/(x(8)^5));
        end
        Pdrop1011 = 0.004;
        Pdrop1213 = 0.004;
        if construction == 1
            Pdrop1314 = f_reh * (8 * mdot^2 * x(10)^2 / (rho_reh * ...
                (x(10) + 1)^4)) * (x(12) / x(11)^5);
        else
            Pdrop1314 = ((0.79*log(4*mdot/(mu_reh*pi*x(11)))-1.64)...
                ^-2)*(8*mdot^2/(rho_reh*pi^2))*(x(12)/(x(11)^5));
        end
        Pdrop1415 = 0.004;
        Pdrop1617 = 0.001;
        Pdrop1718 = fH_reg * (8 * mc_reg^2 * x(4)^2 / (rhoH_reg * (...
            x(4)+1)^4)) * (x(6) / x(5)^5);

% vi)   PRESSURE FIELD
        P1 = P0;                            % Comp.1 inlet
        P2 = pressureRatio * P1;            % Comp.1 outlet
        P3 = P2 * (1 - Pdrop23);            % Intercooler hot inlet
        P4 = P3 - Pdrop34;                  % Intercooler hot outlet
        P5 = P4 * (1 - Pdrop45);            % Comp.2 inlet
        P6 = pressureRatio * P5;            % Comp.2 outlet
        P7 = P6 * (1 - Pdrop67);            % Reg. cold inlet
        P8 = P7 - Pdrop78;                  % Reg. cold outlet
        P9 = P8 * (1 - Pdrop89);            % Rec. inlet
        P10 = P9 - Pdrop910;                % Rec. outlet
        P11 = P10 * (1 - Pdrop1011);        % Turb.1 inlet
        P18 = P0;                           % Reg. hot outlet
        P17 = P18 + Pdrop1718;              % Reg. hot inlet
```

```
        P16 = P17 * (1 + Pdrop1617);      % Turb.2 outlet
        P12 = ((P16 * P11) / ((1 - Pdrop1415) * (1 - Pdrop1213)) + ...
            ...
            Pdrop1314^2/(4 * (1 - Pdrop1213)^2))^(1/2) + Pdrop1314 ...
                / ...
            (2 * (1 - Pdrop1213));        % Turb. 1 outlet
        P13 = P12 * (1 - Pdrop1213);      % Reh. inlet
        P14 = P13 - Pdrop1314;            % Reh. outlet
        P15 = P14 * (1 - Pdrop1415);      % Turb.2 inlet
        P = [P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 P12 P13 P14 P15 P16...
            P17 P18];

% vii)  TEMPERATURE FIELD
        T1 = T0;                                  % Comp.1 inlet
        Kcomp = pressureRatio^((k-1)/k);
        T2 = T1 * (1 + (Kcomp - 1) / eff_comp);   % Comp.1 outlet
        T3 = T2 - T_drop;                         % Intercooler hot...
            inlet
        T4 = T3 - As_ic * hC_ic / (mc_ic * CpH_ic) * ...
            (TsC_ic - T0) - As_ic * Eplate_ic * ...
            sigmaB * (TsC_ic^4 - T0^4) / (mc_ic * CpH_ic); % ...
                Intercooler hot outlet
        Q_ic = mc_ic * CpH_ic * (T3-T4);
        T5 = T4 - T_drop;                         % Comp.2 inlet
        T6 = T5 * (1 + (Kcomp - 1) / eff_comp);   % Comp.2 outlet
        T7 = T6 - T_drop;                         % Reg. cold inlet
        T9 = T8 - T_drop;                         % Rec. inlet
        T10 = T9 + Qnet_rec / (mdot * Cp_rec);    % Rec. outlet
        T11 = T10 - T_drop;                       % Turb.1 inlet
        Kturb = (P15 / P16)^((k-1)/k);
        T12 = T11 * (1 - eff_turb * (1 - 1 / Kturb)); % Turb.1 ...
            outlet
        T13 = T12 - T_drop;                       % Reh. inlet
        T14 = T13 + Qnet_reh / (mdot * Cp_reh);   % Reh. outlet
        T15 = T14 - T_drop;                       % Turb.2 inlet
        T16 = T15 * (1 - eff_turb * (1 - 1 / Kturb)); % Turb.2 ...
            outlet
        T17 = T16 - T_drop;                       % Reg. hot inlet
        T8 = T7 + eff_reg * (T17 - T7);           % Reg. cold ...
            outlet
        T18 = T17 - eff_reg * CpC_reg / CpH_reg * ...
            (T17 - T7);                           % Reg. hot oulet
        T = [T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 T13 T14 T15 T16...
            T17 T18];
        for i = 1:length(T)
            if isnan(T(i)) == 1 || isreal(T(i)) == 0 || ...
                convergenceCounter > 1e2;
                PC = 1;
                outputs = 0;
                Ts = 0;
                I = 0;
                S = 0;
                T = 0;
                P = 0;
                return
```

```
            end
        end

error = abs(T9 - T9_0) / T9_0;
end

% viii) ENTROPY GENERATION
        [Cp_comp1, ~, ~, ~, ~] = getConvectionProperties((T1 + T2) ...
           / 2, propertiesArray);
        Sgen_comp1 = -mdot * Cp_comp1 * log(T1 / T2) + mdot * R * ...
           log(P1 / P2);

        [Cp_duct23, ~, ~, ~, ~] = getConvectionProperties((T2 + T3)...
            / 2, propertiesArray);
        Sgen_duct23 = mdot * Cp_duct23 * log(T3 / T2) - mdot * R * ...
           log(P3 / P2) + mdot * Cp_duct23 * T_drop / T0;

