12-12-2020

Stirling Cycle Machine Analysis

Israel Urieli
Ohio University

Stirling Cycle Machine Analysis

Israel Urieli, PhD,
Associate Professor Emeritus of Mechanical Engineering

Department of Mechanical Engineering
Ohio University

Dedicated to William T. Beale (1928 – 2016), inventor of the Free Piston Stirling Engine, Mentor and Friend

Contents

Preface

This web resource is intended to be totally self contained learning resource for the analysis and development of computer simulation of single phase, piston/cylinder Stirling cycle machines. It includes thermodynamic, heat transfer and fluid flow friction analysis, and until 2012 it was used as resource material for an advanced course for Mechanical Engineering majors. The course structure was based on the book by I.Urieli & D.M.Berchowitz 'Stirling Cycle Engine Analysis' (Adam Hilger, 1984). The computer simulation program modules (originally written in FORTRAN) have all been updated and rewritten in MATLAB, a convenient interactive language which allows direct graphical output - essential for Stirling cycle analysis. A complete set of all the m-files are developed and provided, and they can be augmented and adapted as needed for specific engine/refrigerator configurations.

This learning resource includes a set of tutorial MATLAB computer program modules for simulating specific Stirling engine configurations. The complete set of m-files can be downloaded in compressed format **sea.zip** (**sea** = **s**tirling **e**ngine **a**nalysis). These modules can be augmented and adapted as required to simulate a specific engine design. Currently the engine modules are for Alpha machines, including a Sinusoidal drive, a Ross Yoke-drive and a Ross Rocker-V engine. The heat exchanger types include tubular, annular gap, and slot heat exchangers, and the regenerator matrix types include screen mesh and rolled foil matrices. Working gas types include air, helium, and hydrogen.

Note that the purpose of this learning resource is to develop an appreciation and understanding of the complexity of practical Stirling cycle machine performance simulation, mainly due to the heat transfer processes. It is not intended as an alternative to the **Sage Software** for engineering modeling and optimization of Stirling cycle machines.

Israel Uriel

Stirling Cycle Machine Analysis

Chapter 1: Background and Introduction

The essential background requirements for this resource is a knowledge of thermodynamics, forced convection heat transfer and flow friction, and **MATLAB** computer programing. The ubiquitous **Wikipedia - Stirling engine** website presents an overview and introduction to the Stirling engine which includes some excellent animations giving a clear insight as to the operating principles of these machines.

History

We are currently celebrating the 200th anniversary of Robert Stirling's original patent and many books and articles have been written about the history and development (or reasons for the lack of development) of hot air engines. The most recommended is the book by Theodore Finkelstein and **Allan J. Organ**: '**Air Engines: The History, Science, and Reality of the Perfect Engine**' (ASME Press, 2001). Also recommended are the two books by **Robert Sier**: '**Rev Robert Stirling D.D. - Inventor of the Heat Economiser & Stirling Cycle Engine**' (**L A Mair**, 1995 – Unfortunately out of print), and '**HOT AIR CALORIC and STIRLING ENGINES, Volume One: A History**' (**L A Mair**, 2000). Refer also to Robert Sier's web-page of **Robert Stirling**, which includes an animation of Stirling's original machine. Another interesting web-page of Stirling engine history is that of the **National Museums of Scotland**.

Robert Stirling published his famous patent in 1816, and **Sadi Carnot** published his treatise '**Reflections on the Motive Power of Fire**' in 1824. In this treatise Carnot stated that a heat engine can only attain the ideal maximum efficiency if the heat is transferred isothermally with the source and sink, and proposed the Carnot cycle as an example of the ideal heat engine cycle. The ingenious aspect of Stirling's patent is the regenerator, which allows the non isothermal heat transfer to be done internally, enabling the ideal Stirling engine cycle to attain the ideal maximum efficiency (refer to the fictitious discussion: **A Meeting between Robert Stirling and Sadi Carnot in 1824** (see the additional files area)). Probably one of the main reasons for the lack of development of Stirling engines was that the importance of the regenerator was not understood for about 100 years after Stirling's original patent. The initial attempt at an analysis of the Stirling engine was published in 1871 by **Gustav Schmidt**. The Lehmann machine on which Schmidt based his analysis was not fitted with a regenerator (Refer: C. Lyle Cummins Jr. (1976). *Internal Fire*, Carnot Press/Graphics Arts Center, Portland, Oregon. Chapter 2 - Air Engines, page 23) and it is conceivable that Schmidt did not appreciate its importance. He refers to the

textbook by Zeuner as containing a "complete, simple and clear theory" of air engines, but in the same textbook Zeuner decries the use of regenerators for air engines (Refer: Finkelstein, T., 1959, *Air Engines* in *The Engineer* part 1, 27 March).

From the 1950's through the 1980's there was an ambitious effort to develop automobile Stirling engines in the USA and Europe (refer to **NASA** publications on the Ford Motor Company program and the Mechanical Technology Inc. (MTI) program.) There are a number of reasons why this program was ultimately unsuccessful: A significantly high pressure of Hydrogen gas is required in order to obtain an acceptable specific power output. This lead ultimately to sealing problems for the output crankshaft, hydrogen embrittlement of the casing, and a complex valve system to increase or decrease pressure for acceleration.

The current focus is on fully sealed Stirling cycle machines for electrical power production or refrigeration using Helium or Nitrogen gas. Since the Stirling engine is the only conceivable heat engine that can operate on a low temperature difference, this enables CHP (combined heat and power), low temperature (flat plate) solar, or waste heat recovery systems. Stirling cycle refrigerator systems using Helium gas have higher COP (coefficient of performance) values than regular vapor-compression refrigerators, and can operate at cryogenic temperature levels.

Athens, Ohio

Since the 1970's, Athens, Ohio has been a hotbed of Stirling cycle machine activity, both engines and coolers, and includes three R&D and manufacturing companies:

Sunpower was formed by **William Beale** (Appendix A) in 1974, mainly based on his invention of the free-piston Stirling engine in 1964. Sunpower was mainly an R&D company, licensing its technology globally, and also manufactured Stirling cycle cryogenic coolers for liquifying oxygen. Sunpower developed a 1kW free-piston engine/generator in 1983, and since 1995 this technology was used by British Gas to develop CHP (Combined Heat and Power) units – the 1kW engine/generator is currently manufactured by **Microgen Engine Corporation** (refer to their **History** and **Engine** web pages).
An interesting paper describing the chronology of the development of Free Piston Stirling engine technology was presented by **David Berchowitz** at the 2018 International Stirling Engine Conference (Refer: **A Personal History in the Development of the Modern Stirling Engine**).
In 2013 Sunpower was acquired by **AMETEK, Inc** in Pennsylvania, however continues doing Stirling cycle machine development in Athens, Ohio.

Stirling Technology (note recent company name change: **Combined Energy Technology**) is a spinoff of Sunpower, and was originally formed in order to continue the development of the 3.5 kW ST-5 Air engine. This large Beta type engine uses a bell crank mechanism to obtain the correct displacer phasing, burns biomass fuel (such as sawdust pellets or rice husks), and can function as a cogeneration unit in rural areas.
Currently Stirling Technology is working with **Microgen Engine Corporation**, an international company which produces the MEC 1kW free-piston engine/generator. Stirling Technology has developed a multifuel burner for the engine and is partnering with Microgen to get various systems into the market.

Global Cooling (currently **Stirling Ultracold**) was a spinoff from Sunpower, and was formed in 1995 by **David Berchowitz** mainly in order to develop and commercialize free-piston Stirling cycle coolers for home refrigerator applications. These systems, apart from being significantly more efficient than regular vapor-compression refrigerators, have the added advantage of being compact, portable units using helium as the working fluid (and not the HFC refrigerants such as R134a, having a Global Warming Potential of 1,300). More recently Global Cooling decided to concentrate their development efforts on systems in which there are virtually no competitive systems - cooling between -40°C and -80°C, and they established a new company name: **Stirling Ultracold**.
Update - January 2021: **Stirling Ultracold**'s Ultra-Low Temperature (ULT) freezers meets today's unprecedented COVID-19 deployment challenges.

Sage Software for engineering modeling and optimization of Stirling cycle machines, developed and maintained by David Gedeon.

Solar Heat and Power Cogeneration and Waste Heat Recovery

With the current energy and global warming crises, there is renewed interest in renewable energy systems, such as wind and solar energy, distributed heat and power cogeneration systems and waste heat recovery systems..

Cool Energy, Inc of Boulder, Colorado, developed a complete solar heat and power cogeneration system for home usage incorporating Stirling engine technology for electricity generation. This unique application included evacuated tube solar thermal collectors, thermal storage, hot water and space heaters, and a Stirling engine/generator using nitrogen gas. Currently they are concentrating on low temperature (150°C - 400°C) waste heat recovery systems (Refer: Cool Energy ThermoHeart 25kW Engine Overview).

Various Links

Since 2005 **Siegfried "Zig" Herzog** from Pennsylvania State University (currently retired) has developed a web resource: **Stirling Cycle Analysis** that essentially parallels this web resource, the major difference being that he provides an on-line simulation program, whereas I provide the MATLAB source files which can be modified by the user as required.

Andy Ross of Columbus, Ohio has been developing small air engines with extremely innovative Alpha designs, including the classical Ross-Yoke drive and more recently a balanced "Rocker-V" mechanism. Refer to his book: **Making Stirling Engines** (Ross Experimental, 1993). Matt Keveney has done an animation showing clearly the principles of operation of the **Ross yoke linkage** mechanism. Andy Ross wrote an article on the model Climax locomotive that he built using a small Rocker-V engine: '**A Class A Climax Locomotive**' (Appendix A.3). We will be using both of Andy Ross' engines as examples in our simulation programs.

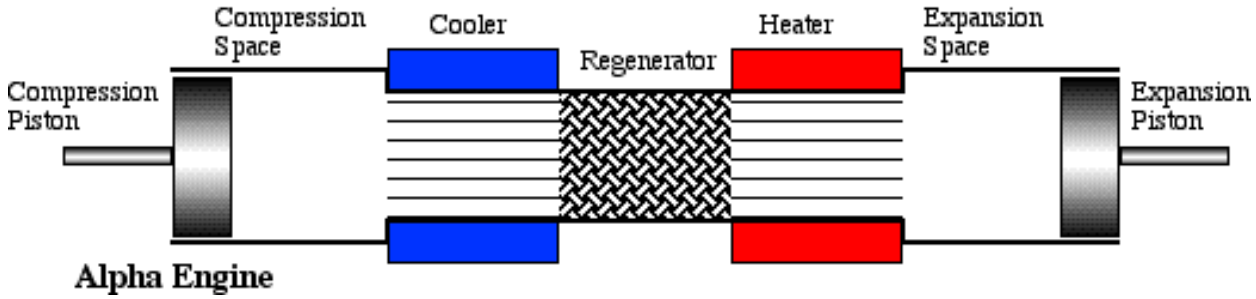
Analysis of Stirling Cycle Machines

The ideal Stirling cycle machine is easily analyzed using basic **thermodynamics**, however the analysis of actual Stirling cycle machines is extremely complex, mainly because of heat transfer between the external heat source/sink and the working gas, the regenerator, and the nonsteady reversing flow of the working gas, requiring sophisticated computer analysis. This learning resource is an attempt to develop this process in stages, from the Ideal Isothermal model followed by the Ideal Adiabatic model (including sections in which there is no heat transfer) through to the complexity of actual heat transfer processes, allowing more practical predictions of actual machine performance, as well as enabling parametric analysis. This does not include dynamic analysis, thus the operating frequency is specified as an independent variable.

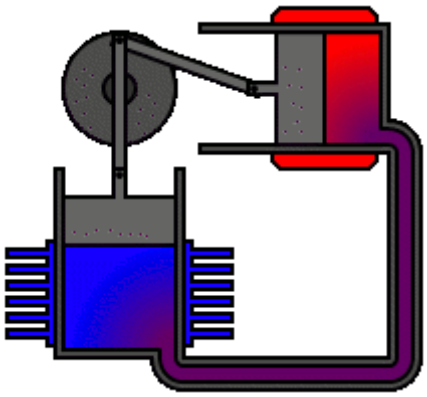
Update 2016: A unique alternative approach was presented by **David Berchowitz** at the 2016 International Stirling Engine Conference (Refer: **A Phasor Description of the Stirling Cycle**). The phasor description includes both thermodynamic processes and mechanical dynamics, resulting in a useful guide to the understanding of these machines.

Chapter 2: Basic Engine Configurations. Part A) Alpha Type Engines

The mechanical configurations of Stirling engines are generally divided into three groups known as the **Alpha**, **Beta**, and **Gamma** arrangements. Alpha engines have two pistons in separate cylinders which are connected in series by a heater, regenerator and cooler. Both Beta and Gamma engines use displacer-piston arrangements, the Beta engine having both the displacer and the piston in an in-line cylinder system, whilst the Gamma engine uses separate cylinders.

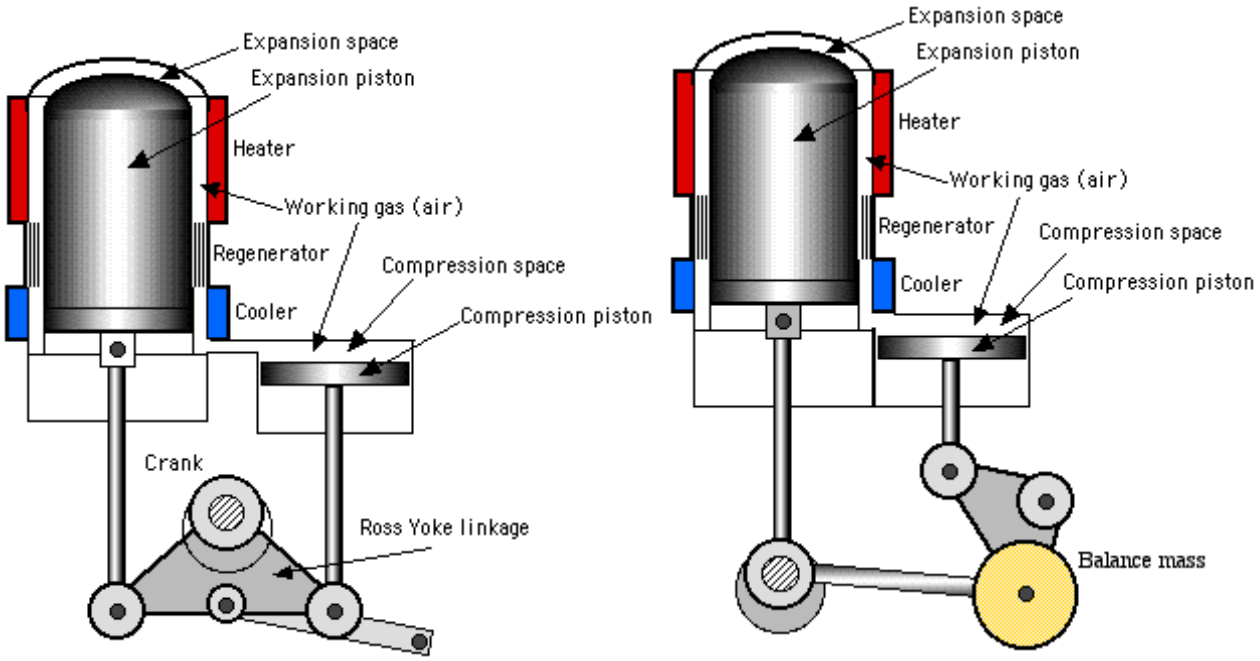


The Alpha engine is conceptually the simplest Stirling engine configuration, however suffers from the disadvantage that both the hot and cold pistons need to have seals to contain the working gas. There are a number of mechanical mechanisms which enable this type of engine to operate correctly with the correct phasing of the two pistons. An excellent animation of the V-type Alpha engine developed by Richard Wheeler (**Zephyris**) of **Wikipedia** is shown below:



Andy Ross of Columbus, Ohio has been designing and building small air engines since the 1970's, including extremely innovative Alpha designs. He is the inventor of the

classical Ross Yoke drive engine as well as a balanced “Rocker-V” mechanism, both shown below.



Refer to Andy Ross' delightful book: **Making Stirling Engines** (Ross Experimental, 1993). The D-90 Yoke drive Alpha Stirling engine described in his book will be used as the primary case study of this web resource. At Ohio University we have a laboratory model of the D-90 Yoke drive engine which is heated electrically in order to accurately determine the heat power input. **Matt Keveney** has done an animation showing clearly the principles of operation of the **Ross yoke linkage** mechanism. This ingenious mechanism for transferring dual piston motion into rotational motion minimizes the piston side forces normally encountered on a regular crankshaft mechanism.

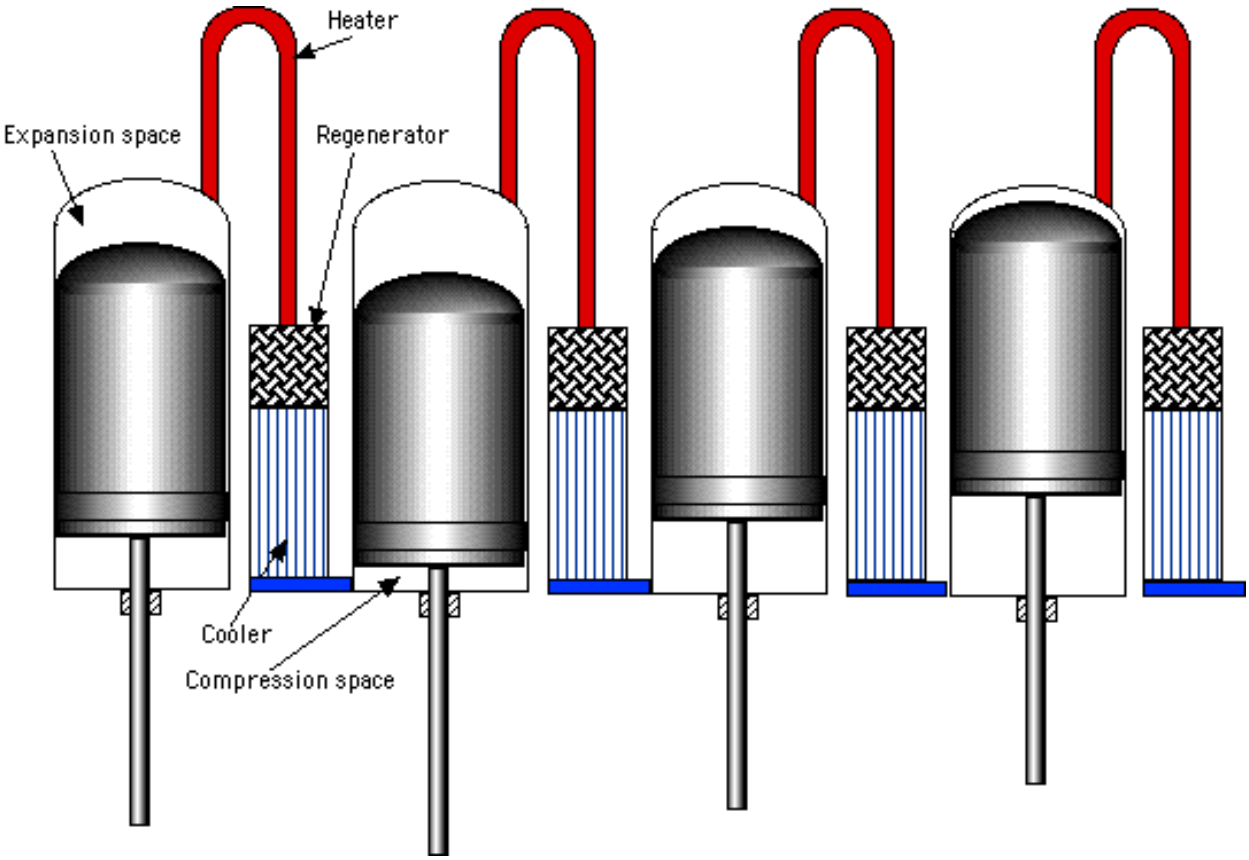
More recently Andy Ross came up with the balanced Rocker-V mechanism design. He has published an article on the model Climax locomotive that he built using a small (20cc) Rocker-V engine, and allowed me to maintain a copy of this article "**A Class A Climax Locomotive**" (Appendix A.3). A number of these Rocker-V engines were built by students for a **Senior Design** class at Ohio University in 2001 (Appendix B), and will also be used as a case study of this web resource. One of Andy Ross' many "YouTube" videos shows a unique balanced **double-V Alpha** engine which avoids the use of a heat exchanger section stretching across the V.

Cool Energy, Inc of Boulder, Colorado have been developing low temperature (150°C – 400°C) Alpha Stirling engine/generator systems since 2006 (Refer to their **Product**

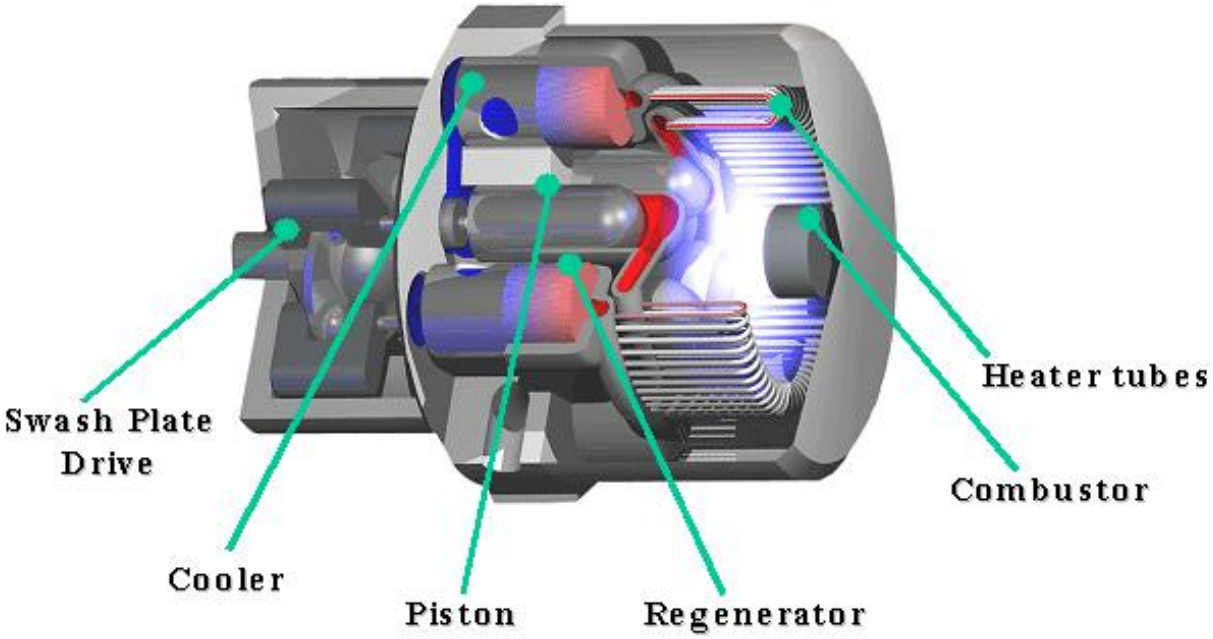
Development History). This included a complete solar heat and power cogeneration system for home usage including evacuated tube solar thermal collectors, thermal storage systems, hot water and space heaters, and the SolarHeart Stirling engine/generator. Currently they are concentrating on waste heat recovery systems (Refer: Cool Energy ThermoHeart 25kW Engine Overview) using the four cylinder Alpha engine as described in the paper presented at the 2016 International Stirling Engine Conference by the Cool Energy team: 25kW Low Temperature Stirling Engine for Heat Recovery, Solar, and Biomass Applications).

Multiple Cylinder Alpha Stirling Engines

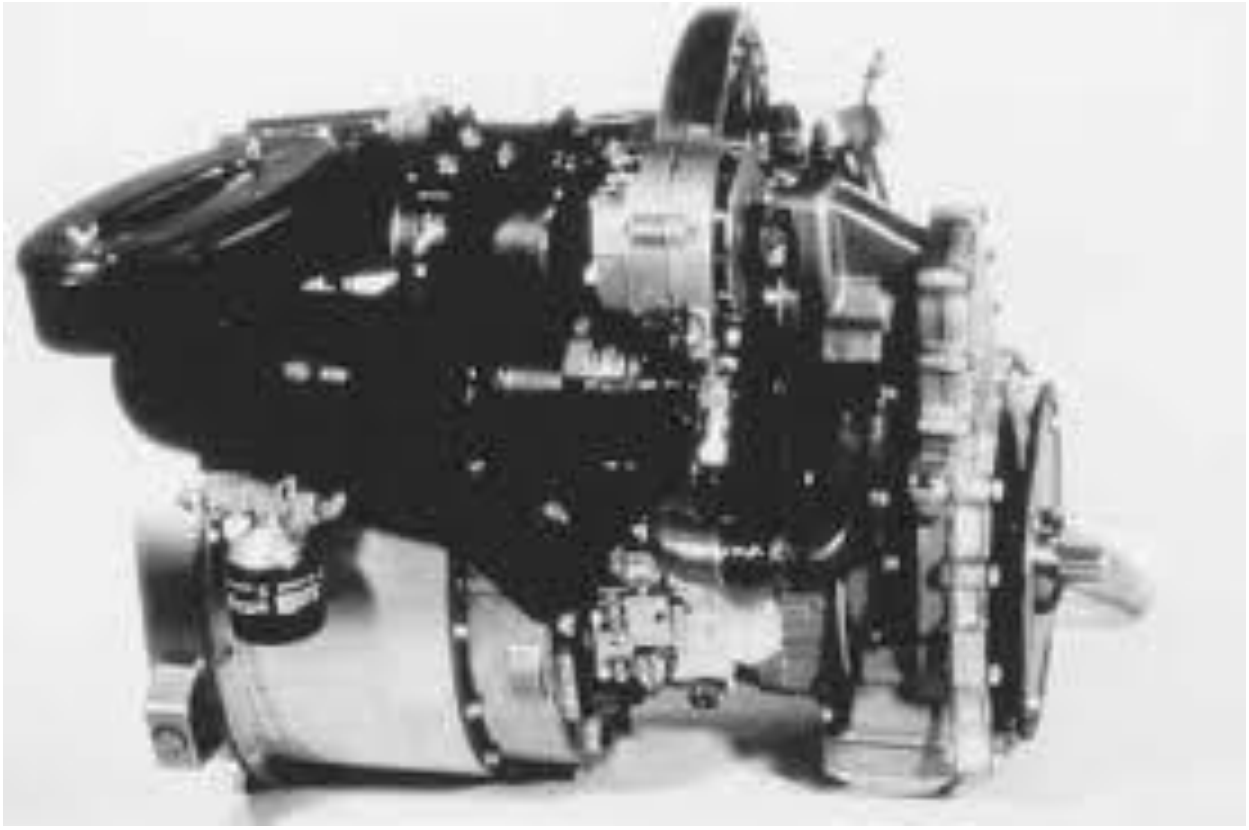
The Alpha engine can also be compounded into a compact multiple cylinder configuration, enabling an extremely high specific power output. A schematic diagram of this configuration is shown below. Notice that the four cylinders are interconnected, so that the expansion space of one cylinder is connected to the compression space of the adjacent cylinder via a series connected heater, regenerator and cooler. The pistons are typically driven by a swashplate, resulting in a pure sinusoidal reciprocating motion having a 90 degree phase difference between the adjacent pistons.



One example of the swashplate 4-cylinder Alpha engine is shown below. This engine was originally developed by **Stirling Thermal Motors** (later **STM Corporation**, however is no longer operational).

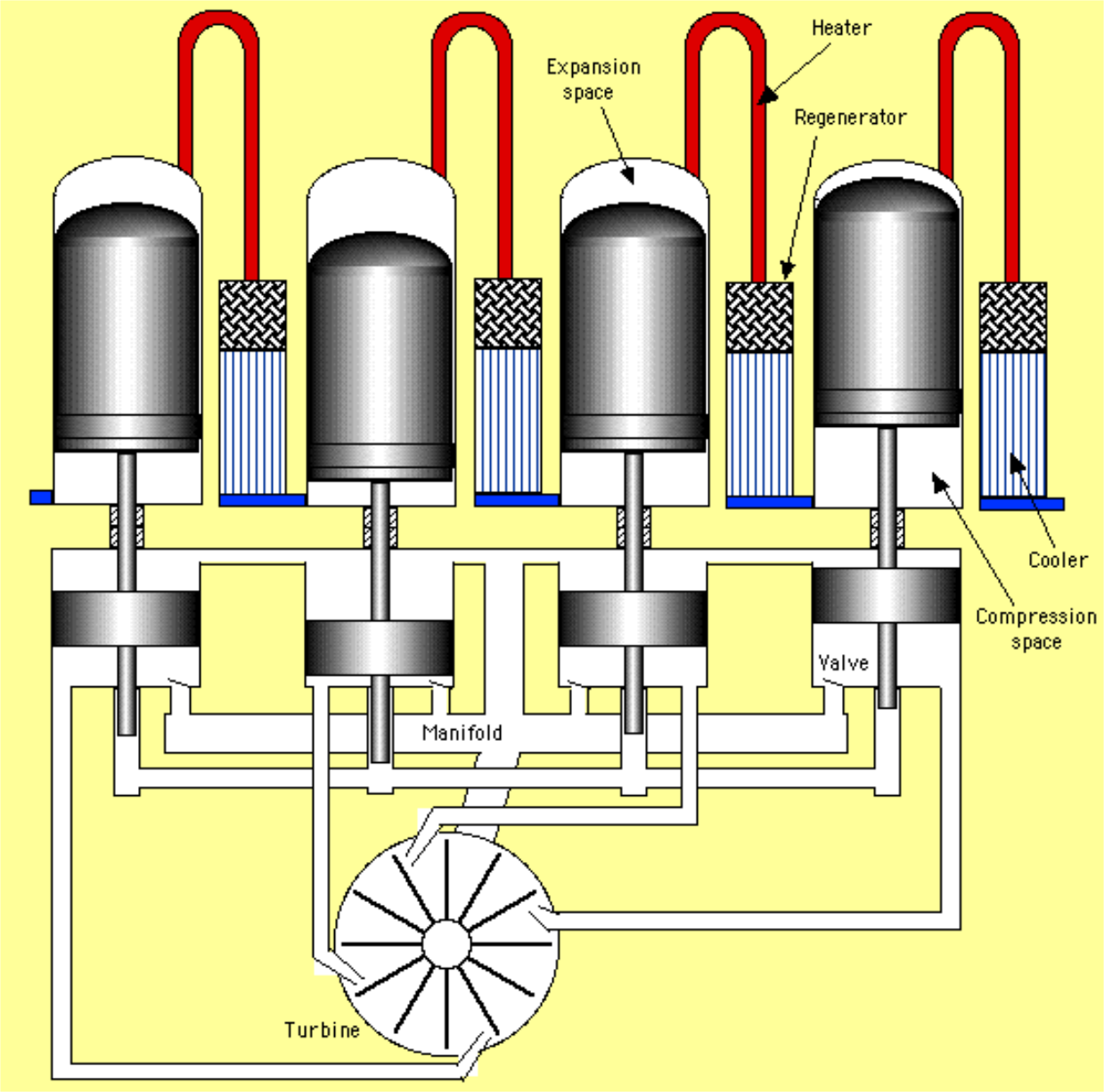


During the 1970's N V Philips, of Holland, and the Ford Motor Company developed an experimental automotive engine – a four-cylinder swashplate-drive engine as shown in the following photograph:



This engine, the Ford-Philips 4-215 engine, is used as a case study in the book by I.Urieli & D.M.Berchowitz – Stirling Cycle Engine Analysis (Adam Hilger, 1984), pages 25 – 31. It will be one of the case studies of this learning resource, and since the book is out of print, these pages have been added here for convenience (Appendix C).

William Beale of **Sunpower, Inc** came up with an interesting configuration, combining a four cylinder free-piston alpha engine with a gas turbine output stage, as shown in the following schematic diagram:

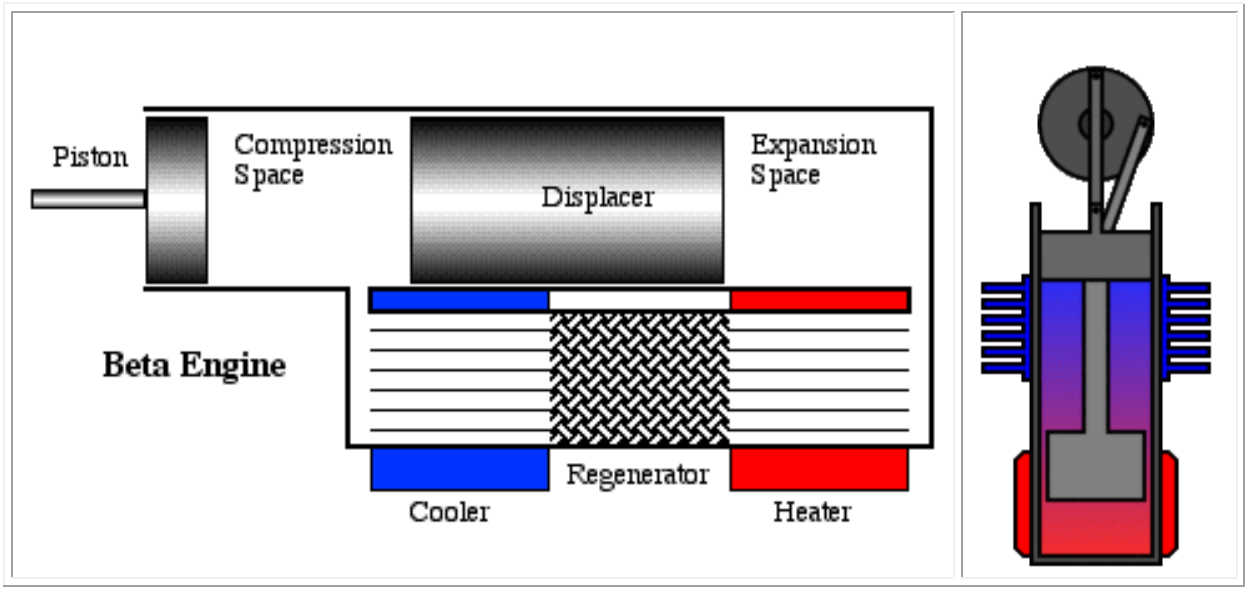


The four cylinders are physically arranged with a 90 degree phase difference with each piston connected to a gas compressor. The gas compressors are then used to drive a gas turbine expander as shown. The main advantage of this system is the promise of a high specific power and most important, high reliability and life resulting from the absence of heavily loaded moving parts, since there are no side loads on any sliding bearings.

The sketch shows single acting gas compressors for simplicity, however the actual machine would use double acting compressors so that there are eight gas pulses on the turbine for each cycle of the four cylinder machine.

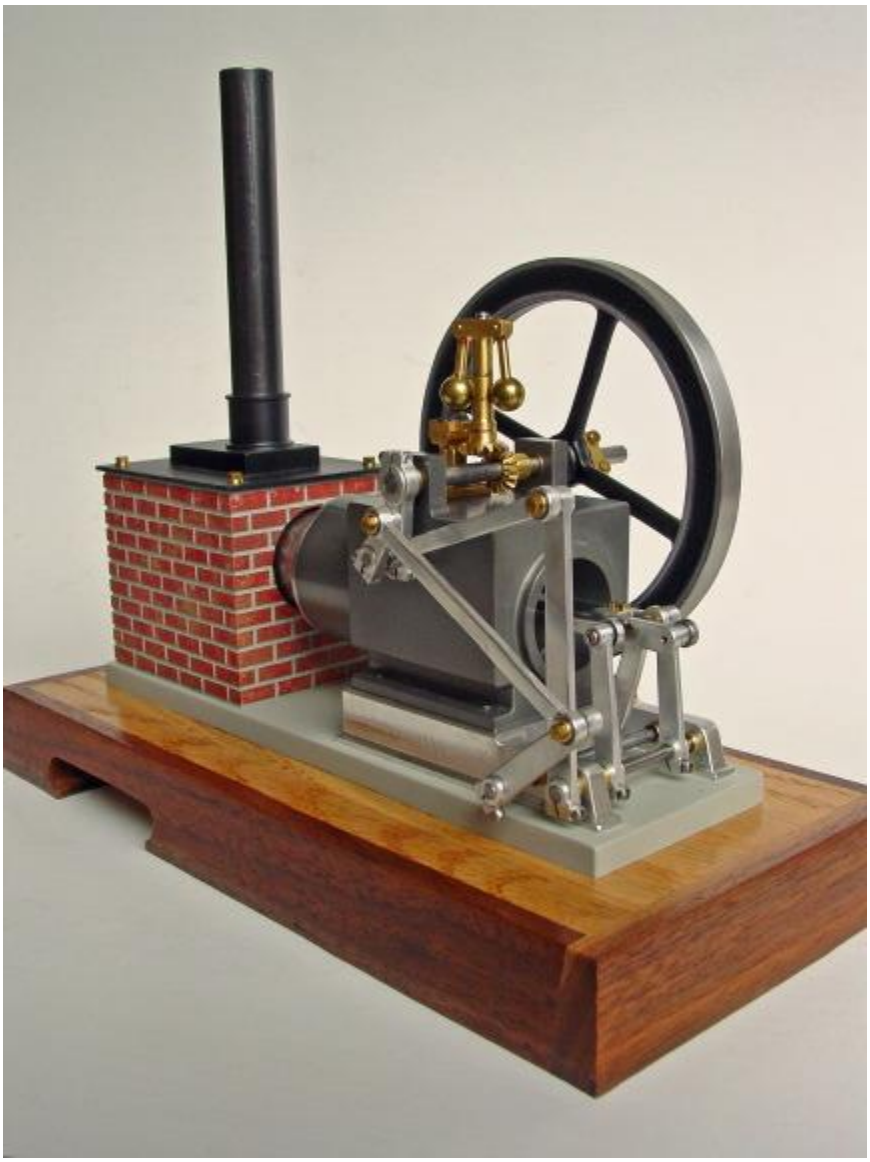
Chapter 2: Basic Engine Configurations. Part B) Beta Type Engines

The Beta configuration is the classic Stirling engine configuration and has enjoyed popularity from its inception until today. Stirling's original engine from his patent drawing of 1816 shows a Beta arrangement. A photograph of Robert Stirling, the original patent drawing, as well as an animated model of Stirling's engine is clearly shown in an interesting website by **Robert Sier**. From the figure below we see that unlike the Alpha machine, the Beta engine has a single power piston and a displacer, whose ideal purpose is to "displace" the working gas at constant volume, and shuttle it between the expansion and the compression spaces through the series arrangement cooler, regenerator, and heater. In actual engines the linkage driving the piston and displacer will move them such that the gas will compress while it is mainly in the cool compression space and expand while in the hot expansion space. This is clearly illustrated in the adjacent animation which was produced by Richard Wheeler (**Zephyris**) of **Wikipedia**.



A detailed description of the ideal Beta machine cycle is presented in the Engineering Thermodynamics - Chapter 3b web resource. Refer also to the animation of the Beta machine by **Matt Keveney** - **Single Cylinder Stirling Engine**, showing clearly the principle of operation.

Apart from Stirling's original engine, an important early Beta engine is Lehmann's machine on which **Gustav Schmidt** did the first reasonable analysis of Stirling engines in 1871. Andy Ross built a small working replica of the **Lehmann machine**



as well as a **model air engine**,

both based on single cylinder Beta configurations.

Rolf Meijer of Philips, Holland, developed his famous and vibrationless **rhombic drive** Beta engines in the early 1960s. In 1965 the General Motors Research Labs developed a 7.5kW rhombic drive Stirling engine/generator set GPU-3 (Ground Power Unit) for the US Army. It is described and analyzed in the book by I.Urieli & D.M.Berchowitz – Stirling Cycle Engine Analysis (Adam Hilger, 1984), pages 30 – 40, and since this book is out of print, these pages have been added here for convenience (Appendix D). Andy Ross built a few small rhombic drive air engines – refer to his book: **Making Stirling Engines** (Ross Experimental, 1993).

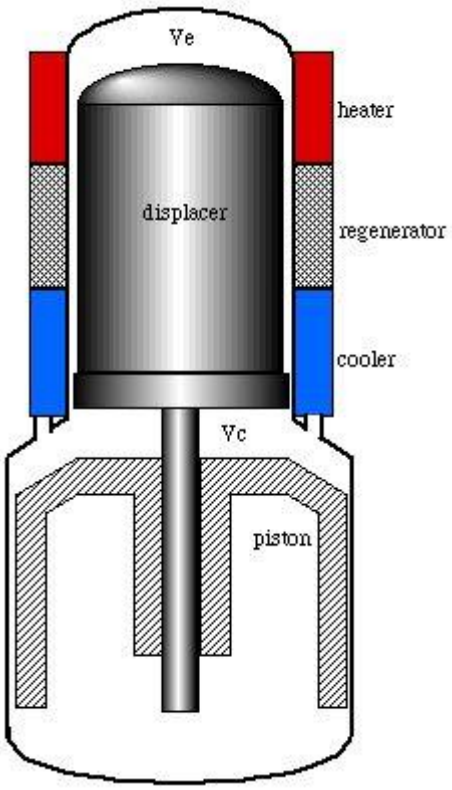
Another interesting website is **Stirling Engine Boats** by **Andrew Hall** describing boats powered mainly by Beta type engines built by modelers in the United Kingdom.

Free Piston Stirling Engines

Probably the most ingenious Stirling engines yet devised are the free-piston engines invented and developed by William Beale at Ohio University in the late 1960s. Legend has it that while teaching about the rhombic drive engine he suddenly realized that "this engine will still run if we simply *throw away* this complex drive mechanism - Eureka!". He then formed the company **Sunpower**, which has been the leader in the development of free-piston Stirling engines and cryocoolers to this day. Most of Sunpower's engines are Beta arrangements and employ no mechanical linkage system. The main aspect of the free piston machine is that the output power can be obtained through a linear alternator, allowing the entire system to be hermetically sealed. In fact, this is the only Stirling configuration to reach

commercialization in any numbers. This is mainly because it avoids the fatal flaws of the crank, proven again and again over the years to be near-insurmountable - sealing and lubrication.

Since 1974 Sunpower has developed Free Piston Stirling Engine/Generators ranging in power levels from $35W_e$ to $7.5kW_e$. Consider for example the EG-1000 engine/generator which is gas fired and has been designed to generate electricity ($1kW_e$) as well as to provide hot water for a private home. The working gas used is helium, which has the advantage of having a low molecular weight and high thermal conductivity compared to air, allowing a significant reduction in size. This engine is shown in the figure below together with a simplified schematic diagram.



The linear electrical generator (not shown in the above schematic) is comprised of powerful rare-earth magnets in the piston cutting a magnetic circuit and coils in the cylinder. This produces 240 Volts at 50 Herz - designed for operation in Europe, and is capable of producing more than one kilowatt of electrical power output at an efficiency of around 90%.

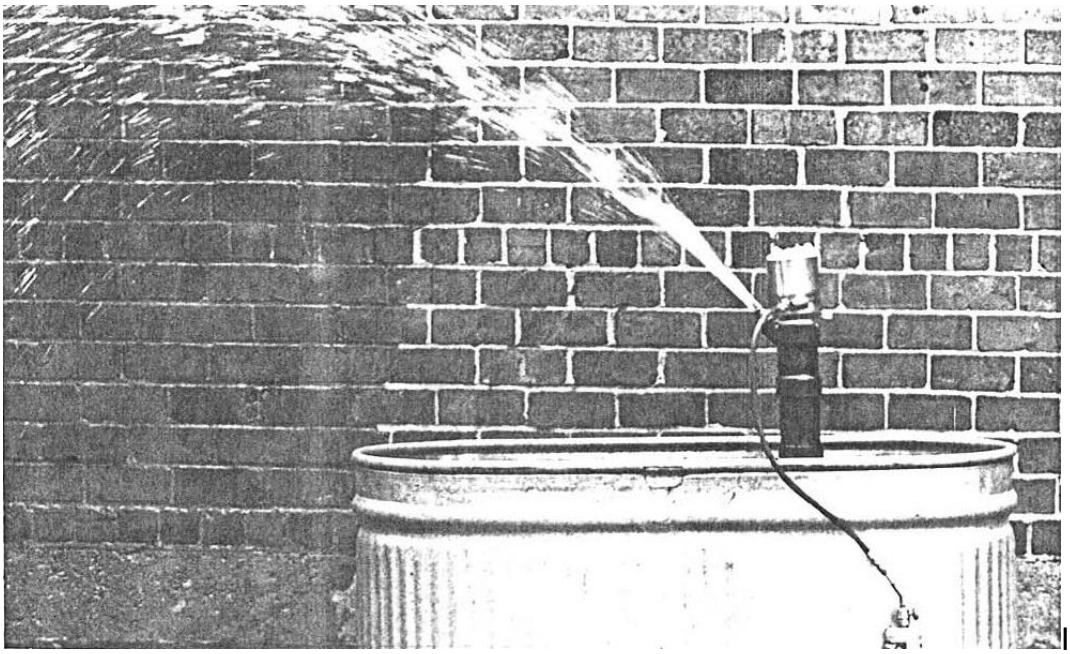
The hot water is provided by operating the cooling water at a temperature of 50°C.

The Sunpower EG-1000 free-piston Stirling engine/generator

In this photograph we see the Sunpower EG-1000 being demonstrated using sawdust pellets as the fuel, and generating more than 1000W of electricity to a light panel. This was done at the Sustainability Fair in the Fairgrounds of Athens Ohio, 2001.

Note that since 1995 the EG-1000 technology was used by British Gas to develop CHP (Combined Heat and Power) units – the 1kW engine/generator is currently manufactured by **Microgen Engine Corporation** (refer to their **History** and **Engine** web pages).

An extremely interesting free piston engine system developed by William Beale is the **free-cylinder water pump**.

 In this engine a heavy internal mass provides the reaction force driving the cylinder which is directly connected to the water pump. It has built in power adjustment and responds to load automatically. All other engines require a transmission and complicated control mechanisms to do this. Furthermore, there is no other mechanical heat engine that I know of that operates from infinite load to zero without either stalling or destroying itself.
Another attractive feature of the free cylinder system is that it can be constructed from inexpensive easy-to-obtain components. In fact, the entire pump housing can be fabricated from ordinary PVC piping and fixtures. The reliability, simplicity and low cost of this engine makes it eminently suitable for application in developing countries, and in the 1970's it was extensively tested both in the field and in the laboratory (Refer to the 1979 presentation by William Beale (Beale W, Rauch J, Lewis R, & Mulej D. (1971). Free cylinder Stirling engines for solar- powered water pumps. *ASME* Pap 71-WA/Sol-11, 8p.)).

Two interesting free piston Stirling powered refrigeration systems have also been investigated – a duplex gas fired refrigerator having only three moving parts, one power piston and two displacer pistons (refer to the paper Penswick, B., & Urieli, I. (1984). Duplex Stirling Machines. *Proceedings of the Intersociety Energy Conversion Engineering Conference*, 1823-1828–1828. and a gas fired free piston CO2 refrigeration system (refer to the paper Berchowitz, D. M., & Kwon, Y. R. (2005). Hermetic gas fired residential heat pump. *In Proc. 8th IEA Heat Pump Conference* (pp. 1-12).

Sunpower was also involved in the manufacture of Stirling cycle cryogenic coolers for liquifying oxygen. Over the years Sunpower has transformed Athens, Ohio into a hotbed of Stirling cycle machine activity, which now includes three R&D/manufacturing companies. In 2013 Sunpower was acquired by **AMETEK, Inc** in Pennsylvania, however continues doing Stirling cycle machine development in Athens, Ohio. Chapter 3 of the book by I.Urieli and D.M.Berchowitz – Stirling Cycle Engine Analysis (Adam Hilger 1984), is entirely about the analysis of Free-Piston machines, and since this book is out of print, this chapter has been added here (Appendix E). Refer also to the paper by R.W.Redlich & D.M.Berchowitz – **Linear dynamics of free-piston Stirling engines** (ImechE 1985), and to the Lecture Notes in Engineering by G.Walker & J.R.Senft – **Free Piston Stirling Engines** (Springer-Verlag 1985, currently available as an eBook).

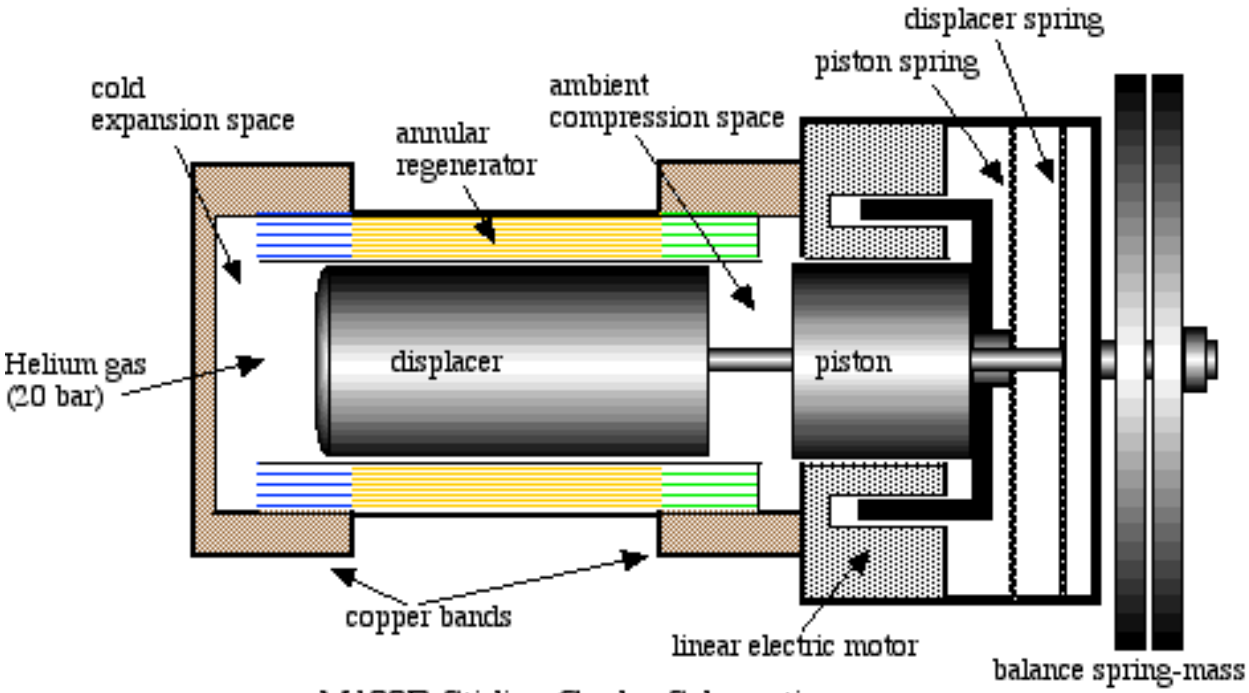
An interesting paper describing the chronology of the development of Free Piston Stirling engine technology was presented by **David Berchowitz** at the 2018 International Stirling Engine Conference (Refer: **A Personal History in the Development of the Modern Stirling Engine**).

The **NASA Glenn Research Center** has been involved in developing free-piston Stirling Engines for deep space missions since the mid-1970's. One of their **experimental units** has recently set a run time record of more than 110,000 hours, running constantly at full power since 2003, and still running without any signs of performance degradation. More recently they have been concentrating on the **Kilopower Reactor Using Stirling TechnologY (KRUSTY)** for up to 10kW power – refer also to the **National Nuclear Security Administration** and their **YouTube video**.

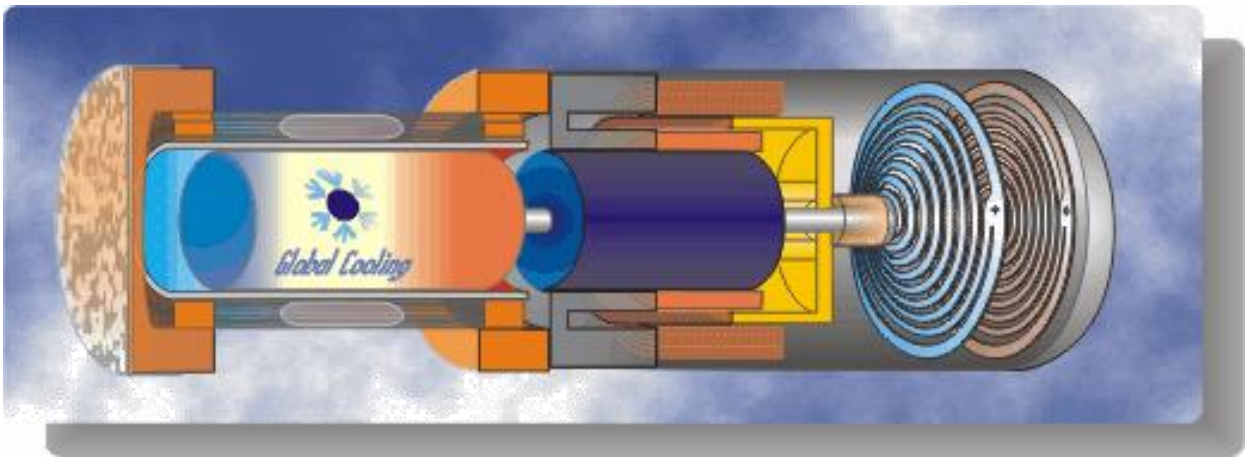
Stirling Technology (note recent company name change: **Combined Energy Technology**) is a spinoff of Sunpower, and was originally formed in order to continue the development and manufacture of the 3.5 kW **ST-5 Air engine**. This large Beta type engine uses a bell crank mechanism to obtain the correct displacer phasing, burns biomass fuel (such as sawdust pellets or rice husks), and can function as a cogeneration unit in rural areas.
Currently Stirling Technology is working with **Microgen Engine Corporation**, an international company which produces the MEC 1kW free-piston engine/generator. Stirling Technology has developed a multifuel burner for the engine and is partnering with Microgen to get various systems into the market.

Global Cooling (currently **Stirling Ultracold**) was a spinoff from Sunpower and was formed in 1995 by **David Berchowitz** mainly in order to develop free-piston Stirling cycle coolers for home refrigerator applications. These systems, apart from being significantly more efficient than regular vapor-compression refrigerators, have the added advantage of being compact, portable units using helium as the working fluid

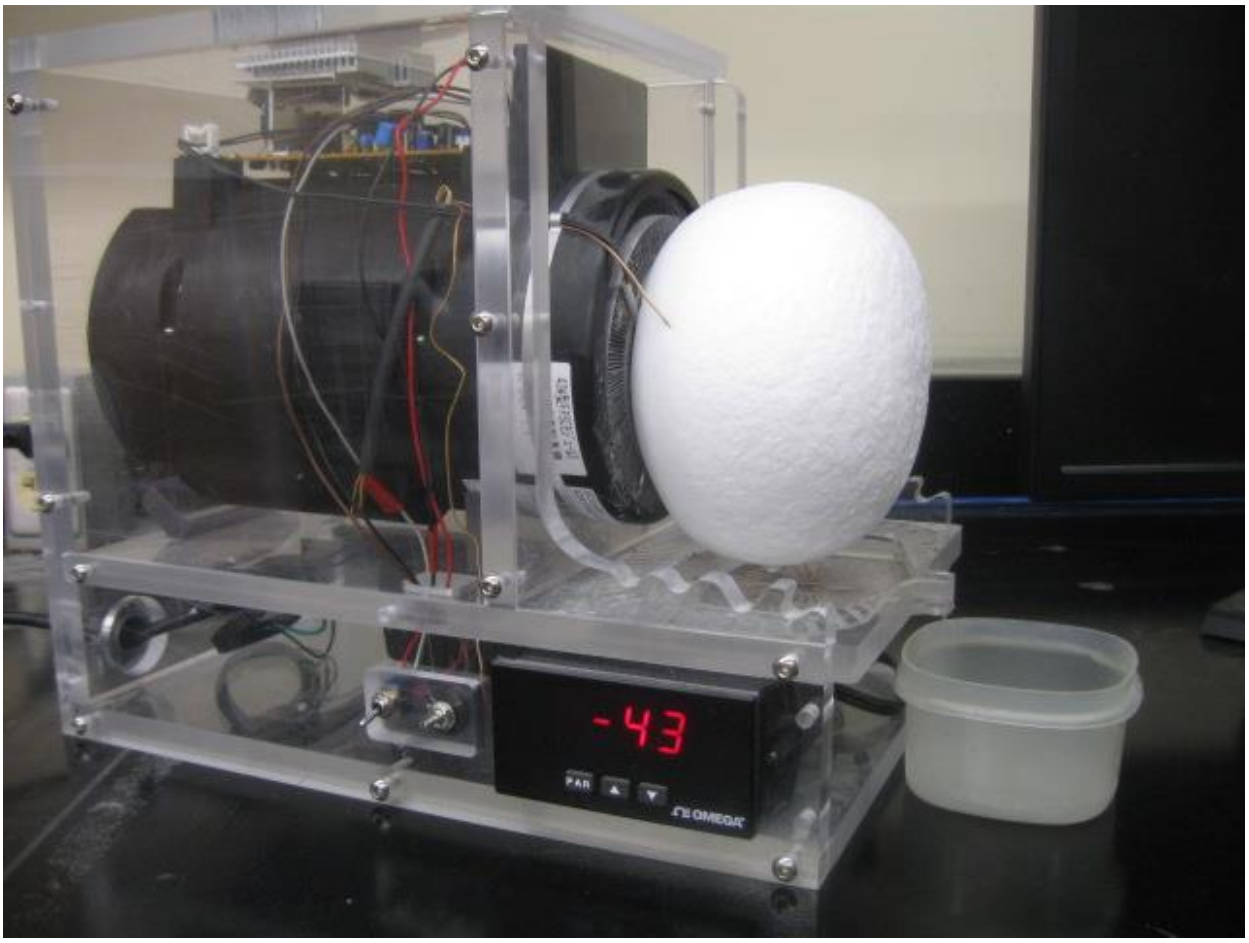
(and not the HFC refrigerants such as R134a, having a Global Warming Potential of 1,300). A schematic diagram followed by an animated schematic of a typical cooler (both courtesy of Global Cooling) are shown below:



M100B Stirling Cooler Schematic



At Ohio University we have a demonstration of the Global Cooling Stirling Cooler shown below. It will normally reach -90°C, however since the ball of ice is covering the entire regenerator section we notice that the temperature has risen to -43°C.

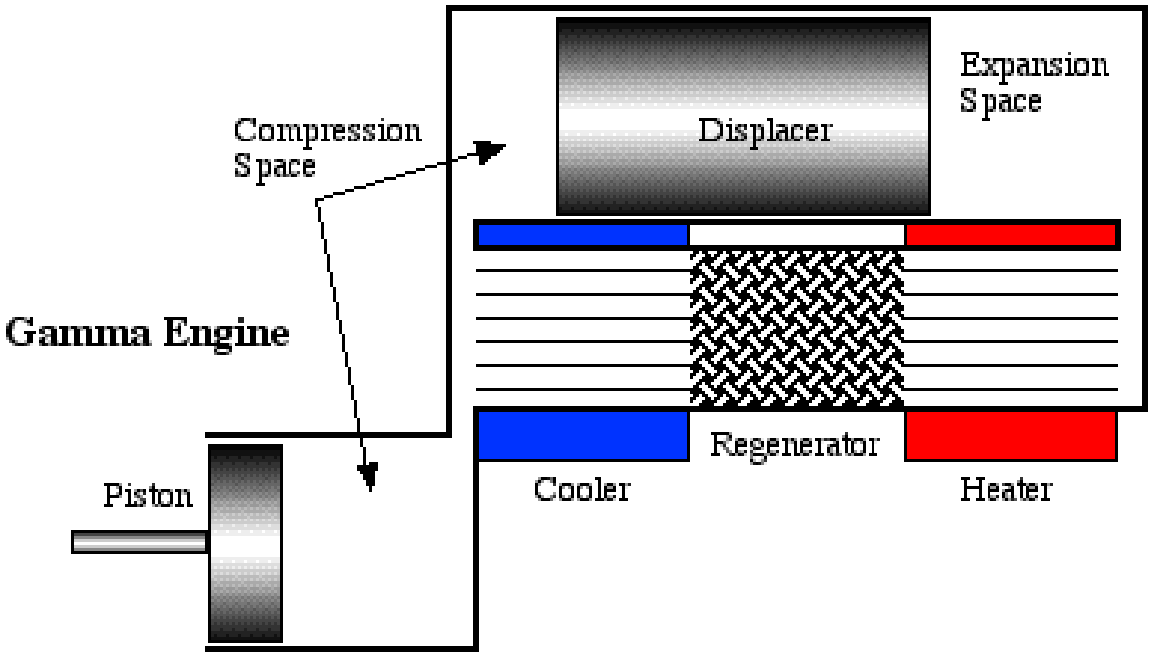


More recently Global Cooling decided to concentrate their development efforts on systems in which there are virtually no competitive systems - cooling between -40°C and -80°C, and they established a new company name: **Stirling Ultracold**.

Update - 2021: Stirling Ultracold's Ultra-Low Temperature (ULT) freezers meets today's unprecedented COVID-19 deployment challenges. Refer to **Walgreens COVID-19 Vaccine Case Study** as well as **Stirling Ultracold to Merge with Biolife Solutions**.

Chapter 2: Basic Engine Configurations. Part C) Gamma Type Engines

Gamma type engines have a displacer and power piston, similar to Beta machines, however in different cylinders. This allows a convenient complete separation between the heat exchangers associated with the displacer cylinder and the compression and expansion work space associated with the piston. Thus they tend to have somewhat larger dead (or unswept) volumes than either the Alpha or Beta engines.



Furthermore during the expansion process some of the expansion must take place in the compression space leading to a reduction of specific power. Gamma engines are therefore used when the advantages of having separate cylinders outweigh the specific power disadvantage.

In the historical web-page of **Robert Stirling** by Robert Sier we notice that one of the engines built by Robert Stirling was a **Gamma** engine. It was donated to the University of Glasgow, and was later used by **William Thomson** (Professor of Natural Philosophy, who later became Lord Kelvin) to show that Stirling's machine worked on a reversible cycle.

A unique Gamma type engine is the Ringbom engine in which the piston is connected to a crankshaft, however the displacer is driven only by the gas pressure

forces on a displacer piston rod. This engine is described in a book **Ringbom Stirling Engines** (Oxford University Press, USA, 1993) by James Senft, formally a professor of mathematics at the University of Wisconsin, River Falls.

James Senft also developed a revolutionary approach to low temperature difference Stirling engine design using the Gamma type engine as described in his book "**An Introduction to Low Temperature Differential Stirling Engines**" (Moriya Press, 1996 – Unfortunately out of print). In this book he describes how to make an engine that runs off the heat of one's hand - a seemingly impossible task. The major innovation of this approach is the extremely large diameter displacer as opposed to an extremely small diameter piston.

Because of the structural convenience of two cylinders that enables a completely independent sizing and construction of the displacer and piston assemblies, the gamma configuration is a favorite among modelers and hobbyists. The website **StirlingSouth** by Roy Rice and Richard Egge, both very innovative modelers, includes a large worldwide selection of engine modelers.

The most amazing and innovative of the Low Temperature (heat-of-your-hand) Gamma type engines are those made by **Kontax Stirling Engines** in the Thames Valley, England. Their website indicates that they produce around thirteen low temperature engines, including the unique **KS90R Black Ross Yoke LTD** engine shown, a twin engine design **KS90T Polished Twin LTD**, and other uniquely innovative engines.	

	Another engine inspired by James Senft is the "heat of your hand" engine, the **model MM-7** Stirling engine made by the **American Stirling Company** - will run on a temperature difference of only 4 degrees C! This is one of a number of innovative gamma type model Stirling engines that they describe on their website, including the **model MM-1** Coffee cup engine.
The **Fizgig** by master modeler Mick Collins. This delightful engine came with a complete set of drawings and instructions which can be downloaded from Mick Collins' website, together with performance characteristics, feedback from modelers who have built the engine, and other interesting information. (Unfortunately the website of Mick Collins is no longer active)	

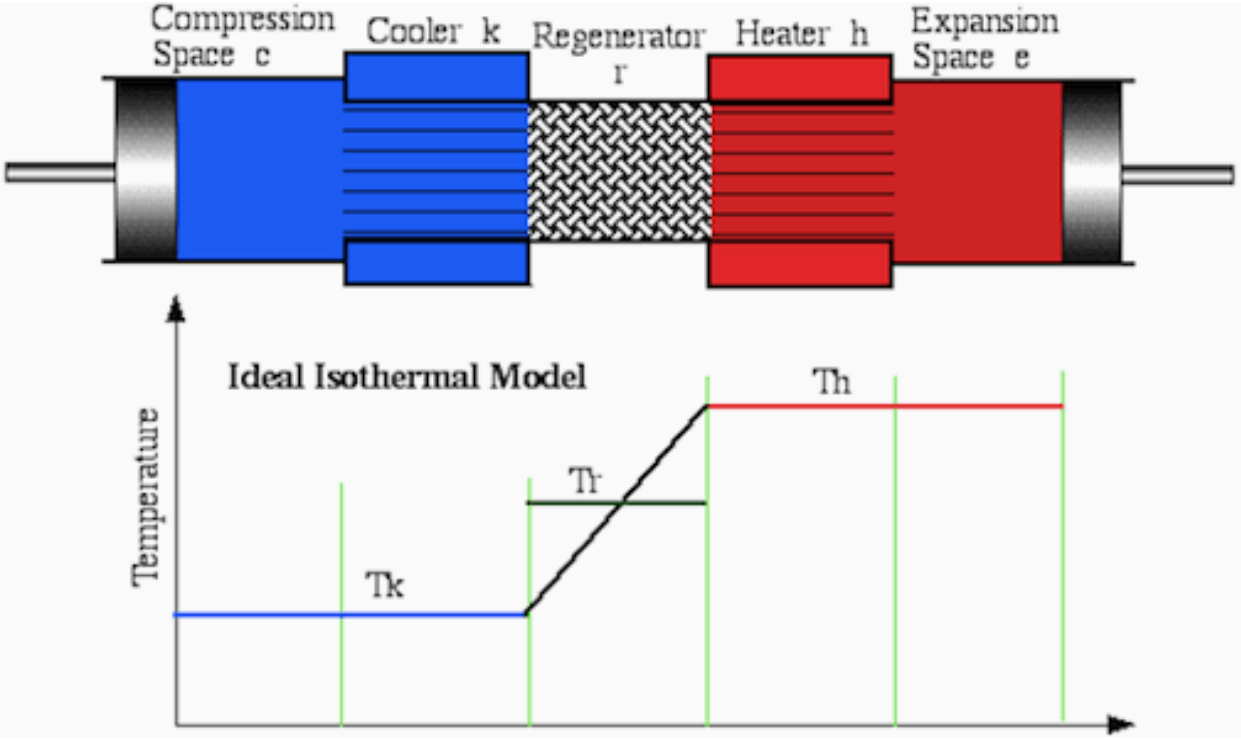
Another approach is that of Boyette White, who converted a Briggs and Stratton engine into a Gamma type Stirling engine by adding the displacer and heat exchanger sections. Unfortunately, Boyette's website is no longer active, however I did want to indicate the option of converting a regular piston IC engine to a Gamma type Stirling engine.