        Sgen_ic = mdot * CpH_ic * log(T4 / T3) - mdot * R * log(P4 ...
           / P3) + mdot * CpH_ic * (T3 - T4) / TsC_ic;

        [Cp_duct45, ~, ~, ~, ~] = getConvectionProperties((T4 + T5)...
            / 2, propertiesArray);
        Sgen_duct45 = mdot * Cp_duct45 * log(T5 / T4) - mdot * R * ...
           log(P5 / P4) + mdot * Cp_duct45 * T_drop / T0;

        [Cp_comp2, ~, ~, ~, ~] = getConvectionProperties((T5 + T6) ...
           / 2, propertiesArray);
        Sgen_comp2 = -mdot * Cp_comp2 * log(T5 / T6) + mdot * R * ...
           log(P5 / P6);

        [Cp_duct67, ~, ~, ~, ~] = getConvectionProperties((T6 + T7)...
            / 2, propertiesArray);
        Sgen_duct67 = mdot * Cp_duct67 * log(T7 / T6) - mdot * R * ...
           log(P7 / P6) + mdot * Cp_duct67 * T_drop / T0;

        T_reg = (T7 + T8 + T17 + T18) / 4;
        [Cp_reg, ~, ~, ~, ~] = getConvectionProperties(T_reg, ...
           propertiesArray);
        Sgen_reg = mdot * Cp_reg * log((T8 * T18) / (T17 * T7) * ((...
           P8 * P18) / (P17 * P7))^((1-k)/k)) + mdot * Cp_reg * ...
           T_drop / T0;

        [Cp_duct89, ~, ~, ~, ~] = getConvectionProperties((T8 + T9)...
            / 2, propertiesArray);
        Sgen_duct89 = mdot * Cp_duct89 * log(T9 / T8) - mdot * R * ...
           log(P9 / P8) + mdot * Cp_duct89 * T_drop / T0;

        Tstar = Tsun;
        beta = pi / 2;
        D_rec = sqrt(3) * d_rec;
        v_rec = (mu_rec / rho_rec);
        Aa_rec = pi * d_rec^2 / 4;
        Gr_D = 9.81 * beta * (Ts_rec - T0) * D_rec^3 / v_rec^2;
        Qdot0_rec = 1.396 * Gr_D^0.209 * (1 + cos(beta))^0.968 * (...
           Ts_rec / T0)^(-0.317) * (d_rec / ...
```

```
    D_rec)^(0.425) * (k_rec * Aa_rec / D_rec) * (Ts_rec - ...
       T0);
Qstar_rec = Qnet_rec + Qdot0_rec;
Sgen_rec = -Qstar_rec / Tstar + Qdot0_rec / T0 + mdot * ...
   Cp_rec * log(T10 / T9) - mdot * R * log(P10 / P9);

[Cp_duct1011, ~, ~, ~, ~] = getConvectionProperties((T10 + ...
   T11) / 2, propertiesArray);
Sgen_duct1011 = mdot * Cp_duct1011 * log(T11 / T10) - mdot ...
   * R * log(P11 / P10) + mdot * Cp_duct1011 * T_drop / T0;

[Cp_turb1, ~, ~, ~, ~] = getConvectionProperties((T11 + T12...
   ) / 2, propertiesArray);
Sgen_turb1 = -mdot * Cp_turb1 * log(T11 / T12) + mdot * R *...
    log(P11 / P12);

[Cp_duct1213, ~, ~, ~, ~] = getConvectionProperties((T12 + ...
   T13) / 2, propertiesArray);
Sgen_duct1213 = mdot * Cp_duct1213 * log(T13 / T12) - mdot ...
   * R * log(P13 / P12) + mdot * Cp_duct1213 * T_drop / T0;

D_reh = sqrt(3) * d_reh;
v_reh = (mu_reh/rho_reh);
Aa_reh = pi * d_reh^2 / 4;
Gr_Dreh = 9.81 * beta * (Ts_reh - T0) * D_reh^3 / v_reh^2;
Qdot0_reh = 1.396 * Gr_Dreh^0.209 * (1 + cos(beta))^0.968 *...
    (Ts_reh / T0)^(-0.317) * ...
    (d_reh / D_reh)^(0.425) * (k_reh * Aa_reh / D_reh) * (...
       Ts_reh - T0);
Qstar_reh = Qnet_reh + Qdot0_reh;
Sgen_reh = -Qstar_reh / Tstar + Qdot0_reh / T0 + mdot * ...
   Cp_reh * log(T14 / T13) - mdot * R * log(P14 / P13);

[Cp_duct1415, ~, ~, ~, ~] = getConvectionProperties((T14 + ...
   T15) / 2, propertiesArray);
Sgen_duct1415 = mdot * Cp_duct1415 * log(T15 / T14) - mdot ...
   * R * log(P15 / P14) + mdot * Cp_duct1415 * T_drop / T0;

[Cp_turb2, ~, ~, ~, ~] = getConvectionProperties((T15 + T16...
   ) / 2, propertiesArray);
Sgen_turb2 = -mdot * Cp_turb2 * log(T15 / T16) + mdot * R *...
    log(P15 / P16);

[Cp_duct1617, ~, ~, ~, ~] = getConvectionProperties((T16 + ...
   T17) / 2, propertiesArray);
Sgen_duct1617 = mdot * Cp_duct1617 * log(T17 / T16) - mdot ...
   * R * log(P17 / P16) + mdot * Cp_duct1617 * T_drop / T0;

Sgen_parts = Sgen_comp1 + Sgen_ic + Sgen_comp2 + Sgen_reg +...
    Sgen_rec + Sgen_turb1 + Sgen_reh + Sgen_turb2;

Sgen_ducts = Sgen_duct23 + Sgen_duct45 + Sgen_duct67 + ...
   Sgen_duct89 + Sgen_duct1011 + Sgen_duct1213 + ...
   Sgen_duct1415 + Sgen_duct1617;
```

```
% ix)    IRREVERSIBILITIES
        I_comp = T0 * (Sgen_comp1 + Sgen_comp2);
        I_ic = T0 * Sgen_ic;
        I_reg = T0 * Sgen_reg;
        I_rec = T0 * Sgen_rec;
        I_reh = T0 * Sgen_reh;
        I_turb = T0 * (Sgen_turb1 + Sgen_turb2);
        I_internal = (I_comp + I_ic + I_reg + I_rec + I_turb + ...
            I_reh) / 1000;
        [Cp, ~, ~, ~, ~] = getConvectionProperties((T1 + T18) / 2, ...
            propertiesArray);
        I_external = -(mdot * Cp * (T1 - T18) - mdot * T0 * Cp * ...
            log(T1 / T18)) / 1000;

% x)     OUTPUT PARAMETERS
        Qnet = Q_abs / 1000;
        Wc = mdot * Cp_comp1 * (T2 - T1) + mdot * Cp_comp2 * (T6 - ...
            T5);
        Wt = mdot * Cp_turb1 * (T11 - T12) + mdot * Cp_turb2 * (T15...
             - T16);
        Wnet1 = (Wt - Wc) / 1000;
        eff_th = Wnet1 / Qnet;
        Sgen = Sgen_parts + Sgen_ducts;
        Qstar = Qstar_rec + Qstar_reh;
        W2 = -T0 * Sgen + (1 - (T0 / Tstar)) * Qstar + mdot * Cp * ...
            (T1 - T18) + mdot * T0 * Cp * log(T18 / T1);
        Wnet2 = W2 / 1000;
        if isreal(W2) == 0
            PC = 1;
            outputs = 0;
            Ts = 0;
            I = 0;
            S = 0;
            T = 0;
            P = 0;
            return
        end
        eff_ex = Wnet2 / Qnet;
        outputs = [mdot, Qnet, Wnet1, Wnet2, eff_th, eff_ex,...
            mc_ic, Ra_ic, fH_ic, ReH_ic, Q_ic, TsC_ic,...
            mc_reg, fC_reg, ReC_reg, fH_reg, ReH_reg, NTU, eff_reg,...
                ...
            Qnet_rec/1000, Qdot0_rec/1000, Ts_rec, f_rec, Re_rec, ...
                Nrec,...
            Qnet_reh/1000, Qdot0_reh/1000, Ts_reh, f_reh, Re_reh, ...
                Nreh];
        Ts      = [Ts_rec, Ts_reh];
        I       = [I_internal, I_external, I_comp, I_turb, I_rec, ...
            I_reh, I_reg, I_ic];
        S       = [Sgen, Sgen_comp1, Sgen_comp2, Sgen_turb1, ...
            Sgen_turb2, Sgen_rec, Sgen_reh, Sgen_reg, Sgen_ic];
end
```

B.4 Optimisation scripts

B.4.1 LFOPC.m

```
function [PC, Xstar] = LFOPC(X0, eX, eg, kmax, nTurbine, ...
   pressureRatio, propertiesArray, microturbineArray, CF1, CF2, ...
   dishApertureArea1, dishApertureArea2, dc1, dc2, construction)
%% PHASE 0
    PHASE = 0;  dTn = 0;
    [PC, XstarMu0, ~, dTf] = LFOP(X0, eX, eg, kmax, PHASE, dTn, ...
       nTurbine, pressureRatio, propertiesArray, microturbineArray,...
        CF1, CF2, dishApertureArea1, dishApertureArea2, dc1, dc2, ...
       construction);
        if PC == 1
            Xstar = 0;
            return
        end
    Xstar = XstarMu0;
    [PC, c] = reheatConstraints(XstarMu0, nTurbine, pressureRatio, ...
       propertiesArray, microturbineArray, CF1, CF2, ...
       dishApertureArea1, dishApertureArea2, dc1, dc2, construction...
       );
        if PC == 1
            Xstar = 0;
            return
        end
    na = 0;     i1 = 0;     Ia = 0;
    for i2 = 1:length(c)
        if c(i2) > 0
            i1 = i1 + 1;
            Ia(i1) = c(i2);
        end
    end
    for i3 = 1:length(Ia)
        if Ia(i3) > 0
            na = na + 1;
        end
    end
    if na == 0 %(unconstrained minimum found)
        return
    end
%% PHASE 2
    [PC, GF] = gradf(XstarMu0,  nTurbine, pressureRatio, ...
       propertiesArray, microturbineArray, CF1, CF2, construction);
        if PC == 1
            Xstar = 0;
            return
        end
    if na == length(XstarMu0) || norm(GF) < eg*1e2
    PHASE = 2;
    X0 = XstarMu0;  dTn = dTf;
    [PC, Xstar, ~, ~] = LFOP(X0, eX, eg, kmax, PHASE, dTn, nTurbine...
       , pressureRatio, propertiesArray, microturbineArray, CF1, ...
```