The **Thumper** engine by **Richard Egge** is a Ringbom type engine in which the displacer piston is moved by pressure forces, and have no mechanical connection to the crankshaft. Richard also made unique **double-Ringbom** and **quad-Ringbom** engines. One of the engines made by **Roy Rice** is **Aquatap**, a water cooled Ringbom type engine inspired by James Senft.

Chapter 3: Ideal Isothermal Analysis.
Part A) Ideal Isothermal Analysis

The invention of the Stirling engine in 1816 was well in advance of all pertinent scientific knowledge of that time. The first attempt at an analysis of the cycle was published in 1871 by Gustav Schmidt. Much as the Otto cycle has become the classic Air standard cycle to describe the spark ignition engine, the cycle described by Schmidt has become the classic ideal Stirling cycle. This is unfortunately mainly because the Schmidt analysis yields a closed form solution rather than its ability to predict the real cycle, however we use it as a starting point to guide us ultimately to a more realistic approach.

Consider the Ideal Isothermal model of a Stirling engine as shown below.



The principal assumption of the analysis is that the gas in the expansion space and the heater is at the constant upper source temperature and the gas in the compression space and the cooler is at the constant lower sink temperature. This isothermal assumption makes it possible to generate a simple expression for the working gas pressure as a function of the volume variations. This expression may then be used to investigate how different drive mechanisms affect the output power. To obtain closed form solutions, Schmidt assumed that the volumes of the working spaces vary sinusoidally.

The assumption of isothermal working spaces and heat exchangers implies that the heat exchangers (including the regenerator) are perfectly effective, with a spatial temperature distribution as indicated in the figure above. The engine is considered as a five component serially connected model, consisting respectively of a compression space c, cooler k, regenerator r, heater h and expansion space e. Each component is considered as a homogeneous entity or cell, the gas therein being represented by its instantaneous mass m, absolute temperature T, volume V and pressure p, with the suffix c, k, r, h, and e identifying the specific cell.

The starting point of the analysis is that the total mass of gas in the machine is constant, thus:

M = mc + mk + mr + mh + me

Substituting the ideal gas law given by

m = p V / R T

we obtain

M = p (Vc / Tk + Vk / Tk + Vr / Tr + Vh / Th + Ve / Th) / R

For the assumed linear temperature distribution in the regenerator we can show that the **effective regenerator temperature** (Appendix F) Tr is given by

Tr = (Th - Tk) / **ln**(Th / Tk)

Thus given the volume variations Vc and Ve we can solve the above equation for pressure p as a function of Vc and Ve.

$$p = \frac{M R}{\left(\frac{Vc}{Tk} + \frac{Vk}{Tk} + \frac{Vr \ \mathbf{ln}(Th / Tk)}{(Th - Tk)} + \frac{Vh}{Th} + \frac{Ve}{Th} \right)}$$

The work done by the system over a complete cycle is given respectively by the cyclic integral of p dV

$$W = We + Wc = \oint p \, dVc + \oint p \, dVe = \oint p \left(\frac{dVc}{d\theta} + \frac{dVe}{d\theta} \right) d\theta$$

On evaluating the **heat transferred over a complete cycle** (Appendix G) to the various cells we find remarkably that the cyclic heat transferred to all three heat exchanger cells is zero! Thus:

Qc = Wc
Qe = We
Qk = 0
Qh = 0
Qr = 0

This rather startling result implies that all the heat exchangers in the ideal Stirling engine are redundant since all the external heat transfer occurs across the boundaries of the compression and expansion spaces. This apparent paradox is a direct result of the definition of the Ideal Isothermal model in which the compression and expansion spaces are maintained at the respective cooler and heater temperatures. Obviously this cannot be correct, since the cylinder walls are not designed for heat transfer. In real machines the compression and expansion spaces will tend to be adiabatic rather than isothermal, which implies that the net heat transferred over the cycle must be provided by the heat exchangers. This will be resolved when we consider the **Ideal Adiabatic Model** in the next section.

The set of pertinent equations is shown in the following table.

$p = \frac{M R}{\left(\frac{Vc}{Tk} + \frac{Vk}{Tk} + \frac{Vr \ln(Th/Tk)}{(Th - Tk)} + \frac{Vh}{Th} + \frac{Ve}{Th} \right)}$	Pressure
$Qe = We = \oint p \left(\frac{dVe}{d\theta} \right) d\theta$	Heat transferred
$Qc = Wc = \oint p \left(\frac{dVc}{d\theta} \right) d\theta$	
$W = Wc + We$	Work done
$\eta = W / Qe$	Efficiency

In order to solve these equations we need to specify the working space volume variations Vc and Ve as well as the respective volume derivatives **d**Vc and **d**Ve with

respect to crankangle θ. One of the case studies of this course is the **Ross Yoke-drive** engine for which we have analyzed the **volume variations** (Appendix H), thus the above equation set can be solved by numerical integration. In 1871 Gustav Schmidt published an analysis in which he obtained closed form solutions for the above equation set for the special case of sinusoidal volume variations. We continue now with the **Schmidt Analysis**.

Chapter 3: Ideal Isothermal Analysis. Part B) The Schmidt Closed Form Solution

In the **previous section** we derived the basic set of equations which describe the Ideal Isothermal model, as shown in the following table.

$$p = \frac{M\,R}{\left(\frac{Vc}{Tk} + \frac{Vk}{Tk} + \frac{Vr\,\ln(Th/Tk)}{(Th - Tk)} + \frac{Vh}{Th} + \frac{Ve}{Th}\right)}$$ Pressure

$$Qe = We = \oint p\left(\frac{dVe}{d\theta}\right)d\theta$$

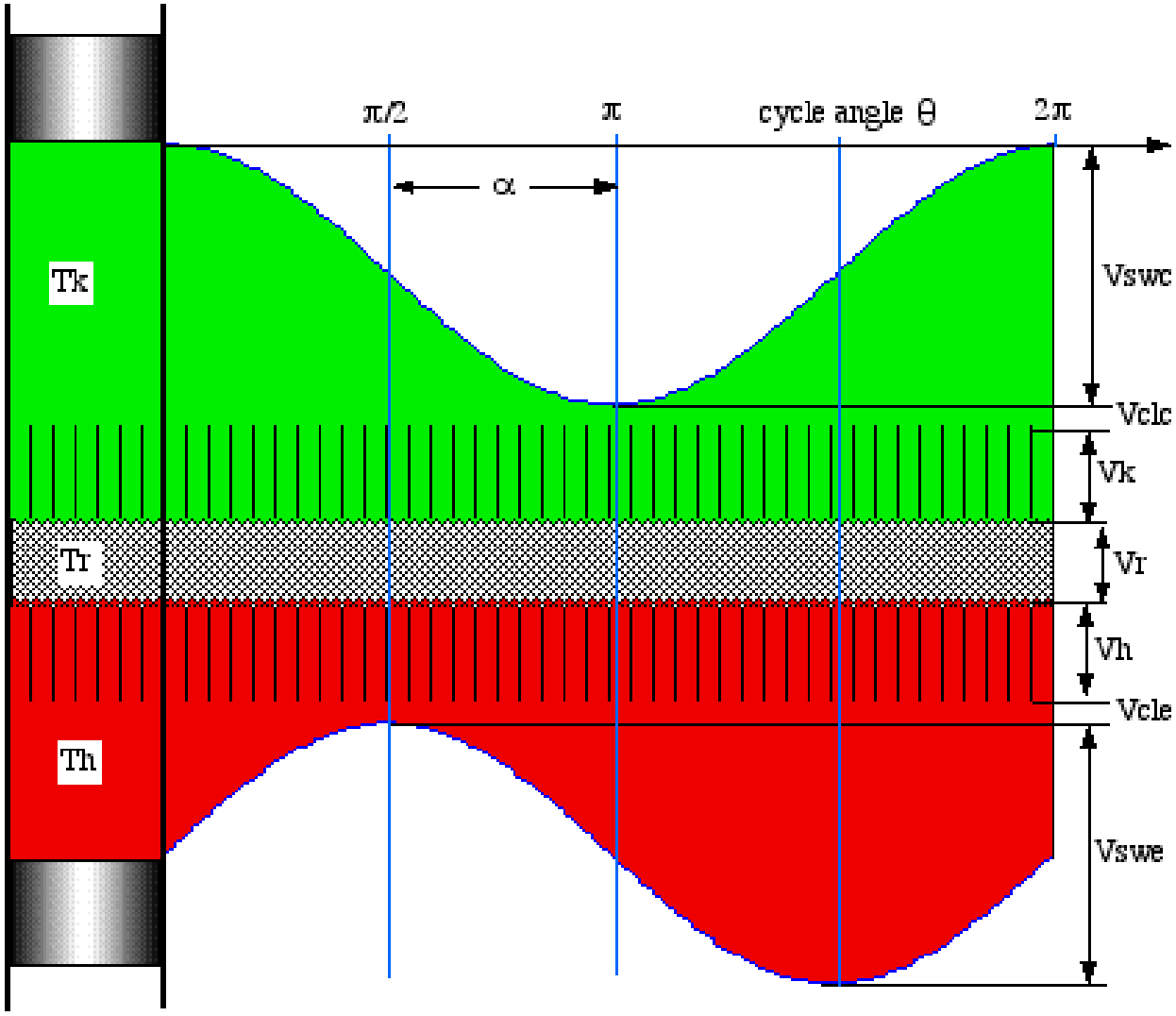
$$Qc = Wc = \oint p\left(\frac{dVc}{d\theta}\right)d\theta$$ Heat transferred

$$W = Wc + We$$ Work done

$$\eta = W / Qe$$ Efficiency

Gustav Schmidt of the German Polytechnic Institute of Prague Published an analysis in 1871 in which he obtained closed form solutions of these equations for the special case of sinusoidal volume variations of the working spaces with respect to the cycle angle θ. This analysis was published in detail in the **appendix** (Appendix I) of the book by Urieli & Berchowitz, "Stirling Cycle Machine Analysis", Adam Hilger 1984, and is repeated here in a condensed form for convenience.

Consider the following diagram showing the volume variations of the compression and expansion spaces (Vc and Ve) for an Alpha engine over a single cycle. Notice the phase advance angle α of the expansion space volume variation with respect to the compression space volume variation:



From the above diagram we note that the sinusoidal volume variations of the compression and expansion spaces are respectively as follows:

Vc = Vclc + Vswc (1 + **cos** θ) / 2

Ve = Vcle + Vswe (1 + **cos**(θ + α)) / 2

where Vcl and Vsw represent respectively clearence and swept volumes, and θ is the cycle angle.

Note that the following Schmidt analysis is done specifically for an Alpha type engine. For Beta or Gamma type engines we examine the equivalent **sinusoidal volume variations** (Appendix J) to determine the effective Vclc, Vswc, Vcle, Vswe, and α required for this analysis.

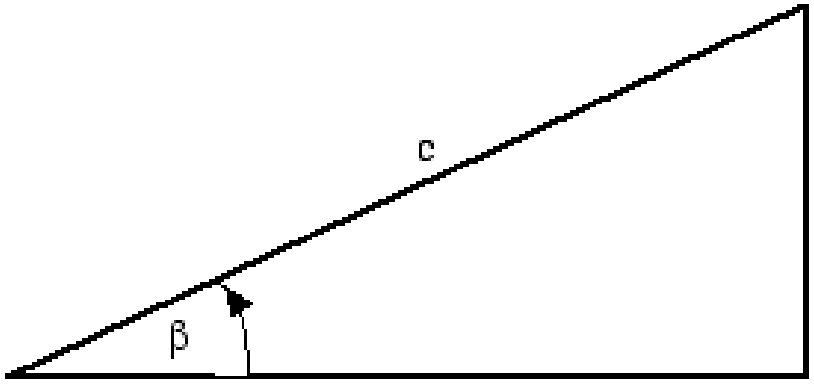
Substituting for Vc and Ve in the pressure equation above and simplifying we obtain

$$p = \frac{M\,R}{\left[s + \left(\frac{Vswe\,\cos\alpha}{2\,Th} + \frac{Vswc}{2\,Tk} \right) \cos\theta - \left(\frac{Vswe\,\sin\alpha}{2\,Th} \right) \sin\theta \right]}$$

where

$$s = \left[\frac{Vswc}{2\,Tk} + \frac{Vclc}{Tk} + \frac{Vk}{Tk} + \frac{Vr\,\ln(Th/Tk)}{(Th - Tk)} + \frac{Vh}{Th} + \frac{Vswe}{2\,Th} + \frac{Vcle}{Te} \right]$$

In order to simplify the pressure equation we now consider a trigonometric substitution of β and c as defined by the following right-angled triangle



$$c\,\sin\beta = \frac{Vswe\,\sin\alpha}{2\,Th}$$

$$c\,\cos\beta = \frac{Vswe\,\cos\alpha}{2\,Th} + \frac{Vswc}{2\,Tk}$$

$$\beta = \tan^{-1}\left(\frac{Vswe\,\sin\alpha / Th}{Vswe\,\cos\alpha / Th + Vswc / Tk} \right)$$

$$c = \frac{1}{2}\sqrt{\left(\frac{Vswe}{Th} \right)^2 + 2\,\frac{Vswe}{Th}\,\frac{Vswc}{Tk}\,\cos\alpha + \left(\frac{Vswc}{Tk} \right)^2}$$

Substituting for β and c in the pressure equation above and simplifying

$$p = \frac{M\,R}{s\,(1 + b\,\cos\phi)}$$

where

$\phi = \theta + \beta$

$b = c / s$

The maximum and minimum values of pressure can now be evaluated for the extreme values of **cos** ϕ

$$p_{\mathbf{min}} = \frac{M\,R}{s\,(1+b)}$$

$$p_{\mathbf{max}} = \frac{M\,R}{s\,(1-b)}$$

The average pressure over the cycle is given by

$$p_{mean} = \frac{1}{2\pi}\int_{0}^{2\pi} p\;\mathbf{d\phi}$$

$$p_{mean} = \frac{M\,R}{2\,\pi\,s}\int_{0}^{2\pi}\frac{1}{(1+b\,\mathbf{cos}\,\phi)}\;\mathbf{d\phi}$$

From tables of integrals, this reduces to

$$p_{mean} = M\,R\Big/\left(s\sqrt{1-b^2}\right)$$

This equation is the most convenient way of relating the total mass of working gas in the cycle to the more conveniently specified mean operating pressure.

The net work done by the engine is the sum of the work done by the compression and expansion spaces. Over a complete cycle

$$Qe = We = \int_0^{2\pi} \left(p \frac{d\,Ve}{d\theta} \right) d\theta$$

$$Qc = Wc = \int_0^{2\pi} \left(p \frac{d\,Vc}{d\theta} \right) d\theta$$

W = Wc + We

The volume derivatives are obtained by differentiating Vc and Ve above

$$\frac{d\,Vc}{d\theta} = -\frac{1}{2} Vswc \sin \theta$$

$$\frac{d\,Ve}{d\theta} = -\frac{1}{2} Vswe \sin(\theta + \alpha)$$

Substituting these and the pressure equation into the equations for Wc and We

$$Wc = -\frac{Vswc\ M\ R}{2\ s} \int_0^{2\pi} \frac{\sin \theta}{1 + b \cos(\beta + \theta)}\ d\theta$$

$$We = -\frac{Vswe\ M\ R}{2\ s} \int_0^{2\pi} \frac{\sin(\theta + \alpha)}{1 + b \cos(\beta + \theta)}\ d\theta$$

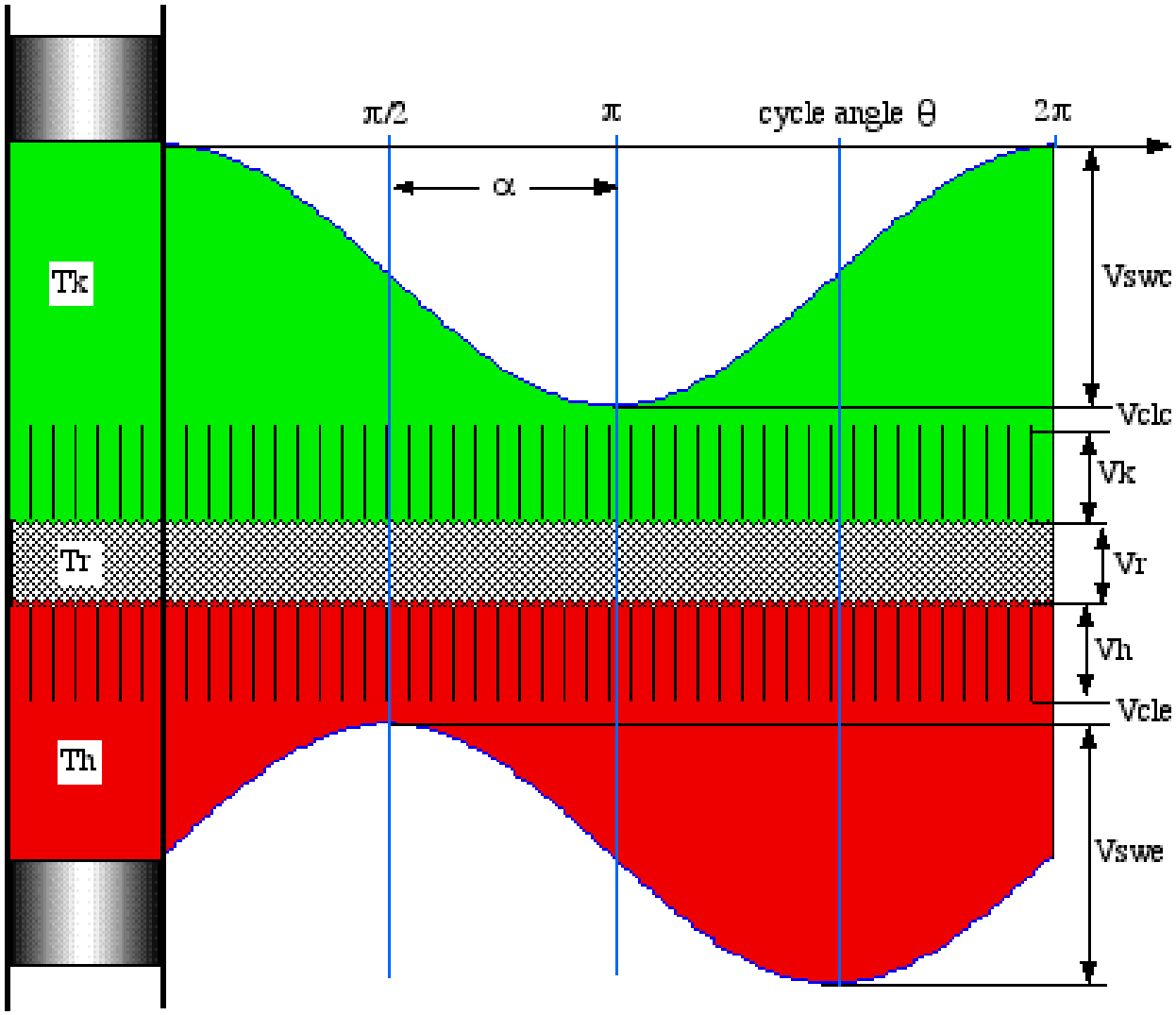
The solution of these integrals requires the judicious use of tables of integrals and is done in the **appendix** (Appendix I) of the book by Urieli & Berchowitz, "Stirling Cycle Machine Analysis", Adam Hilger 1984. Solving these integrals we obtain:

$$Wc = \pi\ Vswc\ p_{mean}\ \sin \beta\ (\sqrt{1 - b^2} - 1)/b$$

$$We = \pi\ Vswe\ p_{mean}\ \sin(\beta - \alpha)\ (\sqrt{1 - b^2} - 1)/b$$

Schmidt Analysis - Equation Summary

Recall that the Schmidt analysis was done specifically for an Alpha type engine. For Beta or Gamma type engines we examine the equivalent **sinusoidal volume variations** (Appendix J) to determine the effective Vclc, Vswc, Vcle, Vswe, and α required for this analysis.



$$Vc = Vclc + Vswc\ (1+\cos\ \theta)/2$$

$$Ve = Vcle + Vswe\ [1+\cos(\theta+\alpha)]/2$$

$$Wc = Qc = \pi\ Vswc\ p_{mean}\ \sin\beta\ (\sqrt{1-b^2}-1)/b$$

$$We = Qe = \pi\ Vswe\ p_{mean}\ \sin(\beta-\alpha)\ (\sqrt{1-b^2}-1)/b$$

$$W = Wc + We$$

$$\eta\ =\ W/Qe\ =\ 1-Tk/Th$$

where

$$\tan\beta = \left(\frac{Vswe\ \sin\alpha/Th}{Vswe\ \cos\alpha/Th + Vswc/Tk}\right)$$

$$c = \frac{1}{2}\sqrt{\left(\frac{Vswe}{Th}\right)^2 + 2\frac{Vswe}{Th}\frac{Vswc}{Tk}\cos\alpha + \left(\frac{Vswc}{Tk}\right)^2}$$

$$s = \left[\frac{Vswc}{2\ Tk} + \frac{Vclc}{Tk} + \frac{Vk}{Tk} + \frac{Vr\ \ln(Th/Tk)}{(Th-Tk)} + \frac{Vh}{Th} + \frac{Vswe}{2\ Th} + \frac{Vcle}{Te}\right]$$

$$b = c/s$$

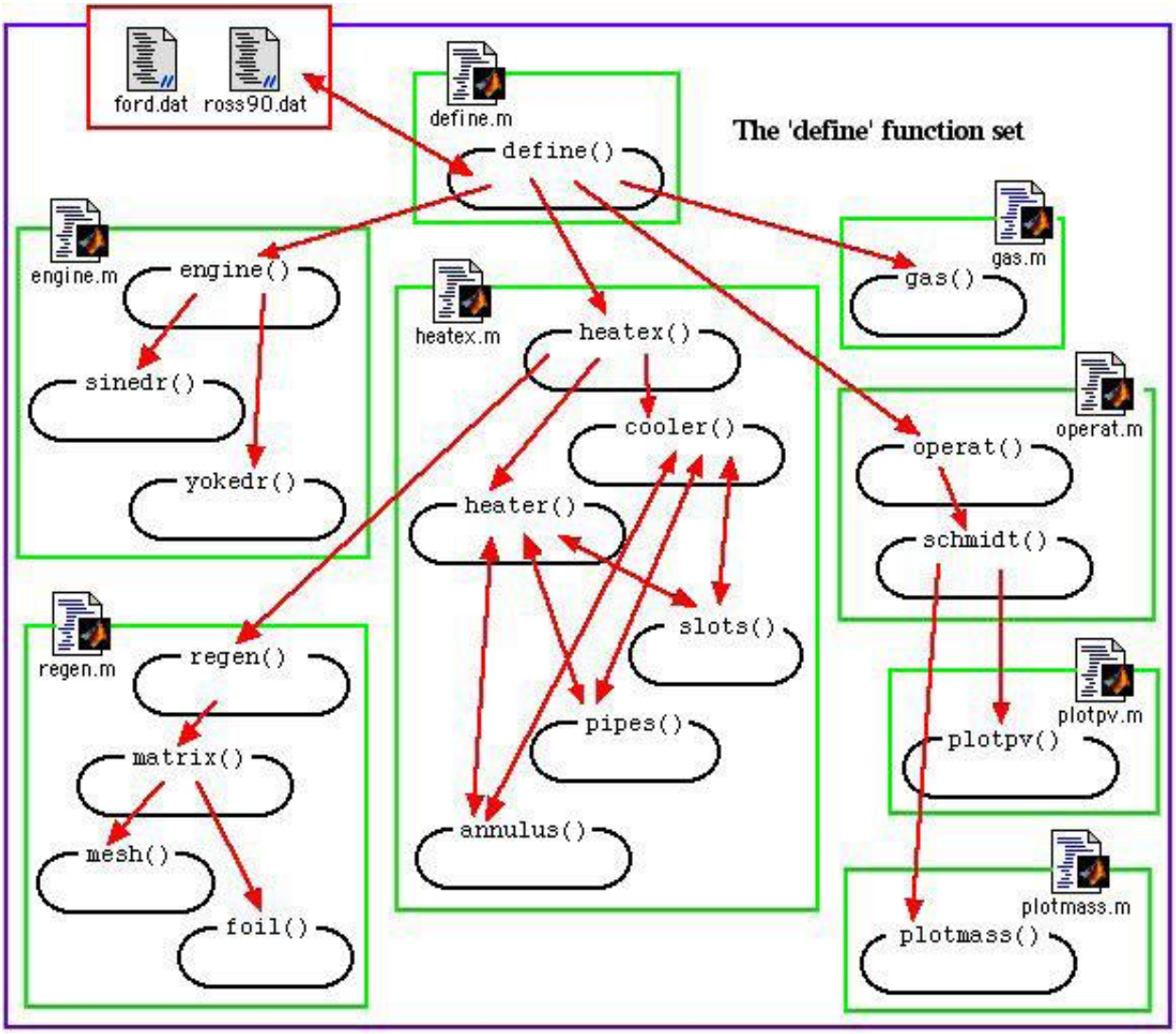
$$p_{mean} = M\ R \Big/ \left(s\sqrt{1-b^2}\right)$$

Chapter 3: Ideal Isothermal Analysis. Part C) Function Set 'Define'

Function set 'define' to define a specific Stirling engine configuration

One of the main purposes of this web resource is to develop a computer simulation of a Stirling engine. Before any simulation can take place we need to define a specific configuration. Thus we need to specify the type and geometry of the engine, the three heat exchangers, the working gas and the operating conditions. Once these have been specified then we can go to three levels of simulation: an Isothermal Schmidt analysis, an Ideal Adiabatic analysis, and a Simple analysis to evaluate the effect of non ideal heat exchangers.

In this section we consider the function set **define**, which includes the system definition and the Schmidt analysis. The program system is written in the MATLAB language and includes eight m-files as shown in the functional block diagram:



This function set has the following purposes:

- Specify values for the global variables in the **define function** (Appendix K) needed to define a specific engine configuration. The **engine function** (Appendix L) includes three alpha engines as case studies - the classic **Ford-Philips 4-215** (Appendix M), the **D-90 Ross Yoke-drive** (Appendix N), and the **Ross Rocker-V** (new - not in the block diagram) (Appendix O). The Ford engine includes tubular heat exchangers and a matrix mesh regenerator, and the D-90 Yoke-drive and the RockerV engines have annular and slotted heat exchangers and wrapped foil regenerators, so the three machines cover all the **heat exchanger** (Appendix P) and **regenerator** (Appendix Q) options of current Stirling engine designs.
- Specify values for the **working gas** (Appendix R) and **operating conditions and Schmidt analysis** (Appendix S) of the engine to evaluate nominal enclosed mass of working gas as well as the Isothermal Schmidt performance of the machine. There is the optional capability of doing a pV or p-theta plot.

- Do a particle trajectory plot using the **plotmass function** (Appendix T), showing particles of equal mass flowing through the engine over a cycle using Natural Coordinates, as defined by Allan Organ: **'Natural' coordinates for analysis of the practical Stirling cycle** (ImechE, 1992). Refer also to Chapter 9 of his book: **The Regenerator and the Stirling Engine** (John Wiley, 1997). This plot enables a better understanding of the process.

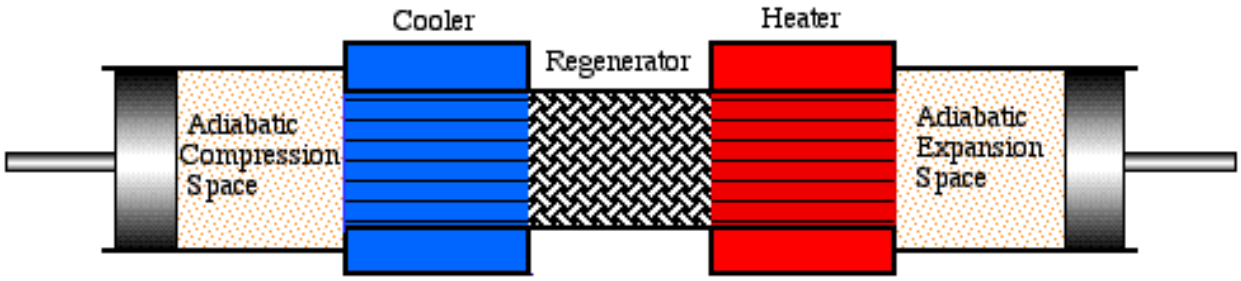
The nineteen functions comprising the set are included in the eight m-files: **define.m**, **engine.m**, **heatex.m**, **regen.m**, **gas.m**, and **operat.m**, and the two plotting routines **plotpv.m** and **plotmass.m**. All the global variables required for the simulation are declared in the header m-file **define.m**, and the main purpose of invoking the function set '**define**' is to assign values to these global variables. These will subsequently be used in the 'adiabatic' and 'simple' simulations following. Notice that the function m-file **engine.m** includes three engine configuration functions: **sindrive** for a basic sinusoidal drive (eg the swashplate drive in the Ford-Philips engine), **yokedrive** for a Ross yoke-drive machine, and **rockerVdrive** for a Ross RockerV engine. This is a tutorial system and user is expected to augment it as required for any specific requirements. To this end the program system has been written in a universal and uniform style with strict rules regarding structure, variable names, and commenting, thus making it a self documenting system. All of the eight m-files shown above can be downloaded and copied onto any system which has MATLAB installed, or the complete program set can be downloaded in compressed format **sea.zip** (**sea** = **s**tirling **e**ngine **a**nalysis)

A typical execution output of the program is shown in the output file **define.out** for the D-90 Ross yoke drive engine, as well as the **pV**, **p-theta**, and **particle mass flow** diagrams. Note that this set of routines was not designed to be executed independently, but as a precursor to the Ideal Adiabatic simulation or the Simple simulation routines, which will be discussed in detail later.

Note that the engine modules are for Alpha machines, including a Sinusoidal drive, a Ross Yoke-drive and a Rocker-V drive machine. The heat exchanger types include tubular, annular gap, and slot heat exchangers, and the regenerator matrix types include screen mesh and rolled foil matrices. Working gas types include air, helium, and hydrogen.
Update 2016: I have decided to include a free piston **Beta drive engine** (Appendix L) configuration assuming that both the piston and displacer motions are sinusoidal. The **betadrive** function has been added to the function m-file **engine.m**.

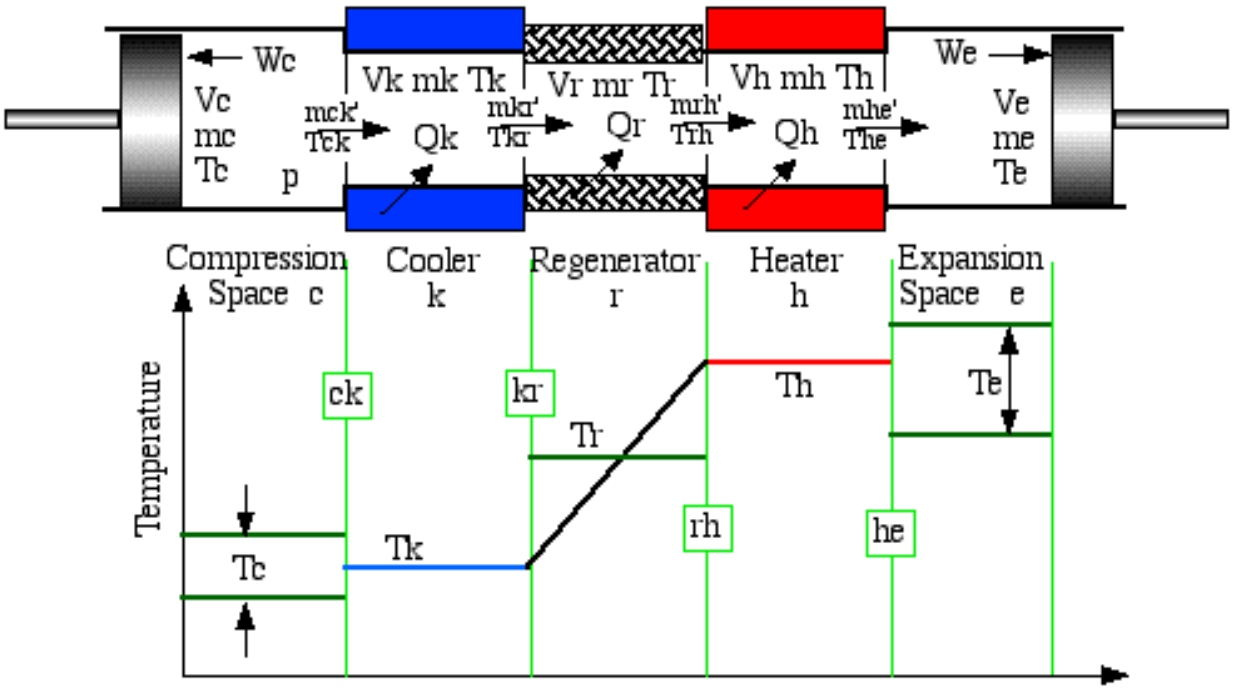
Chapter 4: Ideal Adiabatic Analysis. Part A) Development of the Ideal Adiabatic Equation Set

In the **previous section** we considered an ideal Stirling engine model in which the compression and expansion spaces were maintained at the respective cooler and heater temperatures. This led to the paradoxical situation that neither the heater nor the cooler contributed any net heat transfer over the cycle and hence were redundant. All the required heat transfer occurred across the boundaries of the isothermal working spaces. Obviously this cannot be correct, since the cylinder walls are not designed for heat transfer. In real machines the working spaces will tend to be adiabatic rather than isothermal, which implies that the net heat transferred over the cycle must be provided by the heat exchangers. We thus consider an alternative ideal model for Stirling cycle engines, the Ideal Adiabatic model.



Ideal Adiabatic Model

As before the engine is configured as a five component serially connected model having perfectly effective heat exchangers (including the regenerator) and in this respect is similar to the Ideal Isothermal model defined previously. However both the compression and expansion spaces are adiabatic, in which no heat is transferred to the surroundings. In the following diagram we define the Ideal Adiabatic model nomenclature. Thus we have a single suffix (c, k, r, h, e) representing the five cells, and a double suffix (ck, kr, rh, he) representing the four interfaces between the cells. Enthalpy is transported across the interfaces in terms of a mass flow rate m' and an upstream temperature T. The arrows on the interfaces represent the positive direction of flow, arbitrarily defined from the compression space to the expansion space.



Notice from the temperature distribution diagram that the temperature in the compression and expansion spaces (Tc and Te) are not constant, but vary over the cycle in accordance with the adiabatic compression and expansion occurring in the working spaces. Thus the enthalpies flowing across the interfaces ck and he carry the respective adjacent upstream cell temperatures, hence temperatures Tck and The are conditional on the direction of flow and are defined algorithmically as follows:

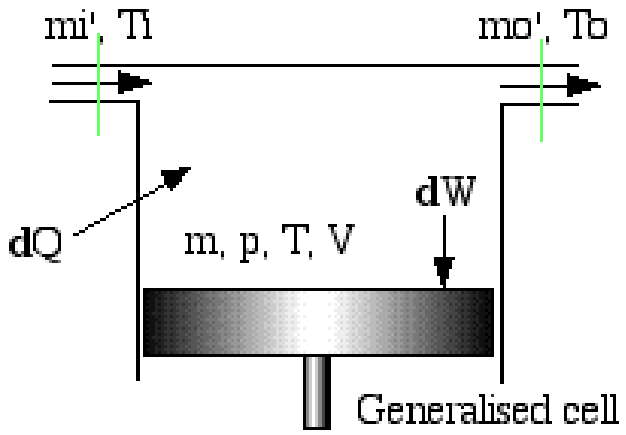
if mck' > 0 **then** Tck = Tc **else** Tck = Tk
if mhe' > 0 **then** The = Th **else** The = Te

In the ideal model there is no gas leakage, the total mass of gas M in the system is constant, and there is no pressure drop, hence p is not suffixed and represents the instantaneous pressure throughout the system.

Work W is done on the surroundings by virtue of the varying volumes of the working spaces Vc and Ve, and heat Qk and Qh is transferred from the external environment to the working gas in the cooler and heater cells, respectively. The regenerator is externally adiabatic, heat Qr being transferred internally from the regenerator matrix to the gas flowing through the regenerator void volume Vr.

Development of the equation set

The general approach for deriving the equation set is to apply the equations of energy and state to each of the cells. The resulting equations are linked by applying the continuity equation across the entire system. Consider first the energy equation applied to a generalised cell which may either be reduced to a working space cell or a heat exchanger cell. Enthalpy is transported into the cell by means of mass flow mi' and temperature Ti, and out of the cell by means of mass flow mo' and temperature To. The derivative operator is denoted by **d**, thus for example **d**m refers to the mass derivative **d**m/**d**θ, where θ is the cycle angle.



The word statement of the energy equation for the working gas in the generalised cell is

rate of heat transfer into the cell	+	net enthalpy convected into the cell	=	rate of work done on the surroundings	+	rate of increase of internal energy in the cell

Mathematically, this word statement becomes

dQ + (cp Ti mi' - cp To mo') = **d**W + cv **d**(m T)

where cp and cv are the specific heat capacities of the gas at constant pressure and constant volume respectively. This equation is the well known classical form of the energy equation for non steady flow in which kinetic and potential energy terms have been neglected.

We assume that the working gas is ideal. This is a reasonable assumption for Stirling engines since the working gas processes are far removed from the gas critical point. The equation of state for each cell is presented in both its standard and differential form as follows

$$p\,V = m\,R\,T$$

$$\mathbf{d}P / p + \mathbf{d}V / V = \mathbf{d}m / m + \mathbf{d}T / T$$

The starting point of the analysis is that the total mass of gas in the machine is constant, thus:

$$mc + mk + mr + mh + me = M$$

Substituting for the mass in each cell from the ideal gas law above

$$p\,(Vc / Tc + Vk / Tk + Vr / Tr + Vh / Th + Ve / Te) / R = M$$

where for the assumed linear temperature profile in the regenerator the **mean effective temperature** Tr (Appendix F) is equal to the log mean temperature difference $Tr = (Th - Tk) / \mathbf{ln}(Th / Tk)$. Solving the above equation for pressure

$$p = M\,R\,/(Vc / Tc + Vk / Tk + Vr / Tr + Vh / Th + Ve / Te)$$

Differentiating the equation for mass above

$$\mathbf{d}mc + \mathbf{d}mk + \mathbf{d}mr + \mathbf{d}mh + \mathbf{d}me = 0$$

For all the heat exchanger cells, since the respective volumes and temperatures are constant, the differential form of the equation of state reduces to

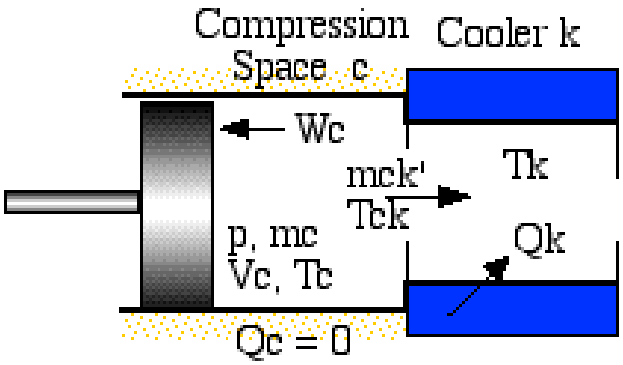
$$\mathbf{d}m / m = \mathbf{d}p / p$$

$$\mathbf{d}m = \mathbf{d}p\ m / p = (\mathbf{d}p / R)\ V / T$$

Substituting in the mass equation above

$$\mathbf{d}mc + \mathbf{d}me + (\mathbf{d}p / R)\,(Vk / Tk + Vr / Tr + Vh / Th) = 0$$

We wish to eliminate **d**mc and **d**me in the above equation so as to obtain an explicit equation in **d**p. Consider the adiabatic compression space (**d**Qc = 0).



Applying the above energy equation to this space we obtain

-cp Tck mck' = **d**Wc + cv **d**(mc Tc)

From continuity considerations the rate of accumulation of gas **d**mc is equal to the mass inflow of gas given by -mck', and the work donw **d**Wc is given by p **d**Vc, thus

cp Tck **d**mc = p **d**Vc + cv **d**(mc Tc)

Substituting the ideal gas relations p Vc = mc R Tc, cp - cv = R, and cp / cv = γ, and simplifying

dmc = (p **d**Vc + Vc **d**p / γ) / (R Tck)

Similarly for the expansion space

dme = (p **d**Ve + Ve **d**p / γ) / (R The)

Substituting for **d**mc and **d**me above and simplifying

$$\mathbf{d}p = \frac{-\gamma\, p\, (\mathbf{d}Vc / Tck + \mathbf{d}Ve / The)}{[Vc / Tck + \gamma\,(Vk / Tk + Vr / Tr + Vh / Th) + Ve / The\,]}$$

From the differential form of the equation of state above we obtain relations **d**Tc and **d**Te

dTc = Tc (**d**p / p + **d**Vc / Vc - **d**mc / mc)

dTe = Te (**d**p / p + **d**Ve / Ve - **d**me / me)

Applying the energy equation above to each of the heat exchanger cells (**d**W = 0, T constant) and substituting for the equation of state for a heat exchanger cell (**d**m = **d**p m / p = (**d**p / R) V / T)

dQ + (cp Ti mi' - cp To mo') = cv T **d**m = V **d**p cv / R

Thus for the three heat exchanger cells we obtain

dQk = Vk **d**p cv / R - cp (Tck mck' - Tkr mkr')

dQr = Vr **d**p cv / R - cp (Tkr mkr' - Trh mrh')

dQh = Vh **d**p cv / R - cp (Trh mrh' - The mhe')

We note that since the heat exchangers are isothermal and the regenerator is ideal, Tkr = Tk and Trh = Th.

Finally the work done in the compression and expansion cells is given by

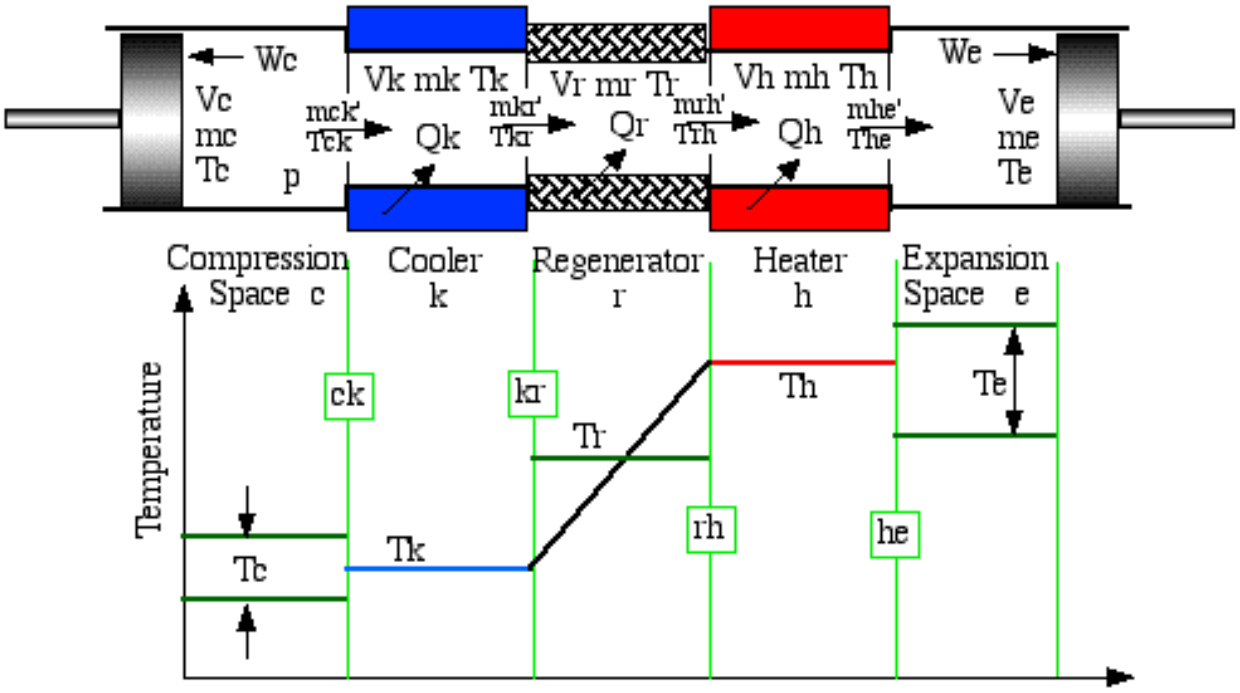
W = Wc + We

dW = **d**Wc + **d**We

dWc = p **d**Vc

dWe = p **d**Ve

Chapter 4: Ideal Adiabatic Analysis. Part B) Equation Summary and Method of Solution



$p = M\ R / (Vc / Tc + Vk / Tk + Vr / Tr + Vh / Th + Ve / Te)$ $\mathbf{d}p = \dfrac{-\gamma\ p\ (\mathbf{d}Vc / Tck + \mathbf{d}Ve / The)}{[Vc / Tck + \gamma(Vk / Tk + Vr / Tr + Vh / Th) + Ve / The]}$	Pressure
$mc = p\ Vc / (R\ Tc)$ $mk = p\ Vk / (R\ Tk)$ $mr = p\ Vr / (R\ Tr)$ $mh = p\ Vh / (R\ Th)$ $me = p\ Ve / (R\ Te)$	Masses
$\mathbf{d}mc = (p\ \mathbf{d}Vc + Vc\ \mathbf{d}p / \gamma) / (R\ Tck)$ $\mathbf{d}me = (p\ \mathbf{d}Ve + Ve\ \mathbf{d}p / \gamma) / (R\ The)$ $\mathbf{d}mk = mk\ \mathbf{d}p / p$ $\mathbf{d}mr = mr\ \mathbf{d}p / p$ $\mathbf{d}mh = mh\ \mathbf{d}p / p$	Mass Accumulations
$mck' = -\mathbf{d}mc$ $mkr' = mck' - \mathbf{d}mk$ $mhe' = \mathbf{d}me$ $mrh' = mhe' + \mathbf{d}mh$	Mass Flow
if $mck' > 0$ **then** $Tck = Tc$ **else** $Tck = Tk$ **if** $mhe' > 0$ **then** $The = Th$ **else** $The = Te$	Conditional Temperatures
$\mathbf{d}Tc = Tc\ (\mathbf{d}p / p + \mathbf{d}Vc / Vc - \mathbf{d}mc / mc)$ $\mathbf{d}Te = Te\ (\mathbf{d}p / p + \mathbf{d}Ve / Ve - \mathbf{d}me / me)$	Temperatures
$\mathbf{d}Qk = Vk\ \mathbf{d}p\ cv / R - cp\ (Tck\ mck' - Tk\ mkr')$ $\mathbf{d}Qr = Vr\ \mathbf{d}p\ cv / R - cp\ (Tk\ mkr' - Th\ mrh')$ $\mathbf{d}Qh = Vh\ \mathbf{d}p\ cv / R - cp\ (Th\ mrh' - The\ mhe')$ $\mathbf{d}Wc = p\ \mathbf{d}Vc$ $\mathbf{d}We = p\ \mathbf{d}Ve$ $\mathbf{d}W = \mathbf{d}Wc + \mathbf{d}We$ $W = Wc + We$	Energy

The method of solution

We now consider the solution of the equation set above. Because of the non-linear nature of the equations (in particular with regards to the Conditional Temperatures)

we have to resort to a numerical solution of specific configurations and operating conditions.

The specific engine configuration and geometry defines Vc, Ve, **d**Vc, and **d**Ve as analytic functions of the crankangle θ, and the heat exchanger geometry defines the void volumes Vk, Vr, Vh. The choice of working gas (typically air, helium or hydrogen) specifies R, cp, cv, and γ. The operating conditions specify Tk and Th, and thus the **mean effective temperature** Tr = (Th - Tk) / **ln**(Th / Tk). Specifying the total mass of working gas M is a problem, since this is not normally a known parameter. The approach we use is to specify the mean operating pressure pmean and then use the **Schmidt Analysi**s to evaluate M. Even though the Ideal Adiabatic model is independent of operating frequency, we nevertheless specify it in order to evaluate power and other time related effects (such as thermal conduction loss in the regenerator housing.)

We notice that apart from the constant parameters specified above, there are 22 variables and 16 derivatives in the equation set, to be solved over a complete cycle (θ = [0, 2π]):

Tc, Te, Qk, Qr, Qh, Wc, We - seven derivatives to be integrated numerically

W, p, Vc, Ve, mc, mk, mr, mh, me - nine analytical variables and derivatives

Tck, The, mck', mkr', mrh', mhe' - six conditional and mass flow variables (derivatives undefined)

We treat this as a "quasi steady-flow" system, thus over each integration interval the four mass flow variables mck', mkr', mrh', and mhe' remain constant and there are no acceleration effects. Thus we consider the problem as that of solving a set of seven simultaneous ordinary differential equations.

The simplest approach to solving a set of ordinary differential equations is to formulate it as an initial-value problem, in which the initial values of all the variables are known and the equations are integrated from that initial state over a complete cycle. The initial value problem can be stated in simple terms. Let the vector **Y** collectively represent the seven unknown variables, thus y[Tc] is the compression space temperature, y[We] is the work done by the expansion space, and so on. Given an initial condition **Y**(θ = 0) = **Y**0 and the corresponding set of differential equations **dY** = **F**(θ, **Y**), evaluate the unknown functions **Y**(θ) that satisfy both the differential equations and the initial conditions. A numerical solution to this problem is

accomplished by first computing the values of the derivatives at θ0 and proceeding in small increments of θ to a new point θ1 = θ0 + Δθ. Thus the solution is composed of a series of short straight-line segments that approximate the true curves **Y**(θ). Among the vast number of methods available for solving initial-value problems, the classical fourth-order Runge-Kutta method is probably the most frequently used.

In order to develop our specific method of solution of the initial-value problem, we have presented a case study in the MATLAB language involving a **large-angle pendulum** (Appendix U). We do not use the MATLAB built-in functions for solving ordinary differential equations, since our method requires overloading features not available in these built-in functions.

Unfortunately the Ideal Adiabatic model is not an initial-value problem, but is instead a boundary-value problem. We do not know the initial values of the working space gas temperatures Tc and Te, which result from the adiabatic compression and expansion processes as well as enthalpy flow processes. The only guidance that we have to their correct choice is that their values at the end of the steady-state cycle should be equal to their respective values at the beginning of the cycle.

However, because of its cyclic nature, the system can be formed as an initial value problem by assigning arbitrary initial conditions, and integrating the equations through several complete cycles until a cyclic steady state has been attained. This is equivalent to the transient "warm-up" operation of an actual machine. Experience has shown that the most sensitive measure of convergence to cyclic steady state is the residual regenerator heat Qr at the end of the cycle, which should be zero.

The compression and expansion space temperatures are thus initially specified at Tk and Th respectively. The system of equations can then be solved through as many cycles as necessary in order to attain cyclic steady state. For most configurations, between five and ten cycles will be sufficient for convergence.

Chapter 4: Ideal Adiabatic Analysis. Part C) Function Set 'Adiabatic'

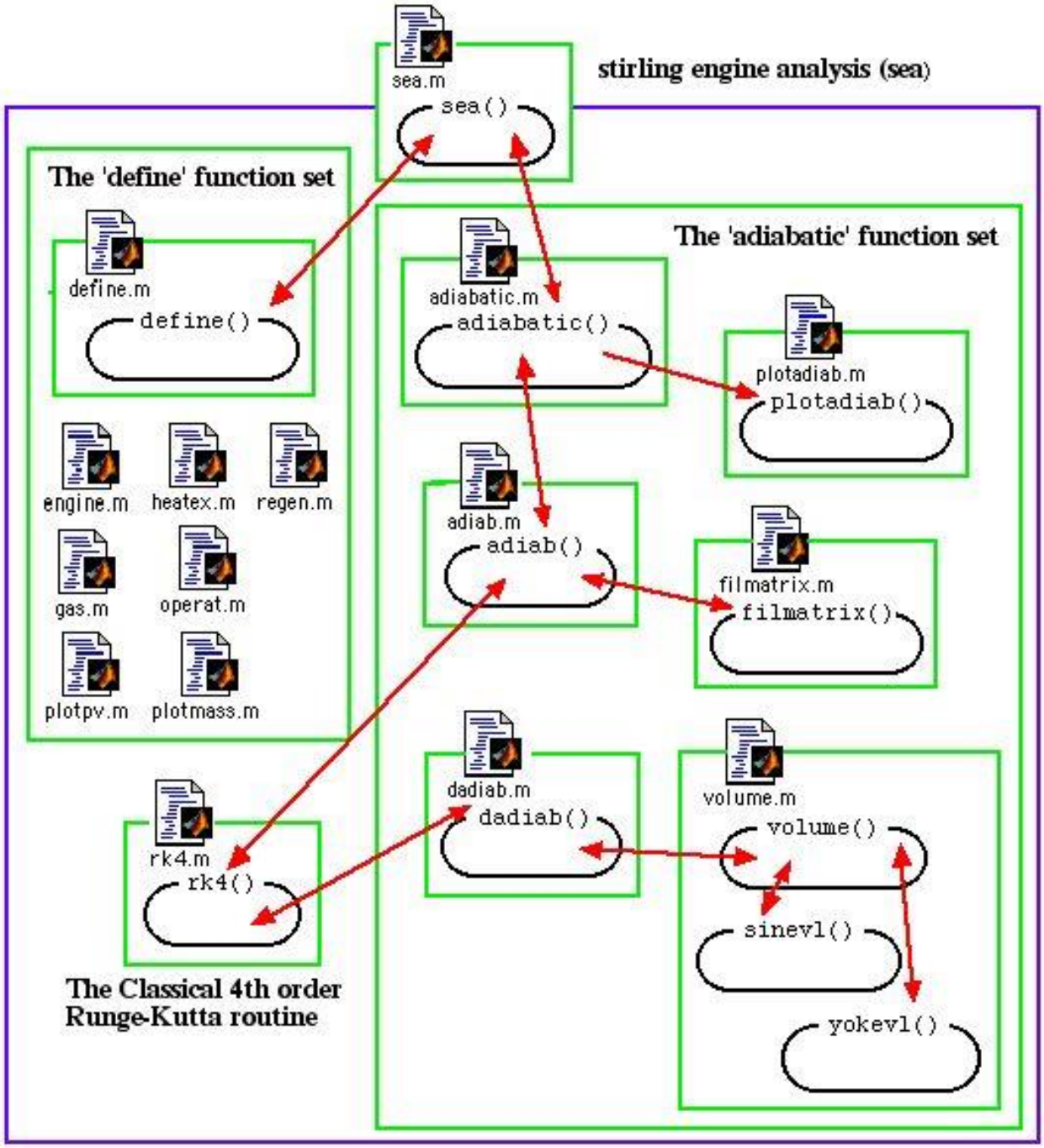
Function set 'adiabatic' - an Ideal Adiabatic simulation of a specific Stirling engine configuration.

Sections in this chapter:

- The main program 'sea' (stirling engine analysis)
- The ideal adiabatic model function 'adiab'
- The ideal adiabatic model derivatives function 'dabiab'

The main program 'sea' (stirling engine analysis)

From the flow diagram below we see that four different systems are invoked to do an Ideal Adiabatic simulation. The main program **sea** (**s**tirling **e**ngine **a**nalysis) first defines the system to be simulated in terms of the set of global variables set up by the **define** (Chapter 3C) set of functions, as described previously. It then invokes function **adiabatic** which in turn invokes function set **adiab** to solve the set of differential equations (function **dadiab**) over a number of cycles until convergence is attained. Function **adiab** then fills in the solution matrix for a complete cycle (function **fillmatrix**) and displays various performance results (power, efficiency). Function **adiabatic** then invokes function **plotadiab** to display various relevant plots. The differential equation set is solved by using the Classical Fourth Order Runge-Kutta method which is which is described separately in a Technical Note on **ordinary differential equations** (Appendix U).



The dynamics of the solution algorithm lies in the function set **adiab**, which initializes the variables, invokes the Runge-Kutta function over a number of cycles, checks for cyclic convergence, then fills in the solution matrix. Notice that the function m-file **volume.m** includes the sinusoidal volume variations (function **sinevol**), the Ross Yoke-drive volume variations (function **yokevol**), and (new - not in the above flow

diagram) the rockerV drive volume variations function **rockvol**). The nine functions of the set **adiabatic** are included in the following seven m-files (refer to the diagram above): [**adiabatic.m**, **adiab.m**, **dadiab.m**, **rk4.m**, **volume.m, filmatrix.m** and **plotadiab.m**]. As before, these can be directly copied from this website and used in a system which has MATLAB installed. It is intended that the user will modify and augment this system as required for specific engine designs.

The main program 'sea' (stirling engine analysis)

We first invoke the **define** function set to specify all the global variables needed to simulate a specific engine system and do a Schmidt analysis, and then invoke either the **ideal adiabatic** or **simple** analysis functions.

```
% sea (stirling engine analysis) - main program
%Israel Urieli 7/20/02
clc;
clear all;
% define a specific engine
define;
choice = 'x';
while(~strncmp(choice,'q',1))
    fprintf('Choose simulation:\n');
    choice = input('a)diabatic, s)imple q)uit: ','s');
    if(strncmp(choice,'a',1))
        [var,dvar] = adiabatic;
    elseif(strncmp(choice,'s',1))
        [var,dvar] = simple;
    end
end
fprintf('quitting simulation...\n');
```

The ideal adiabatic model function 'adiabatic'

The purpose of the adiabatic function set is to determine the numerical solution of the ideal adiabatic model equation set (refer to the **Equation Summary and Method of Solution**). Recall in the equation set that apart from the constants, there are 22 variables and 16 derivatives to be solved over a complete cycle ($\theta = [0, 2\pi]$) as follows:
Tc, Te, Qk, Qr, Qh, Wc, We - seven derivatives to be integrated numerically

W, p, Vc, Ve, mc, mk, mr, mh, me - nine analytical variables and derivatives
Tck, The, mck', mkr', mrh', mhe' - six conditional and mass flow variables (derivatives undefined)

Function **adiabatic** invokes the function set **adiab** to fill in the solution matrices **var** and **dvar** over a complete cycle, and then invokes function **plotadiab** (shown below) to do the required plots of the simulation results.

```
function [var,dvar] = adiabatic
% ideal adiabatic simulation and temperature/energy vs theta plots
% Israel Urieli, 7/20/2002
% Returned values:
%   var(22,37) array of variable values every 10 degrees (0 - 360)
%   dvar(16,37) array of derivatives every 10 degrees (0 - 360)
% Row indices of the var, dvar arrays:
 TC = 1;  % Compression space temperature (K)
 TE = 2;  % Expansion space temperature (K)
 QK = 3;  % Heat transferred to the cooler (J)
 QR = 4;  % Heat transferred to the regenerator (J)
 QH = 5;  % Heat transferred to the heater (J)
 WC = 6;  % Work done by the compression space (J)
 WE = 7;  % Work done by the expansion space (J)
 W  = 8;  % Total work done (WC + WE) (J)
 P  = 9;  % Pressure (Pa)
 VC = 10; % Compression space volume (m^3)
 VE = 11; % Expansion space volume (m^3)
 MC = 12; % Mass of gas in the compression space (kg)
 MK = 13; % Mass of gas in the cooler (kg)
 MR = 14; % Mass of gas in the regenerator (kg)
 MH = 15; % Mass of gas in the heater (kg)
 ME = 16; % Mass of gas in the expansion space (kg)
 TCK = 17; % Conditional temperature compression space / cooler (K)
 THE = 18; % Conditional temeprature heater / expansion space (K)
 GACK = 19; % Conditional mass flow compression space / cooler (kg/rad)
 GAKR = 20; % Conditional mass flow cooler / regenerator (kg/rad)
 GARH = 21; % Conditional mass flow regenerator / heater (kg/rad)
 GAHE = 22; % Conditional mass flow heater / expansion space (kg/rad)
% Size of var(ROWV,COL), dvar(ROWD,COL)
```

```
 ROWV = 22; % number of rows in the var matrix
 ROWD = 16; % number of rows in the dvar matrix
 COL = 37; % number of columns in the matrices (every 10 degrees)
%=============================================================
global freq % cycle frequency [herz]
global tk tr th % cooler, regenerator, heater temperatures [K]
global vk % cooler void volume [m^3]
global vr % regen void volume [m^3]
global vh % heater void volume [m^3]
% do ideal adiabatic analysis:
[var,dvar] = adiab;
% Print out ideal adiabatic analysis results
eff = var(W,COL)/var(QH,COL); % engine thermal efficency
Qkpower = var(QK,COL)*freq;   % Heat transferred to the cooler (W)
Qrpower = var(QR,COL)*freq;   % Heat transferred to the regenerator (W)
Qhpower = var(QH,COL)*freq;   % Heat transferred to the heater (W)
Wpower = var(W,COL)*freq;     % Total power output (W)
fprintf('========== ideal adiabatic analysis results ============\n')
fprintf(' Heat transferred to the cooler: %.2f[W]\n', Qkpower);
fprintf(' Net heat transferred to the regenerator: %.2f[W]\n', Qrpower);
fprintf(' Heat transferred to the heater: %.2f[W]\n', Qhpower);
fprintf(' Total power output: %.2f[W]\n', Wpower);
fprintf(' Thermal efficiency : %.1f[%%]\n', eff*100);
fprintf('=========================================================\n')
% Various plots of the ideal adiabatic simulation results
plotadiab(var,dvar);
```

Function **plotadiab** is invoked both by function **adiabatic** (above) and by function **simple**.

```
function plotadiab(var,dvar)
% various plots of ideal adiabatic simulation results
% Israel Urieli, 7/21/2002 (corrected temp plots 12/3/2003)
% Arguments:
%   var(22,37) array of variable values every 10 degrees (0 - 360)
%   dvar(16,37) array of derivatives every 10 degrees (0 - 360)
```

```
% Row indices of the var, dvar arrays:
 TC = 1;  % Compression space temperature (K)
 TE = 2;  % Expansion space temperature (K)
 QK = 3;  % Heat transferred to the cooler (J)
 QR = 4;  % Heat transferred to the regenerator (J)
 QH = 5;  % Heat transferred to the heater (J)
 WC = 6;  % Work done by the compression space (J)
 WE = 7;  % Work done by the expansion space (J)
 W  = 8;  % Total work done (WC + WE) (J)
 P  = 9;  % Pressure (Pa)
 VC = 10; % Compression space volume (m^3)
 VE = 11; % Expansion space volume (m^3)
 MC = 12; % Mass of gas in the compression space (kg)
 MK = 13; % Mass of gas in the cooler (kg)
 MR = 14; % Mass of gas in the regenerator (kg)
 MH = 15; % Mass of gas in the heater (kg)
 ME = 16; % Mass of gas in the expansion space (kg)
 TCK = 17; % Conditional temperature compression space / cooler (K)
 THE = 18; % Conditional temeprature heater / expansion space (K)
 GACK = 19; % Conditional mass flow compression space / cooler (kg/rad)
 GAKR = 20; % Conditional mass flow cooler / regenerator (kg/rad)
 GARH = 21; % Conditional mass flow regenerator / heater (kg/rad)
 GAHE = 22; % Conditional mass flow heater / expansion space (kg/rad)
% Size of var(ROWV,COL), dvar(ROWD,COL)
 ROWV = 22; % number of rows in the var matrix
 ROWD = 16; % number of rows in the dvar matrix
 COL = 37; % number of columns in the matrices (every 10 degrees)
%============================================================
global tk tr th % cooler, regenerator, heater temperatures [K]
global vk % cooler void volume [m^3]
global vr % regen void volume [m^3]
global vh % heater void volume [m^3]
choice = 'x';
while(~strncmp(choice,'q',1))
    fprintf('Choose plot type:\n');
    fprintf('   p - for a PV diagram\n');
    fprintf('   t - for a temperature vs crank angle plot\n');
    fprintf('   e - for an energy vs crank angle plot\n');
```

```
fprintf('   q - to quit\n');
choice = input('p)vdiagram, t)emperature, e)nergy, q)uit: ','s');
if (strncmp(choice,'p',1))
    figure
    vol = (var(VC,:) + vk + vr + vh + var(VE,:))*1e6; % cubic centimeters
    pres = (var(P,:))*1e-5; % bar
    plot(vol,pres, 'k')
    grid on
    xlabel('Volume (cc)')
    ylabel('Pressure (bar [1bar = 100kPa])')
    title('P-v diagram')
elseif (strncmp(choice,'t',1))
    figure
    x = 0:10:360;
    Tcomp = var(TC,:);
    Texp = var(TE,:);
    plot(x,Tcomp,'b-',x,Texp,'r-');
    hold on
    x = [0,360];
    y = [tk,tk];
    plot(x,y,'b-')
    y = [tr,tr];
    plot(x,y,'g-')
    y = [th,th];
    plot(x,y,'r-')
    hold off
    grid on
    xlabel('Crank angle (degrees)');
    ylabel('Temperature (K)');
    title('Temperature vs crank angle');
elseif (strncmp(choice,'e',1))
    figure
    x = 0:10:360;
    Qkol = var(QK,:); % [J]
    Qreg = var(QR,:); % [J]
    Qhot = var(QH,:); % [J]
    Work = var(W,:); % [J]
    Wcom = var(WC,:); % [J]
```

```
        Wexp = var(WE,:); % [J]
        plot(x,Qkol,'b-',x,Qreg,'g-',x,Qhot,'r-',x,Work,'k-.',x,Wcom,'b--',x,Wexp,'r--');
        grid on
        xlabel('Crank angle (degrees)');
        ylabel('Energy [Joules]');
        title('Energy vs crank angle');
    end
end
fprintf('quitting ideal adiabatic plots...\n');
```

The ideal adiabatic model function 'adiab'

The purpose of function set **adiab** is to fill in the arrays **var** (variable values) and **dvar** (derivative vaues) every 10 degrees over a cycle. The function set includes the derivatives function **dadiab**, the Classical fourth order Runge-Kutta function **rk4**, and the function **filmatrix** (both shown below).
Recall from the **Equation Summary and Method of Solution** that apart from the constants, there are 22 variables and 16 derivatives to be solved as follows:
Tc, Te, Qk, Qr, Qh, Wc, We - seven derivatives to be integrated numerically
W, p, Vc, Ve, mc, mk, mr, mh, me - nine analytical variables and derivatives
Tck, The, mck', mkr', mrh', mhe' - six conditional and mass flow variables (derivatives undefined).
Notice how we have specified the indices of the arrays for improved readability.

```
function [var,dvar] = adiab
% ideal adiabatic model simulation
% Israel Urieli, 7/6/2002
% Returned values:
%   var(22,37) array of variable values every 10 degrees (0 - 360)
%   dvar(16,37) array of derivatives every 10 degrees (0 - 360)
global tk th % cooler, heater temperatures [K]
% Row indices of the var, dvar matrices, and the y,dy variable vectors:
 TC = 1;  % Compression space temperature (K)
 TE = 2;  % Expansion space temperature (K)
 QK = 3;  % Heat transferred to the cooler (J)
 QR = 4;  % Heat transferred to the regenerator (J)
```

```
QH = 5;  % Heat transferred to the heater (J)
WC = 6;  % Work done by the compression space (J)
WE = 7;  % Work done by the expansion space (J)
W  = 8;  % Total work done (WC + WE) (J)
P  = 9;  % Pressure (Pa)
VC = 10; % Compression space volume (m^3)
VE = 11; % Expansion space volume (m^3)
MC = 12; % Mass of gas in the compression space (kg)
MK = 13; % Mass of gas in the cooler (kg)
MR = 14; % Mass of gas in the regenerator (kg)
MH = 15; % Mass of gas in the heater (kg)
ME = 16; % Mass of gas in the expansion space (kg)
TCK = 17; % Conditional temperature compression space / cooler (K)
THE = 18; % Conditional temeprature heater / expansion space (K)
GACK = 19; % Conditional mass flow compression space / cooler (kg/rad)
GAKR = 20; % Conditional mass flow cooler / regenerator (kg/rad)
GARH = 21; % Conditional mass flow regenerator / heater (kg/rad)
GAHE = 22; % Conditional mass flow heater / expansion space (kg/rad)
% Size of var(ROWV,COL), y(ROWV), dvar(ROWD,COL), dy(ROWD)
ROWV = 22; % number of rows in the var matrix
ROWD = 16; % number of rows in the dvar matrix
COL = 37; % number of columns in the matrices (every 10 degrees)
%=========================================================
fprintf('============Ideal Adiabatic Analysis====================\n')
fprintf('Cooler Tk = %.1f[K], Heater Th = %.1f[K]\n', tk, th);
epsilon = 1;  % Allowable error in temperature (K)
max_iteration = 20;  % Maximum number of iterations to convergence
ninc = 360; % number if integration increments (every degree)
step = ninc/36; % for saving values in var, dvar matrices
dtheta = 2.0*pi/ninc; % integration increment (radians)
% Initial conditions:
y(THE) = th;
y(TCK) = tk;
y(TE) = th;
y(TC) = tk;
iter = 0;
terror = 10*epsilon; % Initial error to enter the loop
% Iteration loop to cyclic convergence
while ((terror >= epsilon)&(iter < max_iteration))
```

```
    % cyclic initial conditions
    tc0 = y(TC);
    te0 = y(TE);
    theta = 0;
    y(QK) = 0;
    y(QR) = 0;
    y(QH) = 0;
    y(WC) = 0;
    y(WE) = 0;
    y(W) = 0;
    fprintf('iteration %d: Tc = %.1f[K], Te = %.1f[K]\n',iter,y(TC),y(TE))
    for(i = 1:1:ninc)
        [theta,y,dy] = rk4('dadiab',7,theta,dtheta,y);
    end
    terror = abs(tc0 - y(TC)) + abs(te0 - y(TE));
    iter = iter + 1;
end
if (iter >= max_iteration)
     fprintf('No convergence within %d iteration\n',max_iteration)
end
% Initial var and dvar matrix
var = zeros(22,37);
dvar = zeros(16,37);
% a final cycle, to fill the var, dvar matrices
theta=0;
y(QK)=0;
y(QR)=0;
y(QH)=0;
y(WC)=0;
y(WE)=0;
y(W)=0;
[var,dvar] = filmatrix(1,y,dy,var,dvar);
for(i = 2:1:COL)
     for(j = 1:1:step)
         [theta,y,dy] = rk4('dadiab',7,theta,dtheta,y);
     end
     [var,dvar] = filmatrix(i,y,dy,var,dvar);
end
```

The function **rk4** (below) was developed in the Technical Note on **ordinary differential equations**.