```
        CF2, dishApertureArea1, dishApertureArea2, dc1, dc2, ...
        construction);
         if PC == 1
             Xstar = 0;
             return
         end
    return
    end
%% PHASE 1-0
    PHASE = 1;
    X0 = XstarMu0;  dTn = dTf;
    [PC, XstarMu0, ~, dTf] = LFOP(X0, eX, eg, kmax, PHASE, dTn, ...
        nTurbine, pressureRatio, propertiesArray, microturbineArray,...
         CF1, CF2, dishApertureArea1, dishApertureArea2, dc1, dc2, ...
        construction);
    Xstar = XstarMu0;
         if PC == 1
             Xstar = 0;
             return
         end
    [PC, c] = reheatConstraints(XstarMu0, nTurbine, pressureRatio, ...
        propertiesArray, microturbineArray, CF1, CF2, ...
        dishApertureArea1, dishApertureArea2, dc1, dc2, construction...
        );
         if PC == 1
             Xstar = 0;
             return
         end
    na = 0;      i1 = 0;      Ia = 0;
    for i2 = 1:length(c)
         if c(i2) > 0
             i1 = i1 + 1;
             Ia(i1) = c(i2);
         end
    end
    for i3 = 1:length(Ia)
         if Ia(i3) > 0
             na = na + 1;
         end
    end
    if na == 0 %(unconstrained minimum found)
         return
    end
%% PHASE 1-2
    PHASE = 2;
    X0 = XstarMu0;  dTn = dTf;
    [PC, Xstar, ~, ~] = LFOP(X0, eX, eg, kmax, PHASE, dTn, nTurbine...
        , pressureRatio, propertiesArray, microturbineArray, CF1, ...
        CF2, dishApertureArea1, dishApertureArea2, dc1, dc2, ...
        construction);
         if PC == 1
             Xstar = 0;
             return
         end
end
```

B.4.2 LFOP.m

```
function [PC, Xstar, X, dT] = LFOP(X0, eX, eg, kmax, PHASE, dTn, ...
    nTurbine, pressureRatio, propertiesArray, microturbineArray, CF1...
    , CF2, dishApertureArea1, dishApertureArea2, dc1, dc2, ...
    construction)

%% Initialisation
     delta    = 1;         % maximum step size (default)
     ix       = 0;
     ixm      = 2;
     is       = 0;
     ism      = 3;
     id       = 0;
     idm      = 5;
     p        = 1;
     k        = 1;
     deltaT   = 0.001;
     X(k,:)     = X0;

     % Penalty values
     if PHASE == 0
         Xmu      = 1e2;  % initial penalty value (default is 1e2)
         gamma    = 1;
     elseif PHASE == 1
         Xmu      = 1e4;  % maximum penalty value (default is 1e4)
         gamma    = 1;
     elseif PHASE == 2
         Xmu      = 1e4;
         gamma    = 0;
     end

     % Penalty gradient
     [PC, c]   = reheatConstraints(X(k,:), nTurbine, pressureRatio, ...
        propertiesArray, microturbineArray, CF1, CF2, ...
        dishApertureArea1, dishApertureArea2, dc1, dc2, construction...
        );
         if PC == 1
             Xstar = 0;
             X = 0;
             dT = 0;
             return
         end
     [PC, GC] = gradc(X(k,:), nTurbine, pressureRatio, ...
        propertiesArray, microturbineArray, CF1, CF2, ...
        dishApertureArea1, dishApertureArea2, dc1, dc2, construction...
        );
         if PC == 1
             Xstar = 0;
             X = 0;
             dT = 0;
             return
         end
```

```
    [PC, GF] = gradf(X(k,:), nTurbine, pressureRatio, ...
       propertiesArray, microturbineArray, CF1, CF2, construction);
        if PC == 1
            Xstar = 0;
            X = 0;
            dT = 0;
            return
        end
    penalty = zeros(1,length(X(k,:)));
    for i1 = 1:length(c)
        if c(i1) > 0
            if PHASE == 0 || PHASE == 1
                alpha = Xmu * (norm(gamma * GF) + 1) / norm(GC(:,i1...
                   ))^2;
            else
                alpha = Xmu / norm(GC(:,i1));
            end
            penalty = penalty + 2 * alpha * c(i1) * GC(:,i1)';
        end
    end
    gradP = gamma * GF + penalty;
    if gamma == 1 && norm(gradP) > norm(gamma*GF) + 1
        gradP = (norm(gamma*GF) + 1)*gradP/norm(gradP);
    end

    if PHASE == 0
        dT = sqrt(delta / (5*norm(gradP)));
    else
        dT = dTn;
    end

    a(k,:)     = -gradP;
    v(k,:)     = a(k,:) * dT;


%% Iterate
    if v(k,:) == 0
        v(k+1,:) = v(k,:) + ones(1,length(v(k,:)));
    else
        v(k+1,:) = 2 * v(k,:);
    end
    vk0 = norm(v(k,:));
    vk = norm(v(k+1,:));

    repeat = 1;
    while repeat == 1

        while vk > vk0

            dX(k,:) = norm(v(k,:)) * dT;

            if dX(k,:) < delta
                id  = 0;
                p = p + deltaT;
                dT  = p * dT;
```

```
else
    id   = id + 1;
    v(k,:) = delta * v(k,:) / (dT * norm(v(k,:)));
end

if is < ism
    X(k+1,:) = X(k,:) + v(k,:)*dT;
else
    r = 1 - 80 * eX / delta;
    v(k,:) = (v(k,:) + r * v(k-1,:)) / 4;
    X(k,:) = (X(k,:) + X(k-1,:)) / 2;
    dT = dT/2;
    is = 0;
    X(k+1,:) = X(k,:) + v(k,:)*dT;
end