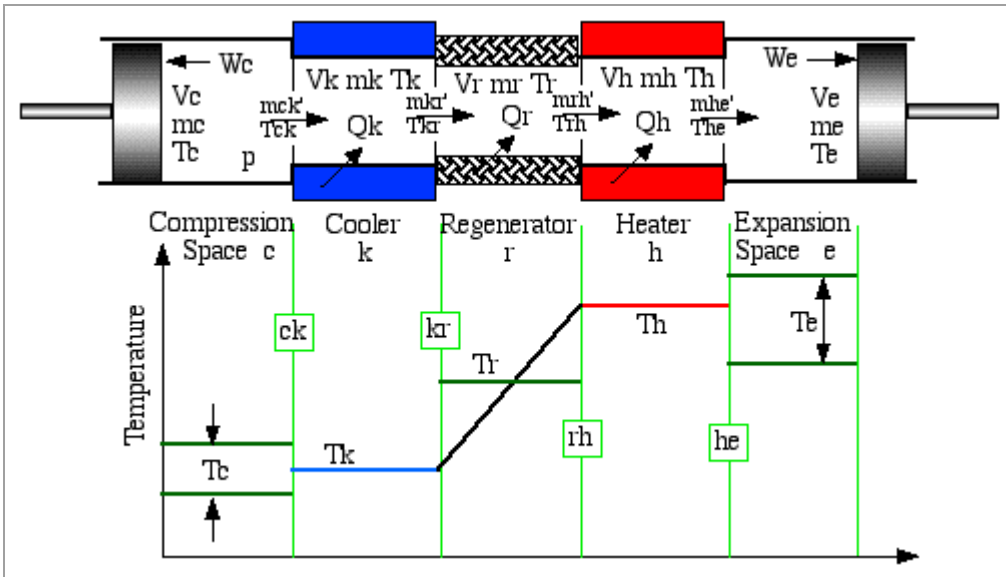
```
function [x, y, dy] = rk4(deriv,n,x,dx,y)
%Classical fourth order Runge-Kutta method
%Integrates n first order differential equations
%dy(x,y) over interval x to x+dx
%Israel Urieli - Jan 21, 2002
x0 = x;
y0 = y;
[y,dy1] = feval(deriv,x0,y);
for i = 1:n
      y(i) = y0(i) + 0.5*dx*dy1(i);
end
xm = x0 + 0.5*dx;
[y,dy2] = feval(deriv,xm,y);
for i = 1:n
      y(i) = y0(i) + 0.5*dx*dy2(i);
end
[y,dy3] = feval(deriv,xm,y);
for i = 1:n
      y(i) = y0(i) + dx*dy3(i);
end
x = x0 + dx;
[y,dy] = feval(deriv,x,y);
for i = 1:n
      dy(i) = (dy1(i) + 2*(dy2(i) + dy3(i)) + dy(i))/6;
      y(i) = y0(i) + dx*dy(i);
end
```

```
function [var,dvar]=filmatrix(j,y,dy,var,dvar);
% Fill in the j-th column of the var, dvar matrices with values of y, dy
% Israel Urieli, 7/20/2002
% Arguments:  j - column index (1 - 37, every 10 degrees of cycle angle)
%          y(ROWV) - vector of current variable values
```

```
%          dy(ROWD) vector of current derivatives
%          var(ROWV,37) - matrix of current variables vs cycle angle
%          dvar(ROWD,37) - matrix of current derivatives vs cycle angle
% Returned values:
%          var(ROWV,37) - matrix of updated variables vs cycle angle
%          dvar(ROWD,37) - matrix of updated derivatives vs cycle angle
ROWV = 22; % number of rows in the var matrix
ROWD = 16; % number of rows in the dvar matrix
for (i = 1:1:ROWV)
  var(i,j) = y(i);
end
for (i = 1:1:ROWD)
  dvar(i,j) = dy(i);
end
```

The ideal adiabatic model derivatives function 'dabiab'

We consider the numerical solution of the previously derived ideal adiabatic model equation set (refer to the **Equation Summary and Method of Solution**).



Note in the equation set that apart from the constants, there are 22 variables and 16 derivatives to be solved over a complete cycle (θ = [0, 2π]) as follows:

Tc, Te, Qk, Qr, Qh, Wc, We - seven derivatives to be integrated numerically

W, p, Vc, Ve, mc, mk, mr, mh, me - nine analytical variables and derivatives

Tck, The, mck', mkr', mrh', mhe' - six conditional and mass flow variables (derivatives undefined)

$p = M\ R\ /\ (Vc\ /\ Tc + Vk\ /\ Tk + Vr\ /\ Tr + Vh\ /\ Th + Ve\ /\ Te)$ $\mathbf{d}p = \dfrac{-\gamma\ p\ (\mathbf{d}Vc\ /\ Tck + \mathbf{d}Ve\ /\ The)}{[Vc\ /\ Tck + \gamma\ (Vk\ /\ Tk + Vr\ /\ Tr + Vh\ /\ Th) + Ve\ /\ The\ ]}$	Pressure
$mc = p\ Vc\ /\ (R\ Tc)$ $mk = p\ Vk\ /\ (R\ Tk)$ $mr = p\ Vr\ /\ (R\ Tr)$ $mh = p\ Vh\ /\ (R\ Th)$ $me = p\ Ve\ /\ (R\ Te)$	Masses
$\mathbf{d}mc = (p\ \mathbf{d}Vc + Vc\ \mathbf{d}p\ /\ \gamma)\ /\ (R\ Tck)$ $\mathbf{d}me = (p\ \mathbf{d}Ve + Ve\ \mathbf{d}p\ /\ \gamma)\ /\ (R\ The)$ $\mathbf{d}mk = mk\ \mathbf{d}p\ /\ p$ $\mathbf{d}mr = mr\ \mathbf{d}p\ /\ p$ $\mathbf{d}mh = mh\ \mathbf{d}p\ /\ p$	Mass Accumulations
$mck' = -\mathbf{d}mc$ $mkr' = mck' - \mathbf{d}mk$ $mhe' = \mathbf{d}me$ $mrh' = mhe' + \mathbf{d}mh$	Mass Flow
if $mck' > 0$ **then** $Tck = Tc$ **else** $Tck = Tk$ **if** $mhe' > 0$ **then** $The = Th$ **else** $The = Te$	Conditional Temperatures
$\mathbf{d}Tc = Tc\ (\mathbf{d}p\ /\ p + \mathbf{d}Vc\ /\ Vc - \mathbf{d}mc\ /\ mc)$ $\mathbf{d}Te = Te\ (\mathbf{d}p\ /\ p + \mathbf{d}Ve\ /\ Ve - \mathbf{d}me\ /\ me)$	Temperatures
$\mathbf{d}Qk = Vk\ \mathbf{d}p\ cv\ /\ R - cp\ (Tck\ mck' - Tk\ mkr')$ $\mathbf{d}Qr = Vr\ \mathbf{d}p\ cv\ /\ R - cp\ (Tk\ mkr' - Th\ mrh')$ $\mathbf{d}Qh = Vh\ \mathbf{d}p\ cv\ /\ R - cp\ (Th\ mrh' - The\ mhe')$ $\mathbf{d}Wc = p\ \mathbf{d}Vc$ $\mathbf{d}We = p\ \mathbf{d}Ve$ $\mathbf{d}W = \mathbf{d}Wc + \mathbf{d}We$ $W = Wc + We$	Energy

Our approach is to define two vectors, y and dy containing all the variables and derivatives. The first 7 variables are to be integrated numerically, and the remainder

are evaluated algebraically in the derivative function **dadiab** shown below. This approach is called 'overloading' of the vectors, and the differential equation set is solved by using the Classical Fourth Order Runge-Kutta method as described in the Technical Note on **ordinary differential equations**. Notice how we have specified the indices of the vectors for improved readability.

```
function [y,dy] = dadiab(theta,y)
% Evaluate ideal adiabatic model derivatives
% Israel Urieli, 7/6/2002
% Arguments:  theta - current cycle angle [radians]
%             y(22) - vector of current variable values
% Returned values:
%             y(22) - updated vector of current variables
%             dy(16) vector of current derivatives
% Function invoked : volume.m
% global variables used from "define" functions
global vk % cooler void volume [m^3]
global vr % regen void volume [m^3]
global vh % heater void volume [m^3]
global rgas % gas constant [J/kg.K]
global cp % specific heat capacity at constant pressure [J/kg.K]
global cv % specific heat capacity at constant volume [J/kg.K]
global gama % ratio: cp/cv
global mgas % total mass of gas in engine [kg]
global tk tr th % cooler, regen, heater temperatures [K]
% Indices of the y, dy vectors:
 TC = 1;  % Compression space temperature (K)
 TE = 2;  % Expansion space temperature (K)
 QK = 3;  % Heat transferred to the cooler (J)
 QR = 4;  % Heat transferred to the regenerator (J)
 QH = 5;  % Heat transferred to the heater (J)
 WC = 6;  % Work done by the compression space (J)
 WE = 7;  % Work done by the expansion space (J)
 W  = 8;  % Total work done (WC + WE) (J)
 P  = 9;  % Pressure (Pa)
 VC = 10; % Compression space volume (m^3)
 VE = 11; % Expansion space volume (m^3)
 MC = 12; % Mass of gas in the compression space (kg)
```

```
MK = 13; % Mass of gas in the cooler (kg)
MR = 14; % Mass of gas in the regenerator (kg)
MH = 15; % Mass of gas in the heater (kg)
ME = 16; % Mass of gas in the expansion space (kg)
TCK = 17; % Conditional temperature compression space / cooler (K)
THE = 18; % Conditional temeprature heater / expansion space (K)
GACK = 19; % Conditional mass flow compression space / cooler (kg/rad)
GAKR = 20; % Conditional mass flow cooler / regenerator (kg/rad)
GARH = 21; % Conditional mass flow regenerator / heater (kg/rad)
GAHE = 22; % Conditional mass flow heater / expansion space (kg/rad)
%=======================================================
% Volume and volume derivatives:
[y(VC),y(VE),dy(VC),dy(VE)] = volume(theta);
% Pressure and pressure derivatives:
vot = vk/tk + vr/tr + vh/th;
y(P) = (mgas*rgas/(y(VC)/y(TC) + vot + y(VE)/y(TE)));
top = -y(P)*(dy(VC)/y(TCK) + dy(VE)/y(THE));
bottom = (y(VC)/(y(TCK)*gama) + vot + y(VE)/(y(THE)*gama));
dy(P) = top/bottom;
% Mass accumulations and derivatives:
y(MC) = y(P)*y(VC)/(rgas*y(TC));
y(MK) = y(P)*vk/(rgas*tk);
y(MR) = y(P)*vr/(rgas*tr);
y(MH) = y(P)*vh/(rgas*th);
y(ME) = y(P)*y(VE)/(rgas*y(TE));
dy(MC) = (y(P)*dy(VC) + y(VC)*dy(P)/gama)/(rgas*y(TCK));
dy(ME) = (y(P)*dy(VE) + y(VE)*dy(P)/gama)/(rgas*y(THE));
dpop = dy(P)/y(P);
dy(MK) = y(MK)*dpop;
dy(MR) = y(MR)*dpop;
dy(MH) = y(MH)*dpop;
% Mass flow between cells:
y(GACK) = -dy(MC);
y(GAKR) = y(GACK) - dy(MK);
y(GAHE) = dy(ME);
y(GARH) = y(GAHE) + dy(MH);
% Conditional temperatures between cells:
y(TCK) = tk;
```

```
if (y(GACK)>0)
    y(TCK) = y(TC);
end
y(THE) = y(TE);
if (y(GAHE)>0)
    y(THE) = th;
end
% 7 derivatives to be integrated by rk4:
% Working space temperatures:
dy(TC) = y(TC)*(dpop + dy(VC)/y(VC) - dy(MC)/y(MC));
dy(TE) = y(TE)*(dpop + dy(VE)/y(VE) - dy(ME)/y(ME));
% Energy:
dy(QK) = vk*dy(P)*cv/rgas - cp*(y(TCK)*y(GACK) - tk*y(GAKR));
dy(QR) = vr*dy(P)*cv/rgas - cp*(tk*y(GAKR) - th*y(GARH));
dy(QH) = vh*dy(P)*cv/rgas - cp*(th*y(GARH) - y(THE)*y(GAHE));
dy(WC) = y(P)*dy(VC);
dy(WE) = y(P)*dy(VE);
% Net work done:
dy(W) = dy(WC) + dy(WE);
y(W) = y(WC) + y(WE);
```

The function **volume** is used to evaluate the working space volume variations and derivatives of the specific engine configuration as a function of the crank angle θ (theta). Recall from the **engine function** that we have included alpha engines as well as a free-piston beta drive engine in this resource, and it is intended that the function **volume** shown below be augmented as required.

```
function [vc,ve,dvc,dve] = volume(theta)
% determine working space volume variations and derivatives
% Israel Urieli, 7/6/2002
% Modified 2/14/2010 to include rockerV (rockvol)
% Modified 6/14/2016 to include beta free piston (sinevol)
% Argument:  theta - current cycle angle [radians]
% Returned values:
%   vc, ve - compression, expansion space volumes [m^3]
%   dvc, dve - compression, expansion space volume derivatives
```

```
global engine_type % s)inusoidal, y)oke r)ockerV (all alpha engines)

if (strncmp(engine_type,'s',1))
      [vc,ve,dvc,dve] = sinevol(theta);
elseif (strncmp(engine_type,'y',1))
      [vc,ve,dvc,dve] = yokevol(theta);
elseif (strncmp(engine_type,'r',1))
      [vc,ve,dvc,dve] = rockvol(theta);
elseif (strncmp(engine_type,'b',1))
      [vc,ve,dvc,dve] = sinevol(theta);
end
%=======================================================
function [vc,ve,dvc,dve] = sinevol(theta)
% sinusoidal drive volume variations and derivatives
% Israel Urieli, 7/6/2002
% Argument:  theta - current cycle angle [radians]
% Returned values:
%   vc, ve - compression, expansion space volumes [m^3]
%   dvc, dve - compression, expansion space volume derivatives
global vclc vcle % compression,expansion clearence vols [m^3]
global vswc vswe % compression, expansion swept volumes [m^3]
global alpha % phase angle advance of expansion space [radians]

 vc = vclc + 0.5*vswc*(1 + cos(theta+pi));
 ve = vcle + 0.5*vswe*(1 + cos(theta + alpha+pi));
 dvc = -0.5*vswc*sin(theta+pi);
 dve = -0.5*vswe*sin(theta + alpha+pi);
%=======================================================
function [vc,ve,dvc,dve] = yokevol(theta)
% Ross yoke drive volume variations and derivatives
% Israel Urieli, 7/6/2002
% Argument:  theta - current cycle angle [radians]
% Returned values:
%   vc, ve - compression, expansion space volumes [m^3]
%   dvc, dve - compression, expansion space volume derivatives
global vclc vcle % compression,expansion clearence vols [m^3]
global vswc vswe % compression, expansion swept volumes [m^3]
global alpha % phase angle advance of expansion space [radians]
```

```
global b1 % Ross yoke length (1/2 yoke base) [m]
global b2 % Ross yoke height [m]
global crank % crank radius [m]
global dcomp dexp % diameter of compression/expansion pistons [m]
global acomp aexp % area of compression/expansion pistons [m^2]
global ymin % minimum yoke vertical displacement [m]

 sinth = sin(theta);
 costh = cos(theta);
 bth = (b1^2 - (crank*costh)^2)^0.5;
 ye = crank*(sinth + (b2/b1)*costh) + bth;
 yc = crank*(sinth - (b2/b1)*costh) + bth;
 ve = vcle + aexp*(ye - ymin);
 vc = vclc + acomp*(yc - ymin);
 dvc = acomp*crank*(costh + (b2/b1)*sinth + crank*sinth*costh/bth);
 dve = aexp*crank*(costh - (b2/b1)*sinth + crank*sinth*costh/bth);
%=====================================================
function [vc,ve,dvc,dve] = rockvol(theta)
% Ross Rocker-V drive volume variations and derivatives
% Israel Urieli, 7/6/2002 & Martine Long 2/25/2005
% Argument:  theta - current cycle angle [radians]
% Returned values:
%   vc, ve - compression, expansion space volumes [m^3]
%   dvc, dve - compression, expansion space volume derivatives
global vclc vcle % compression,expansion clearence vols [m^3]
global crank % crank radius [m]
global acomp aexp % area of compression/expansion pistons [m^2]
global conrodc conrode % length of comp/exp piston connecting rods [m]
global ycmax yemax % maximum comp/exp piston vertical displacement [m]

 sinth = sin(theta);
 costh = cos(theta);
 beth = (conrode^2 - (crank*costh)^2)^0.5;
 bcth = (conrodc^2 - (crank*sinth)^2)^0.5;
 ye = beth - crank*sinth;
 yc = bcth + crank*costh;
 ve = vcle + aexp*(yemax - ye);
 vc = vclc + acomp*(ycmax - yc);
```

```
dvc = acomp*crank*sinth*(crank*costh/bcth + 1);
dve = -aexp*crank*costh*(crank*sinth/beth - 1);
```

Chapter 4: Ideal Adiabatic Analysis. Part D) Case Study: D-90 Ross Yoke-drive Engine

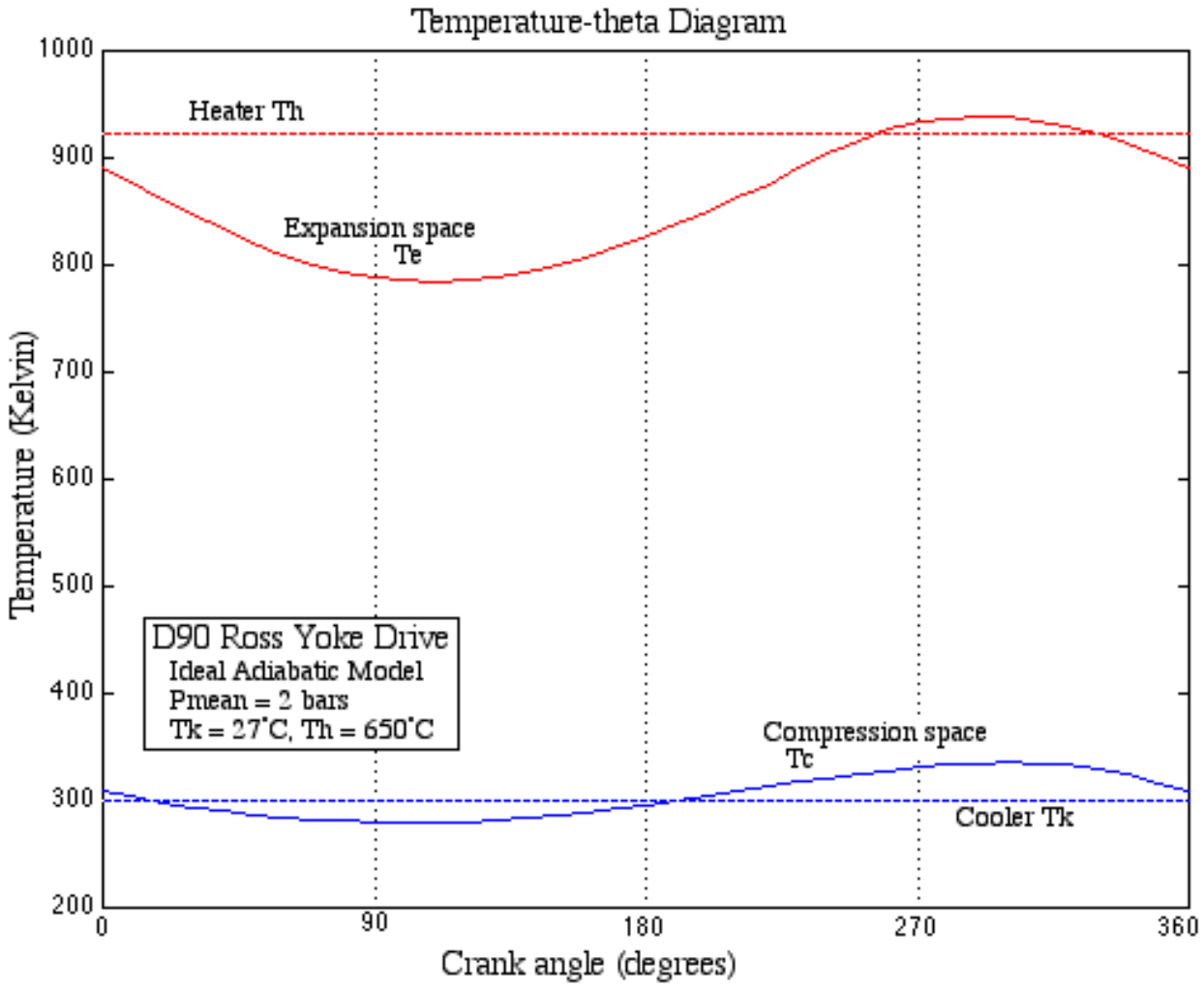
The 90 cc D-90 engine is fully described in Andy Ross' fascinating book **Making Stirling Engines** (pages 41 - 45). At Ohio University we have a **laboratory model** of the D-90 Yoke drive engine which is heated electrically in order to accurately determine the heat power input.

In this section we examine the results of performing an Ideal Adiabatic simulation of the D-90 engine under specific typical operating conditions as follows

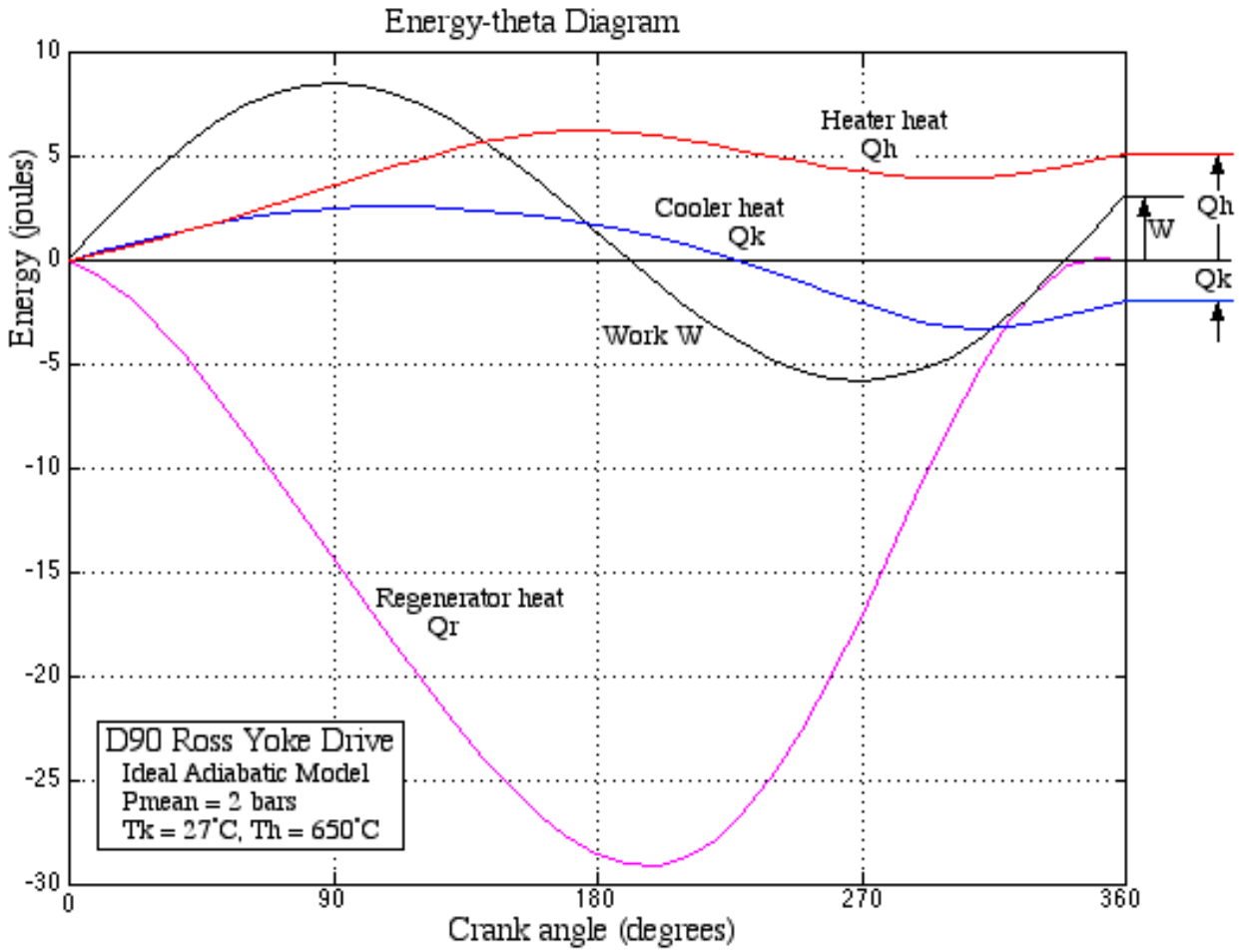
- Mean operating pressure - pmean = 2 bar.
  (The crankcase is sealed, and the output shaft power is obtained by a magnetic coupling. Andy typically pressurizes the crankcase with a bicycle pump to about 2 bar.)
- Cooler temperature 27 degrees Celsius (300 K), and heater temperature 650 degrees Celsius (923 K)
- Operating frequency 50 Hz. (Note that the Ideal Adiabatic model is independent of operating speed - all results are presented per cycle)

In order to simulate the engine by means of the **Ideal Adiabatic model equation set** given previously, we require the equations for the **Yoke-drive volume variations** (Appendix H) and derivatives Vc, Ve, **d**Vc and **d**Ve (all functions of crank angle θ), as well as the void volumes of the heat exchangers Vk, Vr, and Vh.

The cyclic convergence behavior of the Ideal Adiabatic model is extremely good, and using 360 increments over the cycle, the system effectively converges within 5 cycles. The convergence criterion chosen is that after a complete cycle both variable temperatures Te and Tc must be within one degree Kelvin of their initial values. We now consider the solution of the temperature variables Tc and Te, the heat energy variables Qk, Qr, Qh, and the work energy variables Wc, We, and the net work done W. These results are presented as plots showing the variation of these parameters with the crank angle θ.



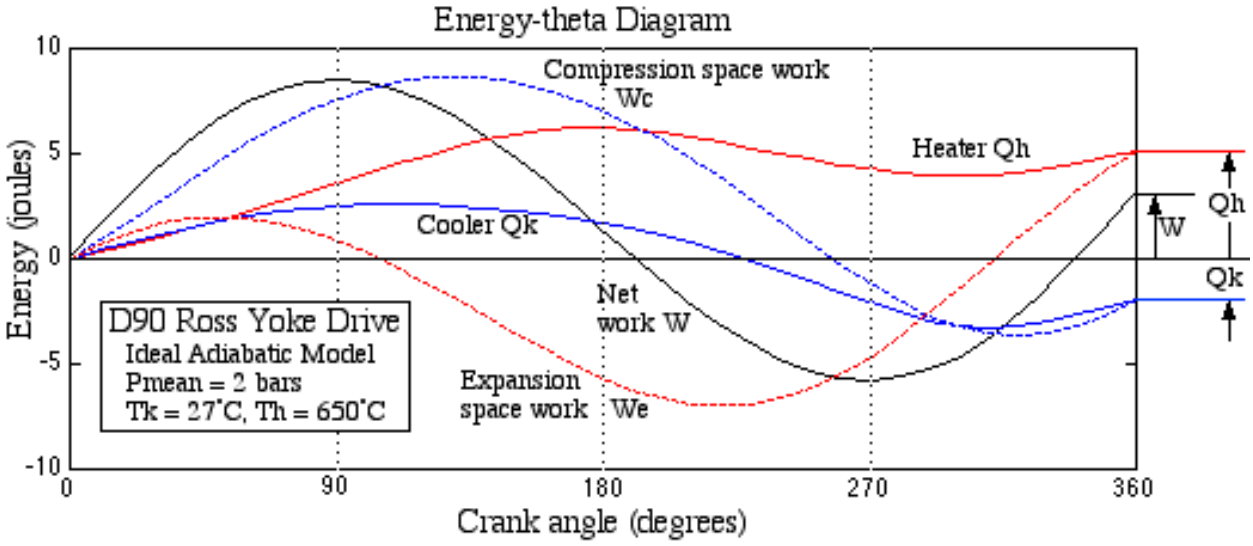
In the temperature-theta diagram we observe a large cyclic temperature variation of the gas in the expansion space (> 100 K), its mean value being less than that of the heater temperature of 923 K. Similarly the mean gas temperature in the compression space is higher than the coller temperature. This suggests that the adiabatic working spaces effectively reduce the temperature limits of operation, thus reducing the thermal efficiency to less than that of the Carnot efficiency.



The energy-theta diagram shows the accumulated heat transferred and work done over the cycle. Notice that the work done W starts with the (positive slope) expansion process then the compression process, and again returning to the expansion process, Thus the total work excursion is almost 15 joules, however the net work done at the end of the cycle is only 3 joules. The most significant aspect of the energy-theta diagram is the considerable amount of heat transferred in the regenerator over the cycle, almost ten times that of the net work done per cycle. This tends to indicate that the engine performance depends critically on the regenerator effectiveness and its ability to accommodate high heat fluxes. This aspect will be revisited in the section on the **Simple analysis** (Chapter 5), when we examine the effect of imperfect heat exchangers on Stirling engine performance. Significantly the energy rejected by the gas to the regenerator matrix in the first half of the cycle is equal to the energy absorbed by the gas from the matrix in the second half of the cycle, thus the net heat transfer to the regenerator over a cycle is zero. It is for this reason that the importance of the regenerator was not understood for about 100 years after Stirling's original patent describing the function and importance of the regenerator. The Lehmann machine on which Schmidt did his

analysis was apparently not fitted with a regenerator, and it is conceivable that Schmidt did not appreciate its importance, He refers to the textbook by Zeuner as containing a "complete, simple and clear theory" of air engines, but in the same textbook Zeuner decries the use of regenerators for air engines (Finkelstein, T., 1959, *Air Engines* in *The Engineer* part 1, 27 March.)

It is of interest to examine the two components, Wc and We, which added together gives the net work done W. These are shown as dashed lines in the following diagram.



Notice in particular that the expansion space work done (We) undergoes a vastly different process from that of heat transferred to the heater (Qh), however at the end of the cycle they have equal values (Qh = We). Similarly for the compression space work done (Wc) and the heat transferred to the cooler (Qk). In retrospect this must be so in order to retain an energy balance, however it did catch us unawares and surprised us when we first noticed this. The ideal regenerator thus behaves as the perfect isolator, isolating the energy balance of the heater and expansion space from that of the cooler and compression space. Thus for the Ideal Adiabatic model over a complete cycle

Qh = We; (Qe = 0)
Qk = Wc; (Qc = 0)
W = Wc + We

Recall that for the Ideal Isothermal model

Qe = We; (Qh = 0)
Qc = Wc; (Qk = 0)
W = Wc + We

Furthermore the Ideal Adiabatic model in itself does not give results which are significantly different from those of the Ideal Isothermal model. The pressure-volume diagram is of similar form, and the power output and efficiency are quantitatively similar (albeit the efficiency of the Ideal Adiabatic model is about 10% lower for reasons described above). However the behaviour of the Ideal Adiabatic model is more realistic, in that the various results are consistent with the expected limiting behavior of real machines. Thus the heat exchangers become necessary components without which the engine will not function. The required differential equation approach to solution reveals the considerable amount of heat transferred in the regenerator, indicating its importance in the cycle, and provides a natural basis for extending the analysis to include non-ideal heat exchangers (**Simple analysis**). Thus the solution of the Ideal Adiabatic model equations is equivalent to a simulation of the engine behavior in all respects, from setting up the initial conditions until convergence to cyclic steady state is attained. Throughout this process all the variables of the system are available as by-products of the simulation and can be used for extending the analysis. Thus for example the mass flow rates through all the heat exchangers can be used in order to evaluate the heat transfer and flow friction effects over the cycle.

Chapter 5: Simple Analysis.

Once we have done an Ideal Adiabatic analysis on a specific Stirling engine, we would like to evaluate the heat transfer and flow-friction effects of the three heat exchangers on the performance of the engine. This will enable us to do a parametric sensitivity analysis as required for design optimization.

Forced convection heat transfer is fundamental to Stirling engine operation. Heat is transferred from the external heat source to the working fluid in the heater section, cyclically stored and recovered in the regenerator, and rejected by the working fluid to the external heat sink in the cooler section. All of this is done in compact heat exchangers (large wetted area to void volume ratio) so as to limit the "dead space" to an acceptable value and thus allow for a reasonable specific power output of the engine. We find that effective heat exchange comes at a price of increased flow friction, resulting in the so-called "pumping loss". This loss refers to the mechanical power required to "pump" the working fluid through the heat exchangers, and thus reducing the net power output of the engine.

The theory and analysis of these effects is extremely complex, and we find that we can only rely on the plethora of documented experimental and empirical studies (e.g. Kays & London ,"**Compact Heat Exchangers**"). Almost all of this vast body of work is based on steady flow conditions and is thus not directly applicable to the oscillating flow conditions that apply to Stirling engines. In this section we adopt a "Quasi-Steady Flow" approach, in that we assume that at each instant of the cycle the fluid behaves as though it is in steady flow. Thus we have called this analysis a "Simple" analysis because it is a gross simplification of an extremely complex process. At this stage there is still a major controversy about this approach, and we need to treat the results of this analysis with a healthy measure of skepticism. An alternative for design is the recent "Similarity and Scaling" approach which has been developed by Allan Organ and is presented in his book "**The Regenerator and the Stirling Engine**".

Chapter 5: Simple Analysis. Part A) Scaling Parameters (Flow Friction and Convective Heat Transfer)

Forced convection heat transfer is fundamental to Stirling engine operation. Heat is transferred from the external heat source to the working fluid in the heater section, cyclically stored and recovered in the regenerator, and rejected by the working fluid to the external heat sink in the cooler section. All of this is done in compact heat exchangers (large wetted area to void volume ratio) so as to limit the "dead space" to an acceptable value and thus allow for a reasonable specific power output of the engine. We find that effective heat exchange comes at a price of increased flow friction, resulting in the so-called "pumping loss". This loss refers to the mechanical power required to "pump" the working fluid through the heat exchangers, and thus reducing the net power output of the engine.

When we try to design a machine for a specific performance, we find that there are a large number of parameters involved which affect the performance-in non intuitive ways. The Scaling Parameter approach uses dimensional analysis to reduce the number of parameters to a basic set of dimensionless scaling groups, and thus allow experimental data to be used in various contexts. This is a standard technique in forced convection heat transfer analysis and can be reviewed in many heat transfer texts. We have based our analysis on the book "**Compact Heat Exchangers**" by Kays & London, both the 1955 and 1964 editions. An extremely lucid discussion of the basic scaling parameters involved and their applicability to the oscillating flow conditions of Stirling engines is found in the book " **The Regenerator and the Stirling Engine**" by Allan Organ (1997), and in particular Chapter 3: "Heat Transfer - and the Price". In this section we assume that you are familiar with the methods of convective heat transfer analysis and we simply define and discuss the various standard dimensionless scaling parameters for each heat exchanger section in the machine, based on a basic (exhaustive) set of variables as follows:

d - the Hydraulic Diameter (m). This variable represents ratio of the two important size parameters of a heat exchanger - the void volume V and the wetted area Awg. It is defined by:

$$d \equiv \frac{4V}{Awg}$$

The factor 4 is included for convenience. For flow in a circular pipe (or a homogeneous bundle of circular pipes) the Hydraulic Diameter thus becomes equal to the pipe internal diameter.
Note that some researchers (e.g. Allan Organ) use the so-called Hydraulic Radius (rh) as their scaling parameter. This is simply defined as rh = V/Awg, thus d = 4 rh.

μ - the working gas dynamic viscosity (Pa s). This is defined in terms of Newtons Law of Viscosity in the section on **Pumping Loss** (Chapter 5D).

u - the mean bulk velocity of the flowing fluid (m / s)

ρ - the density of the working gas (kg / cu.m)

h - the convective heat transfer coefficient (W / sq.m K). This is defined in the section on **Heater and Cooler Simple analysis**. (Chapter 5C)

k - the working gas thermal conductivity (W / m K)

cp - the working gas specific heat capacity at constant pressure (J / kg K)

Reynolds Number (Nre)

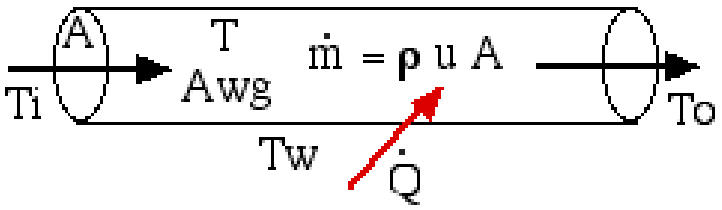
$$Nre \equiv \frac{\rho u d}{\mu}$$

This grouping is obtained by considering the ratio of the inertial forces to the viscous forces. The value of Nre determines the flow regime, whether laminar or turbulent. Both the friction factor and the heat transfer coefficient are strongly dependent on the flow regime, thus Nre is invariably used as the independent variable in the presentation of flow-friction and heat transfer data. Note that by definition Nre is always positive, independent of the direction of fluid flow.

Stanton Number (Nst)

$$Nst \equiv \frac{h}{\rho\ u\ cp}$$

This grouping is one of the two standard methods for the presentation of heat transfer by convection. The physical significance of Nst is that it can be related to the ratio of the convective heat transfer to the thermal capacity of the flowing fluid. It has found favour because of the ease in which it can be obtained from experimental data. Thus from an energy balance of a heated (or cooled) fluid flowing through a heat exchanger:



$$\dot{Q} = h\ Awg\ (Tw - T) = cp\ \dot{m}\ (To - Ti) = cp\ \rho\ u\ A\ (To - Ti)$$

where

$\dot{Q}$ is the rate of heat transfer
$\dot{m}$ is the mass flow rate
Awg is the wall/gas, or wetted area
A is the free flow area (normal to the direction of flow)
Tw, T are the respective wall and bulk fluid temperatures
Ti, To are the respective inlet and outlet fluid temperatures

Substituting for Nst above we obtain:

$$Nst = \frac{A}{Awg}\left(\frac{To - Ti}{Tw - T}\right)$$

Thus the value of Nst can be obtained directly from the heat exchanger dimensions and temperature measurement without reference to the fluid properties.

Another popular grouping named the "Number of Transfer Units", or NTU, can be defined from the above energy balance equation as follows:

$$NTU \equiv \frac{h\ Awg}{\rho\ u\ cp\ A} = Nst\frac{Awg}{A}$$

Note that the NTU is a function of the heat exchanger dimensions and as such is not considered a fundamental heat transfer grouping in the classical sense. However formulation in terms of NTU allows a solution in terms of Nst, and thus avoids the tedium of extracting the heat transfer coefficient h. This approach is used to advantage in the section on **Regenerator Simple analysis** (Chapter 5B).

Prandtl Number (Npr)

$$Npr \equiv \frac{cp\ \mu}{k}$$

This grouping is obtained from the ratio of the kinematic viscosity ν = μ / ρ (sq.m / s) (also known as the momentum diffusivity) to the thermal diffusivity α = k / ρ cp (sq.m / s), and thus represents the ratio of the viscous to the thermal boundary layers. Thus for fluids having a value of Npr close to unity the classic Reynolds Analogy can be used to relate simply between flow-friction and heat transfer data. It involves three fluid properties and is thus itself a property of the fluid, and frequently appears in heat transfer data. For the range of working gases used in Stirling engines and for the temperature range of interest (about 300 to 1000 K), Npr is approximately constant at around a value of 0.7.

Developing the Reynolds Analogy, Allan Organ (in Chapter 3 of his book " **The Regenerator and the Stirling Engine**") shows that one can relate the Stanton Number to the friction factor Cf (defined in the section on **Pumping Loss** (Chapter 5D)) as follows:

Nst = Cf / 2

According to Organ, "this vital result is not quantitatively exact, but serves a more valuable purpose than any precise formula by confirming the inevitable tie between friction factor and Stanton number. It warns against unrealistic expectation of increasing heat transfer without penalty of increased pumping power."

Nusselt Number (Nnu)

$$Nnu \equiv \frac{h\,d}{k}$$

This grouping is often used as an alternative to the Stanton Number for the presentation of heat transfer data, and is usually presented in graphical form in terms of the Prandtl and Reynolds numbers (e.g. Kays & London, " **Compact Heat Exchangers**"). It is not an independent grouping, and can be defined as a function of the other three dimensionless groups as follows:

Nnu = Nst.Npr.Nre

In the foregoing we have not expressed the influence of temperature on the fluid properties. Both dynamic viscosity μ and thermal conductivity k vary significantly with temperature. However, since the specific heat capacity cp and Prandtl Number Npr are approximately constant over the temperature range of interest we see from the definition of Npr that it is sufficient to consider the temperature dependence of the dynamic viscosity μ. This is considered in the section on **Pumping Loss** (Chapter 5D).

Chapter 5: Simple Analysis. Part B) Regenerator Simple Analysis

The first mathematical theories to describe regenerator operation were published in the late 1920s, more than 100 years after its invention by Robert Stirling. Significantly, these and subsequent theories of regenerator operation are based on assumptions which are neither relevant nor applicable to Stirling engine regenerators. More recently Allan Organ published a book "**The Regenerator and the Stirling Engine**" (John Wiley, 1997) which represents a significant step towards "bridging the gap" between Hausen's celebrated regenerator analysis, widely used in the analysis of gas turbine engines, and the unique conditions that apply to Stirling engines.

By definition a regenerator is a cyclic device. On the first part of the cycle the hot gas flows through the regenerator from the heater to the cooler, and in so doing transfers heat to the regenerator matrix. This is referred to as a "single blow". Subsequently during the second part of the cycle the cold gas flows in the reverse direction, absorbing the heat that was previously stored in the matrix. Thus at steady state the net heat transfer per cycle between the working gas and the regenerator matrix is zero.

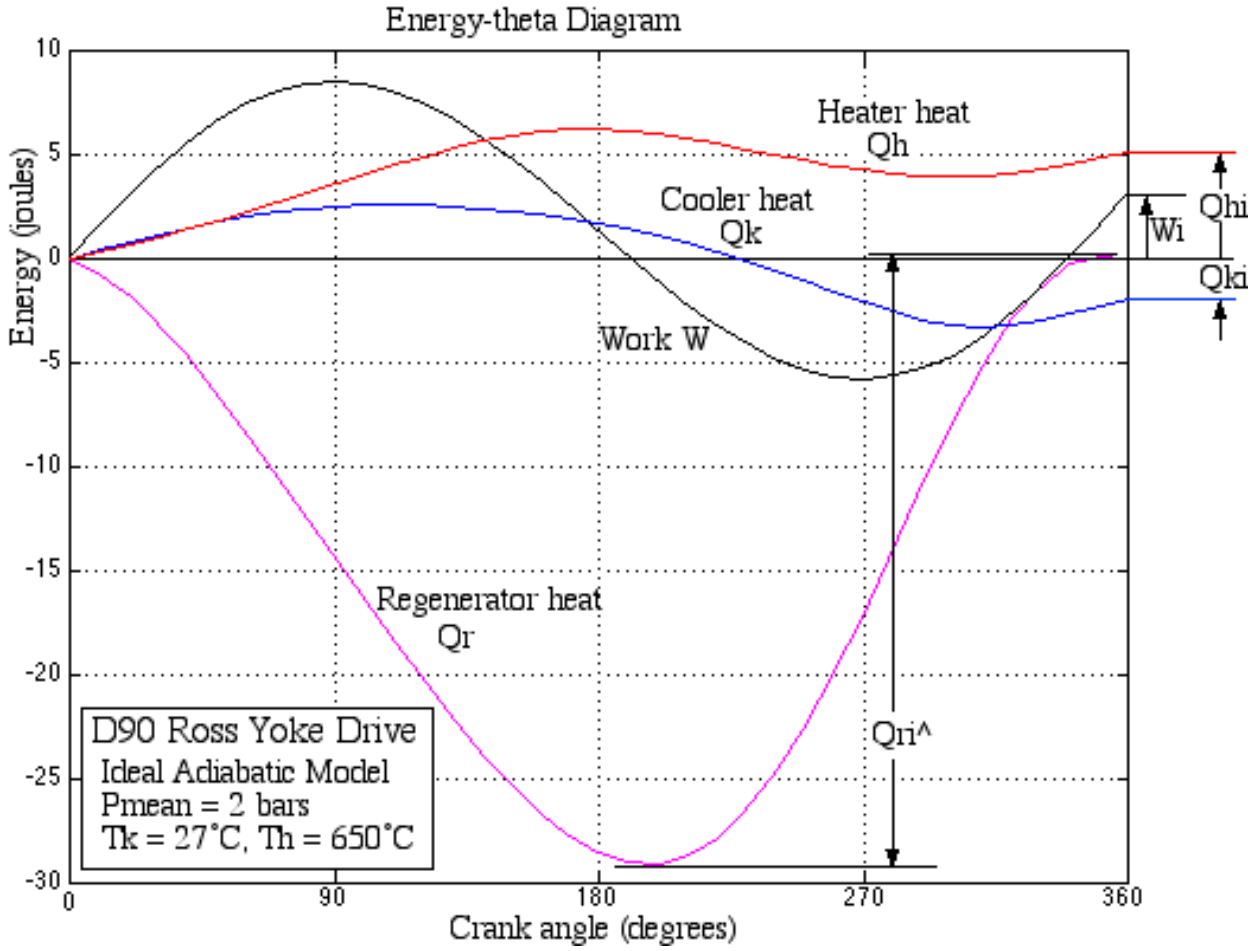
The regenerator quality is usually defined on an enthalpy basis in terms of a regenerator effectiveness ε as follows:

$$\varepsilon \equiv \frac{\text{actual enthalpy change of gas during a single blow through the regenerator}}{\text{equivalent maximum theoretical enthalpy change in an ideal regenerator}}$$

However this definition is not amenable to usage in Stirling engines. We propose an equivalent definition in the context of the **Ideal Adiabatic model** (Chapter 4A), which represents the limiting maximum performance measure, as follows:

$$\varepsilon \equiv \frac{\left(\begin{array}{l}\text{amount of heat transferred from matrix to gas}\\ \text{during a single blow through the regenerator}\end{array}\right)}{\left(\begin{array}{l}\text{equivalent amount of heat transferred in the}\\ \text{regenerator of the Ideal Adiabatic model}\end{array}\right)}$$

The regenerator effectiveness ε thus varies from 1 for an ideal regenerator (as defined in the Ideal Adiabatic model) to 0 for no regenerative action. Consider for example the cyclic energy-theta diagram which we obtained in the **Ideal Adiabatic analysis** (Chapter 4D) of the Ross D-90 engine:



The thermal efficiency of the Ideal Adiabatic cycle (suffix "i") is given in terms of the energy values accumulated at the end of the cycle by:

ηi = Wi / Qhi = (Qhi + Qki) / Qhi

Note from the diagram that Qki is a negative quantity, thus ηi = 0.627 for the D-90 engine as shown. Notice also the significant amount of heat transferred during a single blow of the regenerator given by Qri^. Thus for the D-90 engine as shown the ratio Qri^/Qhi = 5.66.

Now for a system having a non-ideal regenerator, during the single blow when the working gas flows from the cooler to the heater, on exit from the regenerator it will

have a temperature somewhat lower than that of the heater. This will result in more heat being supplied externally over the cycle by the heater in increasing the temperature of the gas to that of heater and can be written quantitatively as follows:

Qh = Qhi + Qri^(1 - ε)

Similarly, when the working gas flows from the heater to the cooler, then an extra cooling load will be burdened on the cooler, as follows:

Qk = Qki - Qri^(1 - ε)

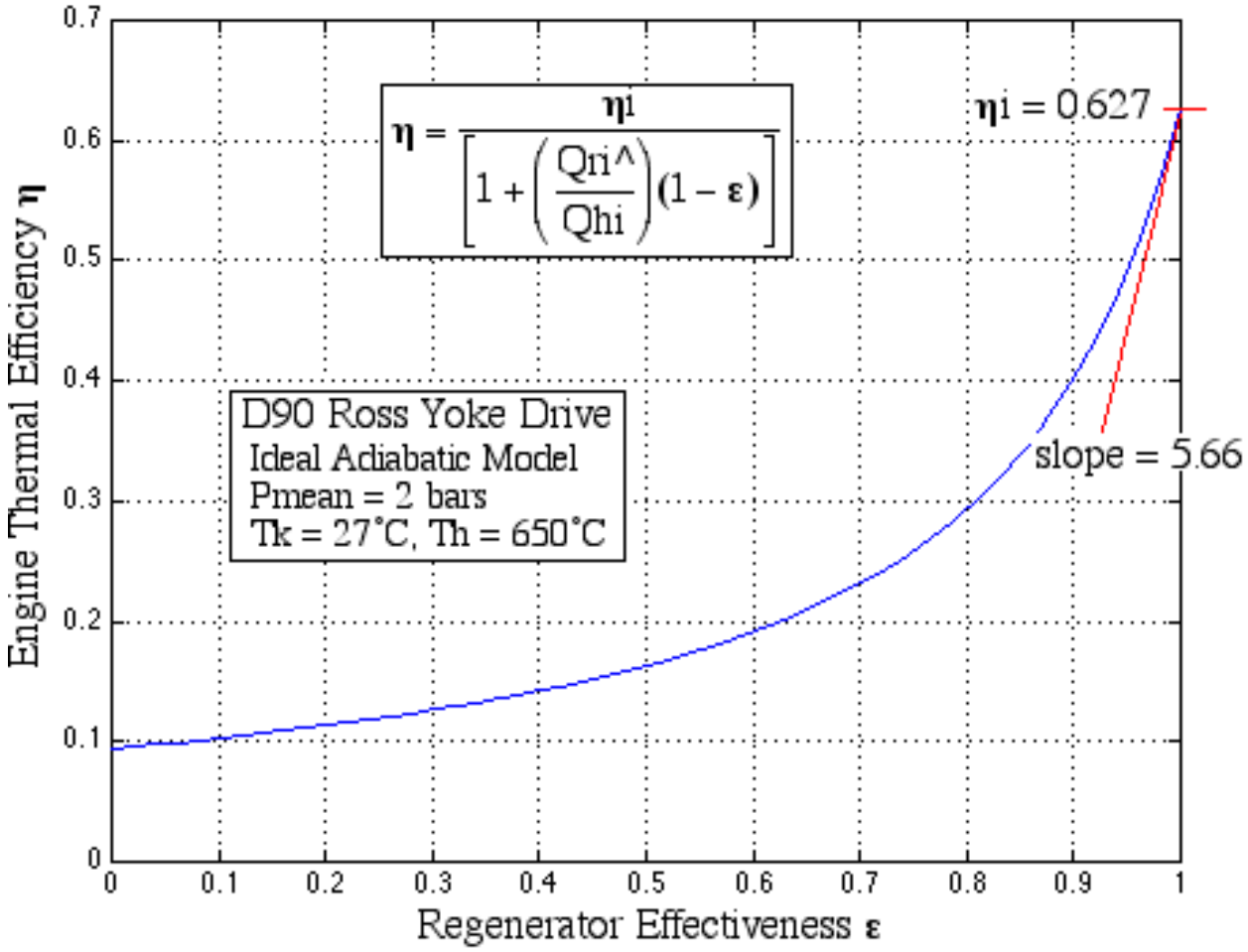
The thermal efficiency of the non-ideal engine (without the suffix "i") is given by:

η = W / Qh = (Qh + Qk) / Qh

Substituting for Qh, Qk, and ηi from the above equations we obtain:

$$\eta = \frac{\eta^{i}}{\left[1 + \left(\frac{Qri^{\wedge}}{Qhi}\right)(1 - \varepsilon)\right]}$$

The following diagram shows a plot of the above equation for the specific case of the D-90 engine and shows the effect of regenerator effectiveness ε on thermal efficiency η.



Notice that as ε varies from 1 for an ideal regenerator (Ideal Adiabatic cycle) to 0 for no regenerative action, the thermal efficiency η drops from more than 60% to less than 10%. Furthermore, differentiating the efficiency equation η with respect to ε, and substituting ε = 1 we obtain:

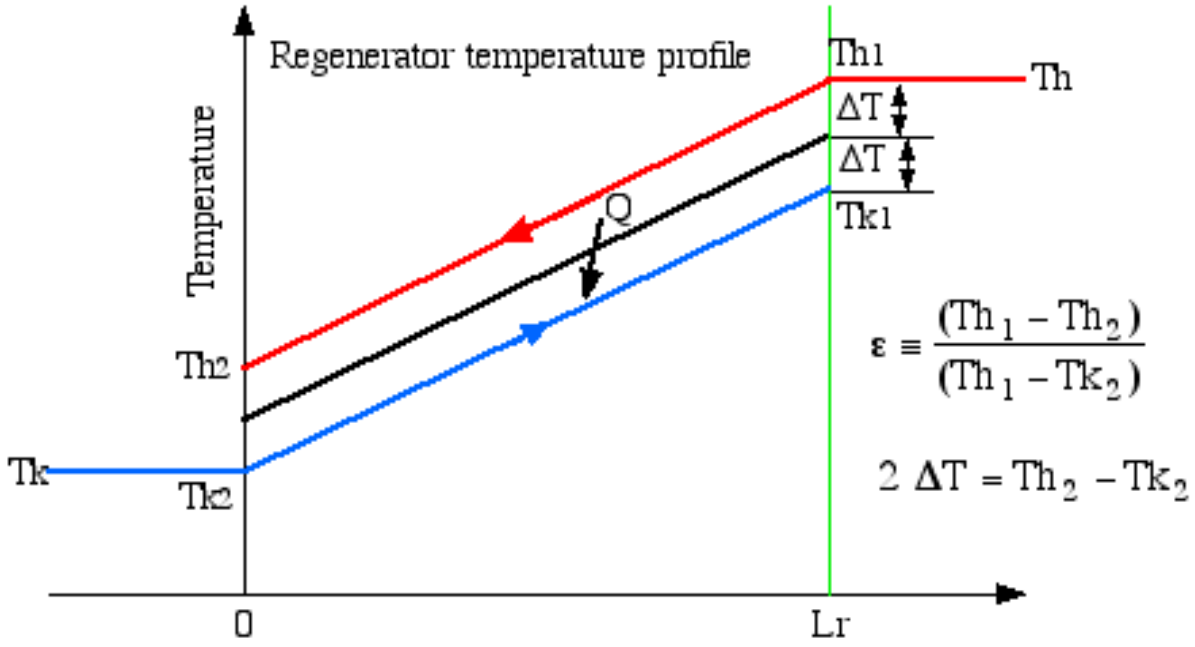
$$\left.\frac{d\eta}{d\varepsilon}\right|_{\varepsilon=1} = \eta i \frac{Qri^{\wedge}}{Qhi}$$
$$= 5.66\ \eta i \quad \text{for the D-90 Ross Yoke-drive engine.}$$

Thus we see that for highly effective regenerators (close to ε = 1) a 1% reduction in regenerator effectiveness results in a more than 5% reduction in thermal efficiency η. Furthermore we see that if one has a regenerator that has an effectiveness of 0.8, the thermal efficiency has dropped by half to around 30%. This not only means a significantly less efficient machine, but one that has to have a significantly larger

cooler. Obviously we need to have a means of determining the actual regenerator effectiveness in any specific machine.

Evaluating the Regenerator Effectiveness ε

We now consider the regenerator effectiveness in terms of the temperature profile of the 'hot' and 'cold' gas streams with respect to the regenerator matrix. We assume an equal difference in temperature ΔT on the hot and the cold sides, and linear temperature profiles, leading to the definition of regenerator effectiveness ε in terms of temperatures, as shown.



Combining the two equations in the figure, we obtain:

$$\varepsilon = \frac{1}{\left(1 + \frac{2\,\Delta T}{(Th_1 - Th_2)}\right)}$$

Now from energy balance considerations of the hot stream, the change in enthalpy of the hot stream is equal to the heat transfer from the hot stream to the matrix, and subsequently from the matrix to the cold stream, thus:

$$\dot{Q} = cp\ \dot{m}\ (Th1 - Th2) = 2\ h\ Awg\ \Delta T$$

where $\dot{Q}$(watts) is the heat transfer power, h is the overall heat transfer coefficient (hot stream / matrix / cold stream), Awg refers to the wall/gas, or "wetted" area of the heat exchanger surface, cp the specific heat capacity at constant pressure, and $\dot{m}$ (kg / s) the mass flow rate through the regenerator. Substituting in the effectiveness equation we obtain:

$$\varepsilon = \frac{1}{\left(1 + \frac{cp\,\dot{m}}{h\,Awg}\right)}$$

We now introduce the concept of Number of Transfer Units (NTU) which is a well known measure of heat exchanger effectiveness, and is defined in the section on **Scaling Parameters** (Chapter 5A).

$$NTU \equiv \frac{h\,Awg}{cp\,m}$$

Thus:

$$\varepsilon = \frac{NTU}{(1 + NTU)}$$

Notice that the NTU value is a function of the type of heat exchanger as well as its physical size. It includes the actual wetted area Awg, as well as the actual mass flow through the regenerator$\dot{m}$ (kg/s). In heat exchanger analysis it is more usual to evaluate local heat exchanger parameters in terms of fluid property values which are independent of size. Thus we define a Stanton number (refer to the section on **Scaling Parameters** (Chapter 5A)) as follows:

NST = h / (ρ u cp)

where ρ is the fluid density, and u is the fluid velocity, thus:

$$\dot{m} = \rho\,u\,A$$

where A is the free flow area through the matrix. Tables and graphs of empirical values of Stanton number vs Reynolds number are available from heat exchanger texts for various heat exchanger types. We use two types of matrices in our regenerator analysis - woven mesh and coiled annular foil. The NTU value can then be obtained in terms of the Stanton number as follows:

NTU = NST (Awg / A) / 2

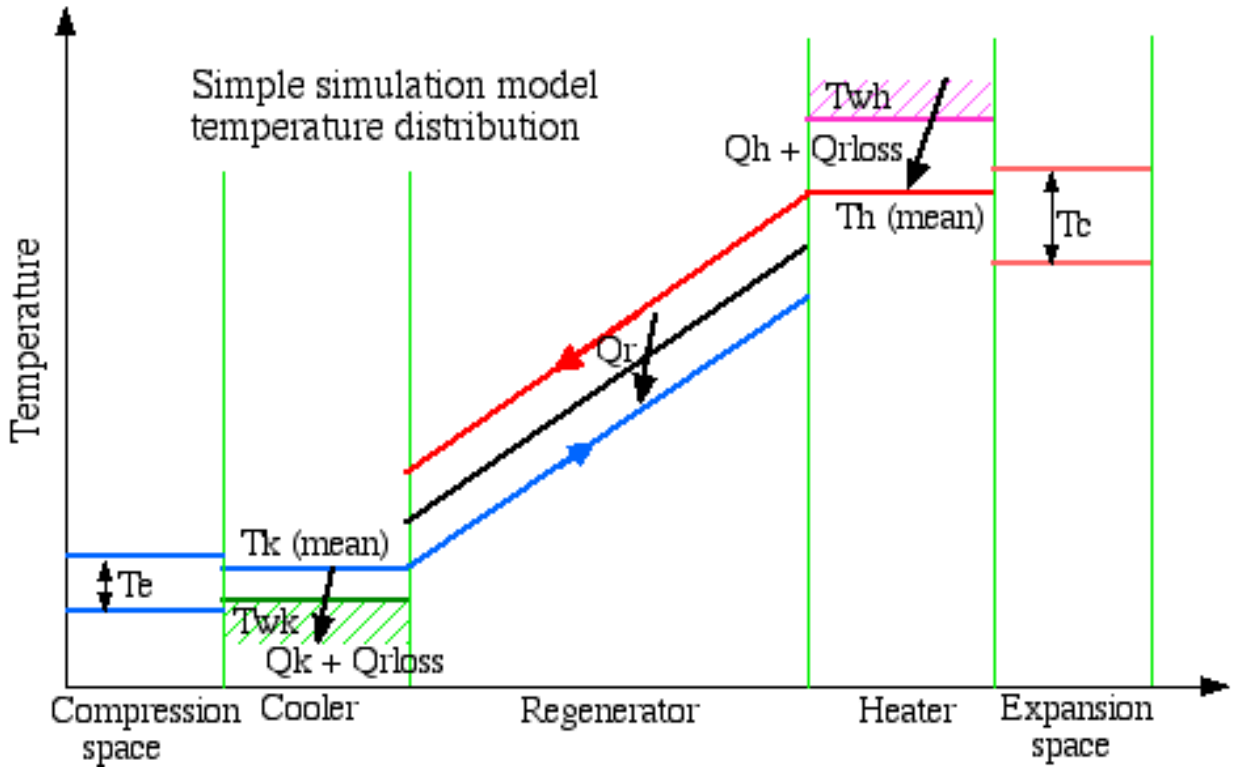
The factor 2 in this equation is unusual, and stems from the fact that the Stanton number is usually defined for the transfer of heat from the gas stream to the matrix alone, whereas the NTU usage in this section is for overall transfer of heat from the hot stream to the regenerator matrix, and subsequently to the cold stream.

Chapter 5: Simple Analysis. Part C) Heater and Cooler Simple Analysis

One normally determines the effectiveness of a heater or cooler heat exchanger in a similar way to that of the regenerator in terms of the following equation:

$$\varepsilon = 1 - e^{-NTU}$$

where ε is the effectiveness of the heat exchanger and NTU is the "Number of Transfer Units" (Refer to "**Compact Heat Exchangers**", Kays & London). Both concepts are described in the section on '**regenerator Simple analysis**' (Chapter 5B). Unfortunately, we are unable to determine a simple relation between the heater and cooler effectiveness and the engine efficiency, as we were able to do with the regenerator. Referring to the temperature profile diagram below we observe that the non-ideal heater and regenerator result in the mean effective temperature of the gas in the heater space (Th) being *lower* than that of the heater wall (Twh). Similarly the non-ideal cooler and regenerator result in the mean effective temperature of the gas in the cooler space (Tk) being *higher* than that of the cooler wall (Twk). This has a significant effect on the engine performance, since it is effectively operating between lower temperature limits than those of the heater and cooler walls. Thus the Simple analysis of the heater and cooler iteratively determines these temperature differences using the convective heat transfer equations, the values of Qh and Qk being evaluated by the Ideal Adiabatic analysis and the value of the regenerator enthalpy loss Qrloss being evaluated in terms of the regenerator effectiveness.



From the basic equation for convective heat transfer we obtain:

$$\dot{Q} = h\ Awg\ (Tw - T)$$

where $\dot{Q}$(watts) is the total heat transfer power (including the regenerator net enthalpy loss), h is the convective heat transfer coefficient, Awg refers to the wall/gas, or "wetted" area of the heat exchanger surface, Tw is the wall temperature, and T the gas temperature. In order to reduce the units of this equation to the net heat transferred over a single cycle Q (joules/cycle) we divide both sides by the frequency of operation (freq), thus:

Qk - Qrloss = hk Awgk (Twk - Tk) / freq

Qh + Qrloss = hh Awgh (Twh - Th) / freq

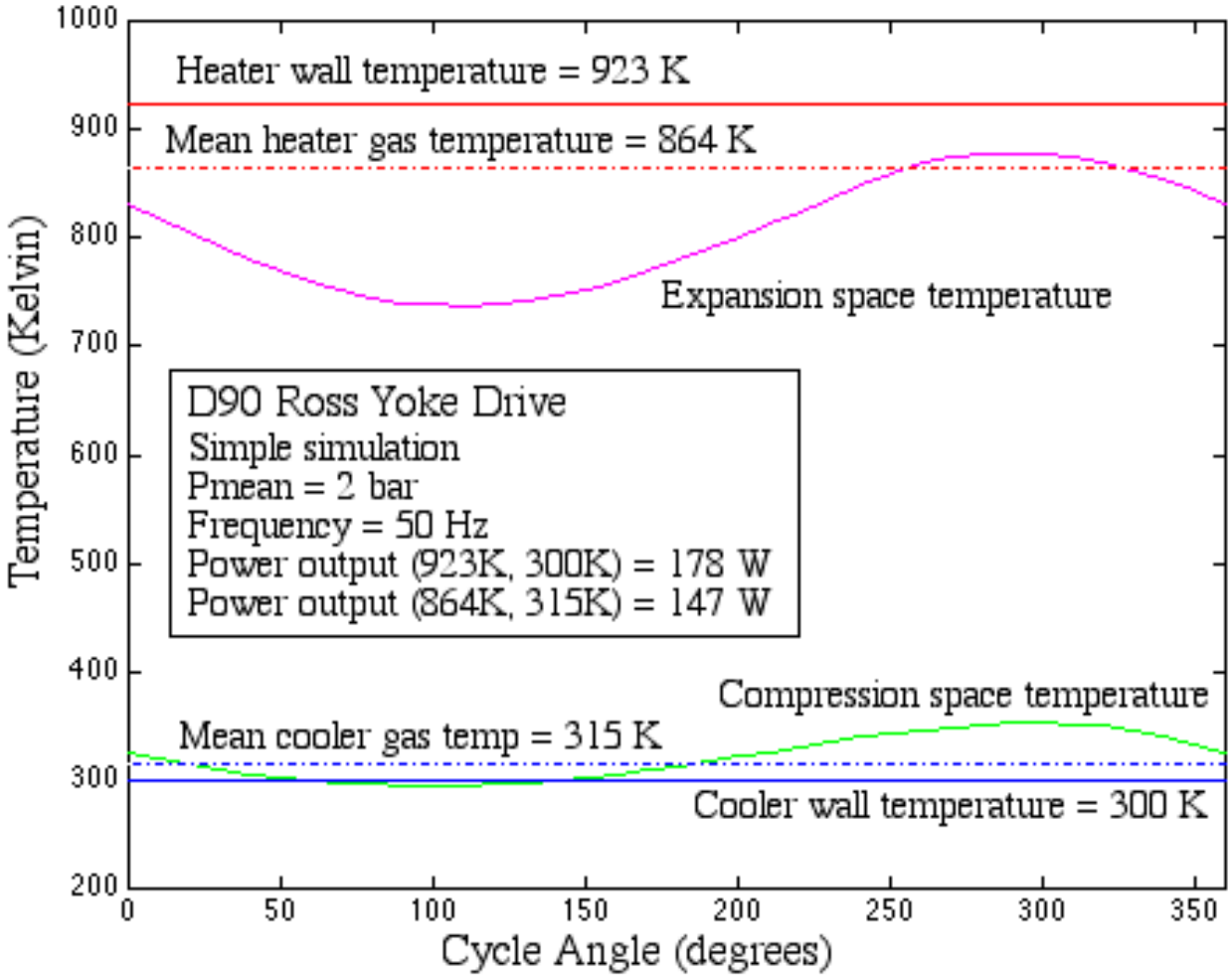
where, as shown in the diagram above, the suffix h refers to the heater, and the suffix k refers to the cooler. We now rewrite these equations to evaluate the respective gas temperatures Tk and Th:

Tk = Twk - (Qk - Qrloss) freq / (hk Awgk)

Th = Twh - (Qh + Qrloss) freq / (hh Awgh)

The Simple solution algorithm requires iterative invoking of the Ideal Adiabatic simulation, each time with new values of Tk and Th, until convergence is attained. After each simulation run values of Qk and Qh are available and Qrloss is determined in terms of the regenerator effectiveness. The mass flow rates through the heater and cooler are used to determine the average Reynolds numbers and thus the heat transfer coefficients in accordance with the methods in the section on **Scaling Parameters** (Chapter 5A). Substituting these values in the above equations yields Tk and Th, and convergence is attained when their successive values are essentially equal.

The Simple simulation of our D90 Ross Yoke-drive engine case study results in the temperature distribution as shown below. Notice that the mean temperature of the gas in the heater space is 59 degrees below that of the heater wall, and similarly the mean temperature of the gas in the cooler space is 15 degrees above that of the cooler wall. This lower temperature range of operation reduced the output power from 178 W to 147 W


Heater wall temperature = 923 K
Mean heater gas temperature = 864 K
Expansion space temperature
D90 Ross Yoke Drive
Simple simulation
Pmean = 2 bar
Frequency = 50 Hz
Power output (923K, 300K) = 178 W
Power output (864K, 315K) = 147 W
Compression space temperature
Mean cooler gas temp = 315 K
Cooler wall temperature = 300 K
Temperature (Kelvin)
Cycle Angle (degrees)
1000
900
800
700
600
500
400
300
200
0
50
100
150
200
250
300
350

Chapter 5: Simple Analysis. Part D) Pumping Loss Simple Analysis

Throughout this analysis we have assumed that at any instant the pressure is constant throughout the engine. However we find that the high heat fluxes required in the heat exchangers in turn requires a large wall/gas, or wetted area Awg. This requirement together with the conflicting requirement of a low void volume will result in heat exchangers with many small diameter passages in parallel. The fluid friction associated with the flow through the heat exchangers will in fact result in a pressure drop across all the heat exchangers which has the effect of reducing the power output of the engine. This is referred to as the "Pumping Loss" and in this section we attempt to quantify this power loss. We first evaluate the pressure drop across all three heat exchangers with respect to the compression space. Subsequently we can determine the new value of work done by integrating over the complete cycle, and isolate the Pumping Loss term as follows:

$$\text{thus:} \quad W = We + Wc = \oint p\, \mathbf{d}Vc + \oint (p - \Sigma\Delta p)\, \mathbf{d}Ve$$

where the summation $\Sigma\Delta p$ is taken over the 3 heat exchangers

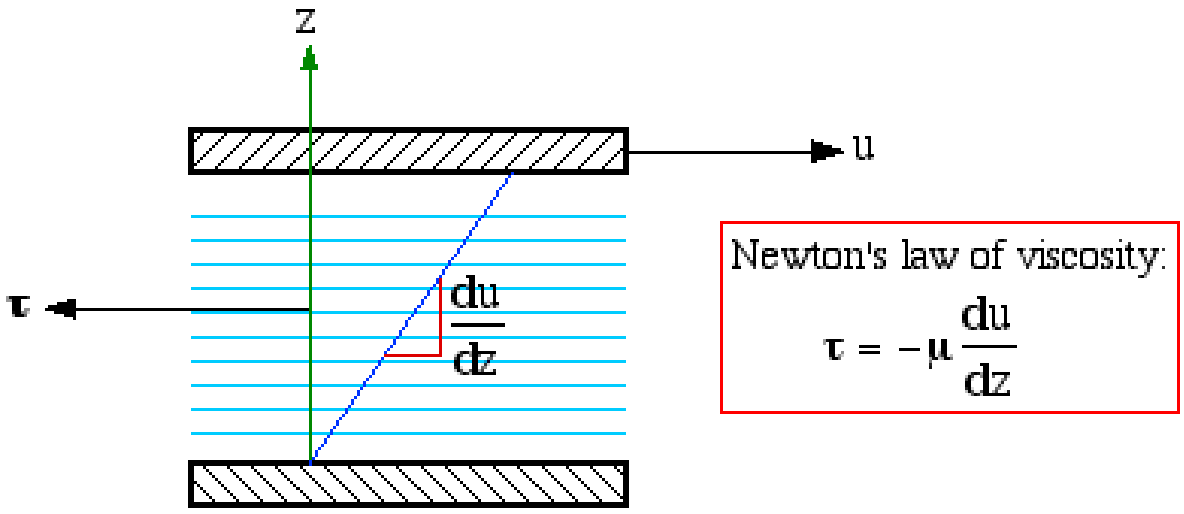
$$\text{thus:} \quad W = \oint p(\mathbf{d}Vc + \mathbf{d}Ve) - \oint \Sigma\Delta p\, \mathbf{d}Ve \quad = Wi - \Delta W$$

where Wi is the Ideal Adiabatic work done per cycle, and
ΔW is the pressure drop loss or pumping loss per cycle

thus:

$$\Delta W = \int_0^{2\pi} \left(\sum_{i=1}^{3} \Delta p_i \frac{\mathbf{d}\,Ve}{\mathbf{d}\theta} \right) \mathbf{d}\theta$$

The pressure drop Δp is due to fluid friction as it flows through the heat exchanger sections. Our model has assumed one-dimensional flow throughout, however the fundamental concepts of fluid friction paradoxically break down under one-dimensional flow. Newton's law of viscosity states that the shear stress τ between adjacent layers of fluid is proportional to the velocity gradient (**d**u/**d**z) in these layers normal to the flow direction, as shown:

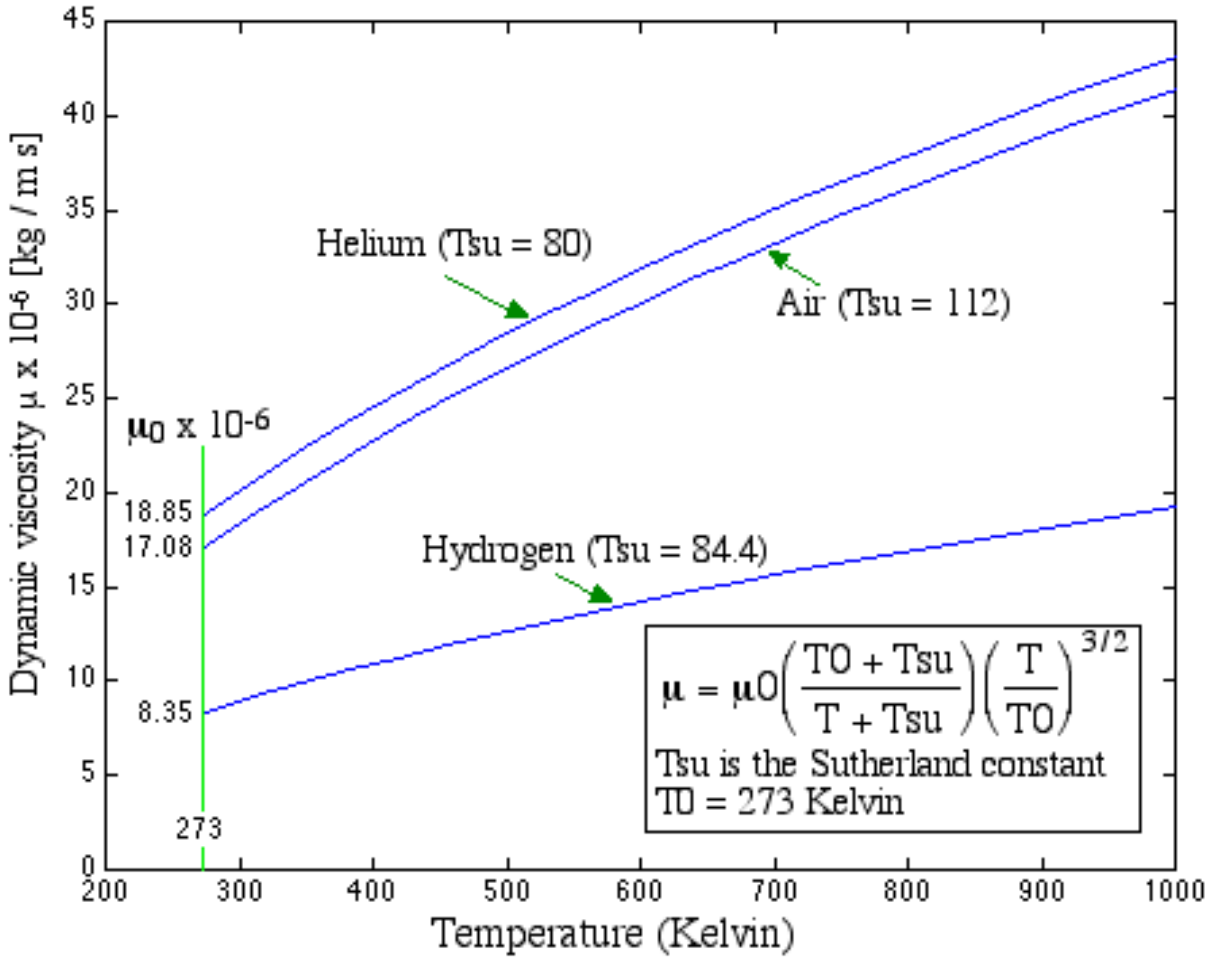


Newton's law of viscosity:

$$\tau = -\mu \frac{du}{dz}$$

From the equation we see that a Newtonian fluid cannot sustain a shear stress unless the flow is two dimensional. This paradox is bypassed by stating that the flow is not strictly one-dimensional, but rather represented by its mean bulk mass flow rate. The dynamic viscosity μ is basically a measure of the internal friction which occurs when the molecules of the fluid in one layer collide with molecules in adjacent layers travelling at different speeds, and in so doing transfer their momentum.

Over the pressure range of interest the dynamic viscosity μ is independent of pressure. Its temperature dependence for the gasses of interest is obtained as in the diagram below (Refer: Bretsznajder, A, 1971, "Prediction of the transport and Other Physical Properties of Fluids", International Series of Monographs in Chemical Engineering, II, Oxford: Pergamon)



The frictional drag force F is related to the shear stress τ as follows:

F = τ Awg

where Awg is the wall/gas, or wetted area of the heat exchanger.

In setting up the working expressions to describe pumping loss we introduce the concept of a "hydraulic diameter" d, which describes the ratio of the two important variables of a heat exchanger - the void volume V and the wetted area Awg:

d = 4 V / Awg

The factor 4 is included for convenience. For flow in a circular pipe (or a homogeneous bundle of circular pipes) the hydraulic diameter thus becomes equal to the pipe internal diameter. Substituting in the force equation above:

F = 4 τ V / d

We now define a Coefficient of Friction Cf as the ratio of the shear stress τ to the "dynamic head" (Refer to "**Compact Heat Exchangers**", Kays & London):

$$Cf \equiv \frac{\tau}{\frac{1}{2}\rho u^2}$$

where ρ is the fluid density and u is the fluid bulk velocity. Thus substituting for τ in the force equation we obtain the frictional drag force in terms of the Coefficient of Friction:

$$F = 2\ Cf\ \rho\ u^2\ V/d$$

Under the quasi-steady flow assumption (no acceleration or deceleration forces) the frictional drag force is equal and opposite to the pressure drop force, thus:

$F + \Delta p\ A = 0$

where A is the cross sectional (free flow) area. Substituting for F, the pressure drop Δp is given by:

$$\Delta p + 2\ Cf\ \rho\ u^2\ V/(d\ \ A) = 0$$

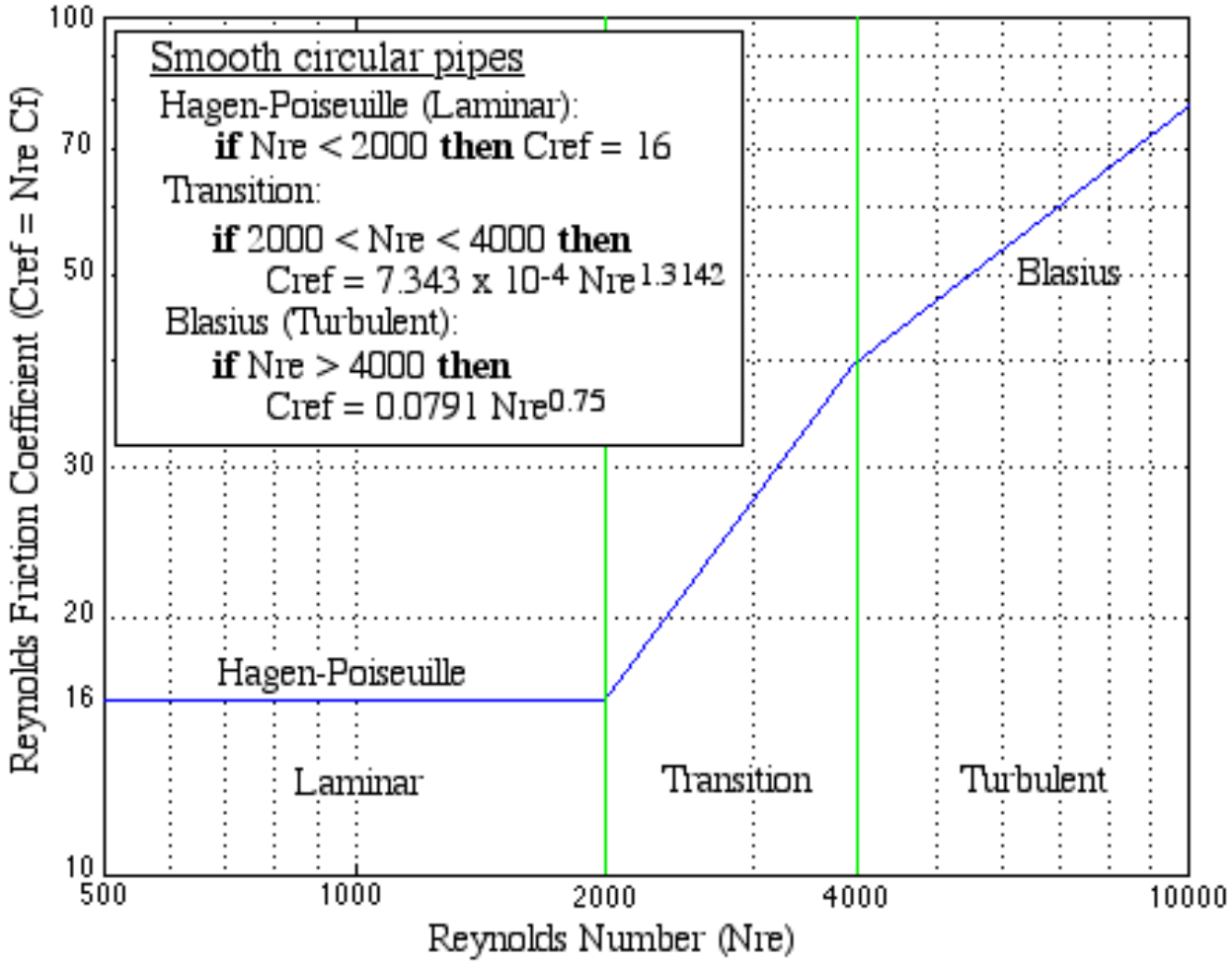
Note that Δp can be positive or negative, depending on the direction of flow. However the second term in this equation is always positive, and thus the equation violates the momentum conservation principle in the case of reversing flow. We resolve this by defining a "Reynolds Friction Coefficient" (Cref) by multiplying the Reynolds Number by the Coefficient of Friction as follows:

$Cref = Nre\ Cf$

Where $Nre = \rho\ u\ d\ /\ \mu$ is the Reynolds Number, defined and discussed in the section **Scaling Parameters** (Chapter 5A). By definition, the Reynolds Number is always positive, independent of the direction of flow. Thus finally:

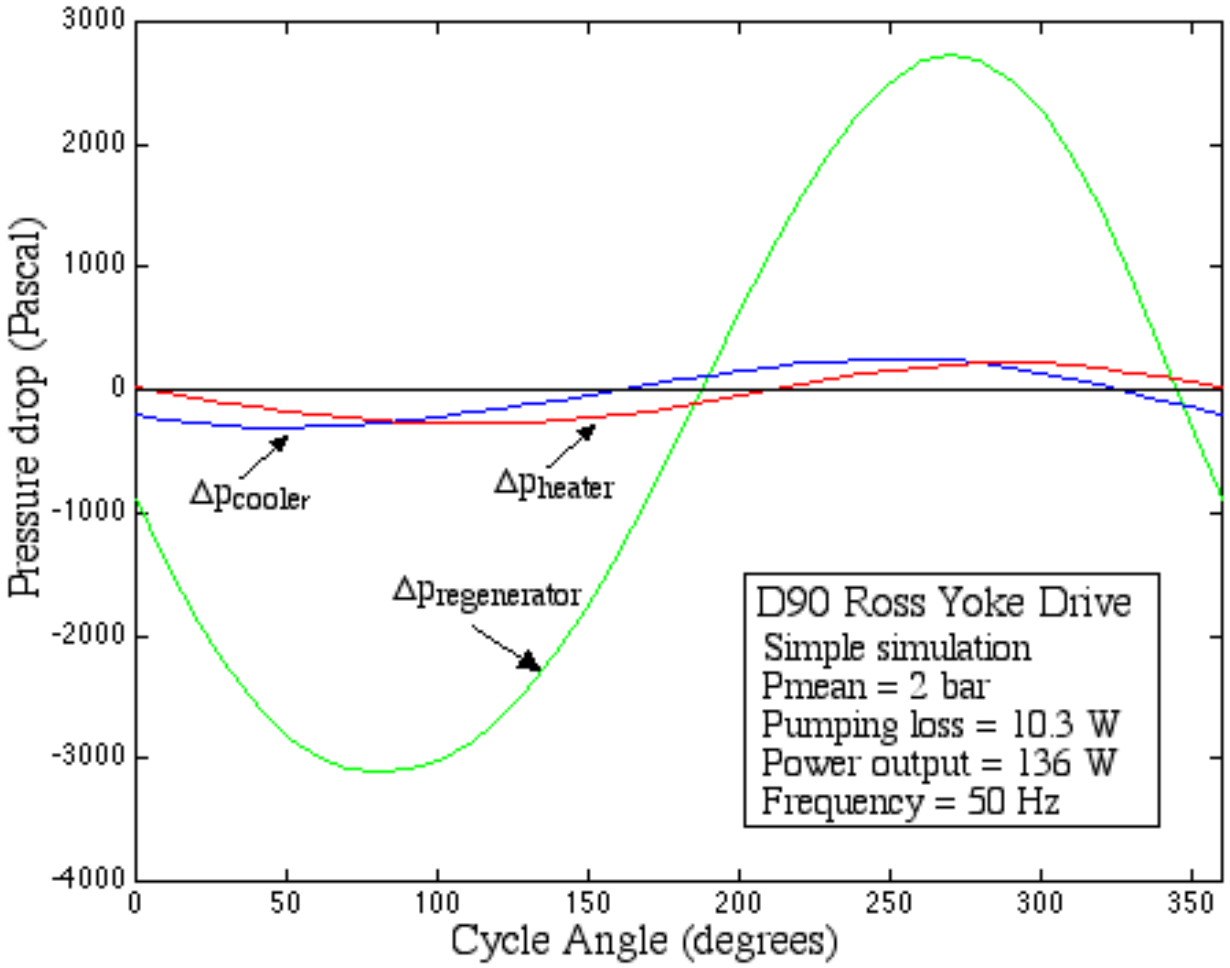
$$\Delta p = \frac{-2\ Cref\ \mu\ u\ V}{d^2\ A}$$

This equation satisfies the momentum conservation principle for both positive and reversed flow, since the sign of Δp is always correctly related to the sign of the velocity u. Since all current empirical data on the Coefficient of Friction is presented as a function of Reynolds Number, it is a simple matter to convert that data to the required Reynolds Friction Coefficient. For example, the Coefficient of Friction vs Nre curves for circular pipes (Moody Diagram) have been in widespread use over the past half a century. These curves have been simplified and rearranged in terms of the Reynolds Friction Coefficient Cref as follows:

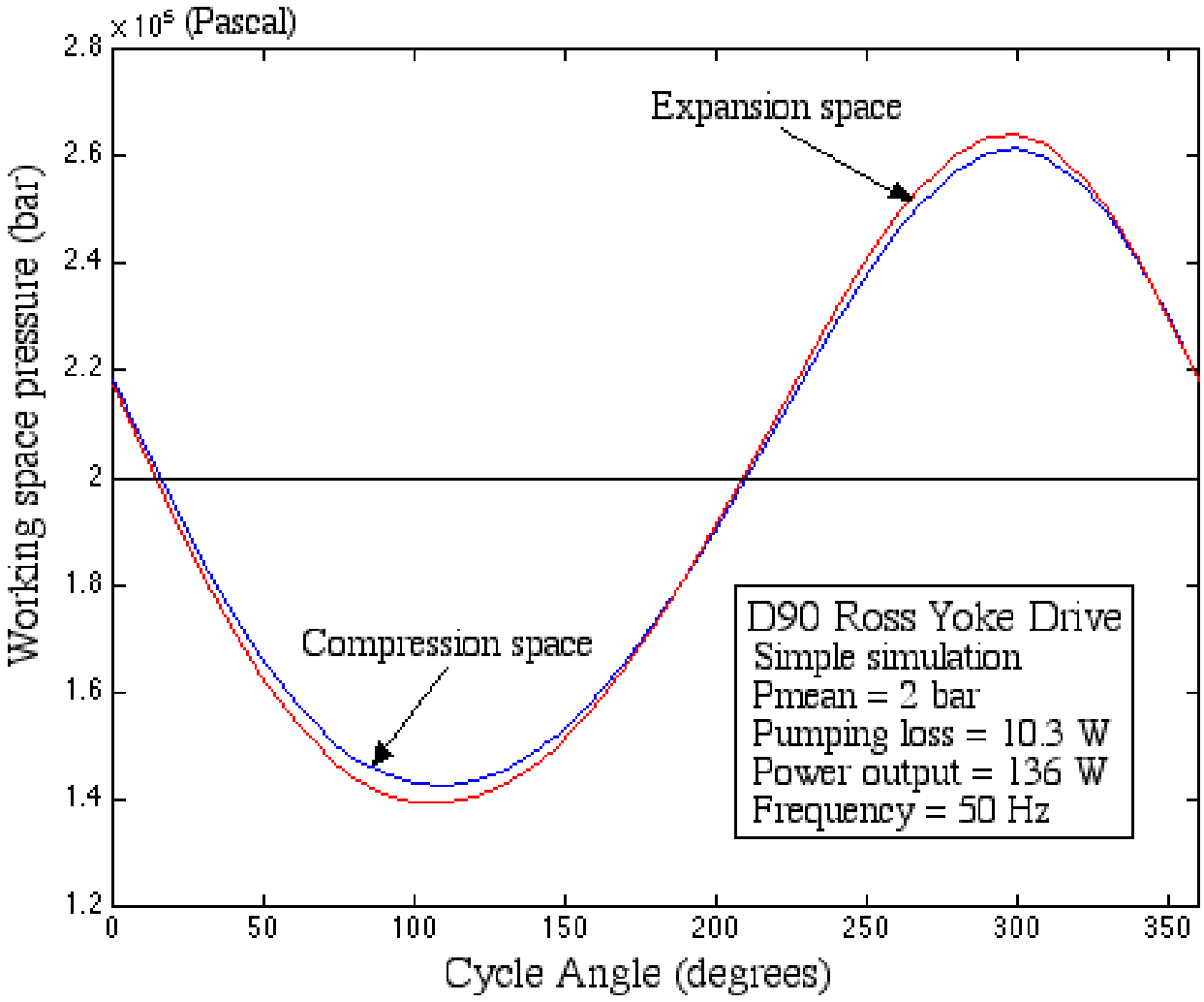


Similar formulations can be done for the various heat exchanger and regenerator types of interest. (Refer to " **Compact Heat Exchangers**", Kays & London).

The Simple simulation of our D90 Ross Yoke-drive engine case study results in the following pressure vs crankangle plots. The first plot shows the pressure drop across the three heat exchangers. Note the relative magnitude (as well as the phase) of the regenerator pressure drop with respect to those of the heater and cooler.



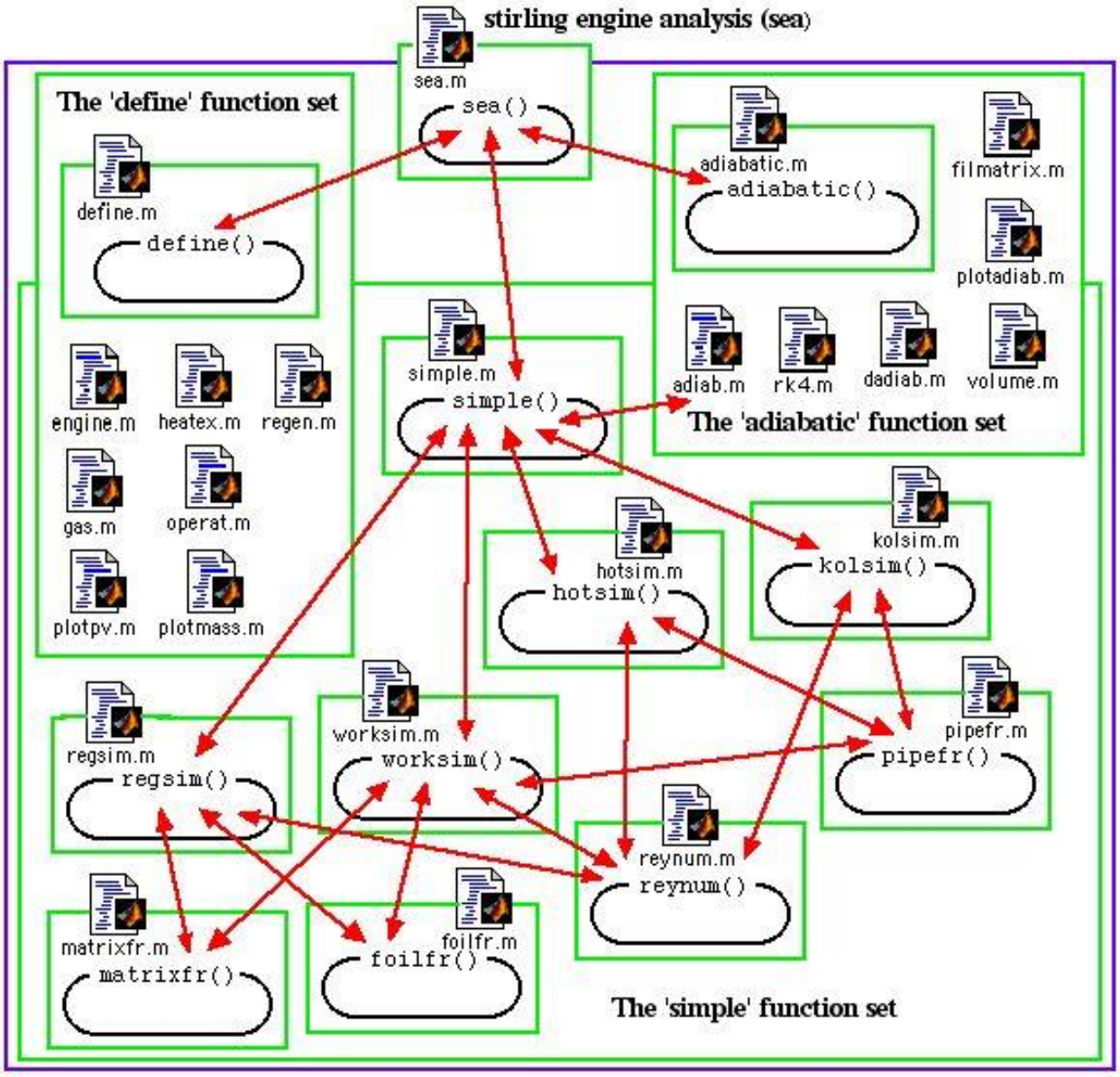
The following plot shows the expansion and compression space pressures vs crankangle. Under these conditions the pumping loss is 10.3 W, or about 7.5% of the net output power.



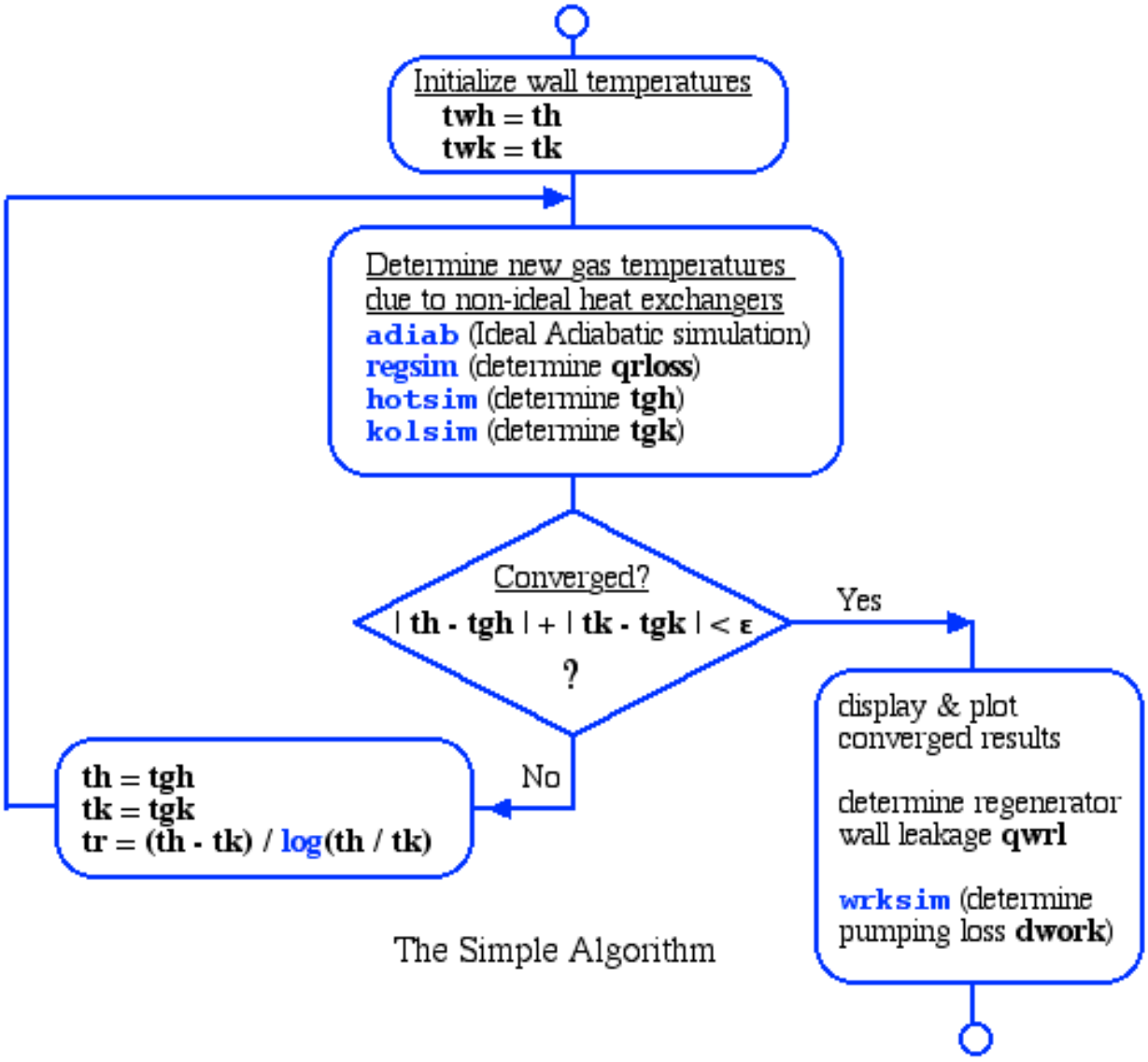
The complete simulation package including the pressure drop (pumping) loss, heat transfer and regenerator loss is described in the following section "Function set '**simple**'".

Chapter 5: Simple Analysis. Part E) Function Set 'Simple'

From the flow diagram below we see that the main program **sea** (stirling engine analysis) (Appendix V.1) first defines the system to be simulated in terms of the set of global variables set up by the **define** (Chapter 3C) set of functions, as described previously. It then invokes either the function **adiabatic** (Appendix W) which does an Ideal Adiabatic simulation, or the function **simple** (Appendix X) to do a Simple simulation to evaluate the heat transfer and pressure drop loss effects. Function set 'simple' includes nine functions, all contained in separate m-files as shown. The four main functions are **hotsim** (Appendix V.2) and **kolsim** (Appendix V.3) to respectively evaluate the heater and cooler gas temperatures, **regsim** (Appendix V.4) to evaluate the regenerator effectiveness and resulting enthalpy loss, and **worksim** (Appendix V.5) to evaluate the pumping loss. The **heat-transfer/flow-friction** (Appendix Y) function set includes function **reynum** to evaluate the instantaneous Reynolds Number, and the three functions **pipefr**, **foilfr** and **matrixfr** to determine the various flow friction and heat transfer coefficients.



The dynamics of the solution algorithm lies in function **simple** (Appendix X), as shown in the following flow diagram. Thus the function set **define** (Chapter 3C) specifies the operating conditions, including the temperature bounds **th** and **tk**. Since the temperature bounds of the working gas affect both the power output and efficiency, the simple routine invokes **adiab** (Appendix W), **regsim** (Appendix V.4), **hotsim** (Appendix V.2), and **kolsim** (Appendix V.3) in a loop until convergence of the gas temperatures is attained.



The Simple Algorithm

The nine functions of the set **simple** are included in the following m-files (refer to the flow diagram above): [**simple.m**, **hotsim.m**, **kolsim.m**, **regsim.m**, **reynum.m** , **worksim.m**, **pipefr.m**, **foilfr.m** and **matrixfr.m**]. As before, these can be directly copied from this website and used in a system which has MATLAB installed.
Notice that there are a limited number of heat exchanger configurations specified. It is intended that the user will modify and augment this system as required for specific systems, and as more updated heat transfer correlation data becomes available.

Appendices

Appendix A: William T. Beale

Appendix A.1: William T. Beale (1928 – 2016)

William Beale was a **mechanical engineering professor at Ohio University** in the 1960's. In 1964 he invented one of the most ingenious external heat source engines yet devised – the Free Piston Stirling Engine. Based on this invention he founded the company **Sunpower, Inc**. in 1974, which over the years developed many innovative, high efficiency heat engines and cryocoolers, using various external heat sources, including solar and wood pellets.

William Beale was a man of ideas and actions whose mind never stopped looking for ways to improve the world even to the end. He lived what he preached.

At the memorial service held on 24 September 2016, William's son, Dan Beale, presented the following “Remarks at the memorial service for my father” (Appendix A.2).

William T. Beale – 1977

William T. Beale – 2016

I visited William a few weeks before his passing, and he sent me the following statement which he requested that I present at the 2016 **ISEC**, to which I have added some links:

Statement by William (July 2016):

I was happy to see that there are so many keynote speeches scheduled at ISEC 2016 from free piston Stirling companies. I gather that free piston products dominate certain niche markets. My goal in this statement is to point out that mass markets are now within reach.

It seems to me possible to dramatically simplify your free piston designs, stamp out excess cost and attack much larger markets. Some obvious steps present themselves.

The first - examine what free piston products performed best in the past in terms of balance, sealing, centering, heat exchanger designs and so forth. Look at what NASA chose for deep space power - and the remarkable performance they achieved.

Consider the free piston alpha, which beats Diesel as a vehicle power plant, according to **NASA technical memorandum 82992**, and also provides high efficiency in a very wide range of sizes from 100 Watts to megawatts, and quick response to load.

Revisit the 1985 text "**Free Piston Stirling Engines**" by Walker and Senft, to see how well some of the cheap early machines performed, and how versatile these engines were in many applications. An example from the book is the simple hydrogen-charged free cylinder engine, with its automatic load matching and high specific power. Also look up **Biowatt**, an early wood-pellet fired system.

By learning from such past efforts cost may be reduced greatly, creating big mass market opportunities. Foremost among these, in the carbon-constrained world of the near future, is the biomass-fired home power plant which complements photovoltaics. Any fuel, with or without pre-processing, could be used to power a free piston Stirling that automatically recharges a home battery bank when charge levels get low. If pyrolysis is used for pre-processing and the resulting biochar is returned to the soil, one has the ideal future power plant: un-interruptible, maintenance-free and carbon-negative in all sizes from domestic on up.

The free piston Stirling community history gives examples of opportunities to simplify designs and slash costs to a fraction of the current values. If you succeed in that, you'll see that the age of commercial Stirling is just beginning.

Best wishes to all
William T. Beale (1928-2016)

Based on discussions with David Berchowitz, I wish to add the following to William's statement concerning the **free-cylinder water pump**. In this engine a heavy internal mass provides the reaction force driving the cylinder which is directly connected to the water pump. It has built in power adjustment and responds to load automatically. All other engines require a transmission and complicated control mechanisms to do this. Furthermore, there is no other mechanical heat engine that we know of that operates from infinite load to zero without either stalling or

destroying itself.
Another attractive feature of the free cylinder system is that it can be constructed from inexpensive easy-to-obtain components. In fact, the entire pump housing can be fabricated from ordinary PVC piping and fixtures. The reliability, simplicity and low cost of this engine makes it eminently suitable for application in developing countries, and in the 1970's it was extensively tested both in the field and in the laboratory (Refer to the 1979 presentation by William **A Free-Cylinder Stirling Engine for Solar Powered Water Pumps** (https://www.osti.gov/biblio/5070310)).

Appendix A.2: Remarks at the Memorial Service for My Father

Lean and Data-Driven: Remembering the
Founder of a World-Class Renewable
Energy Company
Remarks at the Memorial Service for my father

Dan Beale
24 Sept 2016

I had the good fortune to work with my father on a weekly basis, and sometimes on a daily basis over the past 10 years, first in my role as Sunpower Chair of Board and later as a collaborator on engineering projects. Through many conversations over the years I learned that my father never worked alone, but that what he accomplished, he accomplished because he had the support of a lot of people, many of whom I see sitting here today. Thank you to those who worked with my father to achieve great things.

In thinking about my father and what he meant to me, I found two statements from those who knew him to be particularly illuminating. The first, from a Sunpower technician, is as follows:

“It is great, but common, for a person to believe that society should be energy efficient and environmentally conscious. But it is greater and more rare for a person to take a stand by actively working to make it happen.”

That's from a technician who worked with my father 20 years ago.

The second statement is from an investment banker who worked with my father only 4 years ago:

“Your Dad was a great guy. I enjoyed talking with him any time he was available. A very fertile mind, and one of the real classic Americans in invention and leadership. He had insights that no one else had, and he acted on those. The first is rare and the second very rare.”

Two very different observers of my father saying similar things: He had insights that no one else had, and he acted on those. The first is rare and the second very rare.

This must be hitting on something important about my father, so I will focus on the sentiment of these two observers during my remarks today.

What I will not do today is talk about his personality or tell you any of his stories - a short video covering such topics is available on the internet[1].

What I want to focus on is an approach I saw my father use again and again, at really tough times in his life and in really triumphant times in his life as well.

And that approach has two parts. The first part is nothing more nor less than the scientific method, applied to human affairs as well as engineering. He had those insights, and he had them because he really, deeply understood the scientific method. He said to me many times, “be as uncertain as your data permits.” And that means two things. It means don't be too sure, don't be over-confident. But it also means – if your data is telling you something you don't want to believe – you must believe it, it's true. My father was ahead of his time in his broad application of the scientific method to human affairs. Others who apply this method broadly have recently revealed new truths in a great many fields, for example in health care[2,3].

The second method is the lean method and by that I mean lean as in lean manufacturing and the Toyota Production System, the disciplined process of minimizing waste in everything we do. Now my father would never use those words because he invented all these things for himself when he was young before lean terminology became popular. Fortunately a lot of other people have caught on in the ensuing decades, and have developed and documented a variety of lean techniques[4,5,6,7,8]. My father employed the lean method throughout his life, and even dictated an essay on one aspect of this subject hours before he died[9].

Let me walk you through some events in my father's life to show you how he really turned lemons into lemonade using what appears to be two very simple ideas: the scientific method and the lean method.

When my father was 8 years old a bank foreclosed on his family's house. My grandfather had built the house himself, so it must have come as a shock. This foreclosure occurred during the depths of the depression, in 1936. My father and his four siblings and my grandparents packed themselves into a sedan. They moved from town to town in the Southern states, even living in a public work relief camp, a CCC camp, for a while. Despite everything my father was able to develop his skills and understanding, setting the stage for later achievements. During this period he really started to demonstrate a feel for machines. He was in charge of maintaining

the engine of his father's motorboat. He learned how to troubleshoot mechanical problems with whatever tools and parts were available. I think it was at this time that he first learned how to apply the scientific method and the lean method.

Conditions in the country got even worse. When he was 14, my father, along with a lot of other

Americans of that time, saw pillars of smoke rising on the horizon after an enemy attack on the United

States. He saw with his own eyes that the oceans no longer protect us. I'm not talking about September 11, 2001 here, I'm talking about December 11, 1942. My father lived in Louisiana at the time. The columns of smoke he saw were from tankers burning out in the Gulf of Mexico after they had been torpedoed by German submarines.

When he was of age he volunteered for the Navy and here, once again, he was able to make good use of an adverse situation. He was very impressed with the new recruit placement process in the Navy, in how they would take in people from various different backgrounds and every conceivable level of ability and match each person to a job appropriate for that individual. Based on my father's high scores on a placement test, he was chosen to operate some of the most advanced technology of the time, such as radar. The Navy had efficiently determined where he belonged in a vast organization; this example of a lean method impressed my father.

Later he was chosen to relayed messages on an aircraft carrier flight deck. He saw several people get killed on that flight deck just because of landing miscues or traffic jams on the deck and he said – and this is the only time I'm going to use his words – he said “the bloody waste of war impressed me permanently.”

These two early experiences were, I think, the genesis of my fathers' scientific and lean approach.

My father used this approach masterfully a decade after the end of the Second World War. By this time he had obtained a BS degree from Washington State College, where he had been discovered and mentored by a Mechanical Engineering professor named Harry Sorensen. After that he had obtained an masters' degree from Caltech, also in Mechanical Engineering. In 1957 he was in a Ph.D. program at MIT, the appeal of which was summarized in those days with the chant “MIT, Ph.D.,

M.O.N.E.Y.” But as we shall see, the siren song of money did not prevent him from pursuing his own priorities.

My father carefully assessed the relevant factors: jobs were plentiful because of the booming economy and Cambridge Massachusetts was the biggest college town in the United States. He really thought about what had value, what was not a waste of his time, what he really cared about and yes, that led him to drop out of MIT. This freed up time to get involved in the Cambridge dating scene where he met my mother. What a brilliant decision to drop out of MIT. After that things started to really look up.

Dropping out of MIT wasn't his only unorthodox decision at the time. He realized that in order to direct a research program later he'd need to make sure before leaving the ivory tower that he knew Mechanical Engineering like nobody's business. He got a job at

Boston University and taught every course in the Mechanical Engineering curriculum. After all, when you teach something you really learn it. Thus when he

left Massachusetts he was very, very well-versed in Mechanical Engineering.

Fast forward to 1964. His research ability and his friendship with another no-nonsense Navy veteran, Dr. Taylor, Dean of the Ohio University College of Engineering, had served him well. My father had obtained tenure in the Ohio University Mechanical Engineering Department, lack of Ph.D. notwithstanding. By the time he was 36, my father had enough grant money to run a research group exploring renewable energy power plants. What happened next is described in the textbook “Free Piston Stirling Engines”[10], which I will paraphrase as follows:

“Half a dozen people in various parts of the world tried to make a free piston Stirling engine work. But the free piston Stirling engine is a paradox of seeming mechanical simplicity which is in fact difficult and complicated to execute. Most who tried failed completely.

William Beale was a notable exception. With unbelievable persistence Beale continued his efforts and was eventually rewarded with a stable engine. Since those

pioneering days he has contributed as much as anyone to putting the technology on an established foundation."

If I may interject - I have to add something to that last statement from "Free Piston Stirling Engines." I think some of the key contributors are right here in this room today - I see David Berchowitz, David Gedeon, Robi Unger and many more - to all of you, and to Gary Wood and others not here today, this is your accomplishment as well as my fathers'.

Now returning to the passage in the textbook about putting the Stirling technology on an established foundation:

"In all this Carol Beale has supported him beyond all expectation. In recognition of this magnificent and sustained effort this book is dedicated to Carol and William Beale and all the employees of Sunpower of Athens, Ohio."

The development of the free piston Stirling engine was an important achievement. During the 1970's and 1980's, the best work on Stirling engines was being done at Sunpower. Best in the entire world.

We can all take pride in that.

Note that the lean method played a part in this success. My fathers and the others at Sunpower could outlast all those competitors because they had low overhead, low burn rate, and the ability to really focus for as long as it took to get the technology right. All these competitors out there in California and Massachusetts didn't have the needed time and focus. There were other essential elements in

Sunpower's success, of course, notably solid engineering and flashes of brilliance, but the lean method was certainly a contributing factor.

The lean method was again the guiding principle as my father and mother tackled the difficult task of taking the Stirling effort out of Ohio University and founding a new company. Resourcefulness and adaptability were key. By 1974 the United States had withdrawn from the Vietnam war, and those who were at Ohio University only to avoid the draft left school. Because of this, among other factors, enrollment at the university decreased. The Mechanical Engineering department considered closing its (then unused) machine shop and laying off the two mechanics who worked there. My parents arranged to rent the machine shop and hire the mechanics, solving a problem for the university while creating the nucleus around

which Sunpower would grow. My mother taught herself accounting so she could manage the Sunpower books and supervise the non-technical staff.

Now at this point, forgive me but I can't imagine summarizing the life of William Beale without giving those of you who are not in the engineering field some idea of what a free piston Stirling engine is. After all, that is what my father invented[11]. (Danny Briggs, who served as my father's technician after the sale of Sunpower in 2012, holds up a manual water pump in the photo at right). What we have here – is not a free piston Stirling engine but just a regular pitcher pump of the type you'd use with a well or cistern.

Notice that when you push down on the handle, a plunger inside lifts a column of water which then pours out the spout, and also draws more water up from the reservoir below via suction. When you push the handle up, a check valve opens and the plunger goes down into the column of water. As you move the handle up and down, spurts of water come out the spout.

Now for a magic trick. In the spirit of one of my father's thought experiments, I will now turn this ordinary pitcher pump into a free piston Stirling water pump. Imagine this. You have a balloon with some air inside – when you heat the air, the balloon expands, when you cool the air, the balloon contracts. Now this is a magic balloon that is always perfectly spherical and doesn't pop. If I wedge this balloon between the handle and casing of the pitcher pump, and magically attach the balloon to each, then when I heat and cool the balloon it will move the handle up and down, pumping water. If we ignore details such as the regenerator, what we've created is a Stirling water pump. For you Stirling experts in the audience, I contend that it qualifies as a free piston Stirling water pump, as the amplitude of the pump handle can vary. The main point here is this mechanism turns heat energy into mechanical work by alternately heating and cooling a gas confined in an enclosed volume.

Prior to 1964 Stirling engines were not practical. My father and the other Sunpower experts' contribution was to invent and perfect the free piston variant of the Stirling engine, a practical and potentially very cheap Stirling. Prior to 1964, Stirling engines were toys that couldn't compete with gasoline, diesel or gas turbine engines, steam

turbines, fuel cells or electric motors. Now free piston Stirling engines and coolers are commercial products used all over the world. Unlike the other power plants mentioned, free piston Stirling engines are nearly maintenance-free and are well suited for a wide variety of renewable energy sources such as biomass, waste heat and solar energy.

Let me say just one more thing before I wrap up. An important example of both the scientific method and the lean method is the creation of the corporate culture that made Sunpower such a stimulating place to work in the early days.

Long before Google commissioned studies on how best to run a startup, Sunpower got a lot of things right. My father wrote very convincing papers about his invention; his Stirling engine papers attracted recruits from all over the world. We can see that by looking around the room today. For example I see David Berchowitz, Izzi Urieli and Robi Ungar who are from South Africa and Israel.

My father also hired people from all different kinds of backgrounds, even backgrounds that didn't seem to fit. And he gave people a chance because in the lean method – my term not his – in the lean method you don't waste anything. You don't waste time, you don't waste energy and most importantly you don't waste human potential. He could go and look at someone and say “I think this person is capable of more than they themselves think they can do.” He would say “here are your objectives, see if you can do it.” More often than not, the new hire would rise to the occasion. A job candidate whose long rap sheet included running through Athens in his underpants after a wild college party became a world renowned Stirling engine expert. Another candidate who claimed to have a high IQ despite having little formal education became the company's foremost master technician and hardware troubleshooting wizard, and has provided three decades of service to Sunpower so far.

My father took great care to foster the free flow of ideas at Sunpower. When an engineer suggested an idea that was clearly wrong, he told me recently, he would gently introduce a better idea and let the engineer claim it as their own if they wished. He guided group meetings such that they became a respectful, humorous banter about all the technical issues they were facing at any given time. For innovation to flourish, my father believed, the atmosphere had to be respectful. And it was. Anybody could speak their mind in the safe environment of a Sunpower brainstorming session. Google did an extensive study on the best way to make a

team creative and ended up reinventing the Sunpower brainstorming session, 40 years after my father first developed it[12].

Now, what can we learn from this summary of a company founder's life? My father had insights no one else had, and he acted on them. The scientific method yielded those insights, and the lean method gave him the ability to act. Individuals and organizations don't naturally use scientific or lean methods, therefore my father needed to keep things on track with discipline and persistence.

Forgive me, I was instructed to speak today only about my feelings for my father. But in thinking about my father and what he might have wanted, it occurred to me that he didn't just want people to feel things, he wanted people to do things. I will close with my guess about what my father might say to all of you if he were here today. I may be completely wrong – you can be the judge of that.

In conclusion:

For you young people, wouldn't you like to get accurate answers to all your questions – about people, about choices you face? Don't you want to know the unvarnished truth about the issues that matter to you? The scientific method can answer those questions and uncover those truths. The scientific method guided my father to good decisions and it can work of you too. Caveat - for this to work you should employ the lean method also and here's why. If you waste time and money, as so many people do, you'll have no time and money left to pursue the next great idea that comes your way.

For you older people, I implore you to register and vote. We all wish climate change weren't so horribly real, but facts are stubborn things[13]. Engineering solutions are available[14]; the problem is the lack of political will. Our best hope is to do everything we can to elect politicians who will fight climate change.

[1]Two links to the video are:
http://www.communitysolution.org/mediaandeducation/energyandclimateyoutubechannel/home https://vimeo.com/182715288

[2]Atul Gawande, "The Mistrust of Science," New Yorker Magazine, 10 June, 2016.

[3]Julie Schwartz Gottman, "Level 1 Clinical Training," The Gottman Institute, 2016.

[4]Ben Hartman, "The Lean Farm," Chelsea Green Publishing, 2015.

[5]Jeff Sutherland, “Scrum,” Crown Business, 2014.

[6]Jeffrey Liker, “The Toyota Way,” McGraw Hill, 2004.

[7]David Anderson, “Kanban,” Blue Hole Press, 2010.

[8]Kanboard, https://kanboard.net

[9]William Beale, “Level of Requirement,” Community Solutions Blog, 25 July, 2016.

http://www.communitysolution.org/blog/2016/7/25/poppys-dream?rq=level%20of%20requirement [10]Graham Walker & Jim Senft, “Free Piston Stirling Engines,” Springer-Verlag, 1985. h t tp://link.springer.com/book/10.1007%2F978-3-642-82526-2

[11]https://www.ohio.edu/mechanical/stirling/engines/WilliamBeale.html

[12]Charles Duhigg, “What Google Learned From Its Quest to Build the Perfect Team,” New York Times Magazine, Feb. 25, 2016.

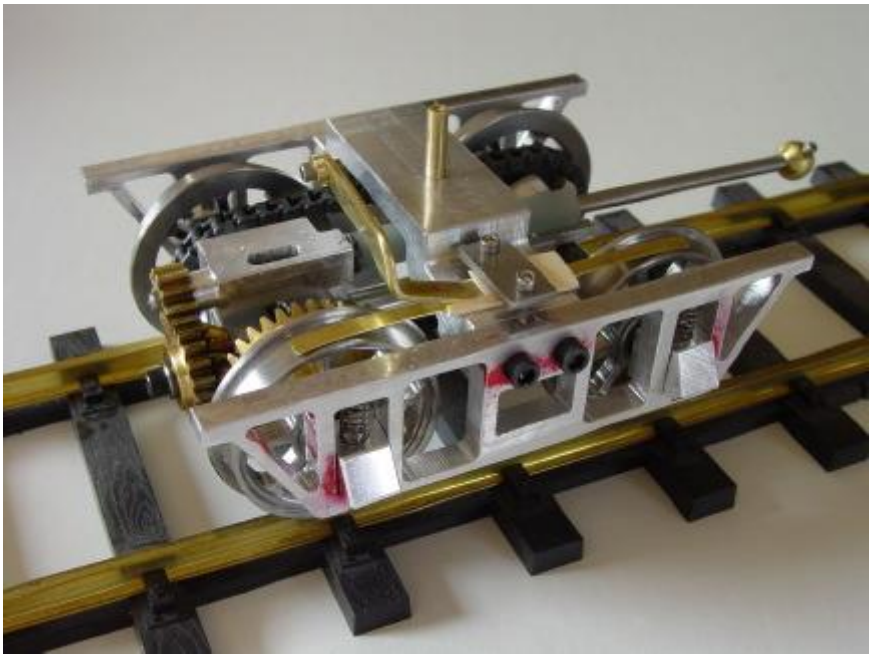
[13]George Marshall, “Don't Even Think About It: Why Our Brains Are Wired to Ignore Climate Change,” Bloomsbury, 2015.

[14]William Beale, “Bottom-Line Thinking Won't Save Our Climate; Solar Thermal Machines,” New York Times, November 30, 1989.

Appendix A.3: A Class A Climax Locomotive by Andy Ross

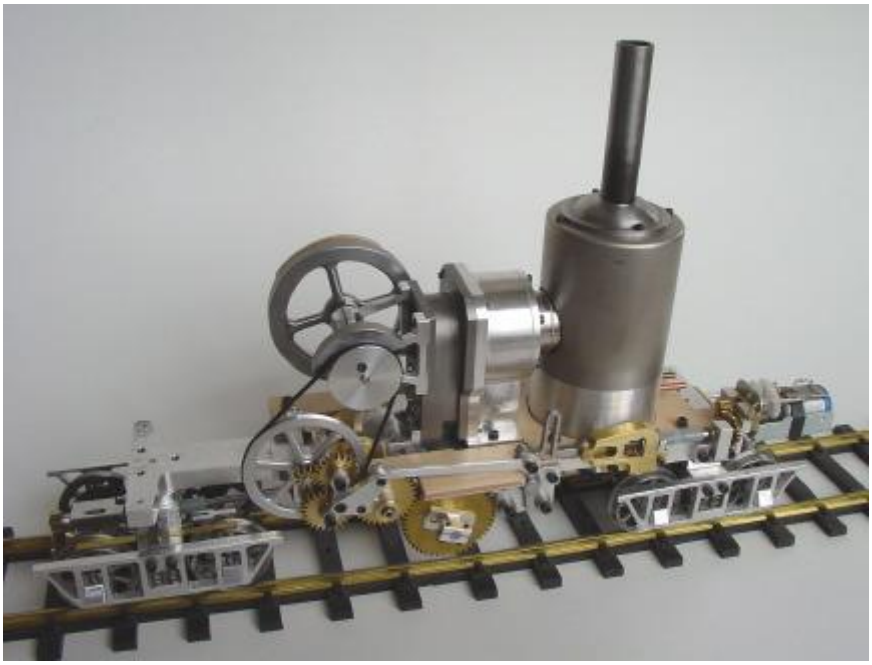
I've been interested in the class A Climax logging locomotives for a long time, but could never find enough information to adequately model one until I discovered The Climax Locomotive six months ago. This magnificent 497 page volume, published by Oso in Hamilton, Montana, is a treasure of rare and wonderful photos, authoritative history, first-hand accounts, technical details, drawings, and much more. It wasn't long before I decided to make a gauge 1 model of a vertical boiler class A.

Although I have made a few live steam engines over the years, I have a special affection for the stirling engine, and decided that my class A would be stirling-powered. Four years ago I completed a gauge 1 model of a four wheel Swiss rail car that was powered by an alcohol-fired stirling engine for long runs, and by compressed air supplied to its steam engines for short demonstration runs (Modeltec magazine, March-April 2002). Now I wanted to revisit the stirling-powered locomotive idea, this time incorporating a gearbox to provide forward, idle, and reverse, with remote control through the track. The aim was to create an unusual, attractive, and practical fire-driven locomotive for a garden railway.