% Penalty gradient
[PC, c]  = reheatConstraints(X(k+1,:), nTurbine, ...
   pressureRatio, propertiesArray, microturbineArray, ...
   CF1, CF2, dishApertureArea1, dishApertureArea2, dc1,...
    dc2, construction);
    if PC == 1
        Xstar = 0;
        X = 0;
        dT = 0;
        return
    end
[PC, GC] = gradc(X(k+1,:), nTurbine, pressureRatio, ...
   propertiesArray, microturbineArray, CF1, CF2, ...
   dishApertureArea1, dishApertureArea2, dc1, dc2, ...
   construction);
    if PC == 1
        Xstar = 0;
        X = 0;
        dT = 0;
        return
    end
[PC, GF] = gradf(X(k+1,:), nTurbine, pressureRatio, ...
   propertiesArray, microturbineArray, CF1, CF2, ...
   construction);
    if PC == 1
        Xstar = 0;
        X = 0;
        dT = 0;
        return
    end
penalty = zeros(1,length(X(k+1,:)));
for i1 = 1:length(c)
    if c(i1) > 0
        if PHASE == 0 || PHASE == 1
            alpha = Xmu * (norm(gamma * GF) + 1) / norm...
                (GC(:,i1))^2;
        else
            alpha = Xmu / norm(GC(:,i1));
        end
```

```
                penalty = penalty + 2 * alpha * c(:,i1) * GC(:,...
                    i1)';
            end
        end
        gradP = gamma * GF + penalty;
        if gamma == 1 && norm(gradP) > norm(gamma*GF) + 1
            gradP = (norm(gamma*GF) + 1)*gradP/norm(gradP);
        end

        a(k+1,:) = -gradP;
        v(k+1,:) = v(k,:) + a(k+1,:)*dT;

        if a(k+1,:)*a(k,:)' > 0
            is = 0;
        else
            if id > idm
                dT = dT / 2;
                id = 0;
                is = is + 1;
                p = 1;
            else
                is = is + 1;
                p = 1;
            end
        end

        if norm(a(k+1,:)) <= eg
            Xstar = X(k+1,:);
            return
        elseif norm(X(k+1,:) - X(k,:)) < eX
            Xstar = X(k+1,:);
            return
        elseif k > kmax
            Xstar = X(k,:);
            return
        else
            vk0 = norm(v(k,:));
            vk = norm(v(k+1,:));
                if vk > vk0
                    ix = 0;
                    k = k + 1;
                end
        end
    end

    while vk <= vk0
        X(k+2,:) = (X(k+1,:) + X(k,:))/2;
        ix = ix + 1;

        if ix <= ixm
            v(k+1,:) = (v(k+1,:) + v(k,:)) / 4;
            k = k + 1;
        else
            v(k+1,:) = 0;
            k = k + 1;
```

```
    ixm = 1;
end

% Penalty gradient
[PC, c]  = reheatConstraints(X(k+1,:), nTurbine, ...
   pressureRatio, propertiesArray, microturbineArray, ...
   CF1, CF2, dishApertureArea1, dishApertureArea2, dc1,...
    dc2, construction);
    if PC == 1
        Xstar = 0;
        X = 0;
        dT = 0;
        return
    end
[PC, GC] = gradc(X(k+1,:), nTurbine, pressureRatio, ...
   propertiesArray, microturbineArray, CF1, CF2, ...
   dishApertureArea1, dishApertureArea2, dc1, dc2, ...
   construction);
    if PC == 1
        Xstar = 0;
        X = 0;
        dT = 0;
        return
    end
[PC, GF] = gradf(X(k+1,:), nTurbine, pressureRatio, ...
   propertiesArray, microturbineArray, CF1, CF2, ...
   construction);
    if PC == 1
        Xstar = 0;
        X = 0;
        dT = 0;
        return
    end
for i1 = 1:length(c)
    if c(i1) > 0
        if PHASE == 0 || PHASE == 1
            alpha = Xmu * (norm(gamma * GF) + 1) / norm...
                (GC(:,i1))^2;
        else
            alpha = Xmu / norm(GC(:,i1));
        end
        penalty = penalty + 2 * alpha * c(:,i1) * GC(:,...
           i1)';
    end
end
gradP = gamma * GF + penalty;
if gamma == 1 && norm(gradP) > norm(gamma*GF) + 1
    gradP = (norm(gamma*GF) + 1)*gradP/norm(gradP);
end

a(k+1,:) = -gradP;
v(k+1,:) = v(k,:) + a(k+1,:)*dT;

if a(k+1,:)*a(k,:)' > 0
    is = 0;
```

```
            else
                if id > idm
                    dT = dT / 2;
                    id = 0;
                    is = is + 1;
                    p = 1;
                else
                    is = is + 1;
                    p = 1;
                end
            end

            if norm(a(k+1,:)) <= eg
                Xstar = X(k+1,:);
                return
            elseif norm(X(k+1,:) - X(k,:)) < eX
                Xstar = X(k+1,:);
                return
            elseif k > kmax
                Xstar = X(k,:);
                return
            else
                vk0 = norm(v(k,:));
                vk = norm(v(k+1,:));
                if vk > vk0
                    ix = 0;
                    k = k + 1;
                end
            end
        end
    end
end
```

B.4.3 reheatObjective.m

```
function [PC, O] = reheatObjective(x, nTurbine, pressureRatio, ...
   propertiesArray, microturbineArray, CF1, CF2, construction)
    [PC, outputs, ~, ~, ~, ~, ~] = reheatFunction(x, nTurbine, ...
       pressureRatio, propertiesArray, microturbineArray, CF1, CF2,...
        construction);
    if PC == 1
        O = 0;
        return
    end
    Wnet2   = outputs(4);
    O = -Wnet2;
end
```

B.4.4 gradf.m

```
function [PC, GF] = gradf(X,nTurbine,pressureRatio,propertiesArray,...
    microturbineArray,CF1,CF2,construction)
    % reheating only

    GF = zeros(1,9);
    h = 1e-8;