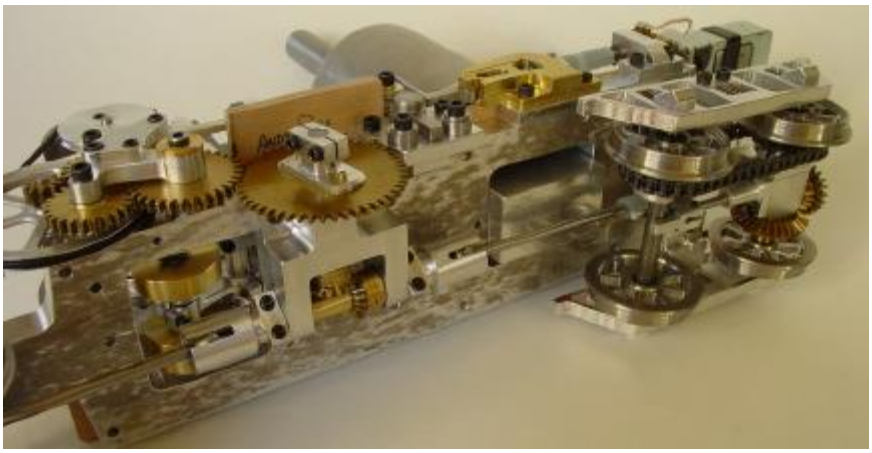


It seemed reasonable to begin with the trucks, since all the other parts rest upon them. I started with some wheels and axles from Hartford Products, then re-machined the wheels' contours and milled out sixty-four little spokes. How to drive those wheels became the next question. Climax used driveshafts located along the locomotive's center-line to transmit power from a marine-style steam engine to the trucks. These driveshafts passed over the axles, driving them with skew gears, which are similar to bevel gears except their shafts are not in the same plane. As I could find no source for such gears, I used stock bevel gears in combination with a pair of helical gears to raise the input shafts above the axles. By locating these gears on the outboard ends of the trucks, the driveshaft connection for each truck could be located immediately below the swivel joint for that truck, minimizing both lateral and fore and aft driveshaft motion. The "universal" joints here are simply lengths of silicone rubber model airplane fuel line fitted over the shafts. The two remaining axles are driven by small plastic sprockets and chain from the gear-driven axles. All axles are sprung independently.

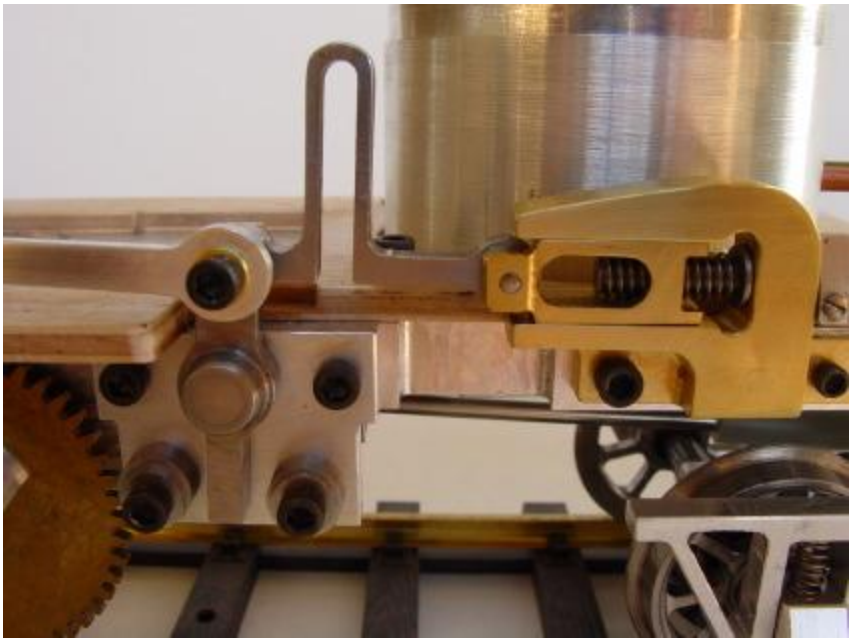
A simpler approach to all-wheel drive would omit powering the trucks' second axles, but rather let those axles float with very light springs, so essentially all the machine's weight would be borne by the two driven axles. The resulting cantilevered forces on the truck bolsters would need to be addressed, of course, but the trucks would be significantly easier to make.



The transmission of power from the stirling engine to the trucks is as follows: A standard rubber O-ring transmits power from a pulley on the engine to a larger pulley on a countershaft. This is a high speed, low torque connection, and there has been no sign of any slippage. The countershaft has a small brass pinion gear, which is the heart of the gear shift mechanism. Below this pinion is a pivoting yoke, containing a train of three brass idler gears. The top two of these gears are located so that one, or the other, or neither, will engage with the countershaft pinion as the yoke is pivoted to and fro on an axle concentric with its bottom (third) gear. This is the same arrangement used on many lathes to provide forward, reverse and neutral to the lead screw. In my model one more gear provides further speed reduction, and drives a set of miter gears which turn the power 90 degrees to align with the driveshafts. It is here that the true universal joints are used. They are of the simplest type, merely a cup with a slot engaging a ball with a pin. These allow for any fore and aft, and angular, movement of the drive-shafts.

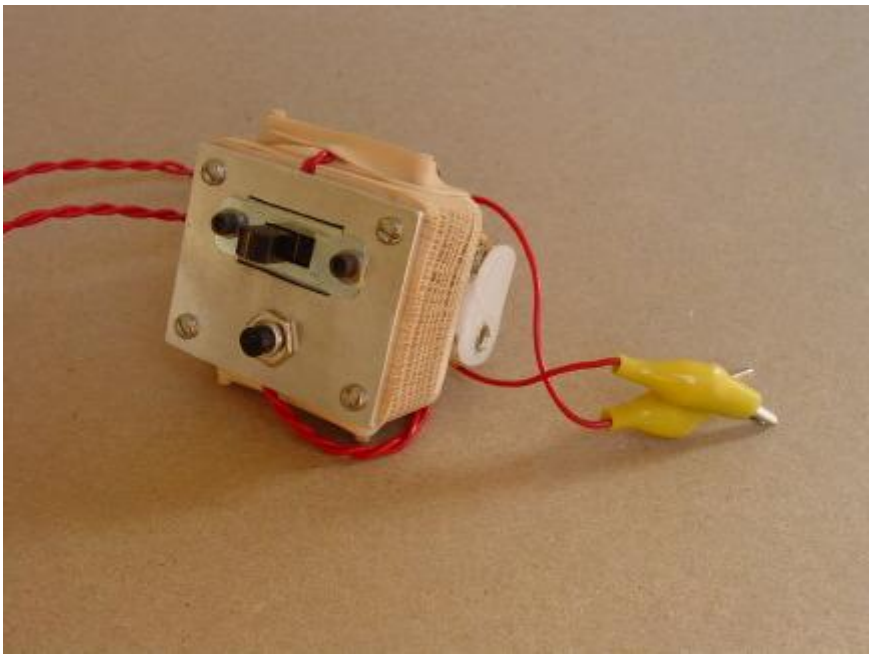


The gears are shifted from forward to neutral to reverse and back by means of a small motor and reduction drive unit (a hobby shop item) connected to a screwjack by yet more silicone rubber fuel line, which provides a necessary high-torque slip clutch for the drive.

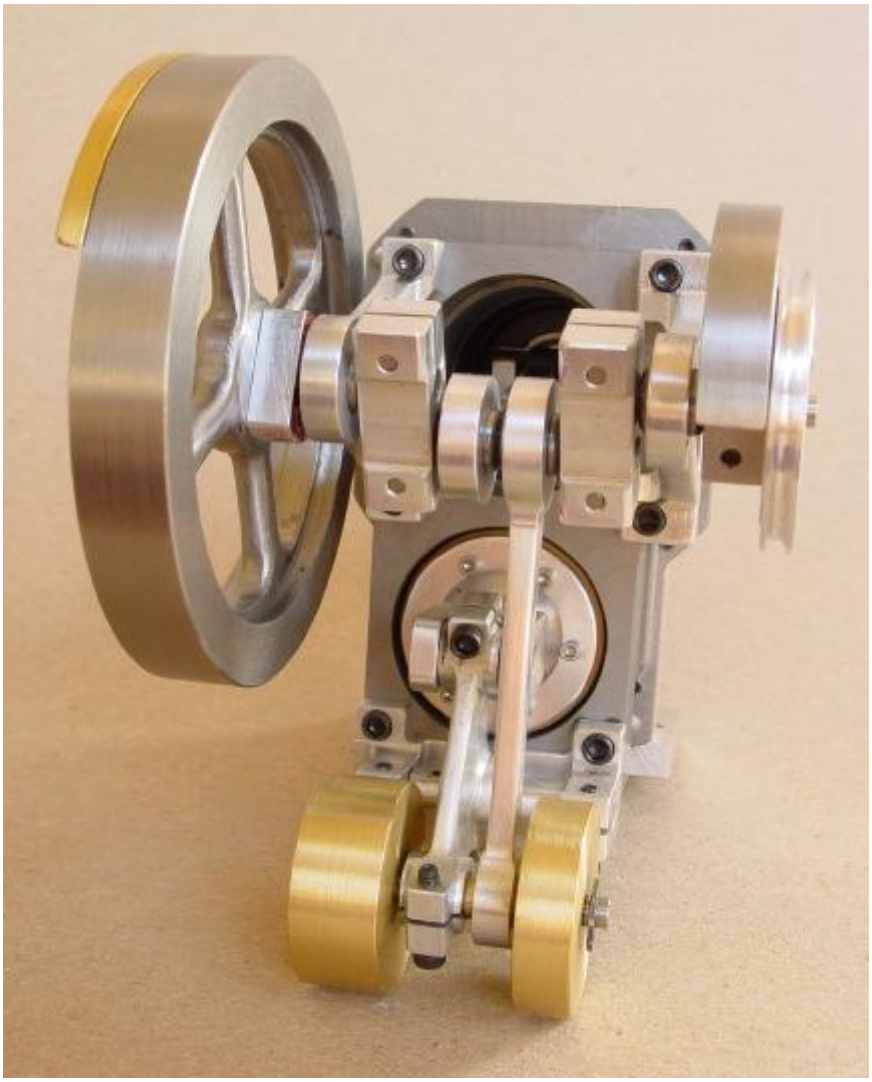


The pivoting yoke needs firm stops to prevent its gears from being driven too far into the countershaft pinion. These are provided by routing the yoke's drive linkage through a steel lever with adjustable steel stops. But the overall rigidity of the drive system would occasionally force the shifting gears together on their outer diameters, rather than in mesh, thereby greatly increasing friction and stalling the engine. Filing the mating gear teeth to a sharper profile, so there would be less likelihood of their outside diameters colliding, minimized but did not solve the problem. Putting a stiff spring (in this case, an oxbow machined into the connecting rod) between the

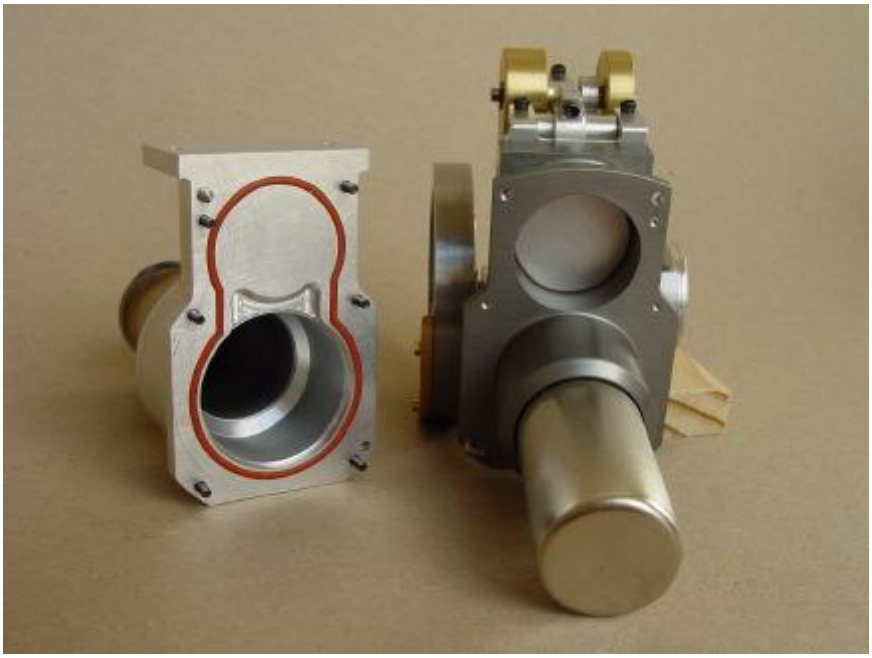
screwjack and the stop lever did solve the problem. The transmission now works smoothly and reliably; but as with the trucks, I do believe a much simpler and more elegant approach could be devised.



The controller for this shifter is a handheld battery box with a reversing switch and a button, and two leads that are attached to the track. The button controls when the current can flow to the servo motor in the locomotive, and the reversing switch controls which way the motor (and therefore the shift linkage) moves. Idle is between the stops of forward and reverse, and it requires a fairly deft finger on the button to stop the shift linkage during the brief time when neither gear is engaged with the pinion. Forward and reverse are easy to engage by merely driving the mechanism to its stops, whereupon the slip clutch lets the motor continue to turn until the operator removes his finger from the button.



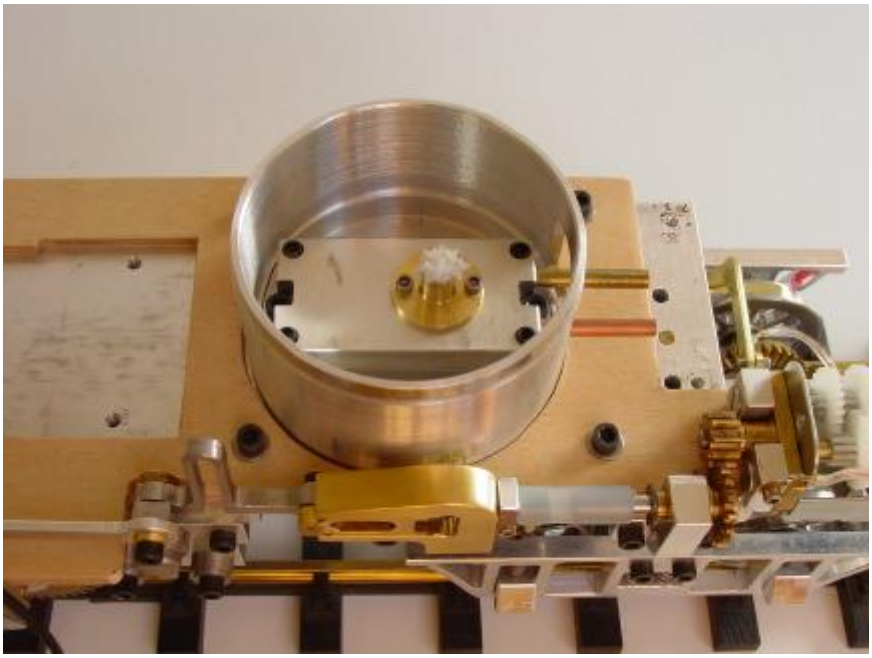
The stirling engine is quite similar to the one I made for the Swiss railcar. It is a two piston (or alpha) type stirling, in which the pistons are linked together so the motion of one leads that of the other by 90 crankshaft degrees. They move in separate parallel cylinders that are connected together by a long annular passage that is divided into three sections of heat exchangers. The section next to the expansion (or "hot") cylinder of the leading piston is the externally-fired heater. The section next to the compression (or "cold") cylinder of the trailing piston is the heat-dissipating cooler. The section of the annular passage between the heater and cooler is heated or cooled only by the gas flowing through it, not by external means. This section serves as a regenerator, which is a heat-conserving device invented by Robert Stirling in 1816. It is interesting to note that the engine bearing Stirling's name was merely one of several ingenious ideas he suggested to illustrate how his regenerator might prove useful.



This arrangement of parts cooperates to accomplish the following four things during each crankshaft revolution:, 1) compress the working gas when most of it is in the cold cylinder and its temperature and pressure are low; 2) transfer most of the working gas through the heat exchangers to the hot cylinder, thereby raising its temperature and pressure; 3) expand the working gas when most of it is in the hot cylinder and its temperature and pressure are high; and 4) transfer most of the working gas through the heat exchangers to cool cylinder, thereby lowering its temperature and pressure. Expanding the high pressure gas produces more power than is consumed by compressing the same gas at a lower pressure, and this excess power is the engine's output. Of course there are many subtleties, but this is the essence of the stirling cycle engine.

The crank drive mechanism is my own design, which I call the Rocker V mechanism because it uses one rocking piston and is balanced like a V-twin. A nice schematic of this simple, strong, and readily-balanced mechanism may be found on Izzy Urieli's web site.

But making a robust model stirling engine depends not so much on choosing the right cylinder arrangement or drive mechanism as it does on; 1) keeping friction very low, 2) keeping internal dead volume low and compression ratio high, 3) making pistons that seal really well, and 4) providing enough surface area for the heat exchangers to transfer heat to and from the working gas rapidly.



Rather than having cooling fins, the engine simply has its cooler section screwed down to the massive aluminum alloy frame, which serves as a heat sink to dissipate waste heat. The burner for the new engine has only 44% of the wick area of the railcar engine, and therefore burns about 44% of the fuel, or about one cubic centimeter of alcohol for each nine minutes of operation. The burner holds ten cubic centimeters when filled. Warm up takes about one minute after light-up.

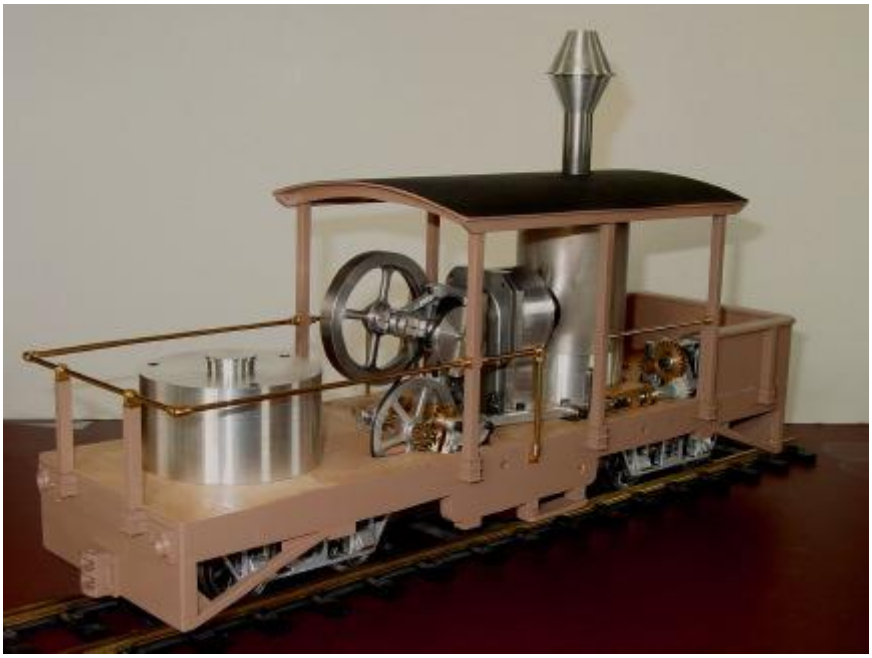
Less wick area of course means less power; this difference is made up by a lower overall gear ratio used in the Climax and its fully sprung all-wheel drive. The railcar has only two of its four wheels driven, and they are unsprung. Overall weights are similar, with the Climax weighing 2.3 Kg (5.07 pounds) and the railcar 2.39 Kg (5.25 pounds). Trials showed the Climax would make 550 grams of pull when tethered to the scale compared to the railcar's 300 grams. Both machines spin their wheels when restrained.

One of the interesting challenges was machining the underside of the little piece that slips over the stack and rests on the curved roof, holding the safety valve and whistle. The radius of this curve is just over 9 inches, and I had no easy way to cut it directly; but the job was readily done with just 8 trigonometrically-calculated step cuts per side with an eighth inch end mill. The tiny steps didn't even need to be filed smooth; the part just fit and clung to the roof perfectly when aligned correctly.

Other interesting parts included the window frames, since their mullions were so long and thin. They were machined (very carefully!) out of aluminum alloy, and their

round corners were then filed square. I also had to consider how to flip the crankshaft over to start the engine, since it was within a wooden cabin. This task became easy after recalling that the flywheel was attached to the crankshaft by being screwed to a clamp mounted behind it. Although these screws were recessed into the flywheel's hub, the hub was easily milled down to expose enough of the screws and crankshaft to permit engaging them with a starting key poked through the open starboard window.

Over the course of building this model there were many modifications and revisions, both major and minor, which I have not described here. The biggest change was in the wooden flatcar and cabin. Originally I modeled an open cabin prototype, but the stirling engine looked far too large to create any illusion of reality. Moreover, the wooden structure could not be removed to get at the mechanical parts without first dismantling the entire transmission.



Since the class A Climaxes were built in so many variations, and modified so routinely by their owners, it was easy to find a suitable enclosed prototype to convert my model into. The new wooden structure is attached to the aluminum alloy frame with just four screws, and may be removed without removing any of the mechanical components.

At last the modifications are complete, and I am now satisfied with the model, both mechanically and aesthetically. Of course that doesn't mean it won't get modified again, sometime in the future. I consider most of my projects to be experimental works-in-progress; and that attitude seems particularly appropriate for a model of a class A Climax.

The End

Appendix B: Senior Design Project - Stirling Powered Dragster

Saturday May 19, 2001 was an exciting day - it was the culmination of the three quarter Senior Design project - ten teams racing their radio-controlled dragsters in pairs down the 100 meter path from Stocker Center to Richland Avenue. These were not regular dragsters - these were externally fired heat engine driven machines in which the entire system including the heat engine was designed and built from scratch by the teams.

There was a significant amount of variation and innovation in the designs - five double-acting steam engines with spool valves, one opposed-piston wobbler steam engine, two impulse steam turbines, two Stirling engines. Transmission mechanisms ranged from direct drive on the wheels to an air propellor drive, and one drive even included a measured fishing line giving a progressively increasing gear ratio The systems had to be completely self contained, including the water and fuel needed for the entire trip. Furthermore the dragsters needed to be remotely operated after a warmup period, including start, steering, braking and fuel shutoff.

Adding to the excitement was the fact that during the Spring quarter the students challenged the professors to participate in the competition. Dr. Urieli accepted the challenge, and within five weeks designed and built a Stirling powered dragster. (shown below) having a radio-controlled pivoting engine which either applies a friction drive to the wheel or a braking force against the wheel. The second channel controls the steering, as well as moving the mascot head in the opposite direction to the steering in order to confuse the competition. The engine is based on a 20 cc Rocker-V Stirling engine designed by Andy Ross. A number of engines were built by students in the context of the winter quarter Stirling engine course. One of the student teams used one of these engines to power their propeller driven machine. In the picture below we see Dr. Urieli in serious conversation with the dragster a few days before the race.

Appendix C: Ford-Philips 4-215 Engine Case Study

Urieli I. & Berchowitz D. M. (1984). *Stirling cycle engine analysis*. A. Hilger. (pp. 25-31)

In the case studies which follow, mechanical drives of varying degrees of complexity are used, from the simple sinusoidal swashplate drive to the more complex rhombic and yoke drives.

2.4 Case study—The Ford–Philips 4-215 engine

The 4-215 engine is a four-cylinder swashplate-drive double-acting engine. Its design and development began in 1972 under a joint program between N V Philips, of Holland, and the Ford Motor Company to develop an experimental automotive engine (van Giesel and Reinink 1977). The operating principle of the engine is shown in figure 2.3. The four cylinders are interconnected, so that the upper expansion space of one cylinder is connected to the lower compression space of the adjacent cylinder via a series-connected heater, regenerator and cooler. The pistons are driven by a swashplate (or alternatively by a 'wobble' plate), resulting in a pure sinusoidal reciprocating motion having a 90° phase difference between the adjacent pistons.

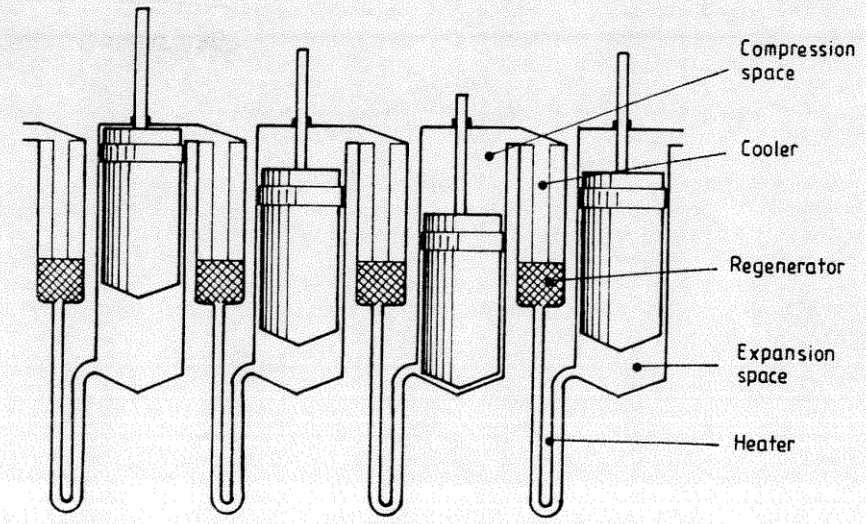


Figure 2.3 Arrangement of the swashplate double-acting engine.

The wobble plate drive was first considered for use in Stirling engines by Sir William Siemens in 1860 (figure 2.4). Babcock (1885), commenting on this engine in 1885 stated:

> No other form of air-engine offers so many advantages, but it has also its peculiar difficulties. If the latter can be overcome, it is likely to become the air-engine of the future.

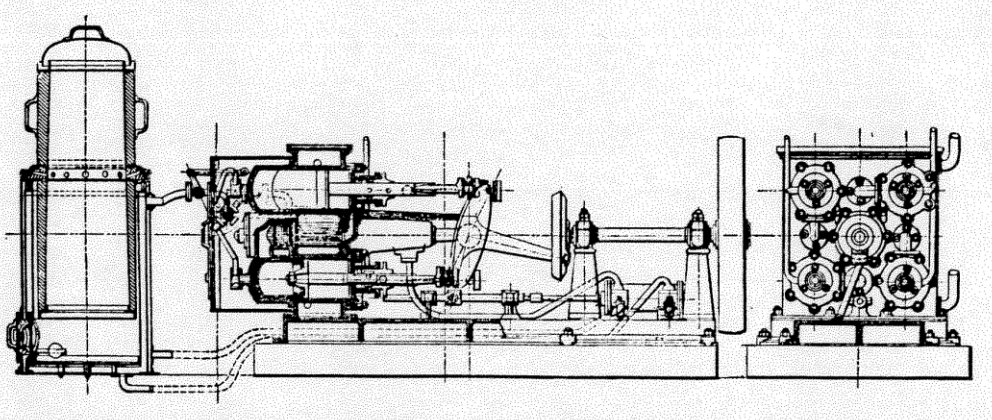


Figure 2.4 Wobble plate engine due to Siemens in 1860 (after Babcock 1885).

The swashplate drive engine was brought out of obsolescence almost a century after its invention. The major advantage of the engine is its high specific power output capability, since there is only one reciprocating component per cycle, and no pressurised crankcase is necessary. Unfortunately, the 'peculiar difficulties' exist even today, being mainly the problem of containing the working gas under the high mean operating pressure.

The Ford–Philips 4-215 engine uses hydrogen as the working gas under a maximum mean pressure of 200 atm, and reached an advanced level of development before the project was cancelled by Ford in 1979.

A general view of the engine is shown in figure 2.5 and a cross section view is shown in figure 2.6. The pertinent data defining the engine configuration and operating parameters are given in table 2.2. However, not all the required geometric data have been published in the open literature as yet, and therefore a number of assumptions have been made. Thus, for example, we have arbitrarily divided the total unaccounted for volume equally between the compression and expansion space clearance volumes.

The performance map is given in table 2.3. We notice that the engine is capable of producing an indicated power output of 200 kW at 4500 rpm.

In simulating the engine we have considered it as having a single cycle, simply multiplying the various volumes per cylinder by four. The sinusoidal volume variations can be stated in a simple form as

$$V_c = V_{clc} + \tfrac{1}{2} V_{swc}(1 + \cos\theta) \tag{2.15}$$

$$V_e = V_{cle} + \tfrac{1}{2} V_{swe}[1 + \cos(\theta + \alpha)] \tag{2.16}$$

where V_{swc} and V_{swe} are the compression and expansion space swept volumes respectively,

V_{clc} and V_{cle} are the compression and expansion space clearance volumes respectively,

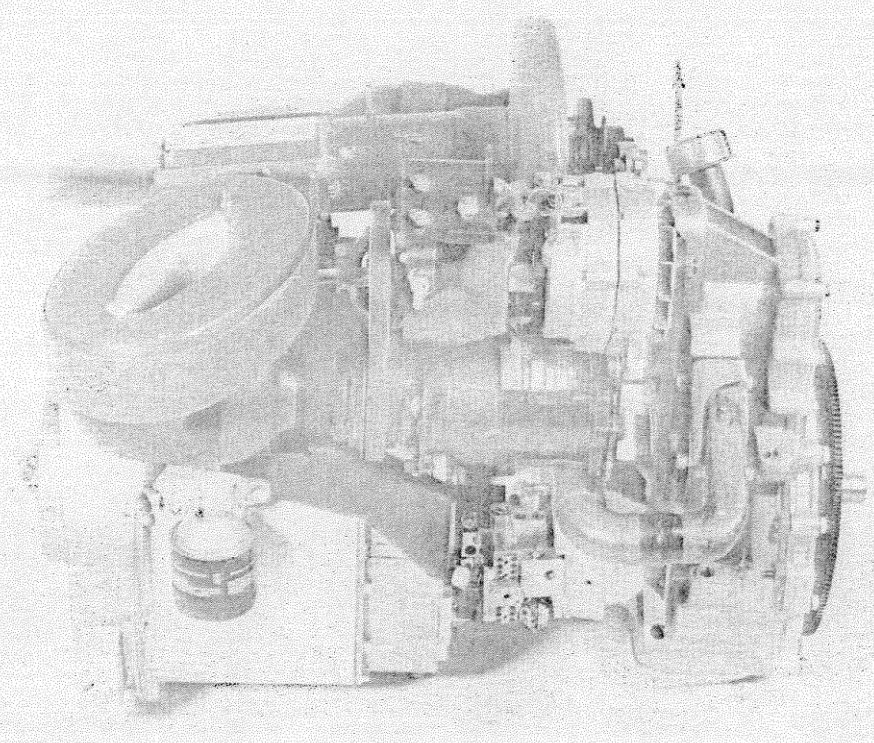


Figure 2.5 General view of the Ford–Philips 4-215 engine (courtesy Ford Motor Company).

θ is the crank angle, arbitrarily chosen to be zero when the compression space volume is a maximum,

α is the phase angle of the expansion space volume variations relative to the compression space volume variations. For the four-cylinder swashplate-drive engine α is 90°.

Differentiating equations (2.15) and (2.16) with respect to the crank angle θ we obtain

$$dV_c/d\theta = -\tfrac{1}{2} V_{swc} \sin\theta \tag{2.17}$$

$$dV_e/d\theta = -\tfrac{1}{2} V_{swe} \sin(\theta + \alpha). \tag{2.18}$$

The data given in table 2.2 have been used to obtain the various simulation parameters shown in table 2.4. Using these parameters on the set of equations given in table 2.1, and integrating numerically over a complete cycle, we obtain the performance characteristics of the Ideal Isothermal model shown in table 2.4. The indicated power output of 212.9 kW is higher than the actual indicated power of 130.1 kW for the same conditions. It is to be expected that the ideal simulation will give an exaggerated performance, mainly because of the use of lossless heat exchanger components in the simulation model. The

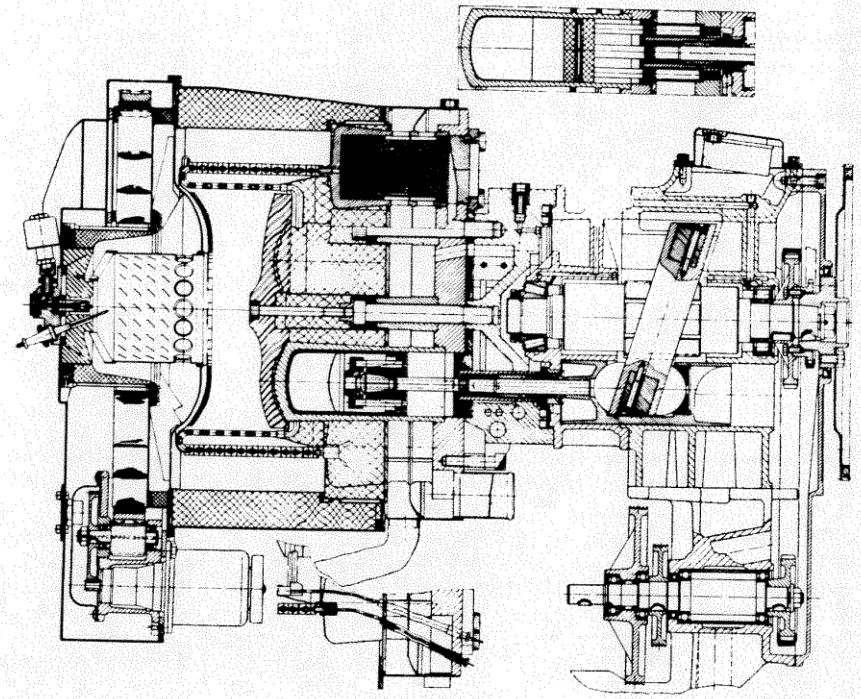


Figure 2.6 Cross section view of the Ford–Philips 4-215 engine (courtesy NASA Lewis Research Center).

thermal efficiency value of 67.1 % is the Carnot efficiency as expected for the Ideal Isothermal model.

The simulation results are shown diagrammatically in figure 2.7. The compression and expansion space volume variations and the resultant pressure variation are shown as functions of the shaft angle θ. The four processes of the ideal Stirling cycle tend to overlap under the sinusoidal volume variations; however, they can be inferred from the diagram. The compression process occurs in the first quadrant (0 to 90°). Some of the compression is seen to occur in the hot expansion space, but most of the working gas is compressed in the compression space. The displacement process occurring in the second quadrant (90 to 180°) is seen to be a constant volume process, to a close approximation. The expansion process occurring over the third quadrant (180 to 270°) is also non-ideal, in that some of the expansion is seen to occur in the compression space, and finally, the displacement process in the fourth quadrant (270 to 360°) is seen to be incomplete in that at the end of this process a significant amount of working gas still remains in the expansion space. Thus the sinusoidal volume variations do not result in the optimum ideal Stirling cycle, but are one of the many practical mechanical realisations of it. The pressure curve is not quite sinusoidal, being peaked towards the high pressure.

Table 2.2 Ford 4-215 engine data (after Belaire and Kitzner 1977).

General	
Working fluid	Hydrogen
Maximum mean pressure	20 MPa
Heater head temperature	1023 K
Cooling water temperature	337 K
Speed	4000 rpm
Bore	73 mm
Stroke	52 mm
Total internal engine volume, including cylinder swept volume, per cylinder	670 cm^3
Heater	
Number of heater tubes/cylinder	22
Heater tube internal diameter	4 mm
Heater tube length	462 mm
Cooler	
Number of cooler tubes/cylinder	742
Cooler tube internal diameter	0.9 mm
Cooler tube length	87 mm
Regenerator	
Regenerator diameter	73 mm
Regenerator length	34 mm
Regenerator matrix wire diameter	36 μm
Matrix mesh size	200
Matrix porosity	0.62
Number of regenerator units per cylinder	2

Table 2.3 Ford 4-215 engine performance map (after Kitzner 1981) giving the indicated power output (kW).

Speed (rpm)	Mean pressure (bar)					
	200	175	150	125	100	50
4500	200.2	179.3	157.0	133.4	108.4	54.2
3900	186.0	166.4	145.6	123.7	100.5	50.6
3300	166.3	148.7	130.1	110.5	89.8	45.4
2700	141.9	126.9	110.9	94.3	76.7	39.0
2100	113.4	101.5	88.8	75.5	61.5	31.5
1900	103.2	92.37	80.9	68.7	56.0	28.8
1700	92.6	82.95	72.6	61.8	50.4	25.9

Table 2.4 Ford–Philips 4-215 engine Ideal Isothermal simulation.

Engine and Operating parameters		
Clearance volumes		$V_{clc} = V_{cle} = 214.2$ cc
Swept volumes		$V_{swc} = V_{swe} = 870.6$ cc
Void volumes:	Cooler	$V_k = 164.3$ cc
	Regenerator	$V_r = 705.8$ cc
	Heater	$V_h = 510.9$ cc
Mean pressure		$p_{mean} = 150$ bar
Mass of gas in engine		$M = 16.2$ g
Hot space temperature		$T_h = 1023$ K
Cold space temperature		$T_k = 337$ K
Ideal Isothermal performance		
Work done		$W = 3870.3$ J/cycle
Heat transferred to gas in hot space		$Q_e = 5771.6$ J/cycle
Heat transferred to gas in cold space		$Q_c = -1901.3$ J/cycle
Thermal efficiency		$\eta = W/Q_e = 67.1\%$
Indicated power output		Power = 212.9 kW at 3300 rpm

Note. The engine is considered as having a single cycle.
All volumes per cylinder are multiplied by four.

The pressure/volume indicator diagram shown in figure 2.7 is the standard means of characterising an engine. The engine size is indicated by the total volume variation and its mass by the maximum pressure, the work done being equal to the area enclosed by the curve.

2.5 Case study—The General Motors GPU-3 engine

2.5.1 The rhombic drive mechanism

The rhombic drive mechanism was invented by R J Meijer of Philips in 1959 (Meijer 1959), and is shown schematically in figure 2.8. The system consists of separate displacer and power pistons which have conveniently separated functions. The displacer separates the hot and cold spaces at approximately the same pressure and is used to shuttle the working gas between these spaces. The power piston provides both the compression and the expansion processes. There are two cranks which are geared together so as to counter-rotate and drive the yokes, to which are attached the displacer and power piston rods. A pressurised buffer space allows the engine to operate at high mean pressures without requiring a pressurised crankcase. Thus only two shaft seals are required on the displacer and power piston rods.

The success of this arrangement is due to the fact that the two pistons have no side forces, as the horizontal components of the forces exerted by each pair

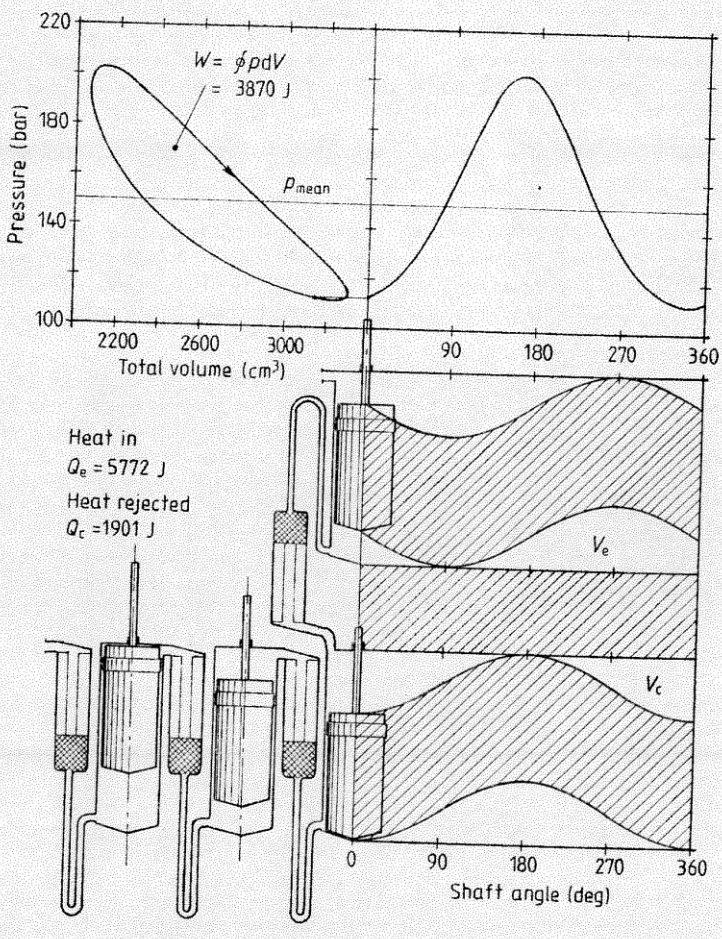


Figure 2.7 Ford–Philips 4-215 engine—Ideal Isothermal simulation.

of connecting rods are exactly balanced at each yoke. Furthermore, complete static and dynamic balancing is attainable in the single cylinder arrangement, allowing vibrationless high-speed operation. The principle disadvantage of the rhombic drive engine is its relative complexity.

The rhombic drive mechanism does not produce sinusoidal volume variations. The actual volume variations are functions of the various geometric parameters of the mechanism and are derived from geometrical considerations, as in figure 2.9. The complete set of equations is summarised in figure 2.10, in which the working space volumes and volume derivatives are shown as functions of the crank angle θ. The initial condition ($\theta = 0$) has been arbitrarily chosen as the point of maximum compression space volume.

2.5.2 The GPU-3 engine

The GPU-3 (Ground Power Unit) is a rhombic drive Stirling engine generator set which was developed by the General Motors Research

Appendix D: General Motors GPU-3 Engine Case Study

Urieli I. & Berchowitz D. M. (1984). *Stirling cycle engine analysis*. A. Hilger. (pp. 30-40)

Table 2.4 Ford–Philips 4-215 engine Ideal Isothermal simulation.

Engine and Operating parameters		
Clearance volumes		$V_{clc} = V_{cle} = 214.2$ cc
Swept volumes		$V_{swc} = V_{swe} = 870.6$ cc
Void volumes:	Cooler	$V_k = 164.3$ cc
	Regenerator	$V_r = 705.8$ cc
	Heater	$V_h = 510.9$ cc
Mean pressure		$p_{mean} = 150$ bar
Mass of gas in engine		$M = 16.2$ g
Hot space temperature		$T_h = 1023$ K
Cold space temperature		$T_k = 337$ K
Ideal Isothermal performance		
Work done		$W = 3870.3$ J/cycle
Heat transferred to gas in hot space		$Q_e = 5771.6$ J/cycle
Heat transferred to gas in cold space		$Q_c = -1901.3$ J/cycle
Thermal efficiency		$\eta = W/Q_e = 67.1\%$
Indicated power output		Power = 212.9 kW at 3300 rpm

Note. The engine is considered as having a single cycle.
All volumes per cylinder are multiplied by four.

The pressure/volume indicator diagram shown in figure 2.7 is the standard means of characterising an engine. The engine size is indicated by the total volume variation and its mass by the maximum pressure, the work done being equal to the area enclosed by the curve.

2.5 Case study—The General Motors GPU-3 engine

2.5.1 The rhombic drive mechanism

The rhombic drive mechanism was invented by R J Meijer of Philips in 1959 (Meijer 1959), and is shown schematically in figure 2.8. The system consists of separate displacer and power pistons which have conveniently separated functions. The displacer separates the hot and cold spaces at approximately the same pressure and is used to shuttle the working gas between these spaces. The power piston provides both the compression and the expansion processes. There are two cranks which are geared together so as to counter-rotate and drive the yokes, to which are attached the displacer and power piston rods. A pressurised buffer space allows the engine to operate at high mean pressures without requiring a pressurised crankcase. Thus only two shaft seals are required on the displacer and power piston rods.

The success of this arrangement is due to the fact that the two pistons have no side forces, as the horizontal components of the forces exerted by each pair

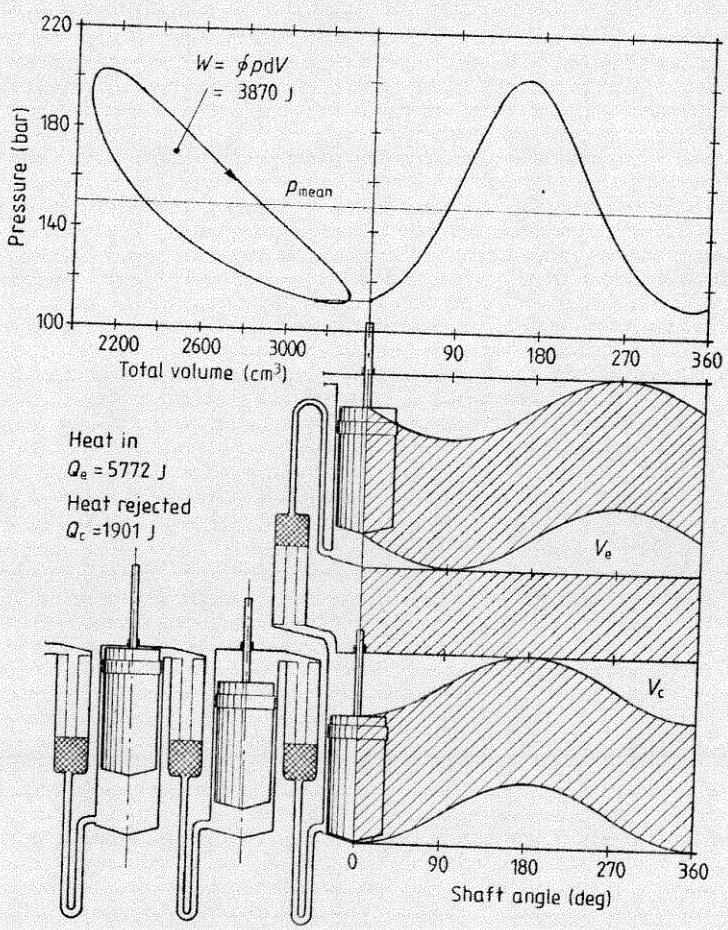


Figure 2.7 Ford–Philips 4-215 engine—Ideal Isothermal simulation.

of connecting rods are exactly balanced at each yoke. Furthermore, complete static and dynamic balancing is attainable in the single cylinder arrangement, allowing vibrationless high-speed operation. The principle disadvantage of the rhombic drive engine is its relative complexity.

The rhombic drive mechanism does not produce sinusoidal volume variations. The actual volume variations are functions of the various geometric parameters of the mechanism and are derived from geometrical considerations, as in figure 2.9. The complete set of equations is summarised in figure 2.10, in which the working space volumes and volume derivatives are shown as functions of the crank angle θ. The initial condition ($\theta = 0$) has been arbitrarily chosen as the point of maximum compression space volume.

2.5.2 The GPU-3 engine

The GPU-3 (Ground Power Unit) is a rhombic drive Stirling engine generator set which was developed by the General Motors Research

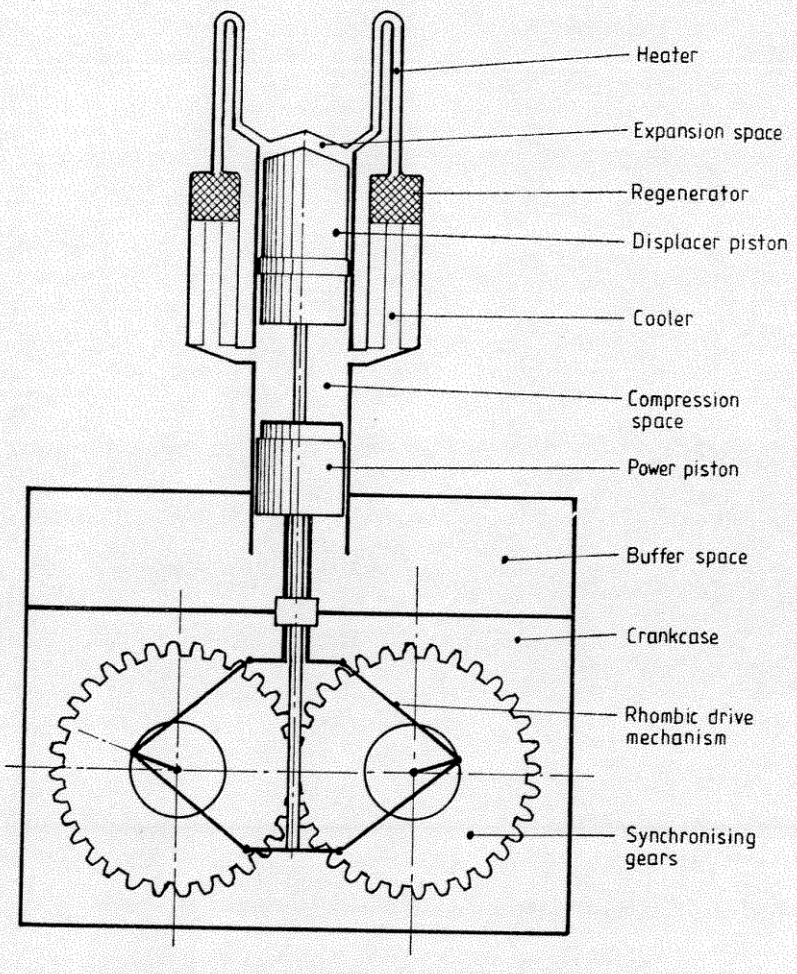


Figure 2.8 Rhombic drive engine—schematic.

Laboratories for the US Army in 1965. The engine is capable of producing a maximum brake output of 7.5 kW using hydrogen as the working fluid pressurised to 69 bar.

Only a few experimental units were actually built by GM. NASA Lewis Research Center has recently restored one unit to operating condition, which is being tested as part of a Stirling engine technology study for the US Department of Energy (Cairelli *et al* 1978). A general view of the GPU-3 is shown in figure 2.11 and a cross section view is shown in figure 2.12.

The rhombic drive engine of the GPU-3 has been well specified in various NASA Technical Memos, such as that by Tew *et al* (1979), from which table 2.5 has been reproduced. This table gives a complete geometric specification of the GPU-3 engine.

Various baseline tests were run at Lewis Research Center to map the engine performance over a range of heater-tube gas temperatures, mean compression-

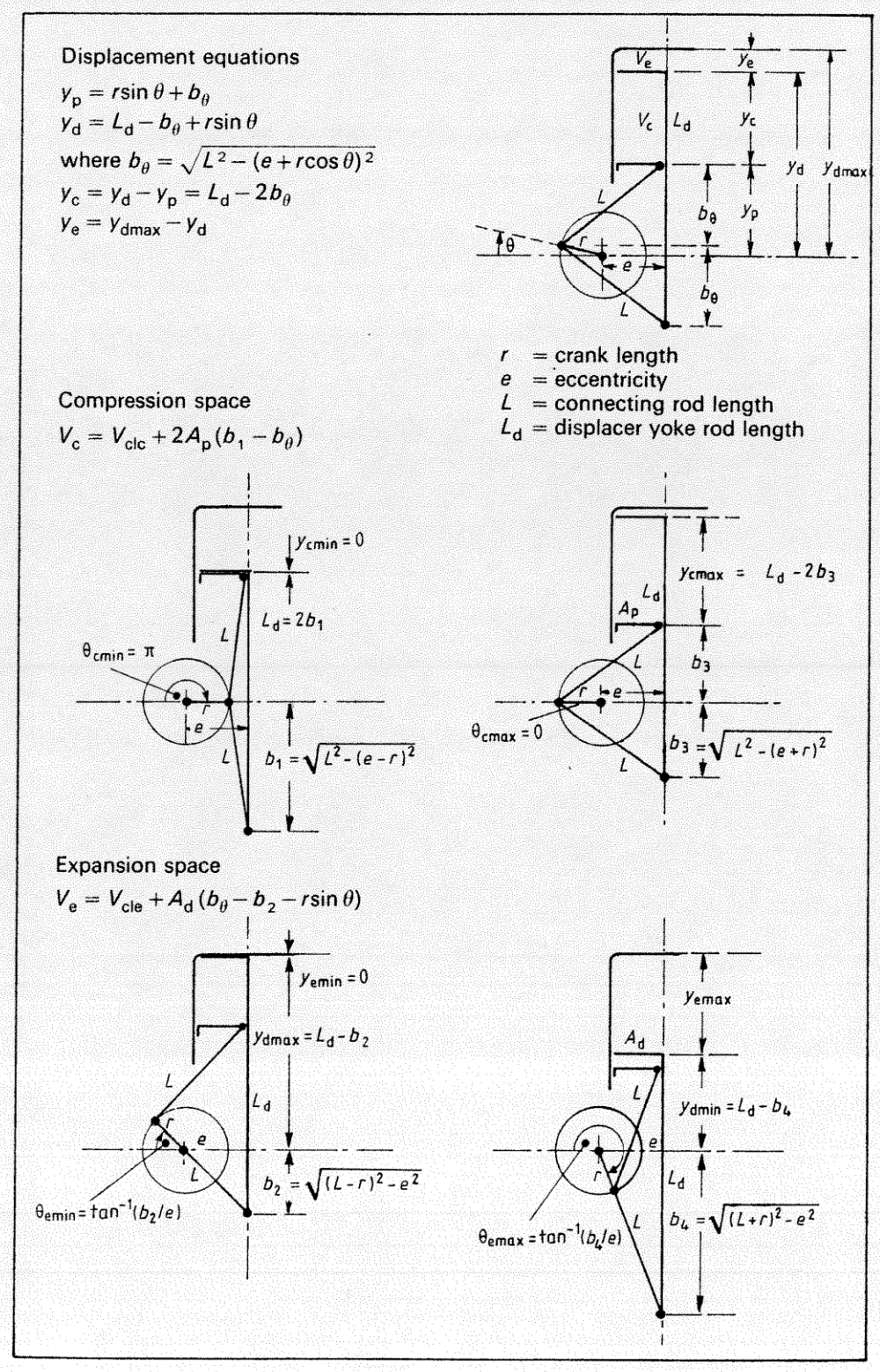


Figure 2.9 Geometric derivation of the rhombic drive equations.

Basic constant parameters

$$b_1 = \sqrt{L^2 - (e-r)^2}$$
$$b_2 = \sqrt{(L-r)^2 - e^2}$$
$$b_3 = \sqrt{L^2 - (e+r)^2}$$
$$b_4 = \sqrt{(L+r)^2 - e^2}$$

Swept volumes

$$V_{swc} = 2A_p(b_1 - b_3)$$
$$V_{swe} = A_d(b_4 - b_2)$$

Volume variations

$$b_\theta = \sqrt{L^2 - (e + r\cos\theta)^2}$$
$$V_c = V_{clc} + 2A_p(b_1 - b_\theta)$$
$$V_e = V_{cle} + A_d(b_\theta - b_2 - r\sin\theta)$$
$$dV_c/d\theta = -2A_p r\sin\theta(e + r\cos\theta)/b_\theta$$
$$dV_e/d\theta = -(dV_c/d\theta)A_d/(2A_p) - A_d r\cos\theta$$

Phase angle advance

$$\alpha \equiv [(\theta_{cmax} - \theta_{emax}) + (\theta_{cmin} - \theta_{emin})]/2$$
$$= \pi - [\tan^{-1}(b_4/e) + \tan^{-1}(b_2/e)]/2$$

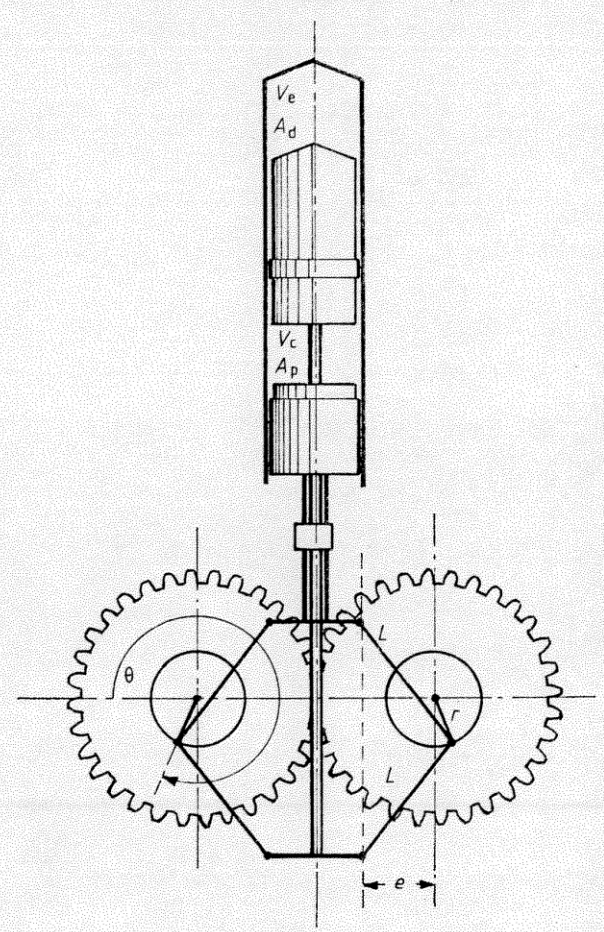


Figure 2.10 Rhombic drive engine equation summary.

space pressures, and engine speeds using both helium and hydrogen as the working fluid. These tests were limited to low power levels since the original alternator used was not capable of absorbing the full engine power output. It was decided to simulate the engine under the same conditions as the sample

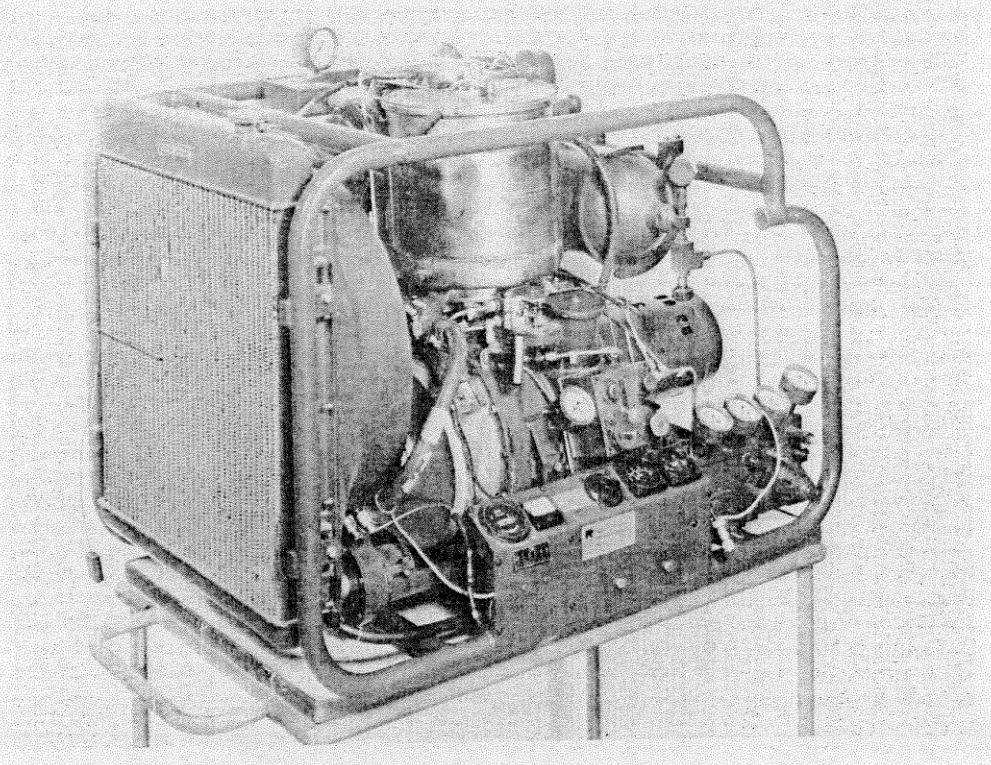


Figure 2.11 The General Motors GPU-3 ground power unit (courtesy NASA Lewis Research Center).

data test point given in Appendix F of the Lewis Research Center report by Thieme (1979). The pertinent operating conditions inferred from the report are summarised as follows:

Working gas:	helium at a mean pressure of 41.3 bar
Operating frequency:	41.72 Hz
Temperatures:	cooler, 288 K; heater, 977 K.

The measured indicated power output was 3958 W at a thermal efficiency of 35%.

The results of simulation using the Ideal Isothermal model are given in table 2.6, and shown diagrammatically in figure 2.13. From the displacement diagram in figure 2.13 we see that the rhombic drive mechanism closely approximates the Ideal Stirling cycle. In particular, the overlapping piston motion allows a large dwell angle during which the compression space volume is almost zero. Thus expansion takes place mainly in the expansion space giving a high specific power output.

The volume variation is highly non-sinusoidal and cannot be described simply. It is difficult even to define the phase angle advance α of the expansion space to compression space volume variations. Measured with respect to the

Table 2.5 GPU-3 engine geometric specification (after Tew *et al* 1979).

GPU 3-2 Engine dimensions and parameters	
Cylinder bore at liner, cm (in)	6.99 (2.751)
Cylinder bore above liner†, cm (in)	7.01 (2.76)
Cooler	
Tube length, cm (in)	4.61 (1.813)
Heat transfer length, cm (in)	3.55 (1.399)
Tube inside diameter, cm (in)	0.108 (0.0425)
Tube outside diameter, cm (in)	0.159 (0.0625)
Number of tubes per cylinder (or number of tubes per regenerator)	312 (39)
Heater	
Mean tube length, cm (in)	24.53 (9.658)
Heat transfer length, cm (in)	15.54 (6.12)
Cylinder tube, cm (in)	11.64 (4.583)
Regenerator tube, cm (in)	12.89 (5.075)
Tube inside diameter, cm (in)	0.302 (0.119)
Tube outside diameter, cm (in)	0.483 (0.19)
Number of tubes per cylinder (or number of tubes per regenerator)	40 (5)
Cold end connecting ducts	
Length, cm (in)	1.59 (0.625)
Duct inside diameter, cm (in)	0.597 (0.235)
Number of ducts per cylinder	8
Cooler end cap, cm^3 (in^3)	0.279 (0.0170)
Regenerators	
Length (inside), cm (in)	2.26 (0.89)
Diameter (inside), cm (in)	2.26 (0.89)
Number per cylinder	8
Material	Stainless steel wire cloth
Number of wires, per cm (per in)	79 × 79 (200 × 200)
Wire diameter, cm (in)	0.004 (0.0016)
Number of layers	308
Filler factor, percent	30.3
Angle of rotation between adjacent screens, deg	5
Drive	
Connecting rod length, cm (in)	4.60 (1.810)
Crank radius, cm (in)	1.38 (0.543)
Eccentricity, cm (in)	2.08 (0.820)
Miscellaneous	
Displacer rod diameter, cm (in)	0.952 (0.375)
Piston rod diameter, cm (in)	2.22(0.875)
Displacer diameter, cm (in)	6.96 (2.740)
Displacer wall thickness, cm (in)	0.159 (0.0625)
Displacer stroke, cm (in)	3.12 (1.23)
Expansion space clearance, cm (in)	0.163 (0.064)
Compression space clearance, cm (in)	0.030 (0.012)
Buffer space maximum volume, cm^3 (in^3)	521 (31.78)
Total working space minimum volume, cm (in)	233.5 (14.25)

†Top of displacer seal is at top of liner at displacer TDC.

Table 2.5 (continued)

GPU-3 Stirling engine dead volumes

(Volumes are given in cm^3 (in^3))

I	**Expansion space clearance volume**	
	Displacer clearance (around displacer)	3.34 (0.204)
	Clearance volume above displacer	7.41 (0.452)
	Volume from end of heater tubes into cylinder	1.74 (0.106)
	Total	12.5 (0.762)
II	**Heater dead volume**	
	Insulated portion of heater tubes next to expansion space	9.68 (0.591)
	Heated portion of heater tubes	47.46 (2.896)
	Insulated portion of heater tubes next to regenerator	13.29 (0.811)
	Additional volume in four heater tubes used for instrumentation	2.74 (0.167)
	Volume in header	7.67 (0.468)
	Total	80.8 (4.933)
III	**Regenerator dead volume**	
	Entrance volume into regenerators	7.36 (0.449)
	Volume within matrix and retaining disks	53.4 (3.258)
	Volume between regenerators and coolers	2.59 (0.158)
	Volume in snap ring grooves at end of coolers	2.18 (0.133)
	Total	65.5 (3.998)
IV	**Cooler dead volume**	
	Volume in cooler tubes	13.13 (0.801)
V	**Compression in space clearance volume**	
	Exit volume from cooler	3.92 (0.239)
	Volume in cooler end caps	2.77 (0.169)
	Volume in cold end connecting ducts	3.56 (0.217)
	Power piston clearance (around power piston)	7.29 (0.445)
	Clearance volume between displacer and power piston	1.14 (0.070)
	Volume at connections to cooler end caps	2.33 (0.142)
	Volume in piston 'notches'	0.06 (0.004)
	Volume around rod in bottom of displacer	0.11 (0.007)
	Total	21.18 (1.293)
	Total dead volume	193.15 (11.787)
	Minimum live volume	39.18 (2.391)
	Calculated minimum total working space volume	232.3 (14.178)
	Measured value of minimum total working space volume (by volume displacement)	233.5 (14.25)
	Change in working space volume due to minor engine modification	2.5 (0.15)
		236.0 (14.40)

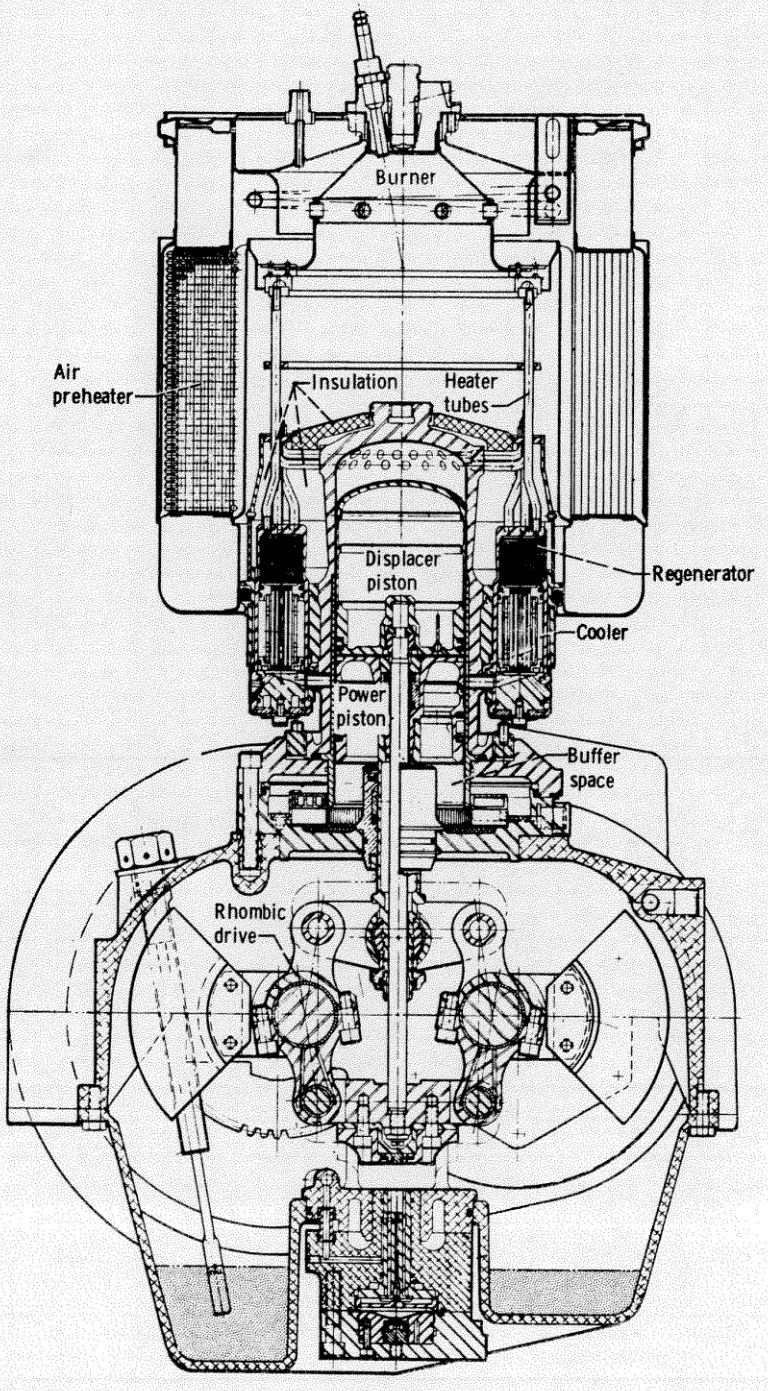


Figure 2.12 Cross section view of the GPU-3 rhombic drive engine (courtesy NASA Lewis Research Center).

Table 2.6 GPU-3 rhombic drive engine Ideal Isothermal simulation.

Engine and operating parameters		
Clearance volumes		$V_{clc} = 28.68$ cc
		$V_{cle} = 30.52$ cc
Swept volumes		$V_{swc} = 114.13$ cc
		$V_{swe} = 120.82$ cc
Void volumes:	Cooler	$V_k = 13.18$ cc
	Regenerator	$V_r = 50.55$ cc
	Heater	$V_h = 70.28$ cc
Mean pressure		$p_{mean} = 41.3$ bar
Mass of gas in engine		$M = 1.1362$ g
Hot space temperature		$T_h = 977$ K
Cold space temperature		$T_k = 288$ K
Ideal Isothermal performance		
Work done		$W = 177.9$ J/cycle
Heat transferred to gas in hot space		$Q_e = 252.3$ J/cycle
Heat transferred to gas in cold space		$Q_c = -74.4$ J/cycle
Thermal efficiency		$\eta = W/Q_e = 70.5\%$
Indicated power output		Power = 7442 W at 41.72 Hz

minimum volume positions α_{min} is 130°, whereas with respect to the maximum volume positions α_{max} is 110°. In figure 2.10 we have arbitrarily defined a 'pseudo phase angle advance' α as being the algebraic mean between α_{min} and α_{max}.

The total mass of gas in the system has been calculated from the mean pressure p_{mean} on the assumption that the volume variations can be simply described in terms of fundamental sinusoids. This is the most convenient way to relate the mass to the mean pressure, and the resulting pressure values in figure 2.13 show it to be reasonable.

The Ideal Isothermal power output and efficiency of 7442 W and 70.5% are seen to be about twice as high as the actual values measured of 3958 W and 35% respectively. In the following chapters we shall attempt to evaluate the various reasons for these discrepancies quantitatively.

2.6 Case study—The Ross yoke drive engine

A Rider type of Stirling engine driven by a unique yoke linkage was recently devised by Andy Ross of Colombus, Ohio. A 25 cc model of the engine using air as the working gas was built and tested by Ross and is described by Martini (1981). Using the component sizing suggested by Ross, a dual engine was built and evaluated at Ormat Turbines, Israel, and is shown in figures 2.14 and 2.15.

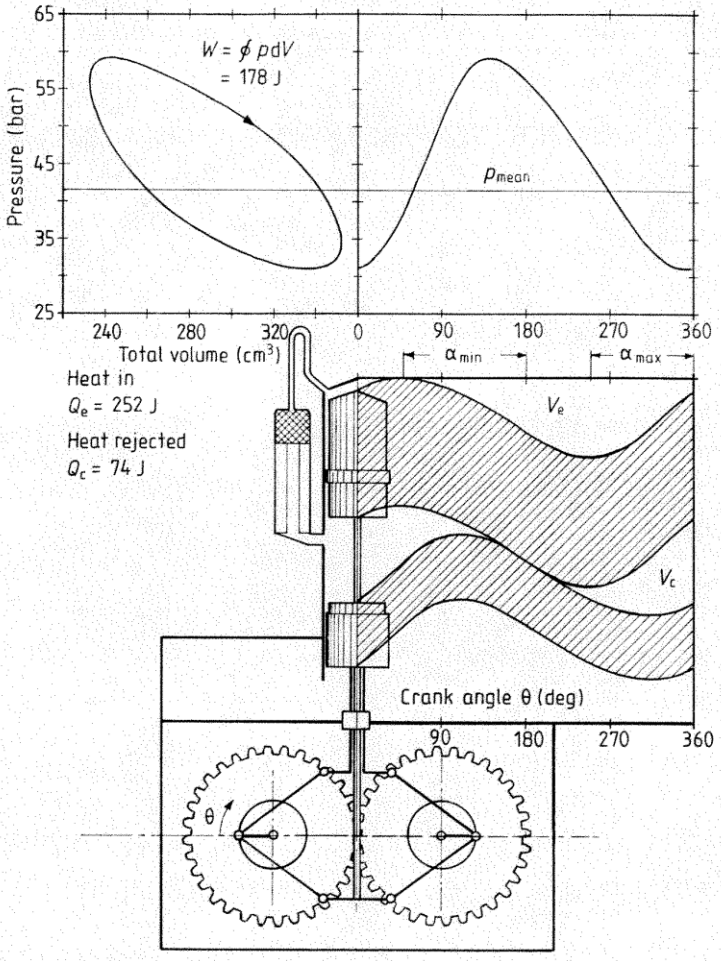


Figure 2.13 GPU-3 rhombic drive engine—Ideal Isothermal simulation.

The engine uses a conventional crank mechanism driving two pistons by means of a yoke linkage. The major feature of this is that there is almost no lateral movement of the connecting rods resulting in very small side forces on the pistons. The basic engine is very similar to the earlier Rider engine described in Chapter 1 in that it includes two working pistons on a single crank mechanism in an Alpha configuration.

With the lack of lateral movement of the connecting rods, there are relatively large unbalanced lateral forces due to the crankshaft counterweight. Ross has a patented gear mechanism which balances the lateral forces by splitting and counter-rotating the counterweights. The Ormat engine overcomes the balancing problem by directly coupling two engine units with a 180° phase difference, as shown in figure 2.15.

The engine is a demonstration model, and as such has an electrical resistance

Appendix E: Free-Piston Machines

Urieli I. & Berchowitz D. M. (1984). *Stirling cycle engine analysis*. A. Hilger. (pp. 51-85)

3

Free-Piston Machines

Hey . . . this engine will work just fine if we simply cut off the rhombic drive . . .

Observation by William T Beale while teaching thermal machinery at the Ohio University, winter 1964

3.1 Background

One of the most novel applications of the Stirling cycle is in free-piston configurations and, indeed, this configuration is the one which holds the most immediate promise.

Free-piston engines operate without physical linkages. They rely only on the gas pressures (and in some cases mechanical springs) to impart the correct motions to the reciprocating elements. Such machines have the advantages of simplicity, low cost, ultra-reliability, and freedom of working gas leakage over conventional Stirling engines. Depending on the particular configuration, these engines may also be designed to operate at constant frequency. They are currently being developed for many diverse applications which include thermally activated heat pumps, solar electric converters, remote area power generators, total energy systems and water pumps (Walker 1980).

The invention of the basic free-piston Stirling engine in the early 1960s is generally attributed to William T Beale (Beale 1969, 1971). Independent inventions of similar types of engines were made by E H Cooke-Yarborough and C West at the Harwell Laboratories of the UK AERE† (Cooke-Yarborough 1967, 1970, Cooke-Yarborough *et al* 1974). G M Benson has also made important early contributions to the field and has patented many novel free-piston Stirling engines (Benson 1973, 1977). Others have since been working on various aspects and modifications of these original ideas (see Martini 1975, Reader 1979, Goldwater and Morrow 1977, Goldberg 1979, Goldberg and Rallis 1979).

An early model free-piston Stirling engine (model M100)‡ produced commercially by Sunpower Inc is shown in figure 3.1. This machine produces a

† Atomic Energy Research Establishment, Harwell, England.

‡ This model is no longer available from the company, having been replaced by a more recently developed one.

nominal 100 W at 50 Hz. From the figure, components typically encountered in a Stirling engine are evident, with the exception of gears, crankshafts and the like, the piston and displacer being mounted on gas springs. This is a common configuration but is certainly not the only one. Many different configurations exist, some of which are analysed later in this chapter.

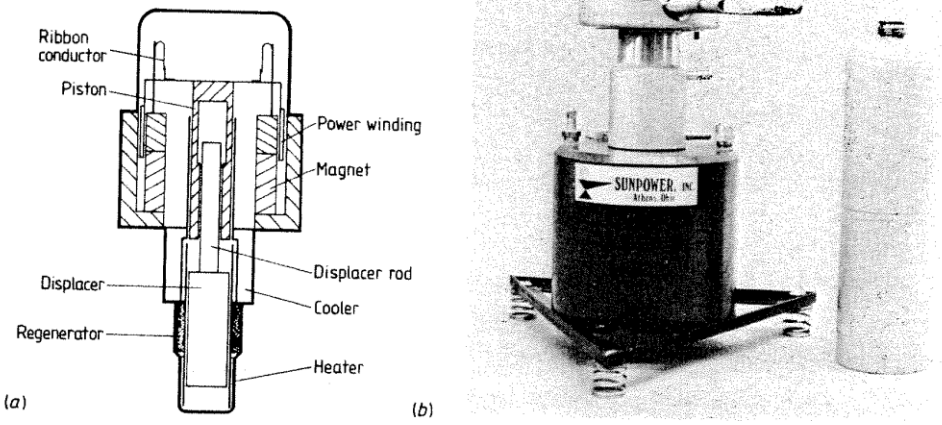


Figure 3.1 Model M100 free-piston Stirling engine (courtesy Sunpower Incorporated).

Consider a free-body diagram (figure 3.2) of the piston and displacer for the model M100. The equations of motion for these two components would be, for the piston:

$$M_P \ddot{x}_p = (p_c - p_b) A_P - C_{pc}(\dot{x}_p + \dot{x}_c) \tag{3.1}$$

where $C_{pc}(\dot{x}_p + \dot{x}_c)$ is the force exerted by the alternator, and for the displacer:

$$M_D \ddot{x}_d = A_D p_e - A_P p_c - A_R p_d \tag{3.2}$$

where $A_P = A_D - A_R$, i.e.

$$M_D \ddot{x}_d = A_D(p_e - p_c) + A_R(p_c - p_d). \tag{3.3}$$

Using p_c as a reference pressure, p_e may be obtained as follows

$$p_e = p_c + \Delta p \tag{3.4}$$

where Δp is the instantaneous pressure difference across the heat exchangers.

Substituting equation (3.4) into equation (3.3) we obtain

$$M_D \ddot{x}_d = A_D \Delta p + A_R(p_c - p_d). \tag{3.5}$$

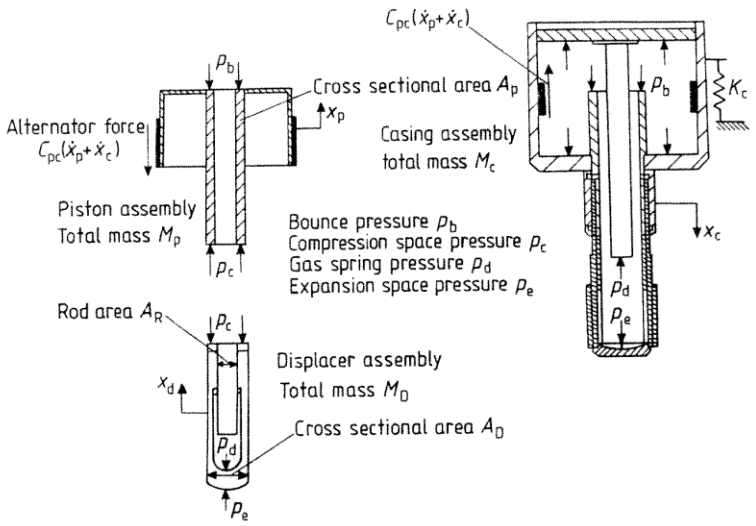


Figure 3.2 Free-body diagram for the Model M100 engine.

Equations (3.1) and (3.5) are then the equations necessary to determine the motions of the piston and displacer respectively. To solve these equations, relationships for p_c and Δp in terms of x_p and x_d are still required. These relationships are obtained from the thermodynamics of the cycle. For the moment we shall assume that the thermodynamic equations are available.

The solution of this set of equations may be attempted by a variety of methods. Possibly the most obvious is by time-stepping integration techniques similar to the simulation method introduced in Chapter 2. This approach has a serious drawback if used for initial sizing, in that the choice of piston and displacer masses, gas springs and other components necessary for the correct dynamic behaviour are not made evident. These parameters would have to be chosen on a more or less arbitrary basis and hence there would be no clear indication, until after the simulation, that a particular engine configuration might work. This trial and error method would clearly prove tedious and expensive, particularly if large scale thermodynamic simulations were used.

The importance of the ability to obtain a preliminary idea of engine configuration in a short space of time cannot be underestimated. It allows the designer to determine at an early stage in the design whether or not the size, performance and operating parameters (such as charge pressure, frequency, etc) are reasonable, and to some extent whether or not a Stirling engine is suitable for the application at hand.

In what follows, a classical dynamic analysis of free-piston Stirling engines is described (Berchowitz and Wyatt-Mair 1979). This analysis is then coupled with the ideal isothermal thermodynamics of Chapter 2 to obtain closed-form solutions for the behaviour, performance and size, as well as the rather narrow criteria under which successful operation may be expected. Furthermore, the

solutions also indicate general characteristics such as the effects of changing load, charge pressure, temperature ratio, etc.

3.2 Generalised analysis

For a given geometry, gas type and temperature distribution, the working gas pressure is ideally a function of the volume variations only (cf Chapter 2, table 2.1). Since the volume variations are, in turn, functions of the piston, displacer and casing motions, the working gas, gas spring and bounce space pressures may be written:

$$p = f(x_p, x_d, x_c). \tag{3.6}$$

As shown in Chapter 2, ideal thermodynamics exclude the possibility of pressure gradients. These pressure gradients, however, strongly influence the dynamic behaviour of the machine in that the reciprocating elements are subjected to increased dissipative (damping) forces. The major effects here are the change in phase between the displacer and piston motions, and the change of displacer to piston amplitude ratio. Together, these two effects significantly alter the power output. Thus, for meaningful results the pressure gradient must be accounted for in some way.

If one considers that the pressure gradient is predominantly a result of viscous friction in the heat exchangers, it is fairly obvious that work against this pressure gradient is dissipative. Therefore, the force due to the pressure gradient influences the dynamics as if it were a damping load. Equivalent linear dampers have been found to work very well in reproducing the effects of the pressure gradient. For the purposes of this analysis, it will therefore be assumed that the pressure gradient may be represented as a function of piston, displacer and casing velocities:

$$\Delta p = f(\dot{x}_p, \dot{x}_d, \dot{x}_c). \tag{3.7}$$

The details of the Δp relationship will be developed later. For the moment, assume that it is of the following form:

$$A_D \Delta p = C_p \dot{x}_p + C_d \dot{x}_d + C_c \dot{x}_c. \tag{3.8}$$

Another important damping effect is the hysteresis loss associated with gas springs. The development of this loss mechanism is covered in Chapter 7. Dynamically, the effects of gas spring hysteresis may also be accounted for by linear dampers. For the piston gas spring

$$f_p = C_{H_{pc}}(\dot{x}_p + \dot{x}_c) \tag{3.9}$$

and for the displacer gas spring

$$f_d = C_{H_{dc}}(\dot{x}_d + \dot{x}_c) \tag{3.10}$$

where $(\dot{x}_p + \dot{x}_c)$ and $(\dot{x}_d + \dot{x}_c)$ are the relative velocities with respect to the casing for the piston and displacer respectively.

The dynamic equations (3.1) and (3.5) should be modified to include gas spring damping as follows

$$M_P \ddot{x}_p = (p_c - p_b) A_P - (C_{pc} + C_{H_{pc}}) \dot{x}_p - (C_{H_{pc}} + C_{pc}) \dot{x}_c \tag{3.11}$$

$$M_D \ddot{x}_d = A_D \Delta p + A_R (p_c - p_d) - C_{H_{dc}} (\dot{x}_d + \dot{x}_c) \tag{3.12}$$

where the gas spring pressures p_b and p_d now represent only the pure spring component of the gas springs, i.e. the ideal pressure swing, which is a function of displacement only.

Substituting for $A_D \Delta p$ from equation (3.8) into equation (3.12):

$$M_D \ddot{x}_d = C_p \dot{x}_p + (C_d - C_{H_{dc}}) \dot{x}_d + (C_c - C_{H_{dc}}) \dot{x}_c + A_R (p_c - p_d). \tag{3.13}$$

If it is assumed that p_c, p_b and p_d oscillate around a mean value of pressure given by p_{mean} (the charge pressure), a linear approximation of any of these pressures would have the form

$$p = p_{mean} (1 + a x_p + b x_d + c x_c). \tag{3.14}$$

Therefore, we may write

$$(p_c - p_b) A_P / M_P = S_{pp} x_p + S_{pd} x_d + S_{pc} x_c \tag{3.15}$$

and

$$(p_c - p_d) A_R / M_D = S_{dp} x_p + S_{dd} x_d + S_{dc} x_c \tag{3.16}$$

where the linear coefficients (i.e. the S) will be developed later.

Substituting equations (3.15) and (3.16) into equations (3.11) and (3.13) respectively, we obtain

$$\ddot{x}_p = S_{pp} x_p + S_{pd} x_d + S_{pc} x_c - (C_{pc} + C_{H_{pc}}) \dot{x}_p / M_P - (C_{pc} + C_{H_{pc}}) \dot{x}_c / M_P \tag{3.17}$$

and

$$\ddot{x}_d = S_{dp} x_p + S_{dd} x_d + S_{dc} x_c + C_p \dot{x}_p / M_D + (C_d - C_{H_{dc}}) \dot{x}_d / M_D + (C_c - C_{H_{dc}}) \dot{x}_c / M_D. \tag{3.18}$$

Grouping the damping terms together, the final forms of the piston and displacer dynamic equations are generalised as follows:

$$\ddot{x}_p = S_{pp} x_p + S_{pd} x_d + S_{pc} x_c + D_{pp} \dot{x}_p + D_{pd} \dot{x}_d + D_{pc} \dot{x}_c \tag{3.19}$$

and

$$\ddot{x}_d = S_{dp} x_p + S_{dd} x_d + S_{dc} x_c + D_{dp} \dot{x}_p + D_{dd} \dot{x}_p + D_{dc} \dot{x}_c \tag{3.20}$$

where the D are the composite damping coefficients. Note that in the particular case of the model M100, D_{pd} is zero.

Since there are three unknown quantities (x_p, x_d, x_c), a third equation is

required before a solution can be attempted. This equation is the dynamic equation for the casing of the engine. By inspection it will be of the same form as equations (3.19) and (3.20):

$$\ddot{x}_c = S_{cp}x_p + S_{cd}x_d + S_{cc}x_c + D_{cp}\dot{x}_p + D_{cd}\dot{x}_d + D_{cc}\dot{x}_c. \tag{3.21}$$

Thus, in the general case, the dynamic equations (also referred to as the equations of motion) may be conveniently represented as

$$\begin{bmatrix} \ddot{x}_p \\ \ddot{x}_d \\ \ddot{x}_c \end{bmatrix} = \begin{bmatrix} S_{pp} S_{pd} S_{pc} \\ S_{dp} S_{dd} S_{dc} \\ S_{cp} S_{cd} S_{cc} \end{bmatrix} \begin{bmatrix} x_p \\ x_d \\ x_c \end{bmatrix} + \begin{bmatrix} D_{pp} D_{pd} D_{pc} \\ D_{dp} D_{dd} D_{dc} \\ D_{cp} D_{cd} D_{cc} \end{bmatrix} \begin{bmatrix} \dot{x}_p \\ \dot{x}_d \\ \dot{x}_c \end{bmatrix} \tag{3.22}$$

or in shorthand:

$$[\ddot{x}] = [\mathscr{S}][x] + [\mathscr{D}][\dot{x}] \tag{3.23}$$

where $[\mathscr{S}]$ and $[\mathscr{D}]$ are the influence coefficients (per unit element mass) of the springs and dashpots, respectively. For example, coefficient S_{pd} accounts for the influence on the piston motions by virtue of a spring coupling to the motions of the displacer, whereas S_{pp} is a spring effect unique to the piston.

In deriving equation (3.22), it is assumed that the pressure terms in the dynamic equations can be linearised without too great a penalty in accuracy. Typically, the non-linearity associated with the working gas pressure is small and can be safely neglected. However, the non-linearity associated with gas springs and pressure gradient terms can be important. A careful check should always be made on the second and higher-order terms. For acceptable results these should never exceed 10% of the linear terms.

The description of the system's characteristic behaviour may be deduced by using standard control theory methods (Gupta and Hasdorff 1970). This behaviour is the way that the three elements (piston, displacer and casing) move for different values of the damping and spring coefficients, namely divergence, convergence or steady oscillation. Since we are only interested in steady oscillation, it is possible to adopt a more straightforward approach by simply assuming that an oscillatory solution exists. By this method, the frequency and conditions for oscillation are obtained directly. However, it should be noted that this approach presupposes that the non-linear elements or effects will be such that a stabilising influence is generated by their presence. A perfectly linear machine would be impossible to operate in reality, since it would be so finely balanced at its operating point that any variation in any parameter, including the load, would either cause the machine to stop or cause its oscillations to grow until collisions between its reciprocating parts occurred. The effects of the non-linearities need, therefore, to be addressed in any load-matching study.

The following solution is assumed for each element

$$x_i = X_i \exp[j(\omega t + \phi_i] \qquad i = p, d, c \tag{3.24}$$

which may be represented in vector form as shown in figure 3.3, where X_i is the amplitude and ϕ_i is the phase.

From equation (3.24), $\dot{x}_i$ and $\ddot{x}_i$ are obtained by successive differentiation:

$$\dot{x}_i = j\omega X_i \exp[j(\omega t + \phi_i)] \tag{3.25}$$

$$\ddot{x}_i = -\omega^2 X_i \exp[j(\omega t + \phi_i)]. \tag{3.26}$$

Before substituting the above solutions into the relevant equations in (3.22), the system may be simplified further by noting that there are two distinct preferred modes of operation: large piston motions relative to casing motions, when power is mainly removed from the piston, or large casing motions relative to piston motions, when power is mainly removed from the casing. If each element is considered as a separate second-order force damped system, we can represent the behaviour of that element as in figure 3.4 (Gupta and Hasdorff 1970).

Before considering figure 3.4 further, it is necessary to introduce the concept of natural frequency. For a hypothetical single degree of freedom system without damping, as shown in figure 3.5, the equation of motion is

$$M\ddot{x} = -kx \tag{3.27}$$

where M is the mass and k is the spring constant.

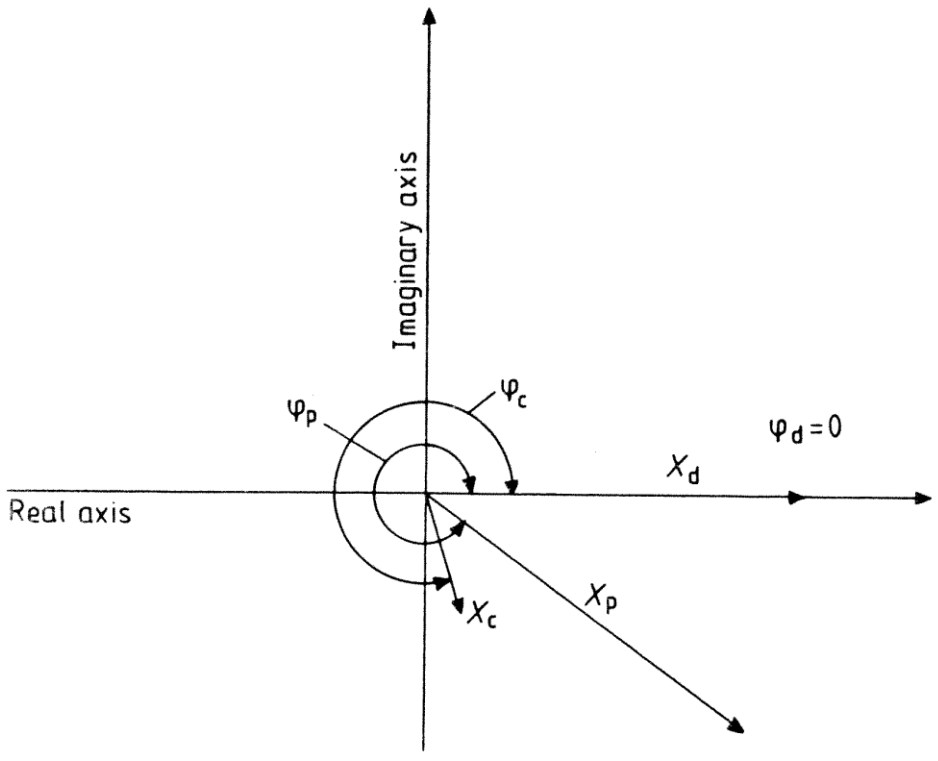


Figure 3.3 Vector notation.

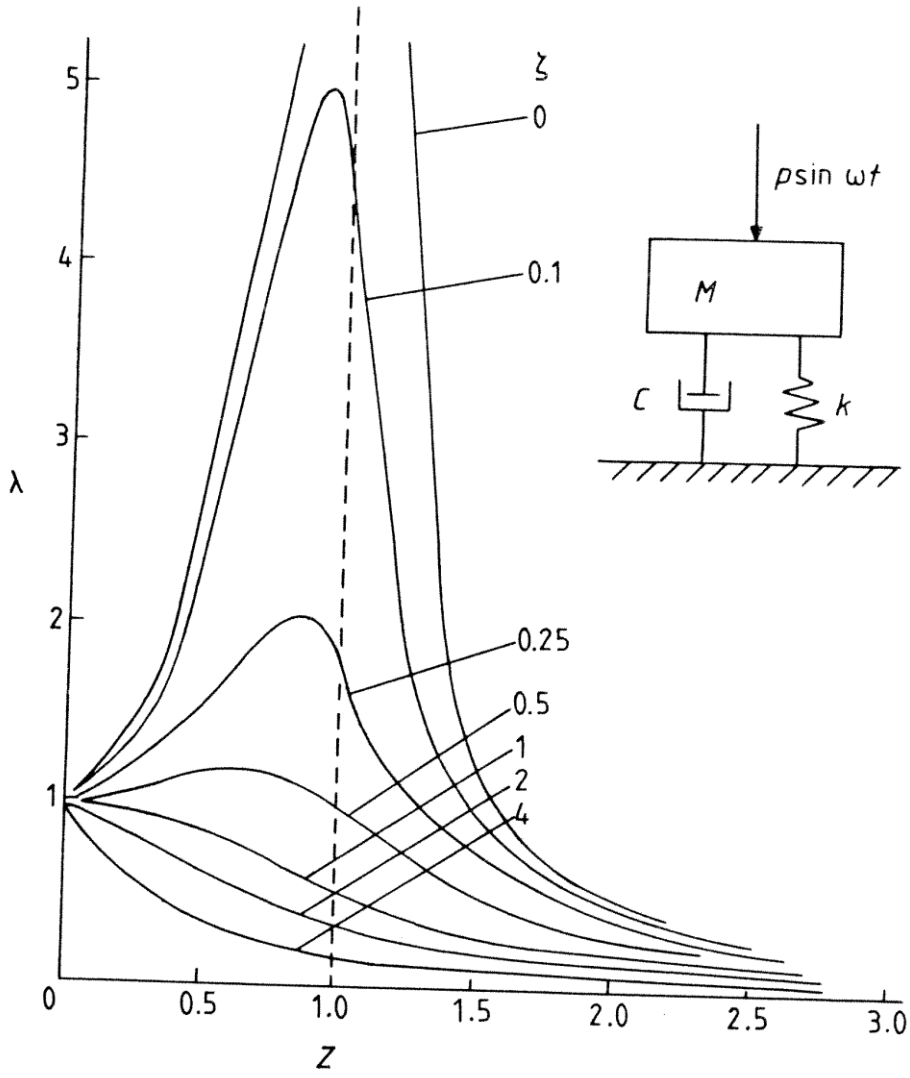


Figure 3.4 Generalised single degree of freedom system:

$$Z = \frac{\omega}{\omega_n} = \frac{\text{applied frequency}}{\text{natural frequency}},$$

$$\lambda = \frac{\text{actual amplitude}}{\text{amplitude due to a static load}},$$

ζ = viscous damping factor $C/(2M\omega_n)$.

Assuming a solution of the form given by equation (3.24), together with equation (3.26), substitution into equation (3.27) yields

$$M\omega^2 = k \tag{3.28}$$

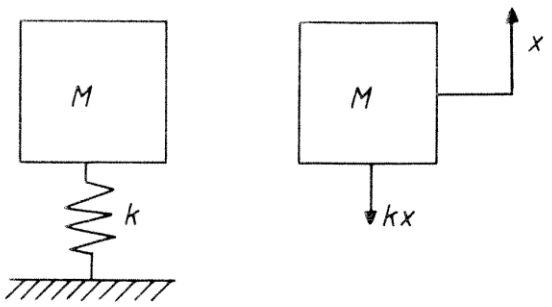


Figure 3.5 Simple spring mass system.

from which the natural frequency follows:

$$\omega_n = \sqrt{k/M}. \tag{3.29}$$

In terms of the notation used in equation set (3.22), this may be written:

$$\omega_n = \sqrt{-S}. \tag{3.30}$$

Thus it is evident that the natural frequencies of the piston, displacer and casing are given by $\sqrt{-S_{pp}}$, $\sqrt{-S_{dd}}$ and $\sqrt{-S_{cc}}$, respectively.

Returning to figure 3.4, it can be seen that operation at a frequency sufficiently higher than the natural frequency of a particular element will cause that element to oscillate at a considerably reduced amplitude.

The conditions for each mode of operation may thus be generalised as follows:

(*a*) predominant piston motion:

natural frequency of casing $\sqrt{-S_{cc}} \ll$ operating frequency ω

(*b*) predominant casing motion:

natural frequency of piston $\sqrt{-S_{pp}} \ll$ operating frequency ω.

It should be sufficient that the unwanted amplitudes be kept to at least one order of magnitude smaller than any of the active amplitudes.

The equation set in (3.22) may now be considerably simplified by ignoring the terms due to the element with small amplitudes but making sure that the conditions for the required operating mode are satisfied. Therefore, for the predominantly piston motion mode the describing equations are

$$\begin{aligned} \ddot{x}_p &= S_{pp}x_p + S_{pd}x_d + D_{pp}\dot{x}_p + D_{pd}\dot{x}_d \\ \ddot{x}_d &= S_{dp}x_p + S_{dd}x_d + D_{dp}\dot{x}_p + D_{dd}\dot{x}_d, \end{aligned} \tag{3.31}$$

subject to $\sqrt{-S_{cc}} \ll \omega$.

Similarly, for the predominantly casing motion mode, the describing equations are

$$\begin{aligned} \ddot{x}_d &= S_{dd}x_d + S_{dc}x_c + D_{dd}\dot{x}_d + D_{dc}\dot{x}_c \\ \ddot{x}_c &= S_{cd}x_c + S_{cc}x_c + D_{cd}\dot{x}_d + D_{cc}\dot{x}_c, \end{aligned} \tag{3.32}$$

this time subject to $\sqrt{-S_{pp}} \ll \omega$.

By substituting equations (3.24), (3.25) and (3.26) into equation (3.31), the following simultaneous algebraic equations are obtained for the piston motion mode

$$(S_{pd} + j\omega D_{pd})X_d e^{j\phi_d} + (\omega^2 + S_{pp} + j\omega D_{pp})X_p e^{j\phi_p} = 0 \tag{3.33}$$

$$(\omega^2 + S_{dd} + j\omega D_{dd})X_d e^{j\phi_d} + (S_{dp} + j\omega D_{dp})X_p e^{j\phi_p} = 0. \tag{3.34}$$

The corresponding equations for the casing motion mode may be inferred from equations (3.33) and (3.34) by simply replacing the subscript p by c. Since the governing equations are generically similar, only the piston motion mode will be analysed further.

Eliminating the amplitudes by substitution yields the characteristic equation

$$\begin{aligned} &\omega^4 + \omega^2(S_{pp} + S_{dd} + D_{dp}D_{pd} - D_{dd}D_{pp}) + S_{dd}S_{pp} - S_{dp}S_{pd} \\ &+ j\omega[\omega^2(D_{dd} + D_{pp}) + D_{dd}S_{pp} + S_{dd}D_{pp} - D_{pd}S_{pd} - S_{dp}D_{pd}] = 0, \end{aligned} \tag{3.35}$$

where both the real and imaginary parts are identically zero, thus

$$\omega^4 + \omega^2(S_{pp} + S_{dd} + D_{dp}D_{pd} - D_{dd}D_{pp}) + S_{dd}S_{pp} - S_{dp}S_{pd} = 0 \tag{3.36}$$

and

$$\omega^2 = (D_{dp}S_{pd} + S_{dp}D_{pd} - D_{dd}S_{pp} - S_{dd}D_{pp})/(D_{dd} + D_{pp}) \tag{3.37}$$

which are of course, subject to $\omega > \sqrt{-S_{cc}}$ by at least a factor of three. Correspondingly, the equations for the casing motion mode (by replacing the subscript p by c) would be subject to $\omega > \sqrt{-S_{pp}}$ by a similar factor.

The operating frequency is obtained from equation (3.37) whilst the physical constraints for operation are obtained by satisfying both equations (3.36) and (3.37) simultaneously.

Note that D_{pp} (and D_{cc}) are load coefficients. Since they appear in the frequency equation, it can be seen that, in general, free-piston Stirling engines have load-dependent frequencies. However, it will be shown that in some designs, with prudent selection of geometry, it is possible to reduce the variation of frequency with load to a negligible level.

For positive power, the piston and casing motions are required to lag behind those of the displacer. Therefore, it is convenient to measure phase displacements relative to the displacer motions. For the case of predominant piston motions, equation (3.24) becomes

$$x_p = X_p \exp[j(\omega t + \phi)] \tag{3.38}$$

$$x_d = X_d \exp(j\omega t) \tag{3.39}$$

where ϕ is defined as negative for piston motions lagging behind those of the displacer.

This allows equation (3.33) to be rearranged thus

$$X_p \exp(j\phi) = -(S_{pd} + j\omega D_{pd})X_d/(\omega^2 + S_{pp} + j\omega D_{pp}) \tag{3.40}$$

from which the phase and amplitude ratio are obtained:

$$\phi = \tan^{-1}\left(\frac{\omega[D_{pp}S_{pd} - D_{pd}(S_{pp} + \omega^2)}{-[S_{pd}S_{pp} + \omega^2(S_{pd} + D_{pd}D_{pp})]}\right) \tag{3.41}$$

$$\frac{X_d}{X_p} = r = \frac{(\omega^2 + S_{pp})^2 + \omega^2 D_{pp}^2}{\{[S_{pd}(\omega^2 + S_{pp}) + \omega^2 D_{pp}D_{pd}]^2 + \omega^2[D_{pd}(\omega^2 + S_{pp}) - D_{pp}S_{pd}]^2\}^{1/2}}. \tag{3.42}$$

Typically, ϕ lies in the third or fourth quadrant ($180° < \phi < 360°$) depending on the size of the denominator in equation (3.41).

Note that both amplitudes cannot be obtained simultaneously. One amplitude must be specified, usually estimated from geometric limitations or other non-linear effects.

Once the motions of the reciprocating elements have been determined, the work transfer may be calculated easily. The cyclic work is divided into two parts: irrecoverable work and useful (or recoverable) work, both of which are evaluated from the damping coefficients.

Cyclic work done against damping is given by

$$W = \oint F\dot{x}\,dx \tag{3.43}$$

where F is the damping coefficient in NSm^{-1}. Assume x is given by

$$x = X \sin \omega t. \tag{3.44}$$

Then equation (3.43) may be integrated over a cycle to give

$$W = \pi F \omega X^2. \tag{3.45}$$

The irrecoverable work is then

$$W_{ir} = -\pi\omega M_D[(D_{dd} + D_{pd}M_P/M_D)X_d^2 + D_{dd}X_p^2] + \pi\omega C_{H_{pc}}X_p^2, \tag{3.46}$$

being the work done against the damping caused by working gas viscous friction and gas spring hysteresis, and the useful work is

$$W_s = -\pi\omega M_P D_{pp}X_p^2 - \pi\omega C_{H_{pc}}X_p^2, \tag{3.47}$$

being the difference between the work done against the damping due to the load and the irrecoverable work due to the piston gas spring. The first term on the right-hand side of equation (3.47) may be equated with the thermodynamic pV power, in order to estimate the piston amplitude. Of course, this would be an iterative procedure unless a simple functional relationship for pV power

were used, for example the Schmidt isothermal power. Again, the corresponding equations for the casing motion mode are obtained by replacing the subscript p by c and M_P by M_C.

The cyclic work may also be obtained by solving the integral $\oint p\mathrm{d}v$ for each working space. However, the above results yield good answers when the linear approximations are reasonable and, furthermore, they are easy to apply. Table 3.1 lists all the pertinent equations derived in this section.

Table 3.1 Set of equations for the generalised dynamic analysis piston motion mode.

Frequency	$\omega^2 = (D_{dp}S_{pd} + S_{dp}D_{pd} - D_{dd}S_{pp} - S_{dd}D_{pp})/(D_{dd} + D_{pp})$
Geometric constraint	$\omega^4 + \omega^2(S_{pp} + S_{dd} + D_{dp}D_{pd} - D_{dd}D_{pp}) + S_{dd}S_{pp} - S_{dp}S_{pd} = 0$
Piston–displacer phase angle	$\phi = \tan^{-1}\left(\dfrac{\omega[D_{pp}S_{pd} - D_{pd}(S_{pp} + \omega^2)]}{-[S_{pd}S_{pp} + \omega^2(S_{pd} + D_{pd}D_{pp})]}\right)$

Piston–displacer amplitude ratio

$$\frac{X_d}{X_p} = r = \frac{(\omega^2 + S_{pp})^2 + \omega^2 D_{pp}^2}{\{[S_{pd}(\omega^2 + S_{pp}) + \omega^2 D_{pp}D_{pd}]^2 + \omega^2[D_{pd}(\omega^2 + S_{pp}) - D_{pp}S_{pd}]^2\}^{1/2}}$$

Irrecoverable work	$W_{ir} = -\pi\omega M_D[(D_{dd} + D_{pd}M_P/M_D)X_d^2 + D_{dp}X_p^2] + \pi\omega C_{H_{pc}}X_p^2$
Useful work	$W_s = -\pi\omega M_P D_{pp}X_p^2 - \pi\omega C_{H_{pc}}X_p^2$

The above equations are subject to $\omega \gg \sqrt{-S_{cc}}$.

The casing motion mode equations are obtained by replacing the subscript p by c and M_P by M_C. The resulting set of equations is subject to $\omega \gg \sqrt{-S_{pp}}$.

3.3 Linearisation

To apply the preceding analysis it is necessary to obtain functions for the pressure variations in the working spaces and the gas springs (also referred to as bounce spaces). The Isothermal analysis lends a convenient closed-form result which is known to be a fairly good approximation to reality. This result, from Chapter 2, table 2.1, is

$$p = MR\left[\frac{V_c}{T_k} + \frac{V_k}{T_k} + \frac{V_r \ln(T_h/T_k)}{(T_h - T_k)} + \frac{V_h}{T_h} + \frac{V_c}{T_h}\right]^{-1}. \tag{3.48}$$

were used, for example the Schmidt isothermal power. Again, the corresponding equations for the casing motion mode are obtained by replacing the subscript p by c and M_P by M_C.

The cyclic work may also be obtained by solving the integral $\oint p\mathrm{d}v$ for each working space. However, the above results yield good answers when the linear approximations are reasonable and, furthermore, they are easy to apply. Table 3.1 lists all the pertinent equations derived in this section.

Table 3.1 Set of equations for the generalised dynamic analysis piston motion mode.

Frequency	$\omega^2 = (D_{dp}S_{pd} + S_{dp}D_{pd} - D_{dd}S_{pp} - S_{dd}D_{pp})/(D_{dd} + D_{pp})$
Geometric constraint	$\omega^4 + \omega^2(S_{pp} + S_{dd} + D_{dp}D_{pd} - D_{dd}D_{pp}) + S_{dd}S_{pp} - S_{dp}S_{pd} = 0$
Piston–displacer phase angle	$\phi = \tan^{-1}\left(\dfrac{\omega[D_{pp}S_{pd} - D_{pd}(S_{pp} + \omega^2)]}{-[S_{pd}S_{pp} + \omega^2(S_{pd} + D_{pd}D_{pp})]}\right)$

Piston–displacer amplitude ratio

$$\frac{X_d}{X_p} = r = \frac{(\omega^2 + S_{pp})^2 + \omega^2 D_{pp}^2}{\{[S_{pd}(\omega^2 + S_{pp}) + \omega^2 D_{pp}D_{pd}]^2 + \omega^2[D_{pd}(\omega^2 + S_{pp}) - D_{pp}S_{pd}]^2\}^{1/2}}$$

Irrecoverable work	$W_{ir} = -\pi\omega M_D[(D_{dd} + D_{pd}M_P/M_D)X_d^2 + D_{dp}X_p^2] + \pi\omega C_{H_{pc}}X_p^2$
Useful work	$W_s = -\pi\omega M_P D_{pp}X_p^2 - \pi\omega C_{H_{pc}}X_p^2$

The above equations are subject to $\omega \gg \sqrt{-S_{cc}}$.

The casing motion mode equations are obtained by replacing the subscript p by c and M_P by M_C. The resulting set of equations is subject to $\omega \gg \sqrt{-S_{pp}}$.

3.3 Linearisation

To apply the preceding analysis it is necessary to obtain functions for the pressure variations in the working spaces and the gas springs (also referred to as bounce spaces). The Isothermal analysis lends a convenient closed-form result which is known to be a fairly good approximation to reality. This result, from Chapter 2, table 2.1, is

$$p = MR\left[\frac{V_c}{T_k} + \frac{V_k}{T_k} + \frac{V_r \ln(T_h/T_k)}{(T_h - T_k)} + \frac{V_h}{T_h} + \frac{V_e}{T_h}\right]^{-1}. \qquad (3.48)$$

By assuming adiabatic gas springs and a perfect gas, the pressure variations in these spaces may be described by

$$p_b = p_{mean}(V_B/V_b)^{\gamma}, \tag{3.49}$$

where V_B is the average gas spring volume and V_b is the instantaneous value.

Referring to figure 3.6 as a general example of a free-piston engine, the volume variations are as follows:

$$V_c = A_P(C_C + x_p) - (A_D - A_R)x_d \tag{3.50}$$

$$V_e = A_D(E_E + x_d + x_c). \tag{3.51}$$

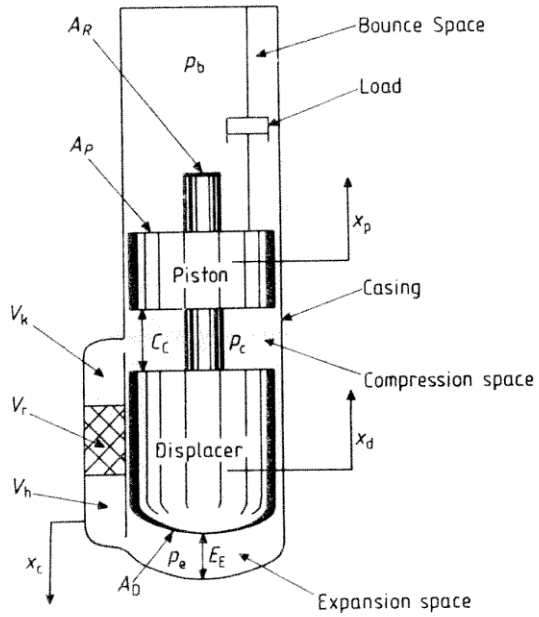


Figure 3.6 Volume variations. In the configuration shown, the displacer rod passes through the piston and is rigidly attached to the displacer. Movement in the positive direction tends to increase working space volumes.

Substituting equations (3.50) and (3.51) into equation (3.48), we obtain

$$p = MR\left(\frac{A_P(C_C + x_p) - (A_D - A_R)x_d}{T_k} + \frac{V_k}{T_k} + \frac{V_r \ln(T_h/T_k)}{(T_h - T_k)} + \frac{V_h}{T_h} + \frac{A_D(E_E + x_d + x_c)}{T_h}\right)^{-1} \tag{3.52}$$

In order to be used in the dynamic analysis, equation (3.52) must be linearised. For the purposes of this analysis the binomial expansion is used.

This method of linearisation has been found to work perfectly adequately and well within the tolerance expected from an analysis such as this. Rewriting equation (3.52)

$$p = \frac{MR}{S}\left(1 + \frac{A_P x_p - (A_D - A_R)x_d}{T_k S} + \frac{A_D(x_d + x_c)}{T_h S}\right)^{-1} \tag{3.53}$$

where

$$S = \frac{A_P C_C}{T_k} + \frac{V_k}{T_k} + \frac{V_r \ln(T_h/T_k)}{(T_h - T_k)} + \frac{V_h}{T_h} + \frac{A_D E_E}{T_h}. \tag{3.54}$$

Using the binomial expansion and neglecting second-order terms we obtain

$$p \simeq \frac{MR}{S}\left(1 - \frac{A_P x_p - (A_D - A_R)x_d}{T_k S} - \frac{A_D(x_d + x_c)}{T_h S} + \ldots\right) \tag{3.55}$$

given the following condition

$$\frac{A_P x_p - (A_D - A_R)x_d}{T_k S} + \frac{A_D(x_d + x_c)}{T_h S} \ll 1,$$

which is usually easily satisfied.

Since the mean pressure p_{mean} in the working spaces is the charge pressure, result (3.55) may be improved further by writing

$$p \simeq p_{mean}\left(1 - \frac{A_D(x_d + x_c)}{T_h S} - \frac{A_P x_p - (A_D - A_R)x_d}{T_k S}\right) \tag{3.56}$$

which implies that all pressure variations occur symmetrically around the charge pressure.

The gas spring pressures are linearised in the same way. These results are not repeated here since they depend on the particular geometry under consideration.

Since the Isothermal analysis does not account for pressure gradients, the pressure variation given by equation (3.56) may be thought of as representing the spatial average pressure at any instant. For the purposes of the dynamic analysis, equation (3.56) is taken to represent the pressure variation in the compression space. Academically, it may be more accurate to assume that this equation describes some intermediate pressure between the compression and expansion spaces. On the other hand, what this pressure variation is referred to does not make any significant difference to the dynamic components and, furthermore, the assumption does simplify the calculations to a great extent. Therefore we assume that

$$p_c \simeq p. \tag{3.57}$$

A result describing the pressure gradient across the heat exchangers is required in addition to the compression space pressure. This result would have

to account for the character of the gas flow which typically may be laminar or turbulent at any instant. The exact form of the pressure gradient result, therefore, is quite complex. Fortunately, in most machines the gas flow is predominantly turbulent in the cooler and heater and laminar in the regenerator. Thus it is convenient to assume that the pressure drop is made up as follows:

$$\Delta p = \Delta p_k + \Delta p_r + \Delta p_h \tag{3.58}$$

where Δp_k, Δp_r and Δp_h are the pressure drop over the cooler, regenerator and heater, respectively. For the cooler and heater, then, the pressure drop may be assumed to be of the form

$$\Delta p_{k,h} = \tfrac{1}{2}\rho(f_t + k_h)\bar{u}|\bar{u}| \tag{3.59}$$

where f_t is the turbulent friction factor and is given by $f_t = 4f_f L/d$, where f_f is the Fanning friction factor, L is the heat exchanger length and d is the hydraulic diameter. k_h is the head loss coefficient due to bends and entrance and exit effects. For the regenerator

$$\Delta p_r = f_l \bar{u} \tag{3.60}$$

where f_l is the laminar friction factor.

Equations (3.59) and (3.60) are both approximate but they do describe the pressure gradients quite accurately in the majority of free-piston Stirling engines. Equation (3.59) is non-linear and must therefore be linearised before it can be incorporated into the dynamic analysis. The variation of density with pressure is neglected since in most high pressure free-piston engines the pressure swing is usually a small fraction of the charge pressure (typically less than 15%). It is therefore assumed that density is only a function of local temperature, which in the isothermal case is constant. Thus the major contribution to the non-linearity is the velocity term.

Since the pressure drop constitutes a damping effect on the reciprocating components, it is necessary to find an equivalent linear damper which will dissipate the same energy. The condition for equivalence to be satisfied is: the energy dissipated per quarter cycle by actual damping must equal the energy dissipated per quarter cycle by an equivalent linear damping (Anvoner 1970). From equation (3.59), the energy dissipated per unit flow area is

$$E_1 = \tfrac{1}{2}\frac{\rho}{\omega}(f_t + k_h)\int_0^{\pi/2} (\bar{u})^2 \bar{u}\,\mathrm{d}\omega t. \tag{3.61}$$

For sinusoidal displacements the flow velocity is approximated by

$$\bar{u} = U\cos\omega t. \tag{3.62}$$

Substituting equation (3.62) into equation (3.61) and integrating, we obtain

$$E_1 = \tfrac{1}{3}\rho(f_t + k_h)U^3/\omega. \tag{3.63}$$

In the case of linear damping, the damping force per unit flow area is given by

$$\Delta p = C\bar{u}. \tag{3.64}$$

In this case the energy dissipated per unit flow area is:

$$E_2 = C\int_0^{\pi/2} (\bar{u})^2 \,\mathrm{d}\omega t. \tag{3.65}$$

Substituting equation (3.62) into equation (3.65) and integrating, we obtain

$$E_2 = \frac{\pi C}{4\omega} U^2. \tag{3.66}$$

Equating (3.63) and (3.66) and solving for C, we obtain

$$C = \tfrac{4}{3}(\rho/\pi)(f_t + k_h)U. \tag{3.67}$$

The turbulent pressure drop may now be represented by

$$\Delta p_{k.\,h} \simeq \tfrac{4}{3}(\rho/\pi)(f_t + k_h)U. \tag{3.68}$$

Thus the total linear pressure drop across the heat exchangers is given by

$$\Delta p = \tfrac{4}{3}(1/\pi)\{[\rho U(f_t + k_h)\bar{u}]_{\text{cooler}} + [\rho U(f_t + k_h)\bar{u}]_{\text{heater}}\} + f_1 \bar{u}_{\text{regenerator}}. \tag{3.69}$$

It is now necessary to find the gas flow velocities in terms of the velocities of the reciprocating elements. This is done by obtaining the net volumetric flow rate through the heat exchanger loop and dividing by the relevant flow area to obtain the corresponding gas velocity. Obviously the volumetric flow rate is not uniform throughout the heat exchangers and therefore the velocities so derived are approximate.

The volumetric flow out of the compression space is given by

$$\dot{V}_c = \mathrm{d}V_c/\mathrm{d}t \tag{3.70}$$

and the volumetric flow out of the expansion space by

$$\dot{V}_e = \mathrm{d}V_e/\mathrm{d}t. \tag{3.71}$$

Therefore, from equations (3.50) and (3.51):

$$\dot{V}_c = A_P\dot{x}_p - (A_D - A_R)\dot{x}_d \tag{3.72}$$

$$\dot{V}_e = A_D(\dot{x}_d + \dot{x}_c). \tag{3.73}$$

Positive values for $\dot{V}_c$ and $\dot{V}_e$ indicate increasing volumes, and therefore the net volumetric flow rate through the heat exchangers is given by

$$\bar{\dot{V}} = \dot{V}_c - \dot{V}_e, \tag{3.74}$$

which from equations (3.72) and (3.73) becomes

$$\bar{\dot{V}} = A_P\dot{x}_p - (2A_D - A_R)\dot{x}_d - A_D\dot{x}_c. \tag{3.75}$$

The approximate velocity through each heat exchanger component is then given by

$$\bar{u}_k = \bar{\dot{V}}/A_k \qquad \text{(cooler velocity)} \tag{3.76}$$

$$\bar{u}_r = \bar{\dot{V}}/A_r \qquad \text{(regenerator velocity)} \tag{3.77}$$

$$\bar{u}_h = \bar{\dot{V}}/A_h \qquad \text{(heater velocity)} \tag{3.78}$$

where A_k, A_r and A_h are the flow areas of the cooler, the regenerator and the heater, respectively.

The pressure drop given by equation (3.69) may now be written

$$\Delta p = \{\tfrac{4}{3}(1/\pi)[\rho_k U_k (f_t + k_h)_k / A_k + \rho_h U_h (f_t + k_h)_h / A_h] + f_l / A_r\} \bar{\dot{V}} \tag{3.79}$$

where ρ_k and ρ_h are given by

$$\rho_k = p_{mean}/(RT_k) \tag{3.80}$$

$$\rho_h = p_{mean}/(RT_h) \tag{3.81}$$

and U_k and U_h are the velocity amplitudes (or peak velocities) in the cooler and the heater and are given by

$$U_k = V_A/A_k \tag{3.82}$$

$$U_h = V_A/A_h \tag{3.83}$$

where V_A is the net volumetric flow rate amplitude and is evaluated from equation (3.75) by assuming sinusoidal displacements. The results for the two operating modes are:

(i) piston motion mode

$$V_A = \omega[(A_P X_p)^2 - 2(2A_D - A_R)A_P X_p X_d \sin\phi + (2A_D - A_R)^2 X_d^2]^{1/2} \tag{3.84}$$

(ii) casing motion mode

$$V_A = \omega[(A_D X_c)^2 + 2(2A_D - A_R)A_D X_c X_d \sin\phi + (2A_D - A_R)^2 X_d^2]^{1/2}. \tag{3.85}$$

From equations (3.75) and (3.79) it can be seen that the pressure drop is now a linear function of the velocities of the reciprocating elements. Recalling the assumed Δp relationship given by equation (3.8), we have

$$A_D \Delta_p = C_p \dot{x}_p + C_d \dot{x}_d + C_c \dot{x}_c. \tag{3.86}$$

Comparing this result with equation (3.79) and equating the respective coefficients of $\dot{x}_p$, $\dot{x}_d$ and $\dot{x}_c$, we have

$$C_p = A_P A_D P_C \tag{3.87}$$

$$C_d = -(2A_D - A_R)A_D P_C \tag{3.88}$$

$$C_c = -A_D{}^2 P_C \tag{3.89}$$

where P_C is given by

$$P_C = \tfrac{4}{3}(1/\pi)[\rho_k U_k (f_t + k_h)_k / A_k + \rho_h U_h (f_t + k_h)_h / A_h] + f_l / A_r, \quad (3.90)$$

which is the final form of the linear damping due to the heat exchanger pressure drop.

A similar pressure drop analysis was done by G Wood at Sunpower Incorporated (Wood 1980). Table 3.2 lists a comparison of his linear results for Δ_p with those predicted by a more complete simulation routine such as that developed in Chapter 5. The agreement between the linear analysis and the simulation can be seen to be perfectly adequate for the purposes set out for the linear analysis.

Table 3.2 Comparison of the Linear and Simulation Δp amplitudes (after Wood 1980).

Displacer amplitude (cm)	Displacer–piston phase (degrees)	Piston amplitude (cm)	Peak Reynolds number C	R	H	Δp (bar) Simulation	Linear
1.27	65.00	1.27	19780	6	5136	0.410	0.345
0.98	21.94	1.12	6311	2	3734	0.126	0.118
0.87	25.00	1.12	7543	2	3351	0.108	0.097
0.45	65.00	1.12	15790	3	1917	0.178	0.146
0.40	90.00	1.12	18440	4	1693	0.230	0.193

The above results were evaluated for a machine operating at 60 Hz: C, cooler; R, regenerator; H, heater.

Finally, we require to calculate the damping effect due to gas spring hysteresis loss. The details of this loss mechanism are outlined in Chapter 7. Therefore, referring to Chapter 7, equation (7.31), the hysteresis loss is given by

$$\bar{W} = \frac{k}{4}\sqrt{\frac{\omega}{2\alpha_0}}\gamma(\gamma-1)T_w\left(\frac{\Delta V}{V_B}\right)^2. \quad (3.91)$$

The work done against damping is given by (equation 3.45)

$$W = \pi C_H \omega X^2 \quad (3.92)$$

where C_H is the gas spring damping coefficient and X is the amplitude of the damped motions.

Expressing (3.92) as a power

$$\dot{W} = \tfrac{1}{2} C_H (\omega X)^2. \quad (3.93)$$

Equating (3.91) and (3.93) and noting that $\Delta V = AX$

$$C_H = \frac{k}{2}\sqrt{\frac{\omega}{2\alpha_0}}\gamma(\gamma-1)T_w A_w\left(\frac{A}{\omega V_B}\right)^2 \quad (3.94)$$

where A is the relevant presented area of the reciprocating element. Equation (3.94) thus gives the damping due to gas spring hysteresis. All the relevant linear equations are listed in table 3.3.

Table 3.3 Set of linearised equations for pressure, pressure drop and gas spring hysteresis damping.

Compression space pressure	$p_c \simeq p_{mean}\left(1-\frac{A_D(x_d+x_c)}{T_h S}-\frac{A_P x_p-(A_D-A_R)x_d}{T_k S}\right)$
	where
	$S=\frac{A_P C_C}{T_k}+\frac{V_k}{T_k}+\frac{V_r \ln(T_h/T_k)}{(T_h-T_k)}+\frac{V_h}{T_h}+\frac{A_D E_E}{T_h}$
Pressure drop	$A_D\Delta_p = C_p\dot{x}_p + C_d\dot{x}_d + C_c\dot{x}_c$
	where
	$C_p = A_P A_D P_C \quad C_d = -(2A_D - A_R)A_D P_C \quad C_c = -A_D{}^2 P_C$
	and
	$P_C = \frac{4}{3}(1/\pi)[\rho_k U_k (f_t + k_h)_k / A_k + \rho_h U_h (f_t + k_h)_h / A_h] + f_l / A_r$
Gas spring hysteresis damping	$C_H = \frac{k}{2}\sqrt{\frac{\omega}{2\alpha_0}}\gamma(\gamma-1)T_w A_w\left(\frac{A}{\omega V_B}\right)^2$

Note that other methods may be used to generate the linear functions. For example, if a sophisticated computer simulation is used to analyse the thermodynamic cycle, the pressure variations will only be known implicitly. In this case, volume variations would be assumed, the resulting pressure variations analysed by Fourier techniques and then, using only the linear terms, the dynamic analysis would be performed to obtain new volume variations. This process is therefore iterative unless one is solving for the reciprocating masses, in which case the solution is obtained directly. The second-order terms neglected should be of the order of 10% or less of the linear terms (Gedeon 1978). Particular examples are now considered.

3.4 The Sunpower RE-1000 engine

The RE-1000 was developed at Sunpower to investigate free-piston Stirling engine applications. The engine is primarily a research machine and is the seventh in a series which began with an engine being built for the American Gas Association (Beale *et al* 1975). The power to load is of the order of 1 kW (figure 3.7).

where A is the relevant presented area of the reciprocating element. Equation (3.94) thus gives the damping due to gas spring hysteresis. All the relevant linear equations are listed in table 3.3.

Table 3.3 Set of linearised equations for pressure, pressure drop and gas spring hysteresis damping.

Compression space pressure	$p_c \simeq p_{mean}\left(1-\frac{A_D(x_d+x_c)}{T_h S}-\frac{A_P x_p-(A_D-A_R)x_d}{T_k S}\right)$
	where
	$S=\frac{A_P C_C}{T_k}+\frac{V_k}{T_k}+\frac{V_r \ln(T_h/T_k)}{(T_h-T_k)}+\frac{V_h}{T_h}+\frac{A_D E_E}{T_h}$
Pressure drop	$A_D\Delta_p = C_p\dot{x}_p + C_d\dot{x}_d + C_c\dot{x}_c$
	where
	$C_p = A_P A_D P_C \quad C_d = -(2A_D - A_R)A_D P_C \quad C_c = -A_D{}^2 P_C$
	and
	$P_C = \frac{4}{3}(1/\pi)[\rho_k U_k (f_t + k_h)_k / A_k + \rho_h U_h (f_t + k_h)_h / A_h] + f_l / A_r$
Gas spring hysteresis damping	$C_H = \frac{k}{2}\sqrt{\frac{\omega}{2\alpha_0}}\gamma(\gamma-1)T_w A_w\left(\frac{A}{\omega V_B}\right)^2$

Note that other methods may be used to generate the linear functions. For example, if a sophisticated computer simulation is used to analyse the thermodynamic cycle, the pressure variations will only be known implicitly. In this case, volume variations would be assumed, the resulting pressure variations analysed by Fourier techniques and then, using only the linear terms, the dynamic analysis would be performed to obtain new volume variations. This process is therefore iterative unless one is solving for the reciprocating masses, in which case the solution is obtained directly. The second-order terms neglected should be of the order of 10% or less of the linear terms (Gedeon 1978). Particular examples are now considered.

3.4 The Sunpower RE-1000 engine

The RE-1000 was developed at Sunpower to investigate free-piston Stirling engine applications. The engine is primarily a research machine and is the seventh in a series which began with an engine being built for the American Gas Association (Beale *et al* 1975). The power to load is of the order of 1 kW (figure 3.7).

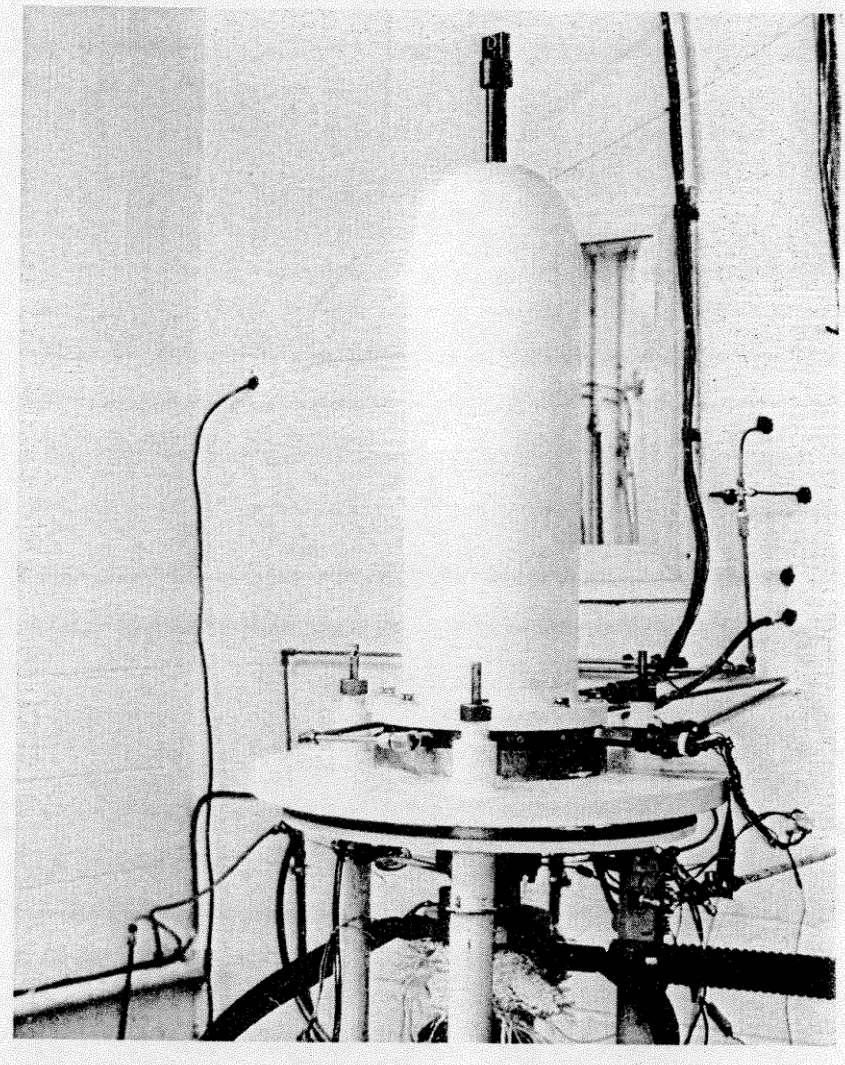


Figure 3.7 Sunpower RE-1000 engine (courtesy Sunpower Incorporated).