    Xgradientp = X + [h 0 0 0 0 0 0 0 0];
    Xgradientn = X - [h 0 0 0 0 0 0 0 0];
    [PC, Fp] = reheatObjective(Xgradientp,nTurbine,pressureRatio,...
        propertiesArray,microturbineArray,CF1,CF2,construction);
    if PC == 1
        return
    end
    [PC, Fn] = reheatObjective(Xgradientn,nTurbine,pressureRatio,...
        propertiesArray,microturbineArray,CF1,CF2,construction);
    if PC == 1
        return
    end
    GF(1) = (Fp - Fn)/(h*2);

    Xgradientp = X + [0 h 0 0 0 0 0 0 0];
    Xgradientn = X - [0 h 0 0 0 0 0 0 0];
    [PC, Fp] = reheatObjective(Xgradientp,nTurbine,pressureRatio,...
        propertiesArray,microturbineArray,CF1,CF2,construction);
    if PC == 1
        return
    end
    [PC, Fn] = reheatObjective(Xgradientn,nTurbine,pressureRatio,...
        propertiesArray,microturbineArray,CF1,CF2,construction);
    if PC == 1
        return
    end
    GF(2) = (Fp - Fn)/(h*2);

    Xgradientp = X + [0 0 h 0 0 0 0 0 0];
    Xgradientn = X - [0 0 h 0 0 0 0 0 0];
    [PC, Fp] = reheatObjective(Xgradientp,nTurbine,pressureRatio,...
        propertiesArray,microturbineArray,CF1,CF2,construction);
    if PC == 1
        return
    end
    [PC, Fn] = reheatObjective(Xgradientn,nTurbine,pressureRatio,...
        propertiesArray,microturbineArray,CF1,CF2,construction);
    if PC == 1
        return
    end
    GF(3) = (Fp - Fn)/(h*2);

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

```
[PC, Fp] = reheatObjective(Xgradientp,nTurbine,pressureRatio,...
   propertiesArray,microturbineArray,CF1,CF2,construction);
if PC == 1
    return
end
[PC, Fn] = reheatObjective(Xgradientn,nTurbine,pressureRatio,...
   propertiesArray,microturbineArray,CF1,CF2,construction);
if PC == 1
    return
end
GF(4) = (Fp - Fn)/(h*2);

Xgradientp = X + [0 0 0 0 h 0 0 0 0];
Xgradientn = X - [0 0 0 0 h 0 0 0 0];
[PC, Fp] = reheatObjective(Xgradientp,nTurbine,pressureRatio,...
   propertiesArray,microturbineArray,CF1,CF2,construction);
if PC == 1
    return
end
[PC, Fn] = reheatObjective(Xgradientn,nTurbine,pressureRatio,...
   propertiesArray,microturbineArray,CF1,CF2,construction);
if PC == 1
    return
end
GF(5) = (Fp - Fn)/(h*2);

Xgradientp = X + [0 0 0 0 0 h 0 0 0];
Xgradientn = X - [0 0 0 0 0 h 0 0 0];
[PC, Fp] = reheatObjective(Xgradientp,nTurbine,pressureRatio,...
   propertiesArray,microturbineArray,CF1,CF2,construction);
if PC == 1
    return
end
[PC, Fn] = reheatObjective(Xgradientn,nTurbine,pressureRatio,...
   propertiesArray,microturbineArray,CF1,CF2,construction);
if PC == 1
    return
end
GF(6) = (Fp - Fn)/(h*2);

Xgradientp = X + [0 0 0 0 0 0 h 0 0];
Xgradientn = X - [0 0 0 0 0 0 h 0 0];
[PC, Fp] = reheatObjective(Xgradientp,nTurbine,pressureRatio,...
   propertiesArray,microturbineArray,CF1,CF2,construction);
if PC == 1
    return
end
[PC, Fn] = reheatObjective(Xgradientn,nTurbine,pressureRatio,...
   propertiesArray,microturbineArray,CF1,CF2,construction);
if PC == 1
    return
end
GF(7) = (Fp - Fn)/(h*2);

Xgradientp = X + [0 0 0 0 0 0 0 h 0];
```

```
    Xgradientn = X - [0 0 0 0 0 0 0 h 0];
    [PC, Fp] = reheatObjective(Xgradientp,nTurbine,pressureRatio,...
       propertiesArray,microturbineArray,CF1,CF2,construction);
    if PC == 1
        return
    end
    [PC, Fn] = reheatObjective(Xgradientn,nTurbine,pressureRatio,...
       propertiesArray,microturbineArray,CF1,CF2,construction);
    if PC == 1
        return
    end
    GF(8) = (Fp - Fn)/(h*2);

    Xgradientp = X + [0 0 0 0 0 0 0 0 h];
    Xgradientn = X - [0 0 0 0 0 0 0 0 h];
    [PC, Fp] = reheatObjective(Xgradientp,nTurbine,pressureRatio,...
       propertiesArray,microturbineArray,CF1,CF2,construction);
    if PC == 1
        return
    end
    [PC, Fn] = reheatObjective(Xgradientn,nTurbine,pressureRatio,...
       propertiesArray,microturbineArray,CF1,CF2,construction);
    if PC == 1
        return
    end
    GF(9) = (Fp - Fn)/(h*2);
end
```