Referring to figure 3.8(*a*), the equations of motion are

$$M_P\ddot{x}_p = A_P(p_c - p_b) - (C_{pc} + C_{H_{pc}})(\dot{x}_p + \dot{x}_c) \tag{3.95}$$

$$M_D\ddot{x}_d = A_D\Delta_p + A_R(p_c - p_d) - C_{H_{dc}}(\dot{x}_d + \dot{x}_c) \tag{3.96}$$

$$M_C\ddot{x}_c = A_D(p_c + \Delta_p - p_b) + A_R(p_c - p_d) - (C_{pc} + C_{H_{pc}})(\dot{x}_c + \dot{x}_p) - K_C x_c \tag{3.97}$$

where the expansion space pressure is taken to be given by

$$p_e = p_c + \Delta_p.$$

This engine has two active gas springs. The pressure variations for these spaces are:

$$p_b = p_{mean}\{V_B/[V_B - A_P(x_p + x_c)]\}^{\gamma} \tag{3.98}$$

$$p_d = p_{mean}\{V_D/[V_D - A_R(x_d + x_c)]\}^{\gamma}. \tag{3.99}$$

Linearising equations (3.98) and (3.99) by the same method that is used for the working gas pressure, the following are obtained:

$$p_b \simeq p_{mean}[1 + \gamma(A_P/V_B)(x_p + x_c)] \tag{3.100}$$

$$p_d \simeq p_{mean}[1 + \gamma(A_R/V_D)(x_d + x_c)]. \tag{3.101}$$

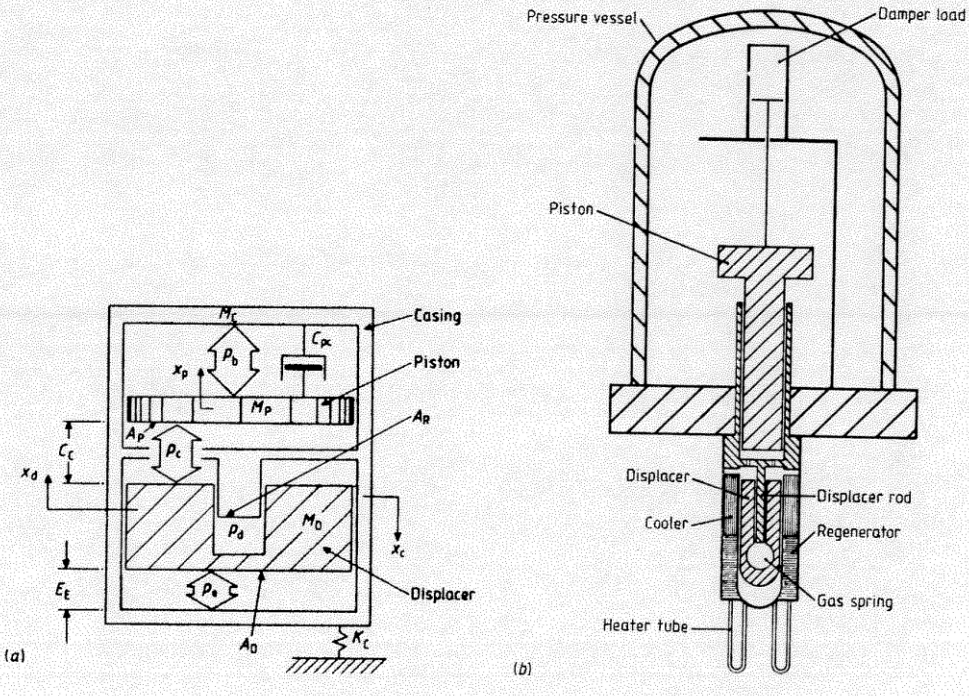


Figure 3.8 Sunpower RE-1000 engine: (*a*) Notation; (*b*) Schematic.

Equations (3.100) and (3.101), together with the linearised equations from table 3.3 are substituted into the equations of motion (3.95), (3.96) and (3.97). Since this is a piston motion mode engine the terms due to the casing motion may be neglected. The casing spring term (S_{cc}), though, is required to ensure that the condition for small casing motions is met. The remaining linear coefficients follow directly by inspection. They are:

$$S_{pp} = -\frac{A_P^2}{M_P}p_{mean}\left(\frac{1}{T_k S} + \frac{\gamma}{V_B}\right) \qquad S_{pd} = -\frac{A_P^2}{M_P S}p_{mean}\left(\frac{1}{T_h} - \frac{(1 - A_R/A_P)}{T_k}\right)$$

$$D_{pp} = -(C_{pc} + C_{H_{pc}})/M_P \qquad D_{pd} = 0$$

$$S_{dp} = -\frac{A_P A_R p_{mean}}{M_D T_k S} \qquad S_{dd} = -\frac{A_R}{M_D} p_{mean}\left(\frac{A_P}{T_h S} - \frac{(A_P - A_R)}{T_k S} + \gamma \frac{A_R}{V_D}\right)$$

$$D_{dp} = C_p / M_D \qquad D_{dd} = (C_d - C_{H_{dc}})/M_D$$

$$S_{cc} = -\frac{A_P{}^2}{M_C} p_{mean}\left(\frac{1}{T_h S} + \frac{\gamma}{V_B} + \frac{A_R/A_P}{T_h S} + \frac{\gamma (A_R/A_P)^2}{V_D}\right) - \frac{K_C}{M_C}.$$

For this particular configuration note that $A_P = A_D$.

From table 3.1 (setting $D_{pd} = 0$) the set of dynamic equations for this configuration becomes:

frequency

$$\omega^2 = (D_{dp} S_{pd} - D_{dd} S_{pp} - S_{dd} D_{pp})/(D_{dd} + D_{pp}) \tag{3.102}$$

geometric constraint

$$\omega^4 + \omega^2 (S_{pp} + S_{dd} - D_{dd} D_{pp}) + S_{dd} S_{pp} - S_{dp} S_{pd} = 0 \tag{3.103}$$

piston–displacer phase angle

$$\phi = \tan^{-1}\left(\frac{\omega D_{pp}}{-(S_{pp} + \omega^2)}\right) \tag{3.104}$$

piston–displacer amplitude ratio

$$\frac{X_d}{X_p} = r = \frac{(\omega^2 + S_{pp})^2 + \omega^2 D_{pp}{}^2}{S_{pd}[(\omega^2 + S_{pp})^2 + \omega^2 D_{pp}{}^2]^{1/2}}. \tag{3.105}$$

From the frequency equation, it can be seen that since the load coefficient D_{pp} appears in this equation, these machines will in general have a load-dependent frequency. However, by careful choice of the geometric parameters, it is possible to alter this characteristic and achieve almost constant frequency for a wide range of loads. Obviously, there are many applications where this might be desirable. It is instructive to see how this engine can be designed for constant frequency operation.

On all free-piston machines the displacer is typically much lighter than the piston, thus

$$D_{dd} \gg D_{pp} \tag{3.106}$$

and by arranging that

$$|D_{dd} S_{pp}| \gg |D_{dp} S_{pd} - S_{dd} D_{pp}| \tag{3.107}$$

equation (3.102) becomes

$$\omega^2 = -S_{pp} \tag{3.108}$$

which is independent of the load and gives the operating frequency as simply the natural frequency of the piston.

The geometric constraint equation must also be satisfied. This may be done in various ways, one of which is to arrange the following:

$$S_{dd} = D_{dd} D_{pp} \tag{3.109}$$

and

$$S_{dd} S_{pp} = S_{dp} S_{pd}, \tag{3.110}$$

thus reducing equation (3.103) to the same result as that expressed by (3.108). Equations (3.106), (3.107), (3.109) and (3.110) specify how the geometry must be arranged to achieve constant frequency operation.

The phase angle under these conditions is also constant:

$$\phi = -90°. \tag{3.111}$$

The amplitude ratio is, however, load† dependent

$$r = \omega D_{pp} / S_{pd} \tag{3.112}$$

which results in a load dependent power.

Note that for constant frequency operation the piston and the displacer are in quadrature, and consequently the optimum phase angle will not in general be satisfied. Thus, when designing this engine for constant frequency operation, there is often a trade-off on available power. On the other hand, if the machine is not designed for constant frequency operation, its capacity to tolerate load variation is much reduced, and such machines tend to operate best as fixed load devices.

The RE-1000 was not specifically designed as a constant frequency device. However, it does operate at essentially constant frequency over a nominal change in load. The design point operating conditions and geometric data are listed in table 3.4. Table 3.5 compares the dynamic operating parameters predicted by the linear analysis with those of a complete simulation and also with experimental values. The agreement can be seen to be good, which indicates that this rather simple analysis can return realistic results. The geometric constraint equation was used only to calculate the bounce space volume V_B.

It is usually important to know how frequency and power will vary with the load damping. This may be done by plotting the frequency and power as a function of D_{pp}. However, to do this presents somewhat of a difficulty, since the geometric constraint equation would typically only be satisfied for a particular value of D_{pp}. In a real machine, what happens is that as the amplitudes and frequency change in response to a load change, the viscous damping in the heat exchangers also changes (approximately proportional to velocity to the 1.75 power). Thus, in a well designed machine the viscous

†The term *load* may be confusing here: the load is the damper to which the engine dissipates its power.

Table 3.4 Sunpower RE-1000 Engine data.

General	
Working fluid	Helium
Mean pressure	$p_{mean} = 71$ bar
Mean hot space temperature	$T_h = 814.3$ K
Mean cold space temperature	$T_k = 322.8$ K
Geometric	
Cooler volume	$V_k = 20.43\ cm^3$
Regenerator volume	$V_r = 56.37\ cm^3$
Heater volume	$V_h = 27.33\ cm^3$
Displacer rod area	$A_R = 2.176\ cm^2$
Piston frontal area	$A_P = 25.69\ cm^2$
Displacer spring volume	$V_D = 37.97\ cm^3$
Piston bounce space volume	$V_B = 2615.0\ cm^3$
Expansion space clearance	$E_E = 18.61$ mm
Compression space clearance	$C_C = 18.30$ mm
Masses	
Piston	$M_P = 6.20$ kg
Displacer	$M_D = 0.426$ kg
Casing	$M_C = 416.0$ kg
Dynamic	
Heat exchanger pressure drop and phase with respect to piston	0.5 bar at 166.4°
Piston amplitude	$X_p = 11.45$ mm
Displacer amplitude	$X_d = 12.33$ mm
Load damping	$C_{pc} = 461.5\ Ns\ m^{-1}$
Displacer gas spring damping	$C_{H_{dc}} = 35.34\ Ns\ m^{-1}$

damping tends to stabilise the motions. To model this effect with the linear analysis it is assumed that D_{dd} and D_{dp} change to effect closure, i.e. that the geometric constraint equation is simultaneously satisfied by allowing D_{dd} and D_{dp} to change. For the RE-1000 the D_{dp} term makes a negligible contribution to the frequency and power calculations and is neglected. Therefore the frequency equation and the geometric constraint equation are simultaneously satisfied by allowing only D_{pp} to vary. The performance characteristics so determined are shown in figure 3.9. It can be seen that for a load change of 100 %, the change in frequency is only 13 %. The power, though, changes quite dramatically.

The power calculation assumes that the piston amplitude remains constant and that the displacer amplitude varies according to the amplitude ratio equation. In reality, both the displacer and the piston change their amplitudes in response to a load change, the ratio of these two amplitudes being given by

Table 3.5 Linear analysis results and comparison with simulation and experiment (RE-1000 engine)

Linear coefficients
$S_{pp} = -5.345 \times 10^4$
$S_{pd} = 2.983 \times 10^4$
$D_{pp} = -74.44$
$S_{dp} = -7.086 \times 10^4$
$S_{dd} = 2.121 \times 10^3$
$D_{dp} = 70.82$
$D_{dd} = -506.4$
$S_{cc} = -547.7$

	Linear analysis	Simulation	Experiment
Frequency (Hz)	33.2	30.0	30.0
Phase angle (°)	−57.9	−43.7	−42.5
Amplitude ratio	0.62	1.08	1.06
Output power (based X_p) (kW)	1.32	1.08	1.00
Output power (based X_d)	2.00	—	—

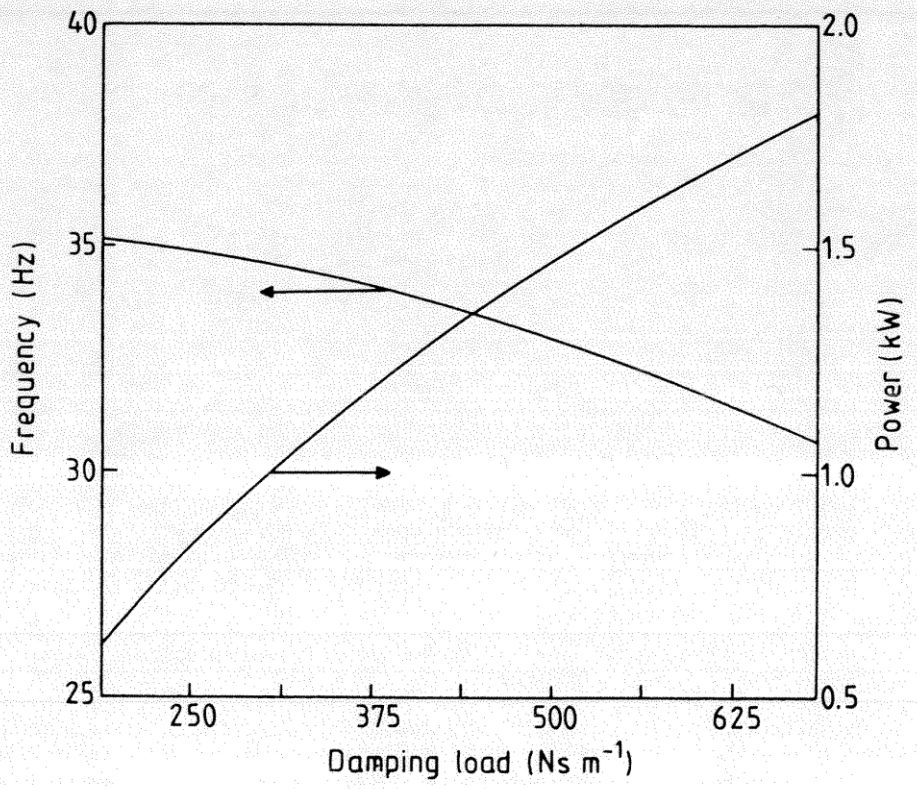


Figure 3.9 Frequency and power versus load for the Sunpower RE-1000 engine.

the amplitude ratio equation. A more accurate way of assessing the power curve is to ensure that the thermodynamic pV power and the power dissipated in the load are approximately equal, by solving for the piston amplitude that will make this so.

3.5 The Sunpower M100 free-cylinder engine

This machine is an example of a casing mode engine developed at Sunpower which has proved to be a very reliable performer (See Beale (1979) and Beale *et al* (1971), and also figure 3.10). The engine is used to drive a water pump by the action of its cylinder. In this form it has been extensively tested both in the field and laboratory. One of the most attractive features of this engine is that it is constructed from inexpensive easy-to-obtain components. In fact, the entire pump housing is fabricated from ordinary PVC piping and fixtures that may be purchased at any reasonably equiped hardware shop. The reliability, simplicity and low cost of this engine make it an eminently suitable device for application in developing countries.

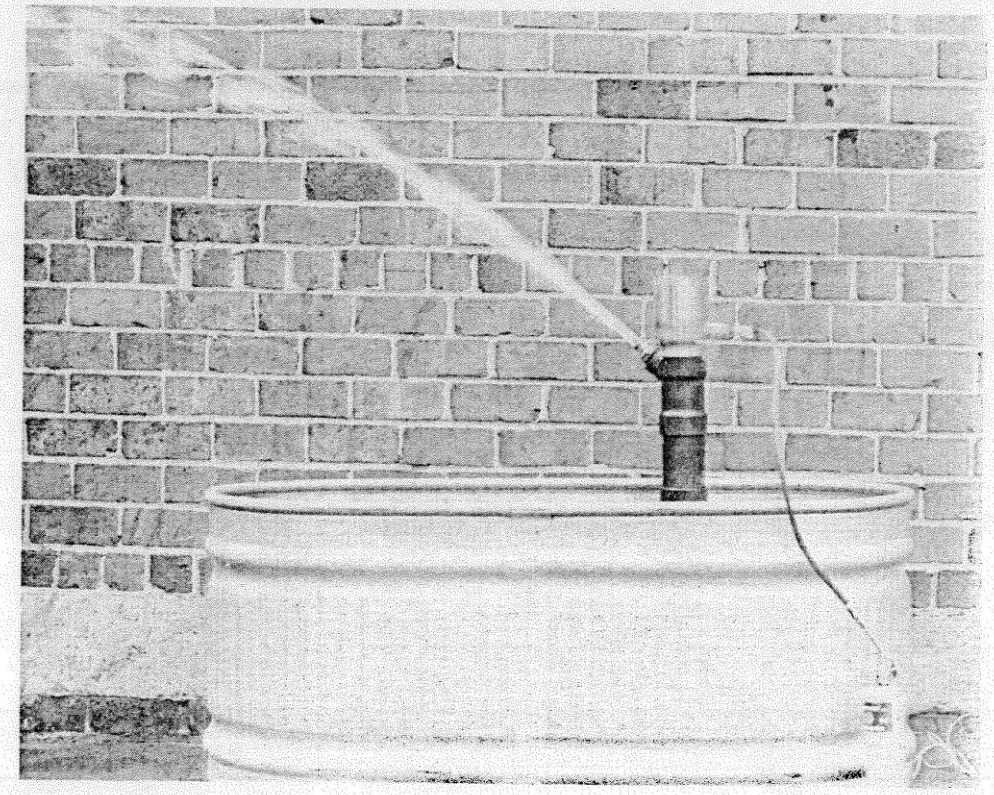


Figure 3.10 Sunpower M100 free-cylinder engine (courtesy Sunpower Incorporated).

Referring to figure 3.11(*a*), the equations of motion for this engine are

$$M_P\ddot{x}_p = A_R p_d + A_P p_c - A_D p_b \tag{3.113}$$

$$M_D\ddot{x}_d = A_R(p_c - p_d) + A_D\Delta p - C_{H_{dp}}(\dot{x}_d - \dot{x}_p) \tag{3.114}$$

$$M_C\ddot{x}_c = A_D(p_c + \Delta p - p_b) - C_L\dot{x}_c \tag{3.115}$$

where positive directions are taken as those which increase the working gas volume and it is assumed that the load may be represented by a damper acting on the casing C_L.

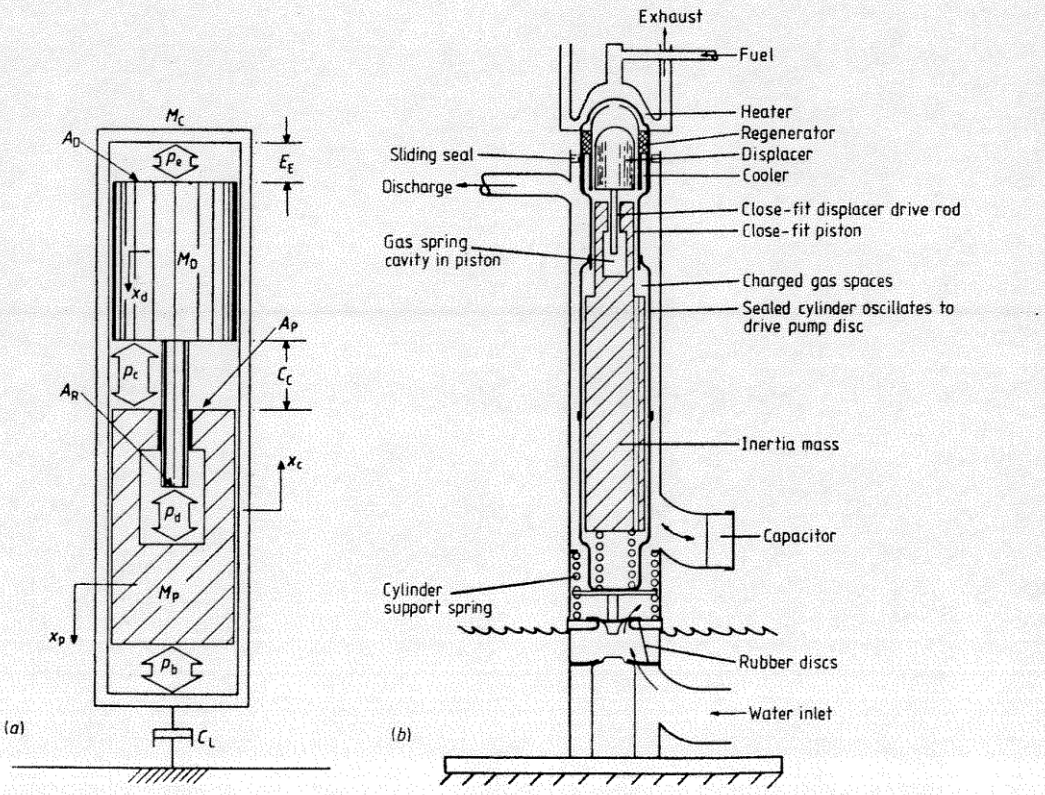


Figure 3.11 Sunpower M100 free-cylinder engine: (*a*) Notation; (*b*) Schematic. NB: The support springs are usually weak compared with the gas springs and make no significant contribution to the spring forces. They have therefore been neglected in the analysis.

This engine has only one active gas spring, the displacer gas spring. The pressure variation for this space is given by

$$p_d = p_{mean}\{V_D/[V_D - A_R(x_d - x_p)]\}^{\gamma}, \tag{3.116}$$

which is linearised to

$$p_d \simeq p_{mean}[1+\gamma(A_R/V_D)(x_d - x_p)]. \tag{3.117}$$

Owing to the large bounce space volume the pressure variation in this space is small and may be safely neglected. Thus

$$p_b \simeq p_{mean}. \tag{3.118}$$

Since this is a casing motion engine it is possible to neglect the motions of the piston if its natural frequency is small compared with the operating frequency. By substituting the linearised equations into the equations of motion, the relevant linear coefficients follow by inspection:

$$S_{pp} = -\frac{p_{mean}}{M_P}\left(\frac{A_P^2}{T_k S} + \gamma\frac{A_R^2}{V_D}\right)$$

$$S_{dd} = \frac{A_R}{M_D}p_{mean}\left(\frac{A_P}{T_k S} - \frac{A_D}{T_h S} - \gamma\frac{A_R}{V_D}\right) \qquad S_{dc} = -\frac{A_D A_R p_{mean}}{M_D T_h S}$$

$$D_{dd} = (C_d - C_{H_{dp}})/M_D \qquad D_{dc} = C_c/M_D$$

$$S_{cd} = \frac{A_D}{M_C S}p_{mean}\left(\frac{A_P}{T_k} - \frac{A_D}{T_h}\right) \qquad S_{cc} = -\frac{A_D^2 p_{mean}}{M_C T_h S}$$

$$D_{cd} = C_d/M_C \qquad D_{cc} = (C_c - C_L)/M_C.$$

Unfortunately, for this case it is not possible to simplify the dynamic equations (table 3.1).

Table 3.6 lists the design point operating conditions and geometric data for the engine. A detailed comparison between the experimental and linear results is not possible, since the machine was never fully instrumented. However, the net pumping power, cylinder stroke and frequency were measured. A comparison of these quantities with the linear results is given in table 3.7. Notwithstanding that the pump efficiency is not known, the correlation is excellent on frequency and certainly of similar magnitude on power. In this example, the geometric constraint equation was used to evaluate the displacer gas spring volume V_D.

The performance characteristics as calculated by this analysis are shown in figure 3.12. It is interesting to note that the power change is quite small (18 %) for a considerable change in load (over 100 %). This is very advantageous for certain applications, for example, as an agricultural water pumping machine. In such a situation it could be expected that the pumping load could vary widely.

The cylinder stroke versus load is shown in figure 3.13. It can be seen that at the higher power levels the amplitude is much reduced. The amount of water pumped is, of course, directly proportional to the stroke.

Table 3.6 Sunpower M100 free-cylinder engine data.

General	
Working fluid	Helium
Mean pressure	$p_{mean} = 22$ bar
Mean hot space temperature	$T_h = 866.6$ K
Mean cold space temperature	$T_k = 322.3$ K
Geometric	
Cooler volume	$V_k = 3.41\ cm^3$
Regenerator volume	$V_r = 25.4\ cm^3$
Heater volume	$V_h = 2.42\ cm^3$
Displacer rod area	$A_R = 1.327\ cm^2$
Piston frontal area	$A_P = 7.744\ cm^2$
Displacer frontal area	$A_D = 9.071\ cm^2$
Displacer spring volume	$V_D = 9.95\ cm^3$
Expansion space clearance	$E_E = 20.0$ mm
Compression space clearance	$C_C = 16.0$ mm
Masses	
Piston	$M_P = 10$ kg
Displacer	$M_D = 78.5$ g
Casing	$M_C = 3.3$ kg
Dynamic	
Displacer amplitude	$X_d = 15.0$ mm
Cylinder amplitude	$X_c = 5.0$ to 7.5 mm
Load damping (approximate)	$C_L = 132.0\ Ns\ m^{-1}$
Displacer gas spring damping (approximate)	$C_{H_{dp}} = 2.3\ Ns\ m^{-1}$

Table 3.7 Linear analysis results and comparison with experiment (M100 free-cylinder engine).

Linear coefficients

$S_{pp} = -3.829 \times 10^3$	$S_{cd} = 6.909 \times 10^3$
$S_{dd} = -3.381 \times 10^4$	$S_{cc} = -5.333 \times 10^3$
$S_{dc} = -3.280 \times 10^4$	$D_{cd} = -0.714$
$D_{dd} = -29.95$	$D_{cc} = -40.0$
$D_{dc} = 15.43$	

	Linear analysis	Experiment
Frequency (Hz)	22.4	23.0
Amplitude ratio	2.25	2.5†
Output power (W)	58.5	40.0†

† Approximate values

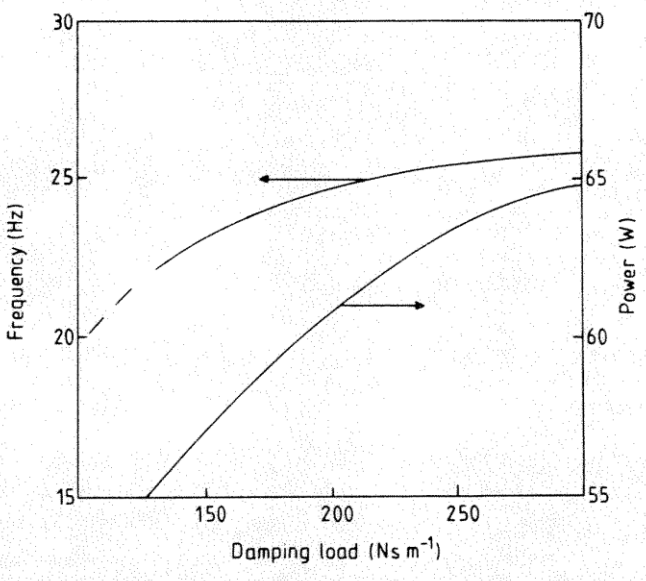


Figure 3.12 Frequency and power versus load for the Sunpower M100 free-cylinder engine.

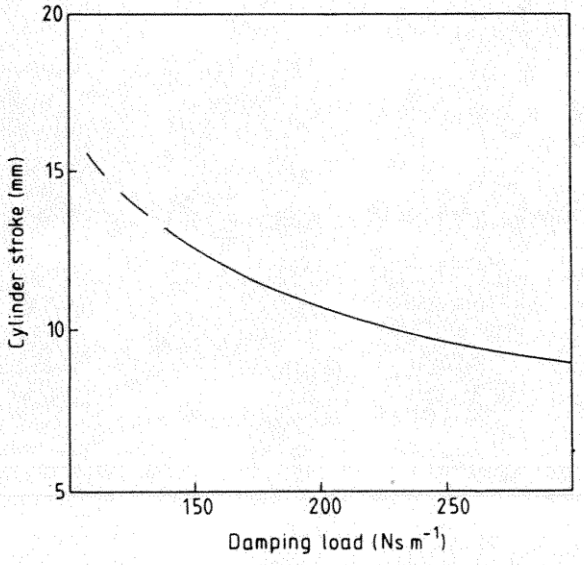


Figure 3.13 Stroke versus load for the Sunpower M100 free-cylinder engine.

3.6 The Harwell Thermo-Mechanical Generator (TMG)

The TMG is a unique free-piston machine in many respects. Instead of a piston, this engine has a diaphragm which flexes to effect the change of working gas volume. In addition, the stiffness of the diaphragm acts as a mechanical spring.

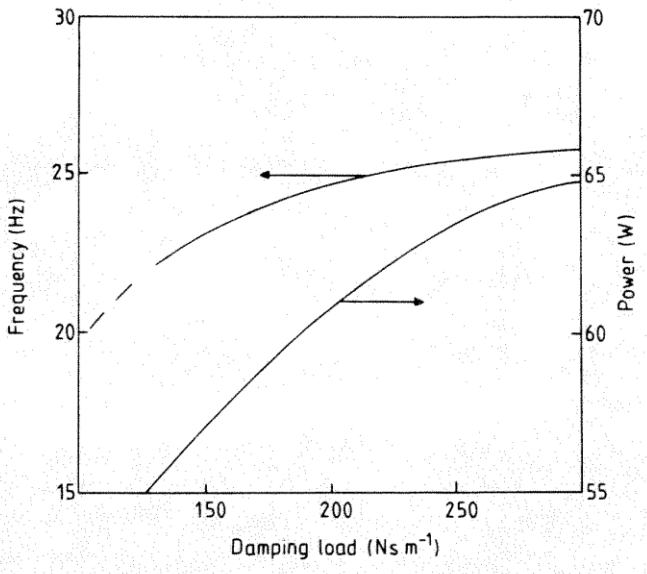


Figure 3.12 Frequency and power versus load for the Sunpower M100 free-cylinder engine.

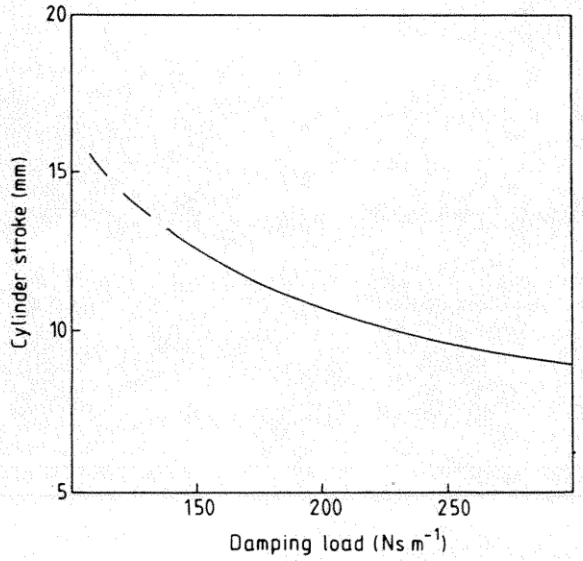


Figure 3.13 Stroke versus load for the Sunpower M100 free-cylinder engine.

3.6 The Harwell Thermo-Mechanical Generator (TMG)

The TMG is a unique free-piston machine in many respects. Instead of a piston, this engine has a diaphragm which flexes to effect the change of working gas volume. In addition, the stiffness of the diaphragm acts as a mechanical spring.

The displacer, too, is supported on a mechanical spring, which is connected to the casing (figure 3.14). The motion of the displacer is not a direct result of the pressure forces but is a result of feedback from the casing motions. Fortunately, it is still possible to neglect the casing motions for the purposes of the linear analysis since the required casing amplitude is considerably smaller than both the diaphragm and displacer amplitudes. A disadvantage of this system is that it is necessary to limit the mechanical stress to below the fatigue limit for the diaphragm material, in order that infinite operating life be obtained. Thus, the amplitude may be typically only of the order of 1 mm or less. This usually requires that the machine operates at high frequencies to achieve any reasonable power output. However, since there is no sliding contact, the reciprocating elements do not suffer any static friction at rest and consequently the machine will self-start reliably. In many applications this feature may be the most important one. Another important feature is that there is no wear, so there is essentially no limitation to operating life.

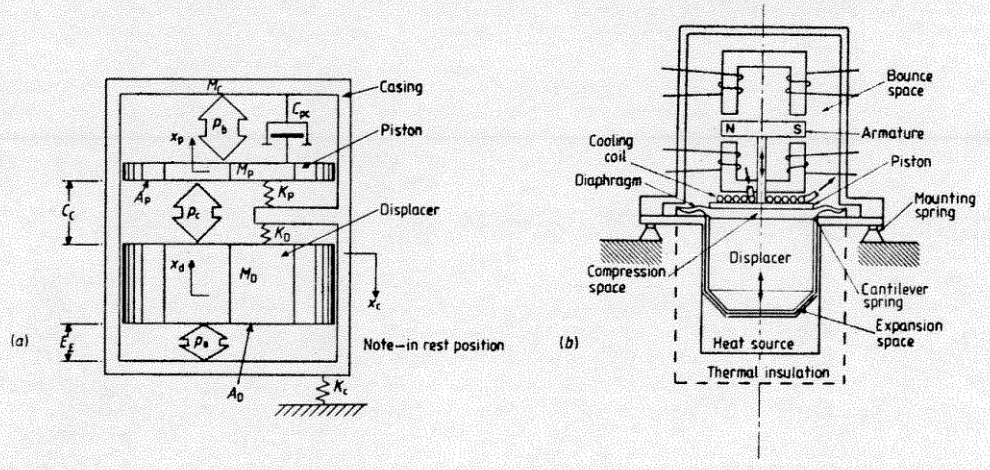


Figure 3.14 The Harwell Thermo-Mechanical Generator: (*a*) Notation; (*b*) Schematic.

A schematic view of the TMG is shown in figure 3.15.

In writing the equations of motion, the diaphragm will be indicated by the subscript p, since it is essentially the piston of the device. Thus, referring to figure 3.14, the equations of motion are

$$M_P\ddot{x}_p = A_P(p_c - p_b) - K_P(x_p + x_c) - C_{pc}(\dot{x}_p + \dot{x}_c) \tag{3.119}$$

$$M_D\ddot{x}_d = -K_D(x_d + x_c) + A_D\Delta p \tag{3.120}$$

$$M_C\ddot{x}_c = A_P(p_c + \Delta p - p_b) - K_P(x_c + x_p) - K_D(x_c + x_d) - K_C x_c - C_{pc}(\dot{x}_c + \dot{x}_p). \tag{3.121}$$

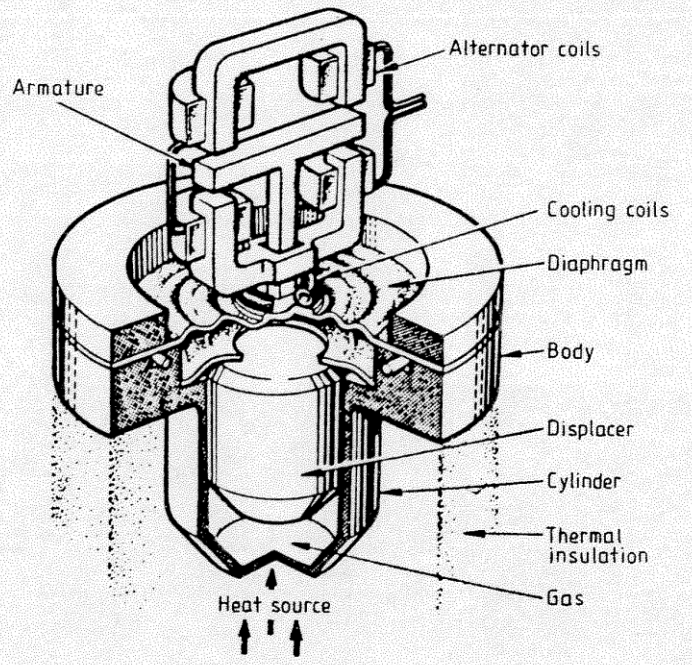


Figure 3.15 The Harwell Thermo-Mechanical Generator—schematic view (courtesy *The Chartered Mechanical Engineer*).

Owing to the generally large bounce space volume compared with the diaphragm (piston) stroke volume, the bounce space pressure may be taken as constant and the gas spring hysteresis neglected. The linear coefficients for this case are therefore:

$$S_{pp} = -\frac{1}{M_P}\left(\frac{p_{mean}A_P^2}{T_k S} + K_P\right) \qquad S_{pd} = p_{mean}\frac{A_P A_D}{M_P S}\left(\frac{1}{T_k} - \frac{1}{T_h}\right)$$

$$D_{pp} = -C_{pc}/M_P \qquad D_{pd} = 0$$

$$S_{dp} = 0 \qquad S_{dd} = -K_D/M_D$$

$$D_{dp} = C_p/M_D \qquad D_{dd} = C_d/M_D$$

$$S_{cc} = -\frac{1}{M_C}\left(\frac{p_{mean}A_P A_D}{T_h S} + K_P + K_D + K_C\right).$$

The frequency equation (from table 3.1) becomes

$$\omega^2 = (D_{dp}S_{pd} - D_{dd}S_{pp} - S_{dd}D_{pp})/(D_{dd} + D_{pp}). \qquad (3.122)$$

Typically, the regenerator on the TMG consists of a series of annular spaces connecting the hot and cold ends. The hot and cold heat exchangers are simply the actual working spaces of the machine. These types of heat exchangers tend to exhibit extremely small pressure drops. It is generally

possible, therefore, to neglect the effect of the heat exchanger pressure drop on the reciprocating elements. Thus we may set

$$D_{dp} \simeq 0 \text{ and } D_{dd} \simeq 0. \tag{3.123}$$

The frequency equation now becomes

$$\omega^2 = -S_{dd} \tag{3.124}$$

which is simply the natural frequency of the displacer, and is independent of the load.

From the geometric constraint equation it can be seen that the following must also hold

$$S_{pp} = S_{dd}. \tag{3.125}$$

Again from table 3.1, the phase and the amplitude ratio are simply

$$\phi = -90^\circ \tag{3.126}$$

$$r = \omega D_{pp}/S_{pd}. \tag{3.127}$$

Unfortunately, data for this engine are not available in the open literature. These data have, however, been made available to the authors on a confidential basis and may not, therefore, be published here. The performance characteristics indicated in figure 3.16 are generated from the data supplied. Actual

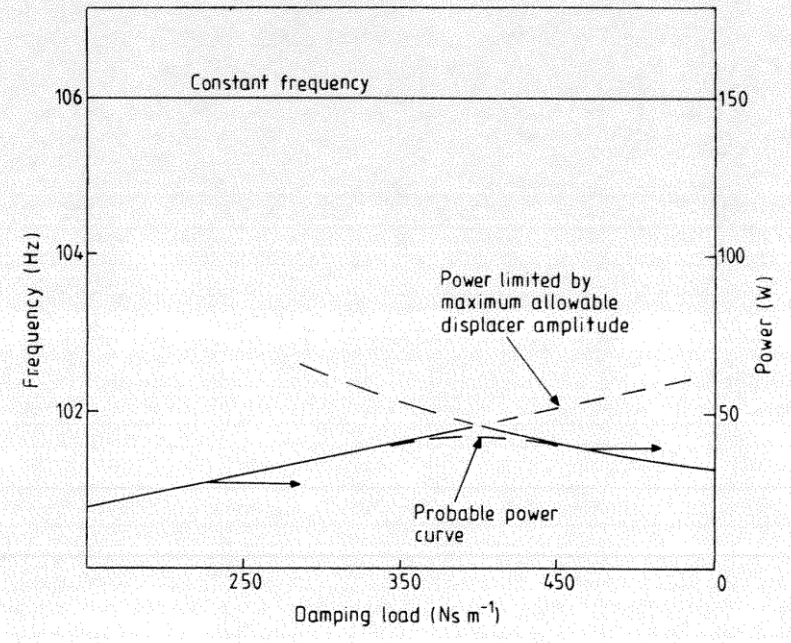


Figure 3.16 Frequency and power versus load for the Harwell Thermo-Mechanical Generator engine.

frequency and power output of the TMG are 106 Hz and 25 W electrical power respectively. Again, it can be seen that the linear analysis agrees with actual results to within an acceptable margin. Note that in this case the increase of power with load is limited at around 400 Ns m^{-1}, when the growth of the displacer amplitude becomes constrained by the geometry of the engine. At this point the displacer amplitude remains fixed at its maximum value and the piston amplitude begins to drop off with further load, thus reducing the power.

References

Anvoner S 1970 *Solution of Problems in Mechanics of Machines* vol 1 (London: Pitman)

Beale W T 1969 Free-piston Stirling Engines—Some Model Tests and Simulations *SAE Paper No 690230*

—— 1971 Stirling Cycle Type Thermal Device *US Patent 3 552 120*

—— 1979 A Free Cylinder Stirling Engine Solar Powered Water Pump *Int. Solar Energy Soc. Int. Congress*

Beale W T, Rankin C F and Wood J G 1975 *Free-piston Stirlilg engine–Inertia compressor prototype gas fired air conditioner* (American Gas Association)

Beale W T, Rauch J, Lewis R and Mulej D 1971 Free Cylinder Stirling Engines for Solar-Powered Water Pumps *ASME Paper No 71-WA/Sol-11*

Benson G M 1973 Thermal Oscillators *Proc. 8th IECEC, Philadelphia, August 13–17 1973* Paper No. 739076 pp 182–29

—— 1977 Thermal Oscillator *US Patent 4,044,558*

Berchowitz D M and Wyatt-Mair G F 1979 Closed-Form Polutions for a Coupled Ideal Analysis of Free-Piston Stirling Engines. *Proc. 14th IECEC, Boston*, pp 1114–9

Cooke-Yarborough E H 1967 A Proposal for a Heat-Powered Nonrotating Electrical Alternator *Harwell Memorandum AERE-M881 UK AERE*

—— 1970 Heat Engines *US Patent 3,548,589*

Cooke-Yarborough E H, Franklin E, Geisow J, Howlett R and West C D 1974 Harwell Thermo-Mechanical Generator *Proc. 9th IECEC, San Francisco August 26–30 1974* Paper No 749156 pp 1132–6

Gedeon D R 1978 The Optimisation of Stirling Cycle Machines, *Proc, 13th IECEC San Diego August 20–25 1978* Paper No. 789193 pp 1784–90

Goldberg L F 1979 *A Computer Simulation and Experimental Development of Liquid Piston Stirling Cycle Engines* MSc Dissertation, University of the Witwatersrand, South Africa

Goldberg L F and Rallis C H 1979 Prototype Liquid-Piston Free-Displacer Stirling Engine *Proc. 14th IECEC, Washington DC* Paper No 799239 pp 1103–7

Goldwater B and Morrow R B 1977 Demonstration of a Free-Piston Stirling Linear Alternator Power Conversion System. *Proc. 12th IECEC, Washington DC* Paper No 779249 pp 1488–95

Gupta S C and Hasdorff L 1970 *Fundamentals of Automatic Control* (Chichester: Wiley)

Martini W R 1975 The Free-Displacer, Free-Piston Stirling engine—Potential Energy Conserver, *Proc. 10th IECEC* Paper No 759149 pp 995–1002

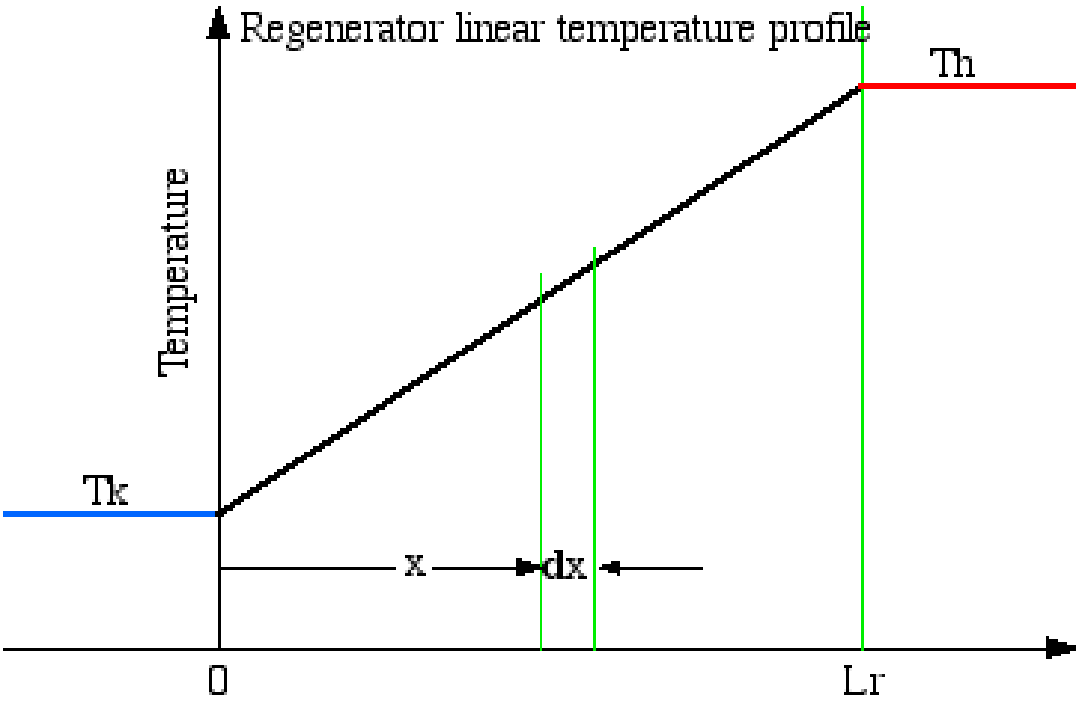
Reader G T 1979 Modes of Operation of a Jet-stream Fluidyne *Proc. 14th IECEC, Boston* Paper No 799238 pp 1098–102

Walker G 1980 *Stirling Engines* (Oxford: Oxford University Press)

Wood J G 1980 *A Program for Predicting the Dynamics of Free Piston Stirling Engines ME Thesis* Mech. Eng. Dept. Ohio University

Appendix F: Regenerator Mean Effective Temperature

In order to correctly evaluate the mass of gas in the regenerator void space, the lengthwise distribution of the gas temperature must be known. We will assume that the ideal regenerator has a linear temperature profile between the cold temperature Tk and the hot temperature Th, as shown in the following figure.



Thus the temperature profile in the regenerator can be described by

$T(x) = (Th - Tk)\ x / Lr + Tk$

where Lr is the regenerator length.

The total mass of gas mr in the regenerator void space Vr is given by

$$mr = \int_0^{Vr} \rho \, dVr$$

where ρ is the density. Now from the ideal gas equation of state, and for a constant free flow area Ar, we have

p = ρ R T

dVr = Ar **d**x

Vr = Ar Lr

Substituting for ρ, Vr, and **d**Vr in the above equation and simplifying

$$mr = \frac{Vr\ p}{R} \int_0^{Lr} \frac{1}{(Th - Tk)\ x + Tk\ Lr}\ \mathbf{d}x$$

Integrating and simplifying

$$mr = \frac{Vr\ p}{R} \frac{\mathbf{ln}(Th / Tk)}{(Th - Tk)}$$

We now define the mean effective temperature Tr of the gas in the regenerator in terms of the ideal gas equation of state

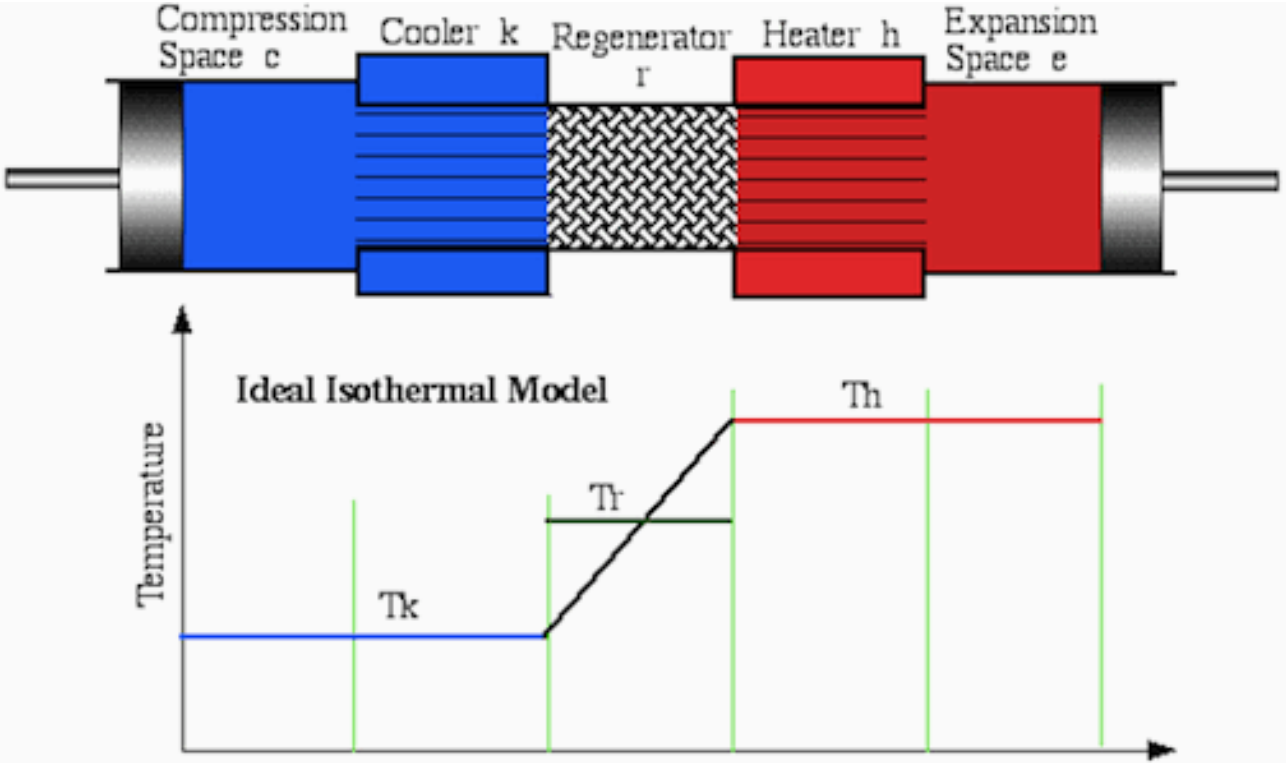
mr = Vr p / (R Tr)

Comparing the above two equations and equating Tr, we obtain

Tr = (Th - Tk) / **ln**(Th / Tk)

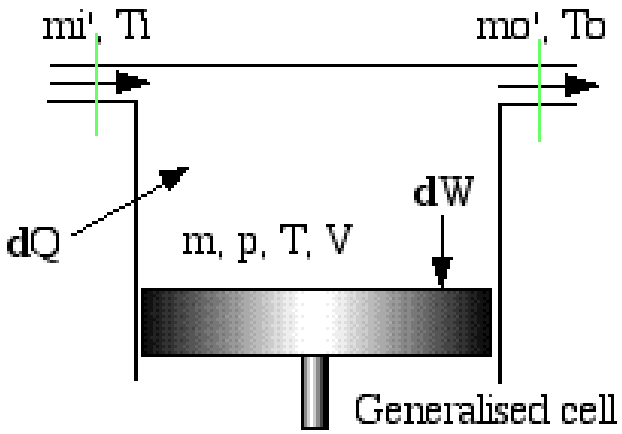
This equation gives the mean effective regenerator Tr as a function of Tk and Th, as required.

Appendix G: Energy Analysis - Ideal Isothermal Model



We now consider the Ideal Isothermal model from the point of view of energy flow. Heat transfer to and from the external environment occurs at the cold and hot temperatures Tk and Th respectively. Typically, Stirling engines will have separate heater and cooler heat exchangers, as indicated in the above figure. To investigate the heat transfer to these spaces, it is necessary to consider the working gas energy equation.

A generalised cell is shown below which may either be reduced to a working space cell or a heat exchanger cell. Enthalpy is transported into the cell by means of mass flow mi' and temperature Ti, and out of the cell by means of mass flow mo' and temperature To. The derivative operator is denoted by **d**, thus for example **d**m refers to the mass derivative **d**m/**d**θ, where θ is the cycle angle.



The word statement of the energy equation for the working gas in the generalized cell is

rate of heat transfer into the cell	+	net enthalpy convected into the cell	=	rate of work done on the surroundings	+	rate of increase of internal energy in the cell

Recall from **thermodynamics** that for an ideal gas, enthalpy and internal energy are functions of temperature only, thus:

specific enthalpy h = cp T

specific internal energy u = cv T

where cp and cv are the specific heat capacities of the gas at constant pressure and constant volume respectively.

Thus substituting in the above word statement we obtain

dQ + (cp Ti mi' - cp To mo') = **d**W + cv **d**(m T)

This equation is the well known classical form of the energy equation for non steady flow in which kinetic and potential energy terms have been neglected. In the isothermal model for the compression and expansion spaces, as well as for the heater and cooler, we have Ti = To = T, a constant value. Furthermore, from mass conservation considerations, the difference in the mass flow (mi' - mo') is equal to the rate of mass accumulation within the cell **d**m. The above equation thus simplifies to

dQ + cp T **d**m = **d**W + cv T **d**m

dQ = **d**W - R T **d**m

where for the assumed ideal gas, the gas constant R = cp - cv.

The net heat transferred to the working gas over the cycle Q is given by cyclic integration of **d**Q. However, the implicit assumption that cyclic steady state has been attained implies that the cyclic change in mass of the working gas (m) is zero for each of the cells. Thus applying the above equation to each of the isothermal cells and integrating over the cycle, we obtain for the working spaces

Qc = Wc

Qe = We

Similarly, for the heat exchanger spaces, in which no work is done

Qk = 0

Qh = 0

Of course, for the ideal regenerator Qr = 0. This is because all heat exchange between the regenerator matrix and the working gas is internal - there is no external heat transferred between the regenerator and the environment.

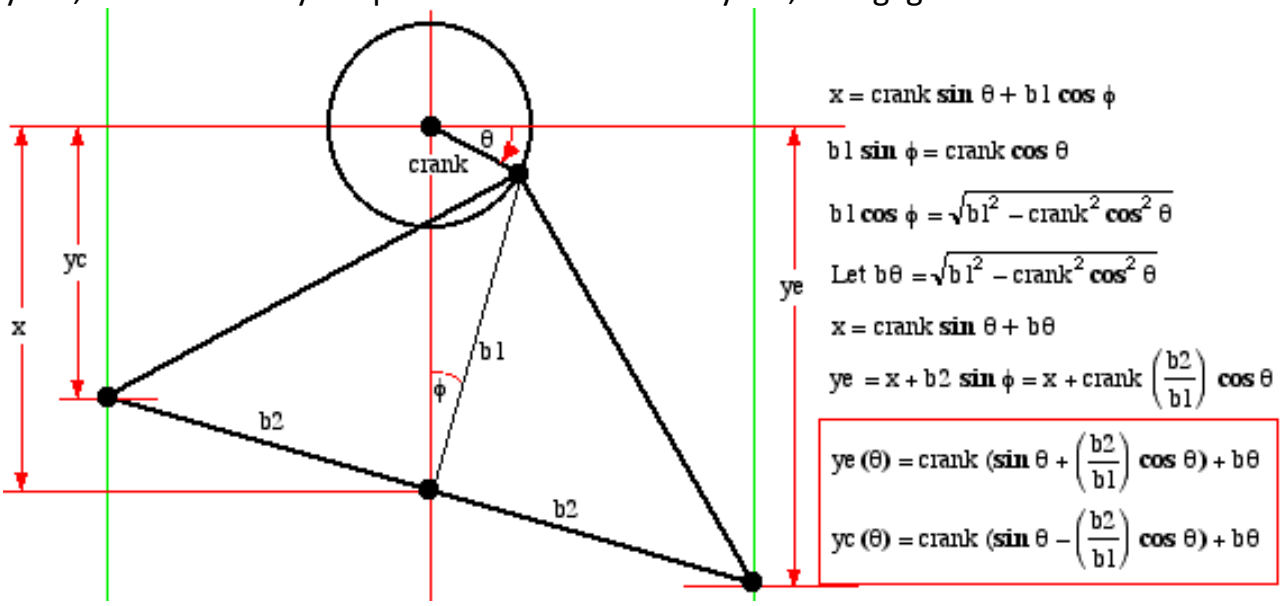
This is an amazing result - it implies that the heat exchangers are redundant, all the required external heat transfer occurring across the boundaries of the working spaces. This apparent paradox is a direct result of the isothermal model in which the compression and expansion spaces are maintained at the respective cooler and heater temperatures. Obviously this cannot be correct, since the cylinder walls are not designed for heat transfer. In real machines the working spaces will tend to be adiabatic rather than isothermal, which implies that the net heat transferred over the cycle must be provided by the heat exchangers.

We thus consider an alternative ideal model for Stirling cycle engines, the Ideal Adiabatic model, which will be developed in Chapter 4.

Appendix H: Volume variations - Ross Yoke-drive engine

The **Ross Yoke-drive engine** was invented by Andy Ross and is unique in that there is virtually no side movement of the two piston connecting rods, and thus no side forces on the pistons. **Matt Keveney** did an excellent animation of the **Ross Yoke-drive engine** showing clearly the principle of operation of this unique design.

In order to begin a simulation of the performance of the Ross Yoke-drive engine we need to derive equations for the volume variations Vc and Ve as well as the volume derivatives **d**Vc and **d**Ve as functions of the crankangle θ. The diagram below shows the two basic components of the Ross Yoke-drive - the crank and the yoke. The crankangle θ has a positive clockwise rotation starting from the horizontal crank arm position as shown. Let yc and ye respectively represent the displacements of the compression and expansion pistons, and x the displacement of the yoke pin at the center of the yoke base, all with respect to the crank axle. The basic assumption of the analysis is that the lateral movement of the connecting rods connected to the two extremeties of the yoke, as well as the yoke pin at the center of the yoke, is negligible.



$$x = \text{crank}\,\sin\theta + b1\,\cos\phi$$

$$b1\,\sin\phi = \text{crank}\,\cos\theta$$

$$b1\,\cos\phi = \sqrt{b1^2 - \text{crank}^2\,\cos^2\theta}$$

$$\text{Let } b\theta = \sqrt{b1^2 - \text{crank}^2\,\cos^2\theta}$$

$$x = \text{crank}\,\sin\theta + b\theta$$

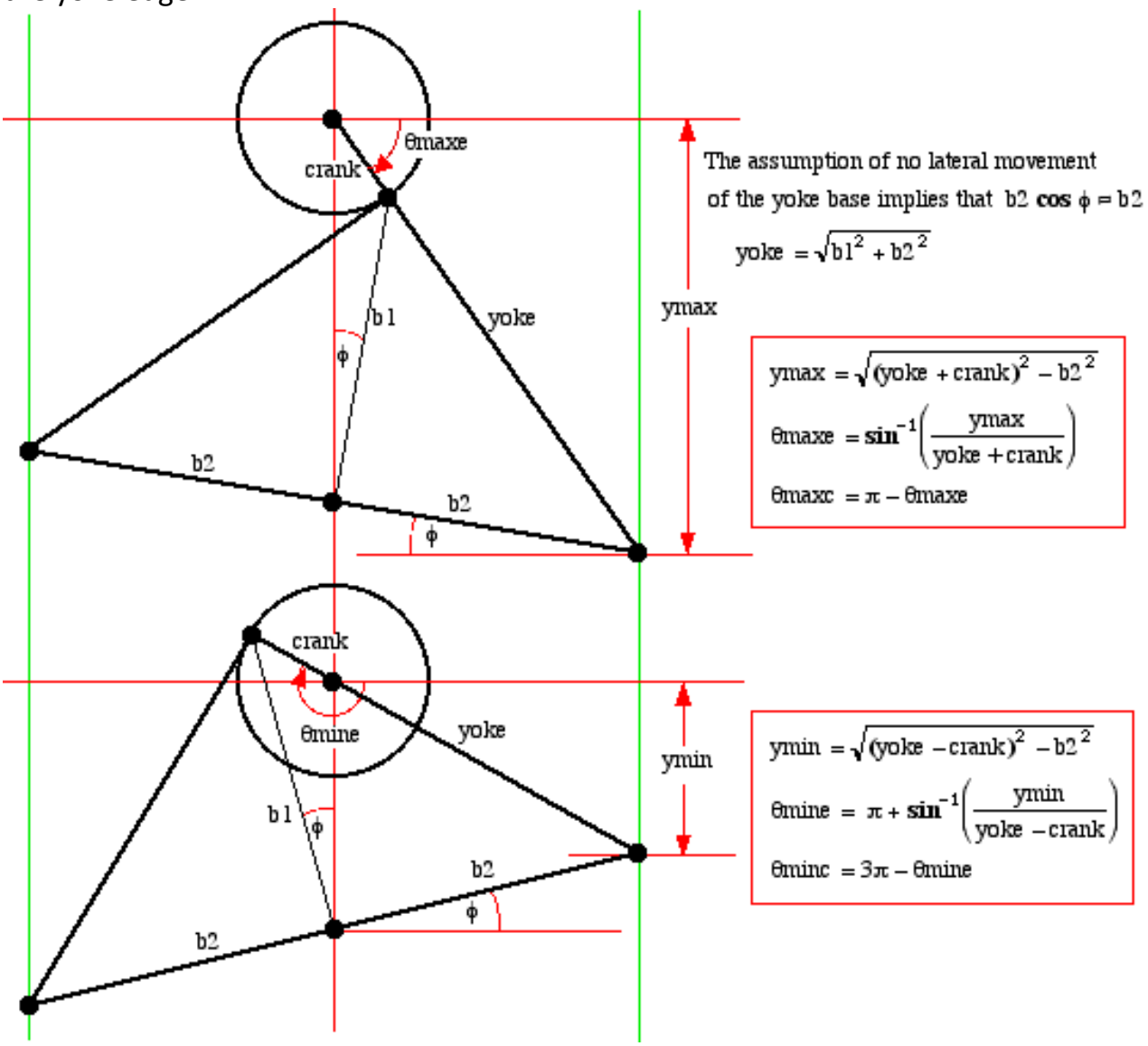
$$ye = x + b2\,\sin\phi = x + \text{crank}\left(\frac{b2}{b1}\right)\cos\theta$$

$$ye(\theta) = \text{crank}\left(\sin\theta + \left(\frac{b2}{b1}\right)\cos\theta\right) + b\theta$$

$$yc(\theta) = \text{crank}\left(\sin\theta - \left(\frac{b2}{b1}\right)\cos\theta\right) + b\theta$$

We now consider the minimum and maximum displacements of the compression and expansion pistons. Notice that they respectively occur when the crank arm lines up with

the yoke edge.



Thus for compression, expansion piston areas Ac, Ae, and clearence volumes Vclc, Vcle, the complete set of equations for evaluating the volumes Ve, Vc and the volume derivatives **d**Ve, **d**Vc are given by:

$$\text{Let } b\theta = \sqrt{b1^2 - \text{crank}^2 \cos^2\theta}$$

$$\text{ymin} = \sqrt{(\text{yoke} - \text{crank})^2 - b2^2}$$

$$ye = \text{crank}\left(\sin\theta + \left(\frac{b2}{b1}\right)\cos\theta\right) + b\theta$$

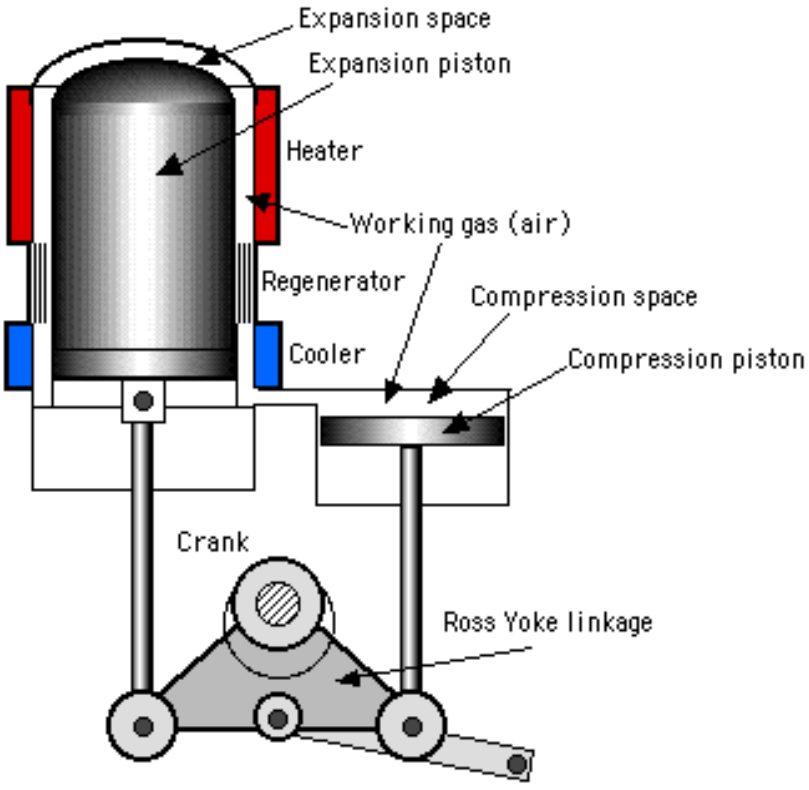
$$yc = \text{crank}\left(\sin\theta - \left(\frac{b2}{b1}\right)\cos\theta\right) + b\theta$$

$$Ve = Vcle + Ae\,(ye - \text{ymin})$$

$$Vc = Vclc + Ac\,(yc - \text{ymin})$$

$$dVe = Ae\ \text{crank}\left(\cos\theta - \left(\frac{b2}{b1}\right)\sin\theta + \frac{\text{crank}\sin\theta\cos\theta}{b\theta}\right)$$

$$dVc = Ac\ \text{crank}\left(\cos\theta + \left(\frac{b2}{b1}\right)\sin\theta + \frac{\text{crank}\sin\theta\cos\theta}{b\theta}\right)$$



Appendix I: Schmidt Analysis for Stirling Engines

SCHMIDT ANALYSIS FOR STIRLING ENGINES

This Analysis was written by David Berchowitz and Israel Urieli and published in their book "Stirling Cycle Engine Analysis". The analysis was written for engines but applies also to coolers. The difference between the two analysis is that the Rejector (k) in the engine is the lower temperature reject sink and in the cooler is the higher temperature reject sink. The Acceptor is the heat accept in both engine and cooler but corresponds to the lower temperature in the cooler and to the higher temperature in the engine.