B.4.5 reheatConstraints.m

```
function [PC, c] = reheatConstraints(X, nTurbine, pressureRatio, ...
   propertiesArray, microturbineArray, CF1, CF2, dishApertureArea1,...
    dishApertureArea2, dc1, dc2, construction)
    if construction == 2
        X(4) = 1;
        X(7) = 1;
    end
    x = [X(1), X(2)/100 , X(3), X(4), X(5)/100, X(6), X(7), X(8)...
       /100, X(9)];

    c(1) = x(5) * x(6) * (x(4) + 1) / 16 - dishApertureArea1/100;
    c(2) = x(8) * x(9) * (x(7) + 1) / 16 - dishApertureArea2/100;

    if construction == 2
        c(3) = 2 * x(5) - (sqrt(3)-1) / 2 * sqrt(x(5) * x(6) / (2 *...
            pi));
        c(4) = 2 * x(8) - (sqrt(3)-1) / 2 * sqrt(x(8) * x(9) / (2 *...
            pi));
    else
        c(3) = x(5) * (x(4) + 1) / 2 - (sqrt(3) - 1) / 2  * (x(5) *...
            x(6) * (x(4) + 1) / (4 * pi))^(0.5);
        c(4) = x(8) * (x(7) + 1) / 2 - (sqrt(3) - 1) / 2  * (x(8) *...
            x(9) * (x(7) + 1) / (4 * pi))^(0.5);
    end

    c(5) = dc1 - sqrt(x(5) * x(6) * (x(4) + 1) / (4 * pi));
    c(6) = dc2 - sqrt(x(8) * x(9) * (x(7) + 1) / (4 * pi));

    c(7) = - x(2);
    c(8) = -x(3);
    c(9) = - x(5);
    c(10) = - x(6);
    c(11) = - x(8);
    c(12) = - x(9);

    c13 = sqrt(dishApertureArea1 / pi);
    c(13) = x(3) - c13;

    [PC, ~, Ts, ~, ~, ~, ~] = reheatFunction(X, nTurbine, ...
       pressureRatio, propertiesArray, microturbineArray, CF1, CF2,...
        construction);
    if PC == 1
        return
    end
    c(14) = Ts(1) - 1200;
    c(15) = Ts(2) - 1200;

    if construction == 1;
        c(16) = 2.5 - x(4);
        c(17) = 2.5 - x(7);
    end
end
```

B.4.6 gradc.m

```
function [PC, GC] = gradc(x, nTurbine, pressureRatio, ...
   propertiesArray, microturbineArray, CF1, CF2, dishApertureArea1,...
    dishApertureArea2, dc1, dc2, construction)
    % reheating only

    if construction == 2;
        GC = zeros(9,15);
    else
        GC = zeros(9,17);
    end

    h = 1e-8;

    Xgradientp = x + [h 0 0 0 0 0 0 0 0];
    Xgradientn = x - [h 0 0 0 0 0 0 0 0];
    [PC, Cp] = reheatConstraints(Xgradientp, nTurbine, ...
       pressureRatio, propertiesArray, microturbineArray, CF1, CF2,...
        dishApertureArea1, dishApertureArea2, dc1, dc2, ...
       construction);
    if PC == 1
        return
    end
    [PC, Cn] = reheatConstraints(Xgradientn, nTurbine, ...
       pressureRatio, propertiesArray, microturbineArray, CF1, CF2,...
        dishApertureArea1, dishApertureArea2, dc1, dc2, ...
       construction);
    if PC == 1
        return
    end
    gradCr1 = (Cp - Cn)/(h*2);
    GC(1,:) = gradCr1;

    Xgradientp = x + [0 h 0 0 0 0 0 0 0];
    Xgradientn = x - [0 h 0 0 0 0 0 0 0];
    [PC, Cp] = reheatConstraints(Xgradientp, nTurbine, ...
       pressureRatio, propertiesArray, microturbineArray, CF1, CF2,...
        dishApertureArea1, dishApertureArea2, dc1, dc2, ...
       construction);
    if PC == 1
        return
    end
    [PC, Cn] = reheatConstraints(Xgradientn, nTurbine, ...
       pressureRatio, propertiesArray, microturbineArray, CF1, CF2,...
        dishApertureArea1, dishApertureArea2, dc1, dc2, ...
       construction);
    if PC == 1
        return
    end
    gradCr2 = (Cp - Cn)/(h*2);
    GC(2,:) = gradCr2;

    Xgradientp = x + [0 0 h 0 0 0 0 0 0];
```

```
Xgradientn = x - [0 0 h 0 0 0 0 0 0];
[PC, Cp] = reheatConstraints(Xgradientp, nTurbine, ...
   pressureRatio, propertiesArray, microturbineArray, CF1, CF2,...
    dishApertureArea1, dishApertureArea2, dc1, dc2, ...
   construction);
if PC == 1
    return
end
[PC, Cn] = reheatConstraints(Xgradientn, nTurbine, ...
   pressureRatio, propertiesArray, microturbineArray, CF1, CF2,...
    dishApertureArea1, dishApertureArea2, dc1, dc2, ...
   construction);
if PC == 1
    return
end
gradCr3 = (Cp - Cn)/(h*2);
GC(3,:) = gradCr3;

Xgradientp = x + [0 0 0 h 0 0 0 0 0];
Xgradientn = x - [0 0 0 h 0 0 0 0 0];
[PC, Cp] = reheatConstraints(Xgradientp, nTurbine, ...
   pressureRatio, propertiesArray, microturbineArray, CF1, CF2,...
    dishApertureArea1, dishApertureArea2, dc1, dc2, ...
   construction);
if PC == 1
    return
end
[PC, Cn] = reheatConstraints(Xgradientn, nTurbine, ...
   pressureRatio, propertiesArray, microturbineArray, CF1, CF2,...
    dishApertureArea1, dishApertureArea2, dc1, dc2, ...
   construction);
if PC == 1
    return
end
gradCr4 = (Cp - Cn)/(h*2);
GC(4,:) = gradCr4;

Xgradientp = x + [0 0 0 0 h 0 0 0 0];
Xgradientn = x - [0 0 0 0 h 0 0 0 0];
[PC, Cp] = reheatConstraints(Xgradientp, nTurbine, ...
   pressureRatio, propertiesArray, microturbineArray, CF1, CF2,...
    dishApertureArea1, dishApertureArea2, dc1, dc2, ...
   construction);
if PC == 1
    return
end
[PC, Cn] = reheatConstraints(Xgradientn, nTurbine, ...
   pressureRatio, propertiesArray, microturbineArray, CF1, CF2,...
    dishApertureArea1, dishApertureArea2, dc1, dc2, ...
   construction);
if PC == 1
    return
end
gradCr5 = (Cp - Cn)/(h*2);
GC(5,:) = gradCr5;
```

```
Xgradientp = x + [0 0 0 0 0 h 0 0 0];
Xgradientn = x - [0 0 0 0 0 h 0 0 0];
[PC, Cp] = reheatConstraints(Xgradientp, nTurbine, ...
   pressureRatio, propertiesArray, microturbineArray, CF1, CF2,...
    dishApertureArea1, dishApertureArea2, dc1, dc2, ...
   construction);
if PC == 1
    return
end
[PC, Cn] = reheatConstraints(Xgradientn, nTurbine, ...
   pressureRatio, propertiesArray, microturbineArray, CF1, CF2,...
    dishApertureArea1, dishApertureArea2, dc1, dc2, ...
   construction);
if PC == 1
    return
end
gradCr6 = (Cp - Cn)/(h*2);
GC(6,:) = gradCr6;

Xgradientp = x + [0 0 0 0 0 0 h 0 0];
Xgradientn = x - [0 0 0 0 0 0 h 0 0];
[PC, Cp] = reheatConstraints(Xgradientp, nTurbine, ...
   pressureRatio, propertiesArray, microturbineArray, CF1, CF2,...
    dishApertureArea1, dishApertureArea2, dc1, dc2, ...
   construction);
if PC == 1
    return
end
[PC, Cn] = reheatConstraints(Xgradientn, nTurbine, ...
   pressureRatio, propertiesArray, microturbineArray, CF1, CF2,...
    dishApertureArea1, dishApertureArea2, dc1, dc2, ...
   construction);
if PC == 1
    return
end
gradCr7 = (Cp - Cn)/(h*2);
GC(7,:) = gradCr7;

Xgradientp = x + [0 0 0 0 0 0 0 h 0];
Xgradientn = x - [0 0 0 0 0 0 0 h 0];
[PC, Cp] = reheatConstraints(Xgradientp, nTurbine, ...
   pressureRatio, propertiesArray, microturbineArray, CF1, CF2,...
    dishApertureArea1, dishApertureArea2, dc1, dc2, ...
   construction);
if PC == 1
    return
end
[PC, Cn] = reheatConstraints(Xgradientn, nTurbine, ...
   pressureRatio, propertiesArray, microturbineArray, CF1, CF2,...
    dishApertureArea1, dishApertureArea2, dc1, dc2, ...
   construction);
if PC == 1
    return
end
```

```
    gradCr8 = (Cp - Cn)/(h*2);
    GC(8,:) = gradCr8;

    Xgradientp = x + [0 0 0 0 0 0 0 0 h];
    Xgradientn = x - [0 0 0 0 0 0 0 0 h];
    [PC, Cp] = reheatConstraints(Xgradientp, nTurbine, ...
       pressureRatio, propertiesArray, microturbineArray, CF1, CF2,...
        dishApertureArea1, dishApertureArea2, dc1, dc2, ...
       construction);
    if PC == 1
        return
    end
    [PC, Cn] = reheatConstraints(Xgradientn, nTurbine, ...
       pressureRatio, propertiesArray, microturbineArray, CF1, CF2,...
        dishApertureArea1, dishApertureArea2, dc1, dc2, ...
       construction);
    if PC == 1
        return
    end
    gradCr9 = (Cp - Cn)/(h*2);
    GC(9,:) = gradCr9;
end
```

Appendix C

Microturbines

Model	Number	Minimum pressure ratio	Maximum pressure ratio	Minimum mass flow rate (kg/s)	Maximum mass flow rate (kg/s)	Maximum turbine efficiency (%)	Maximum compressor efficiency (%)
GT12	1	1.88	2.75	0.057	0.088	59	75
GT15	2	1.43	1.95	0.075	0.122	59	75
GT20	3	1.6	2.45	0.111	0.181	66	79
GT22a	4	1.48	2.22	0.099	0.163	72	80
GT22b	5	1.62	2.07	0.108	0.151	72	76
GT25R	6	1.46	2	0.118	0.175	65	73
GT28R	7	1.43	1.92	0.13	0.188	65	75
GT28RS	8	1.48	2.4	0.13	0.245	68	77
GT30R	9	1.56	2.78	0.179	0.334	72	78
GT32	10	1.64	2.74	0.155	0.285	68	75
GT35a	11	1.62	2.73	0.157	0.285	71	75
GT35b	12	1.64	2.46	0.204	0.316	71	77
GT35R	13	1.32	2.54	0.163	0.394	74	79
GT37a	14	1.92	2.7	0.224	0.306	68	77
GT37b	15	1.76	2.56	0.214	0.347	68	76
GT37c	16	1.46	2.82	0.153	0.326	68	78
GT40a	17	1.48	2.4	0.221	0.367	70	78
GT40b	18	1.5	2.9	0.198	0.379	70	78
GT42a	19	1.78	3	0.253	0.571	69	78
GT42b	20	1.47	2.72	0.294	0.53	69	77
GT60	21	1.5	2.6	0.578	1.085	79	79
GT6041	22	1.3	3	0.607	0.88	78	80

References

Turbo by Garrett. 2002. *Garrett Performance Products*. Honeywell International Inc.