Nomenclature		**Subscripts**	
T	Temperature	c	Compression Space
P	Pressure	e	Expansion Space
M	Mass	r	Regenerator
R	Gas Constant	k	Rejector
V	Volume	h	Acceptor
W	Work	sw	swept

A.1 The Schmidt analysis

A.1.1 Background

The apparent conceptual simplicity of the Stirling engine belies its intractability to mathematical analysis. The difficulty of describing even idealized models of the engine in terms of simple closed-form equations is one of the primary reasons for the widespread skepticism and lack of understanding which exists even today.

In Chapter 2 we derived the basic set of equations which describe the Ideal Isothermal model (table 2.1). Gustav Schmidt of the German Polytechnic Institute of Prague published an analysis in 1871 in which he obtained closedform solutions of these equations for the special case of sinusoidal volume variations of the working spaces (Schmidt 1871). This analysis is still used today as the classic Stirling cycle analysis. It was derived in order to describe the highly successful Lehmann engine shown in figure A. 1. 1. The paper includes a detailed description of the engine and displays a clear insight and appreciation of it. From figure A.1.1 we see that a very long horizontal

cylinder *ABC* was used, in which a concentric displacer L and power piston D reciprocate in accordance with a rather complex driving mechanism. A full size complete Lehmann engine is on permanent display at the Deutscher Museum in Munich, and a clear, modern description of the engine operation has been recently presented by Kolin (1972).

The driving mechanism does not produce sinusoidal motion. Schmidt derived the equivalent mechanism 'which by means of an imagined, infinitely long thrust-rod, would attain the true movement' (sinusoidal motion) of the working and displacer pistons.

The Lehmann engine piston seal was ingeniously constructed similarly to a bicycle pump, allowing limited pressurized operation. 'The working piston is isolated by means of a leather sleeve turned towards the inside. So long as the air inside the machine has a higher pressure than the outside atmosphere, this sleeve effectively prevents the escape of air towards the outside. However as soon as the pressure inside sinks below the ordinary atmospheric pressure, it permits the entrance of external air into the machine.

'The Lehmann arrangement has the advantage that the working piston is in constant contact only with the cooler air, thereby preventing the inward turned leather sleeve from burning!

Schmidt was acutely aware of the advantages of operating the cycle at a higher pressure, and further states. 'Undoubtedly this is the only system which holds any promise for the future, because with high pressure one can use lower temperatures and therefore produce a durable machine.'

Significantly, there is no mention throughout the paper of the importance of the regenerator, even though the use of regenerators in the earlier Ericsson machines was described. In figure A.1.1 we notice that about a third of the cylinder is enclosed by the furnace and the rest of the cylinder by a water jacket. 'The displacer leaves between itself and the wall of cylinder *A,* the intervening piece *B* and the heating pot C exactly so much room that the cross section of the circular intervening space is large enough to allow minimal resistance to the passage of the air, and small enough to produce a thin layer of air in order that heating and cooling may be achieved as rapidly as possible.' The Lehmann machine apparently was not fitted with a regenerator. Now, since over the cycle under cyclic steady conditions the net heat transferred to the regenerator is zero, it is conceivable that Schmidt did not appreciate the importance of the regenerator. He refers to the textbook by Zeuner as containing a 'complete, simple and

clear theory' of air engines, but in the same textbook Zeuner decries the use of regenerators for air engines (Finkelstein 1959).

A.1.2 The analysis

The approach taken by Schmidt for the analysis follows the Isothermal Analysis used in Chapter 2 quite closely, up to the derivation of the pressure relation given by equation (2.5), reproduced as follows:

$$p = MR\left(\frac{V_c}{T_k} + \frac{V_k}{T_k} + \frac{V_r \ln(T_h/T_k)}{(T_h - T_k)} + \frac{V_h}{T_h} + \frac{V_e}{T_h}\right)^{-1} \quad \text{(A. 1. 1)}$$

The sinusoidal volume variations are given as in equations (2.15) and

(2-16) as follows:

$$V_c = V_{clc} + V_{swc}(1 + \cos\theta)/2$$
$$V_e = V_{cle} + V_{swe}[1 + \cos(\theta + \alpha)]/2. \quad \text{(A.1.2) (A.1.3)}$$

Substituting (A.1.2) and (A.1.3) in (A.1.1) and simplifying we obtain

$$p = MR\left[s + \left(\frac{V_{swe}\cos\alpha}{2T_h} + \frac{V_{swc}}{2T_k}\right)\cos\theta - \left(\frac{V_{swe}}{2T_h}\sin\alpha\right)\sin\theta\right]^{-1} \quad \text{(A.1.4)}$$

where

$$s = \left[\frac{V_{swc}}{2T_k} + \frac{V_{clc}}{T_k} + \frac{V_k}{T_k} + \frac{V_r \ln(T_h/T_k)}{(T_h - T_k)} + \frac{V_h}{T_h} + \frac{V_{swe}}{2T_h} + \frac{V_{cle}}{T_h}\right].$$

Referring to figure *A.1.2* we consider the following trigonometric substitutions:

$$c\sin\beta = \frac{V_{swe}\sin\alpha}{2T_h} \quad \text{(A.1.5)}$$

$$c\cos\beta = \frac{V_{swe}\cos\alpha}{2T_h} + \frac{V_{swc}}{2T_k} \quad \text{(A.1.6)}$$

where

$$\beta = \tan^{-1}\left(\frac{V_{swe}\sin\alpha/T_h}{V_{swe}\cos\alpha/T_h + V_{swc}/T_k}\right) \qquad \text{(A.1.7)}$$

and

$$c = \frac{1}{2}\left[\left(\frac{V_{swe}}{T_h}\right)^2 + 2\frac{V_{swe}}{T_h}\frac{V_{swc}}{T_k}\cos\alpha + \left(\frac{V_{swc}}{T_k}\right)^2\right]^{1/2} \qquad \text{(A.1.8)}$$

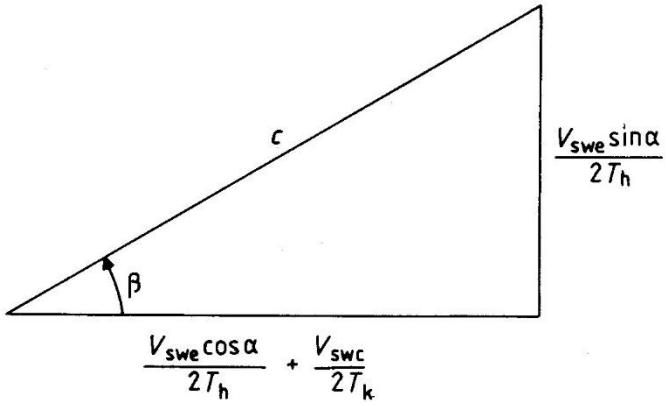


Figure A.1.2 Trigonometric substitutions.

Substituting equations *(A.1.5)* and *(A.1.6)* into equation *(A.1.4)* and simplifying, we obtain

$$p = \frac{MR}{s(1 + b\cos\phi)} \qquad \text{(A.1.9)}$$

where

$$\phi = \theta + \beta \qquad b = c/s.$$

Equation (A. 1.9) is the 'equation of the caloric line' and has essentially the same form as that derived by Schmidt. The maximum and minimum values of pressure are easily evaluated for the extreme values of cos Ø :

$$p_{max} = \frac{MR}{s(1-b)} \qquad \text{(A.1.10)}$$

$$p_{min} = \frac{MR}{s(1+b)}. \qquad \text{(A.1.11)}$$

The average pressure over the cycle is given by

$$p_{\text{mean}} = \frac{1}{2\pi}\int_0^{2\pi} p\,\mathrm{d}\phi$$
$$= \frac{MR}{2\pi s}\int_0^{2\pi} \frac{1}{(1+b\cos\phi)}\,\mathrm{d}\phi. \qquad \text{(A.1.12)}$$

From tables of integrals (Dwight 1961), equation (A.1.12) reduces to

$$p_{\text{mean}} = MR/(s\sqrt{1-b^2}). \qquad \text{(A.1.13)}$$

Equation (A. 1.13) is the most convenient method of relating the total mass of working gas to the more conveniently specified mean operating pressure and is used for this purpose throughout this book.

Work is done by the engine on the surroundings by virtue of the varying volumes of the working spaces Vc and Ve The total work done by the engine is therefore the algebraic sum of the work done by the compression and expansion spaces. Over a complete cycle we have

$$W_c = \oint p\,\mathrm{d}V_c = \int_0^{2\pi} p\frac{\mathrm{d}V_c}{\mathrm{d}\theta}\,\mathrm{d}\theta \qquad \text{(A.1.14)}$$

$$W_e = \oint p\,\mathrm{d}V_e = \int_0^{2\pi} p\frac{\mathrm{d}V_e}{\mathrm{d}\theta}\,\mathrm{d}\theta \qquad \text{(A.1.15)}$$

$$W = W_c + W_e. \qquad \text{(A.1.16)}$$

Differentiating equations (A.1.2) and (A.1.3), the volume derivatives are

$$\frac{\mathrm{d}V_c}{\mathrm{d}\theta} = -\tfrac{1}{2}V_{swc}\sin\theta \qquad \text{(A.1.17)}$$

$$\frac{\mathrm{d}V_e}{\mathrm{d}\theta} = -\tfrac{1}{2}V_{swe}\sin(\theta+\alpha). \qquad \text{(A.1.18)}$$

Substituting equations (A.1.17), (A.1.18) and (A.1.9) into equations (A.1.14) and (A.1.15), we obtain

$$W_c = -\frac{V_{swc}MR}{2s}\int_0^{2\pi}\frac{\sin\theta}{1+b\cos(\beta+\theta)}\,\mathrm{d}\theta \qquad \text{(A.1.19)}$$

$$W_e = -\frac{V_{swe}MR}{2s}\int_0^{2\pi}\frac{\sin(\theta+\alpha)}{1+b\cos(\beta+\theta)}\,\mathrm{d}\theta. \qquad \text{(A.1.20)}$$

The following approach to the solution of integrals (A. 1.19) and (A. 1.20) is somewhat different from that due to Schmidt. However, it is considered by the authors to be more easily comprehended.

The Fourier series expansion of the pressure function is first considered. It is shown that only one of the terms of this expansion will return a non-zero integral. This integral is then evaluated, giving the exact solution.

The Fourier series expansion of p (0) in equation (A. 1.9) is given as follows (Arfken 1970):

$$p(\phi) = p_0 + \sum_{i=1}^{\infty} [p_{ci}\cos(i\phi) + p_{si}\sin(i\phi)] \qquad \text{(A.1.21)}$$

where

$$p_0 = \frac{1}{2\pi}\int_0^{2\pi} p(\phi)\,\mathrm{d}\phi$$

$$p_{ci} = \frac{1}{\pi}\int_0^{2\pi} p(\phi)\cos(i\phi)\,\mathrm{d}\phi$$

$$p_{si} = \frac{1}{\pi}\int_0^{2\pi} p(\phi)\sin(i\phi)\,\mathrm{d}\phi.$$

Now, referring to the graph of equation (A.1.9) for a typical value of b (figure A. 1.3) we observe that p (Ø) is an even function of Ø and can thus be represented exclusively by the cosine terms. Equation (A. 1.2 1) thus reduces to

$$p(\phi) = p_0 + \sum_{i=1}^{\infty} p_{ci}\cos(i\phi). \qquad \text{(A.1.22)}$$

Substituting equations (A.1.22) and (A.1.17) into (A.1.14) we obtain

$$W_c = -\frac{V_{swc}}{2}\int_0^{2\pi}\left(p_0 + \sum_{i=1}^{\infty} p_{ci}\cos(i\phi)\right)\sin\theta\,\mathrm{d}\theta. \qquad \text{(A.1.23)}$$

Expanding equation (A.1.23)

$$W_c = -\frac{V_{swc}p_0}{2}\int_0^{2\pi}\sin\theta\,\mathrm{d}\theta - \frac{V_{swc}}{2}\sum_{i=2}^{\infty} p_{ci}\int_0^{2\pi}\cos[i(\theta+\beta)]\sin\theta\,\mathrm{d}\theta - \frac{V_{swc}p_{c1}}{2}\int_0^{2\pi}\cos(\theta+\beta)\sin\theta\,\mathrm{d}\theta. \qquad \text{(A.1.24)}$$

It can easily be shown that the first two terms on the right-hand side of equation (A.1.24) are zero, resulting in

$$W_c = -\frac{V_{swc} p_{c1}}{2} \int_0^{2\pi} \cos(\theta + \beta) \sin\theta \, d\theta. \qquad \text{(A.1.25)}$$

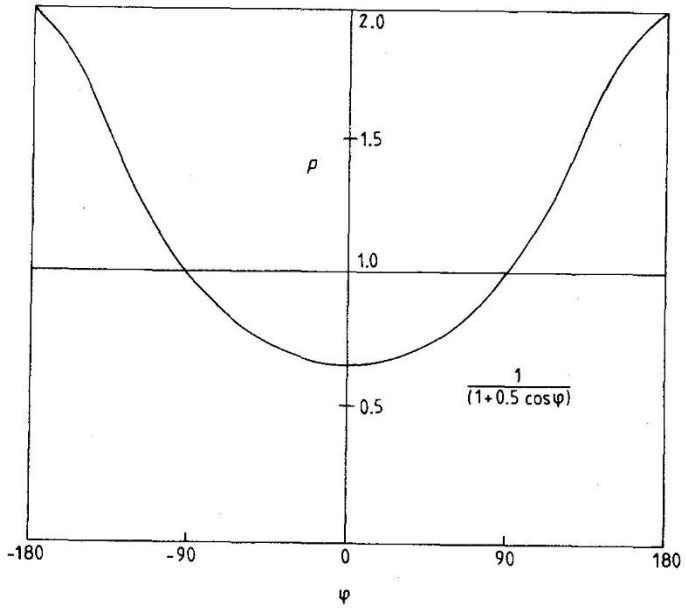


Figure A.1.3 Normalized pressure p versus composite angle 0.

Evaluating the integral of equation (A.1.25)

$$W_c = \tfrac{1}{2}\pi V_{swc} p_{c1} \sin\beta. \qquad \text{(A.1.26)}$$

Similarly, for the expansion space we find

$$W_e = \tfrac{1}{2}\pi V_{swe} p_{c1} \sin(\beta - \alpha). \qquad \text{(A.1.27)}$$

Now, from equations (A.1.21) and (A.1.9)

$$p_{c1} = \frac{MR}{\pi s} \int_0^{2\pi} \frac{\cos\phi}{(1 + b\cos\phi)} d\phi. \qquad \text{(A.1.28)}$$

Equation (A.1.28) can be evaluated in two steps using tables of integrals, as follows (Dwight 1961):

$$p_{c1} = \frac{MR}{\pi s}\left(\frac{2\pi}{b} - \frac{1}{b}\int_0^{2\pi} \frac{1}{(1 + b\cos\phi)}\,d\phi\right)$$
$$= \frac{MR}{\pi s}\left(\frac{2\pi}{b} - \frac{2\pi}{b\sqrt{1-b^2}}\right)$$
$$= \frac{2MR}{sb}\left(1 - \frac{1}{\sqrt{1-b^2}}\right). \qquad \text{(A.1.29)}$$

Substituting equations (A.1.29) and (A.1.13) into equations (A.1.26) and (A.1.27) we finally obtain

$$W_c = \pi V_{swc} p_{mean} \sin\beta(\sqrt{1-b^2} - 1)/b \qquad \text{(A.1.30)}$$
$$W_e = \pi V_{swe} p_{mean} \sin(\beta - \alpha)(\sqrt{1-b^2} - 1)/b. \qquad \text{(A.1.31)}$$

Equations (A.1.30) and (A.1.31) are essentially the same results as those obtained by Schmidt, and constitute the major analytical results of the analysis.

Now, since the Schmidt analysis is based on the Ideal Isothermal model, the thermal efficiency should reduce to the Carnot efficiency. The thermal efficiency is defined by the ratio of the work done by the engine to the heat supplied externally to the engine. In Chapter 2 we showed that the heat supplied externally is equal to the work done by the expansion space (equations (2.12) and (2.13)) thus:

$$\eta = W/W_e = (W_c + W_e)/W_e. \qquad \text{(A.1.32)}$$

Substituting equations (A.1.30) and (A.1.31) into equation (A.1.32)

$$\eta = 1 + \frac{V_{swc}\sin\beta}{V_{swe}\sin(\beta - \alpha)}. \qquad \text{(A.1.33)}$$

Expanding equation (A.1.33) and simplifying

$$\eta = 1 - \frac{V_{swc}}{V_{swe}}\left(\frac{\tan\beta}{\sin\alpha - \tan\beta\cos\alpha}\right). \qquad \text{(A.1.34)}$$

Substituting equation (A.1.7) into equation (A.1.34)and simplifying, we obtain

$$\eta = 1 - T_k/T_h \qquad \text{(A.1.35)}$$

which is the Carnot efficiency.

A.2 Regenerator mean effective temperature

In order to evaluate the total mass of gas in the regenerator void space correctly, the lengthwise distribution of the gas temperature must be known. It has been shown that for real regenerators, the temperature profile is very nearly linear (Urieli 1980), and thus we assume that the ideal regenerator has a linear temperature profile between the cold temperature T, and the hot temperature *Th,* as in figure A.2.1 (Creswick 1965). From figure A.2.1 we observe

$$T(x) = (T_h - T_k)x/L_r + T_k \qquad \text{(A.2.1)}$$

where *L,* is the regenerator length.

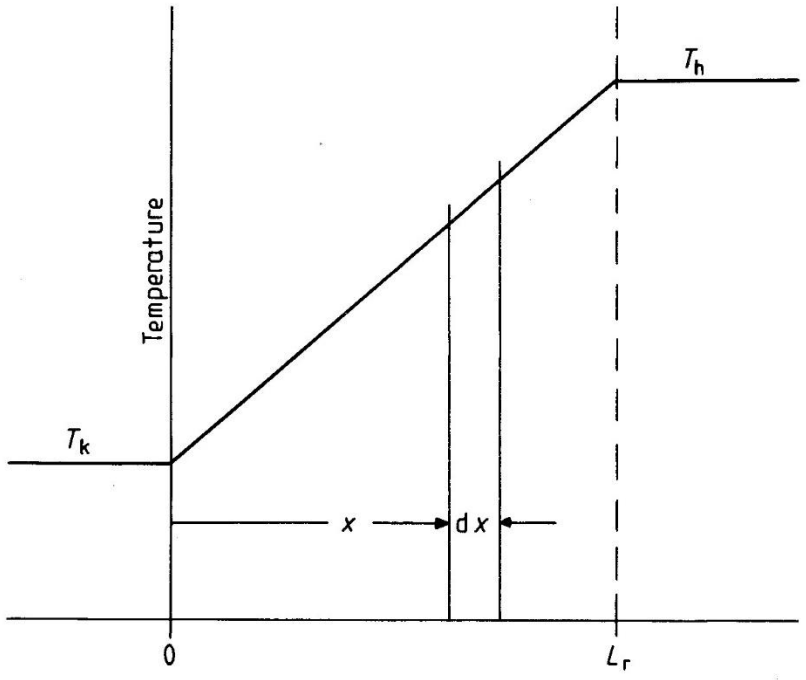


Figure A.2.1 Regenerator linear temperature profile.

The total mass of gas m, in the regenerator void volume is given by

$$m_r = \int_0^{V_r} \rho \, dV_r \qquad \text{(A.2.2)}$$

where p is the density, d V, = A, dx is the lengthwise volume derivative for constant free-flow area *A,* and *Vr =*

ArL,.

Substituting for the ideal gas law p = pRT in equation (A.2.2) and simplifying

$$m_r = \frac{V_r p}{R} \int_0^{L_r} \frac{1}{[(T_h - T_k)x + T_k L_r]} dx. \qquad (A.2.3)$$

Integrating the right-hand side of equation (A.2.3) and simplifying

$$m_r = \frac{V_r p}{R} \frac{\ln (T_h/T_k)}{(T_h - T_k)}. \qquad (A.2.4)$$

We define the mean effective regenerator temperature T, in terms of the ideal gas equation of state:

$$m_r = V_r p/(RT_r). \qquad (A.2.5)$$

Comparing equations (A.2.4) and (A.2.5) we obtain

$$T_r = (T_h - T_k)/\ln(T_h/T_k). \qquad (A.2.6)$$

Equation (A.2.6) gives the mean effective regenerator temperature. Tr as a function of Tk and Th, as required.

References

Arfken G 1970 Mathematical Methods for Physicists 2nd edn (New York: Academic)
Creswick F A 1965 Thermal design of Stirling cycle machines Int. Auto. Eng. Congr. Detroit, SAE 650079

Dwight H B 1961 Tables of integrals and other mathematical data 4th edn (New York: Macmillan)

Finkelstein T 1959 Air Engines in The Engineer part 1, 27 March

Kolin 11972 The Evolution of the Heat Engine in Thermodynamics Atlas 2 (London: Longman)

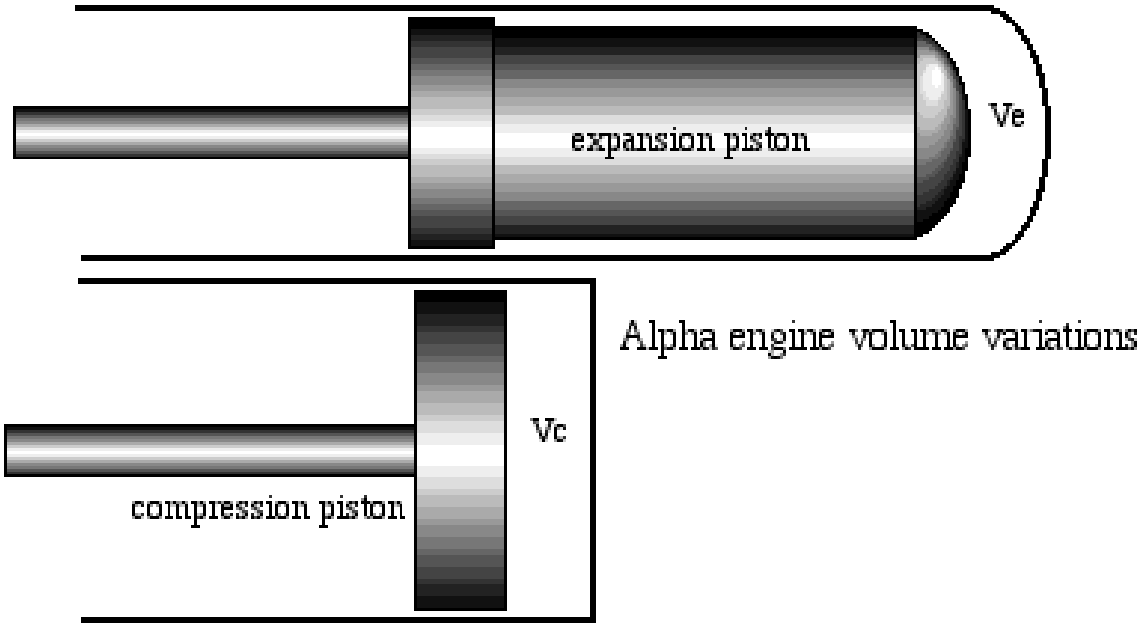
Schmidt G 1871 The Theory of Lehmann's Calorimetric Machine Z. ver. Dtsch. Ing. 15 part I

Urieli I 1980 A general purpose program for Stirling engine simulation Proe. l5th IECEC, Seattle, Washington Paper 809336 pp 1701-5

Appendix J: Sinusoidal Volume Variations

In order to evaluate the performance of a Stirling engine, one must first determine the volume variations of the compression and expansion spaces with respect to the crankangle θ. Gustav Schmidt of the German Polytechnic Institute of Prague Published an analysis in 1871 in which he obtained closed form solutions of the Ideal Isothermal model for the special case of sinusoidal volume variations of the working spaces with respect to the cycle angle θ. In this section we wish to develop the sinusoidal volume relations Vc(θ) and Ve(θ) for Alpha, Beta and Gamma type engines.

The Alpha type engine is the simplest of all three types in that there are two pistons, each independently controlling the respective working space. Thus the expansion piston only affects the volume variation of the expansion space Ve, and the compression piston only affects the volume variation of the compression space Vc. In the following figures we consider only the working spaces and not the interconnecting heat exchanger spaces.



Alpha engine volume variations

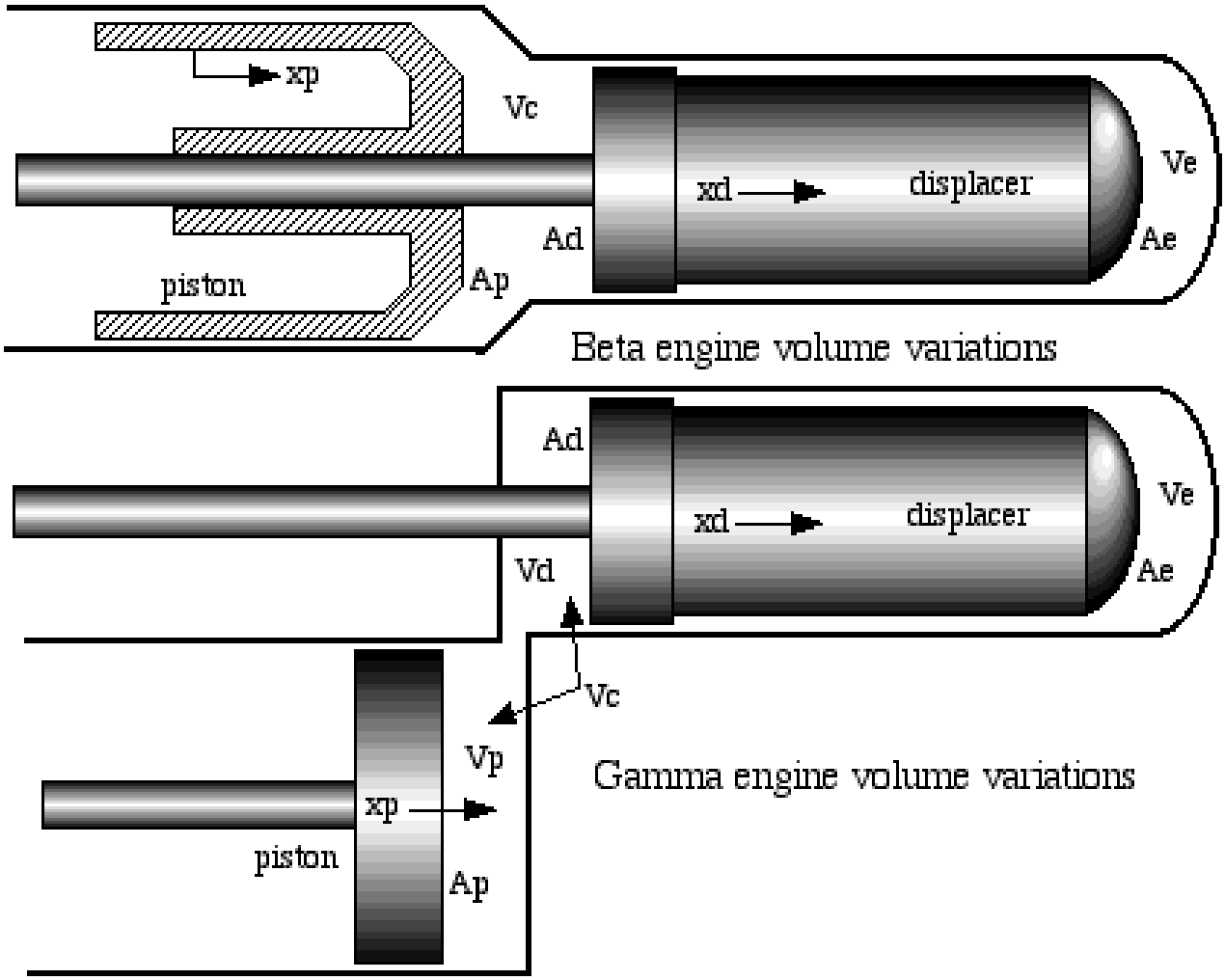
The sinusoidal volume variations of the compression and expansion spaces are respectively as follows:

$$Vc = Vclc + Vswc\ (1 + \mathbf{cos}\ \theta) / 2$$

$$Ve = Vcle + Vswe\ (1 + \mathbf{cos}(\theta + \alpha)) / 2$$

where Vcl and Vsw represent respectively clearence and swept volumes, and θ is the cycle angle. The constant angle α represents the phase advance of the expansion space volume variations with respect to the compression space volume variations.

Both Beta and Gamma type machines have more complex relations in that the movement of both the piston and the displacer affects the volume variation of the compression space Vc.



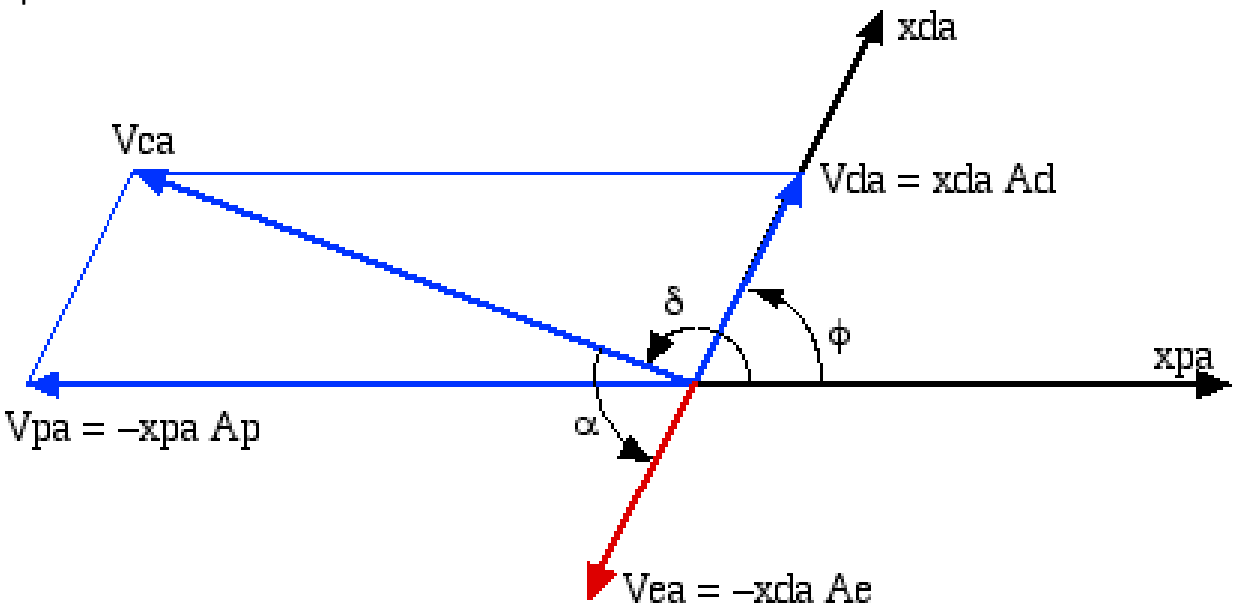
For the analysis that follows, we define the following variables:

Vc0 / Ve0 is the volume of the compression / expansion space (including clearence spaces) when both piston and displacer are in their mean positions.
Vca / Vea is the volume amplitude of the compression / expansion space. Thus Vswc = 2 Vea, Vswe = 2 Vea, Vclc = Vc0 - Vca, and Vcle = Ve0 - Vea.
Vpa / Vda is the volume amplitude in the compression space due to movement of the

piston / displacer.
Ap / Ad is the cross sectional area of the piston / displacer in the compression space, i.e. the respective cylinder area minus the cross sectional area of the displacer rod.
Ae is the cross sectional area of the displacer in the expansion space.
xpa / xda is the amplitude of movement of the piston / displacer.

In the phasor diagram following

ϕ is the phase advance of the displacer with respect to the piston,
δ is the phase advance of the compression space volume with respect to the piston, and
α is the phase advance of the expansion space volume with respect to the compression space volume.

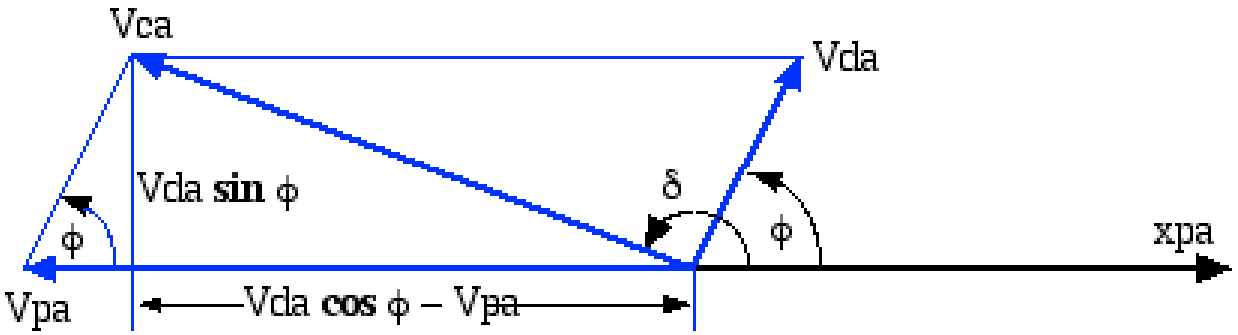


Vc = Vc0 - Vpa **cos** θ + Vda **cos**(θ + ϕ)

Expanding, we have

Vc = Vc0 + (Vda **cos** ϕ - Vpa) **cos** θ - (Vda **sin** ϕ)**sin** θ

Now consider the following phasor diagram



Vc = Vc0 + Vca **cos**(θ + δ)

Vc = Vc0 + Vca (**cos** θ **cos** δ - **sin** θ **sin** δ)

Equating like terms in the above equations for Vc, we obtain

Vca **cos** δ = Vda **cos** ϕ - Vpa

Vca **sin** δ = Vda **sin** ϕ

Thus finally

$$\delta = \arctan\left(\frac{\text{Vda}\,\sin\phi}{\text{Vda}\,\cos\phi - \text{Vpa}}\right)$$

$$\text{Vca} = \sqrt{\text{Vpa}^2 - 2\ \text{Vpa}\ \text{Vda}\cos\phi + \text{Vda}^2}$$

Vc = Vc0 + Vca **cos**(θ + δ) = Vclc + Vswc (1 + **cos**(θ + δ)) / 2

Ve = Ve0 + Vea **cos**(θ + δ + α) = Vcle + Vswe (1 + **cos**(θ + δ + α) / 2

where α = π + ϕ - δ

Appendix K: The Define Function

The purpose of the define function is to invoke the functions which specify the values of all the required global variables subsequently used in the Schmidt analysis, Ideal Adiabatic or Simple simulations. The data is either entered from a keyboard creating a new data file or entered from an previously created data file.

```
function define
% define the stirling engine geometric
% and operational parameters
% Israel Urieli 4/1/02 (April Fool's Day)
% Modified 2/12/2010 to include no-matrix regenerator awgr0
% Modified 7/10/2016 to include betadrive engines
clc;
clear all;
% The set of global variables defined are:
% engine
global engine_type % s)inusoidal, y)oke, r)ockerV, b)etadrive
global vclc vcle % compression,expansion clearence vols [m^3]
global vswc vswe % compression, expansion swept volumes [m^3]
global alpha % phase angle advance of expansion space [radians]
global b1 % Ross yoke length (1/2 yoke base) [m]
global b2 % Ross yoke height [m]
global crank % crank radius [m]
global dcomp dexp % diameter of compression/expansion pistons [m]
global acomp aexp % area of compression/expansion pistons [m^2]
global ymin % minimum yoke vertical displacement [m]
global conrodc conrode % length of comp/exp piston connecting rods [m]
global ycmax yemax % maximum comp/exp piston vertical displacement [m]
% heatex/cooler
global vk % cooler void volume [m^3]
global ak % cooler internal free flow area [m^2]
global awgk % cooler internal wetted area [m^2]
global dk % cooler hydraulic diameter [m]
global lk % cooler effective length [m]
% heatex/heater
global vh % heater void volume [m^3]
```

```
global ah % heater internal free flow area [m^2]
global awgh % heater internal wetted area [m^2]
global dh % heater hydraulic diameter [m]
global lh % heater effective length [m]
% heatex/regenerator
global lr % regenerator effective length [m]
global cqwr % regenerator housing thermal conductance [W/K]
global matrix_type % m)esh f)oil n}o matrix
global vr % regen void volume [m^3]
global ar % regen internal free flow area [m^2]
global awgr0 % no matrix regenerator wetted area [m^2]
global awgr % regen internal wetted area [m^2]
global dr % regen hydraulic diameter [m]
% gas
global rgas % gas constant [J/kg.K]
global cp % specific heat capacity at constant pressure [J/kg.K]
global cv % specific heat capacity at constant volume [J/kg.K]
global gama % ratio: cp/cv
global mu0 % dynamic viscosity at reference temp t0 [kg.m/s]
global t0 t_suth % reference temp. [K], Sutherland constant [K]
global prandtl % Prandtl number
% operat
global pmean % mean (charge) pressure [Pa]
global tk tr th % cooler, regenerator, heater temperatures [K]
global freq omega % cycle frequency [herz], [rads/s]
global mgas % total mass of gas in engine [kg]
% new data file
global new fid
new = input('Create a new data file? (y/n)','s');
if strncmp(new,'y',1)
   filename = input('enter new filename: ','s');
   fid = fopen(filename,'w');
else
   fid = 0;
   while fid < 1
      filename = input('open filename: ','s');
      [fid, message] = fopen(filename,'r');
      if fid == -1
```

```
        display(message)
        display('press ^C to exit')
      end
    end
end
engine
heatex
gas
operat
status = fclose(fid);
```

Appendix L: The Engine Function

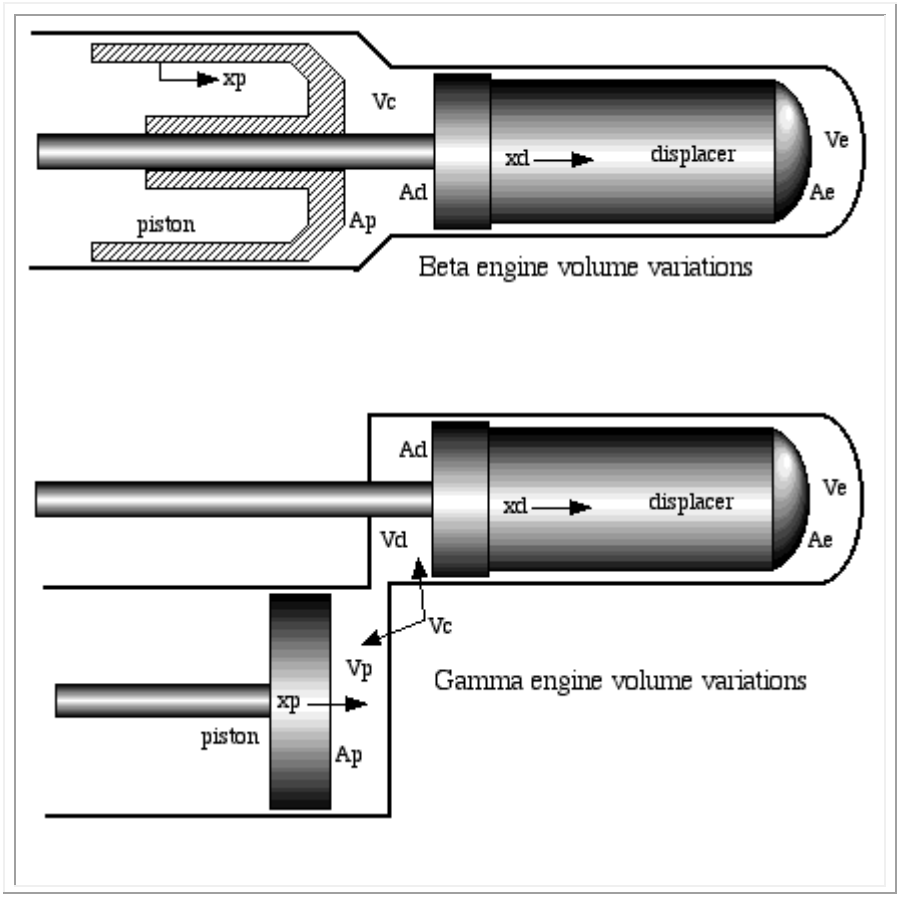
The purpose of the engine function is to select the engine type and invoke the function which specifies the values of the global variables relevant to the working space (compression and expansion) volumes of the selected engine type. We have currently included three alpha engines as case studies - the classic **Ford-Philips 4-215**, the **D-90 Ross Yoke-drive,** and the **Ross Rocker-V** drive engine. Note that the Ford-Philips engine has a swashplate drive mechanism, which has the classical sinusoidal motion of the working spaces.

```
function engine
% Define engine configuration and drive geometric parameters.
% Israel Urieli 4/14/02
% Modified 6/14/2016 to include free-piston beta engine
% s)inusoidal, y)oke r)ocker-V (all alpha engines)
global engine_type
global new fid % new data file
engine_type = 'u';
while(strncmp(engine_type,'u',1))
   if(strncmp(new,'y',1))
      fprintf('Available engine types are:\n');
      fprintf('   s)inusoidal drive\n');
      fprintf('   y)oke drive (Ross)\n');
      fprintf('   r)ocker-V drive (Ross)\n');
      fprintf('   b)eta drive (Free piston)\n');
      engine_type = input('enter engine type ','s');
      fprintf(fid, '%c\n', engine_type(1));
   else
      engine_type = fscanf(fid, '%c',1);
   end
   if(strncmp(engine_type,'s',1))
      sindrive;
   elseif(strncmp(engine_type,'y',1))
      yokedrive;
   elseif(strncmp(engine_type,'r',1))
      rockerVdrive;
   elseif(strncmp(engine_type,'b',1))
      betadrive;
   else
      fprintf('engine type is undefined\n')
      engine_type = 'u';
   end
end
%============================================================
```

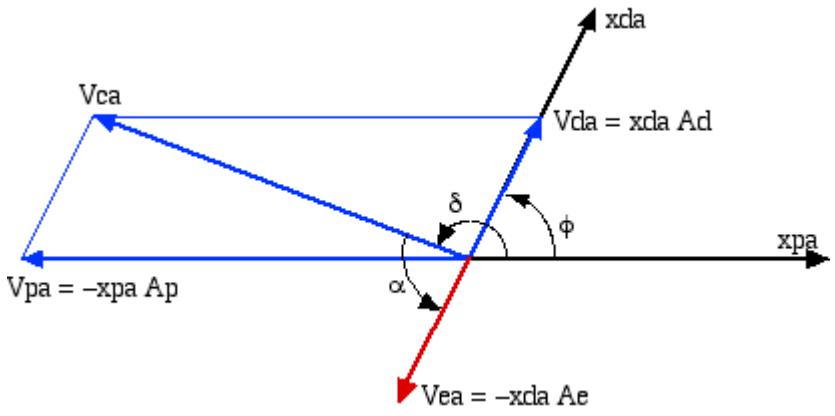
Update 2016: We have recently included a **Free-piston beta drive** engine assuming that both the piston and displacer motions are sinusoidal.

Free Piston Beta drive engine

Referring to the section on **Sinusoidal Volume Variations** (Appendix J) we find that both Beta and Gamma type machines have more complex relations in that the movement of both the piston and the displacer affects the volume variation of the compression space Vc.



In the phasor diagram following:
ϕ is the phase advance of the displacer with respect to the piston,
δ is the phase advance of the compression space volume with respect to the piston, and
α is the phase advance of the expansion space volume with respect to the compression space volume.



In the following MATLAB program we assume that both the piston and displacer motions are sinusoidal.

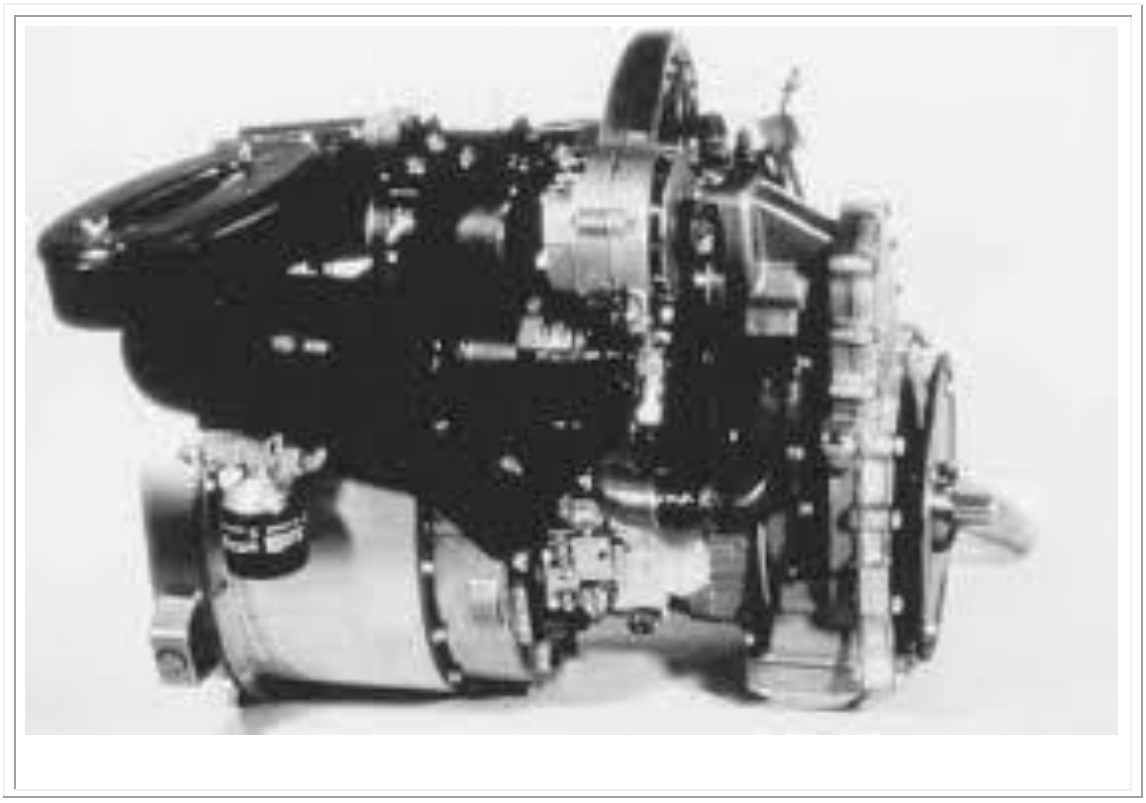
```
function betadrive
% Beta drive engine configuration
% Israel Urieli 1/22/08
global vclc vcle % compression,expansion clearence vols [m^3]
global vswc vswe % compression, expansion swept volumes [m^3]
global alpha % phase angle advance of expansion space [radians]
global new fid % new data file
fprintf('beta drive engine configuration\n')
if (strncmp(new,'y',1))
    xpa = input('enter piston amplitude (m): ');
    xda = input('enter displacer amplitude (m): ');
    phid = input('enter displacer phase angle advance [degrees]: ');
    dp = input('enter piston diameter (m): ');
    dd = input('enter displacer diameter (m): ');
    dr = input('enter displacer rod diameter (m): ');
    vclc = input('enter compression space clearance volume [m^3]: ');
    vcle = input('enter expansion space clearance volume [m^3]: ');
    fprintf(fid, '%.3e\n', xpa);
    fprintf(fid, '%.3e\n', xda);
    fprintf(fid, '%.1f\n', phid);
    fprintf(fid, '%.3e\n', dp);
    fprintf(fid, '%.3e\n', dd);
    fprintf(fid, '%.3e\n', dr);
    fprintf(fid, '%.3e\n', vclc);
    fprintf(fid, '%.3e\n', vcle);
else
    xpa = fscanf(fid,'%e',1);
    xda = fscanf(fid,'%e',1);
    phid = fscanf(fid,'%f',1);
    dp = fscanf(fid,'%e',1);
    dd = fscanf(fid,'%e',1);
    dr = fscanf(fid,'%e',1);
    vclc = fscanf(fid,'%e',1);
    vcle = fscanf(fid,'%e',1);
end
ap = pi*dp*dp/4; % piston area (m^2)
ad = pi*dd*dd/4; % displacer area (m^2)
ar = pi*dr*dr/4; % displacer rod area (m^2)
```

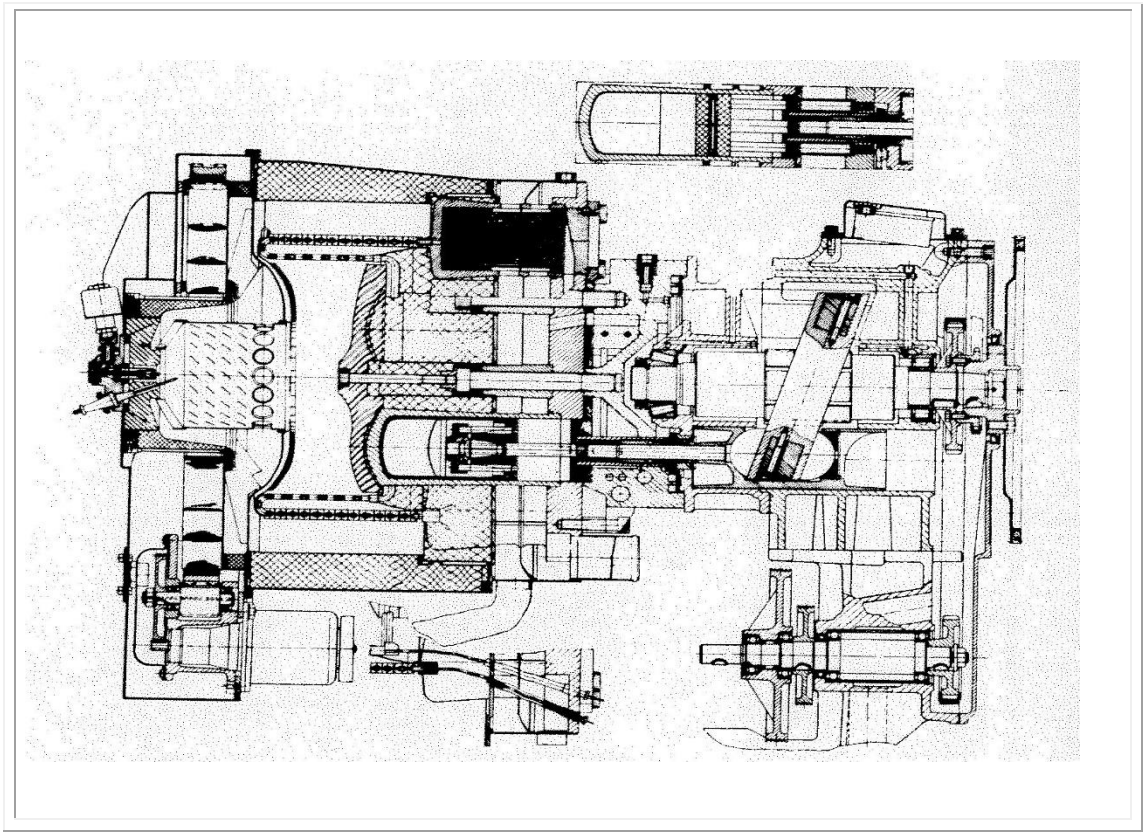
```
vpa = xpa*(ap - ar); % (piston - rod) volume aplitute {m^3)
vda = xda*(ad - ar); % (displacer - rod) volume amplitude(m^3)
vea = xda*ad; % displacer volume aplitute {m^3)
phi = phid*pi/180; % radians
delta = atan2(vda*sin(phi),(vda*cos(phi) - vpa));
  % compression space volume to piston amplitude phase advance
vca = sqrt(vpa*vpa - 2*vpa*vda*cos(phi) + vda*vda);
  % compression space volume amplitude (m^3)
vswc = 2*vca; % compression space swept volume (m^3)
vswe = 2*vea; % expansion space swept volume (m^3)
alpha = pi + phi - delta; % expansion phase angle advance (radians)
fprintf('\nbeta drive engine data summary:\n');
fprintf(' comp clearence,swept vols %.1f, %.1f [cm^3]\n', vclc*1e6,vswc*1e6);
fprintf(' exp clearence,swept vols %.1f, %.1f [cm^3]\n', vcle*1e6,vswe*1e6);
fprintf(' expansion phase angle advance %.1f[degrees]\n\n', alpha*180/pi);
%=============================================================
```

Appendix M: Ford-Philips 4-215 Engine

From the 1950's through the 1980's there was an ambitious effort to develop automobile Stirling engines in the USA and Europe (refer to **NASA** publications on the **Ford Motor Company** program and the **Mechanical Technology Inc.** (MTI) program.)

During the 1970's N V Philips, of Holland, and the Ford Motor Company developed the Ford-Philips 4-215 engine – a four-cylinder swashplate-drive engine as shown below:





This engine was used as a case study in the book by I.Urieli & D.M.Berchowitz – Stirling Cycle Engine Analysis (Adam Hilger, 1984), pages 25 – 31, and since the book is out of print, these pages have been added here for convenience (Appendix C). All the required data for the simulation is available from this file.

The swashplate drive is a pure sinusoidal motion, and with four pistons the volumetric phase angle advance is 90 degrees. Thus the global variables data required is simply entered from the keyboard to create a data file, or read from a previously created data file, and no calculations are required. The MATLAB function **sindrive** follows:

```
function sindrive
% Sinusoidal drive engine configuration
% Israel Urieli 4/14/02

global vclc vcle % compression, expansion clearance vols [m^3]
global vswc vswe % compression, expansion swept volumes [m^3]
global alpha % phase angle advance of expansion space [radians]
global new fid % new data file

fprintf('sinusoidal drive engine configuration\n');
if(strncmp(new,'y',1))
    vclc = input('enter compression space clearance volume [m^3]: ');
    vswc = input('enter compression space swept volume [m^3]: ');
    vcle = input('enter expansion space clearance volume [m^3]: ');
    vswe = input('enter expansion space swept volume [m^3]: ');
    phase = input('enter expansion phase angle advance [degrees]: ');
    fprintf(fid, '%.3e\n', vclc);
    fprintf(fid, '%.3e\n', vswc);
    fprintf(fid, '%.3e\n', vcle);
    fprintf(fid, '%.3e\n', vswe);
    fprintf(fid, '%.1f\n', phase);
else
    vclc = fscanf(fid, '%e', 1);
    vswc = fscanf(fid, '%e', 1);
    vcle = fscanf(fid, '%e', 1);
    vswe = fscanf(fid, '%e', 1);
    phase = fscanf(fid, '%f', 1);
end

fprintf('\nsinusoidal drive engine data summary:\n');
fprintf('comp clearance,swept vols %.1f, %.1f [cm^3]\n', vclc*1e6, vswc*1e6);
fprintf('exp clearance,swept vols %.1f, %.1f [cm^3]\n', vcle*1e6, vswe*1e6);
fprintf('expansion phase angle advance %.1f[degrees]\n', phase);
alpha = phase * pi/180;
%============================================================
```

Notice that the main purpose of this function is to fill in values of the global variables concerning the working space (compression and expansion) volumes. The remaining

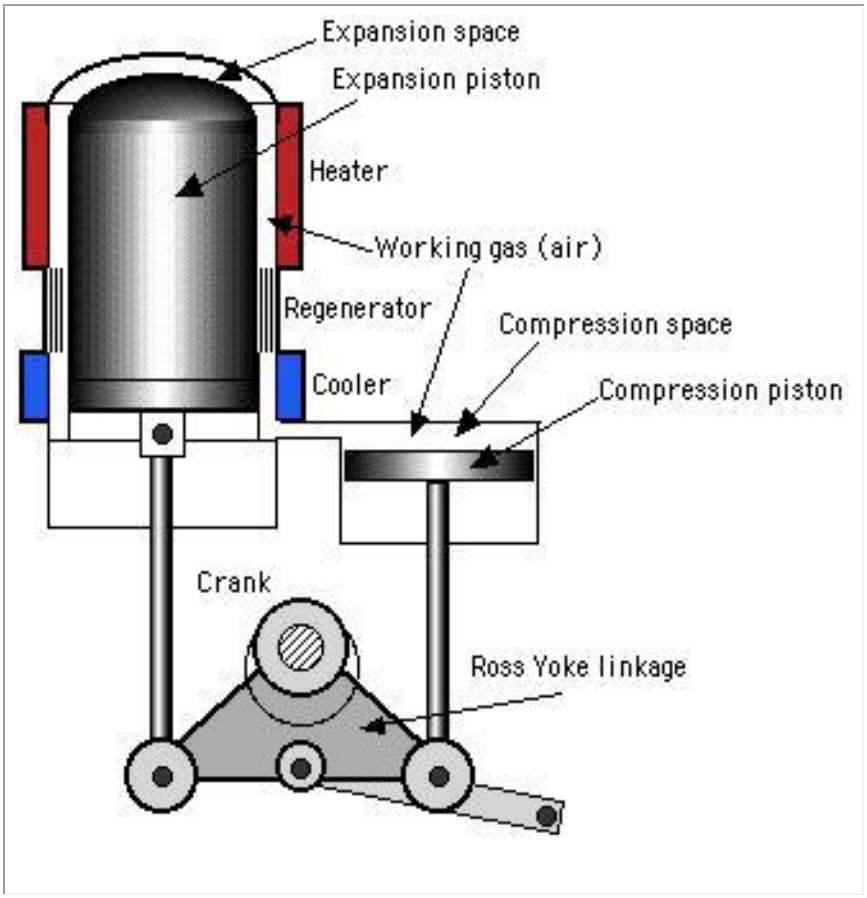
global variables are concerning the heat exchangers, regenerator, working gas and operating conditions are determined in the various functions of the define function set.

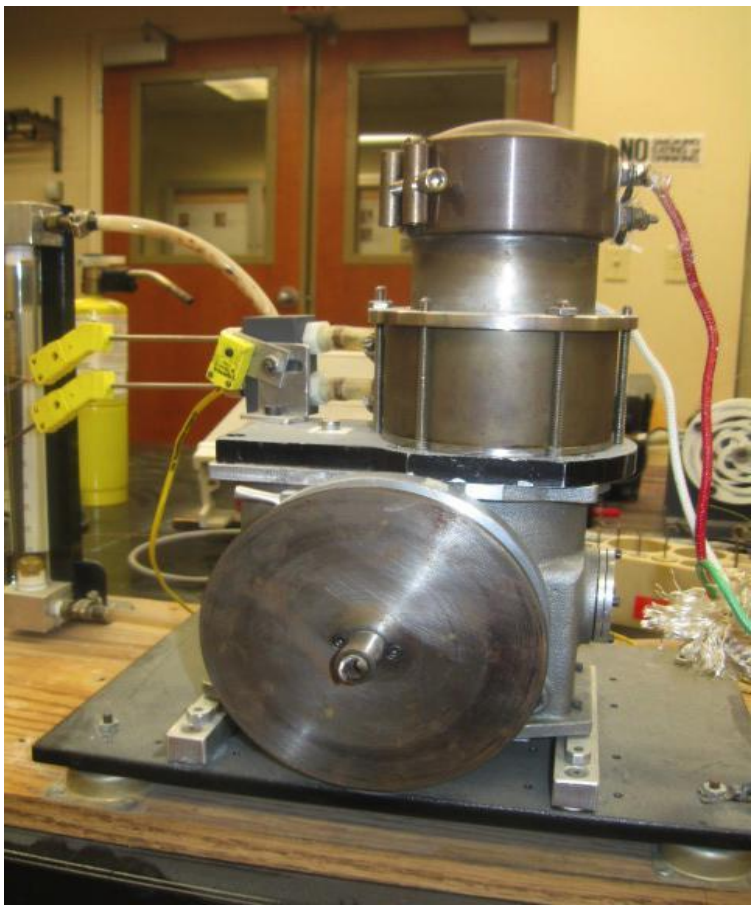
At Ohio University we have one of the four pistons and one of the eight cooler heat exchanger and mesh regenerator tubes of the Ford-Philips 4-215 engine. The engine has a total of 2968 cooler tubes with a 0.9mm internal diameter and 87mm length, and extremely fine wire diameter regenerator mesh (36µm) leading to more than 100 square meters of wetted area. This indicates the extreme importance of the heat exchanger section of the engine which we will cover in the Chapter 5 of this learning resource.

The rest of the define function modules include the heat exchangers, regenerator, operating parameters, working gas parameters and Schmidt analysis, and all are independent of the engine type. The complete data file that defines the Ford-Philips 4-215 engine is **ford.dat**.

Appendix N: D-90 Ross Yoke-drive Engine

Andy Ross of Columbus, Ohio has been designing and building small air engines since the 1970's, including extremely innovative Alpha designs. Refer to Andy Ross' delightful book: **Making Stirling Engines** (Ross Experimental, 1993). The D-90 Yoke-drive Alpha Stirling engine described in his book is used as the primary case study of this web resource. **Matt Keveney** has done an animation showing clearly the principles of operation of the **Ross yoke linkage** mechanism. This ingenious mechanism for transferring dual piston motion into rotational motion minimizes the piston side forces normally encountered on a regular crankshaft mechanism.





The D-90 Ross Yoke-drive engine at Ohio University

In order to do a simulation of the performance of the Ross Yoke-drive engine we need to evaluate the compression and expansion volume variations Vc and Ve as well as the volume derivatives **d**Vc and **d**Ve as functions of the crankangle θ. The equations shown here were derived in the section on **Ross Yoke-drive volume variations** and the main purpose of the function below is to fill in the values of the global variables required in order to enable evaluation of these equations.

$$\text{Let } b\theta = \sqrt{b1^2 - crank^2 \cos^2\theta}$$

$$ymin = \sqrt{(yoke - crank)^2 - b2^2}$$

$$ye = crank\left(\sin\theta + \left(\frac{b2}{b1}\right)\cos\theta\right) + b\theta$$

$$yc = crank\left(\sin\theta - \left(\frac{b2}{b1}\right)\cos\theta\right) + b\theta$$

$$Ve = Vcle + Ae\,(ye - ymin)$$

$$Vc = Vclc + Ac\,(yc - ymin)$$

$$dVe = Ae\ crank\left(\cos\theta - \left(\frac{b2}{b1}\right)\sin\theta + \frac{crank\sin\theta\cos\theta}{b\theta}\right)$$

$$dVc = Ac\ crank\left(\cos\theta + \left(\frac{b2}{b1}\right)\sin\theta + \frac{crank\sin\theta\cos\theta}{b\theta}\right)$$

```
function yokedrive
% Ross yoke drive engine configuration
% Israel Urieli 4/14/02

global vclc vcle % compression, expansion clearance vols [m^3]
global vswc vswe % compression, expansion swept volumes [m^3]
global alpha % phase angle advance of expansion space [radians]
global b1 % Ross yoke length (1/2 yoke base) [m]
global b2 % Ross yoke height [m]
global crank % crank radius [m]
global dcomp dexp % diameter of compression/expansion pistons [m]
global acomp aexp % area of compression/expansion pistons [m^2]
global ymin % minimum yoke vertical displacement [m]
global new fid % new data file

fprintf('Ross yoke drive engine configuration\n');
```

```
if(strncmp(new,'y',1))
    vclc = input('enter compression space clearance volume [m^3]: ');
    vcle = input('enter expansion space clearance volume [m^3]: ');
    b1 = input('enter Ross yoke height b1 [m]: ');
    b2 = input('enter Ross yoke length b2 (1/2 yoke base) [m]: ');
    crank = input('enter crank radius [m]: ');
    dcomp = input('enter compression piston diameter [m]: ');
    dexp = input('enter expansion piston diameter [m]: ');
    fprintf(fid, '%.3e\n', vclc);
    fprintf(fid, '%.3e\n', vcle);
    fprintf(fid, '%.3e\n', b1);
    fprintf(fid, '%.3e\n', b2);
    fprintf(fid, '%.3e\n', crank);
    fprintf(fid, '%.3e\n', dcomp);
    fprintf(fid, '%.3e\n', dexp);
else
    vclc = fscanf(fid, '%e', 1);
    vcle = fscanf(fid, '%e', 1);
    b1 = fscanf(fid, '%e', 1);
    b2 = fscanf(fid, '%e', 1);
    crank = fscanf(fid, '%e', 1);
    dcomp = fscanf(fid, '%e', 1);
    dexp = fscanf(fid, '%e', 1);
end

acomp = pi*dcomp^2/4.0;
aexp = pi*dexp^2/4.0;
yoke = sqrt(b1^2 + b2^2);
ymax = sqrt((yoke + crank)^2 - b2^2);
ymin = sqrt((yoke - crank)^2 - b2^2);
vswc = acomp*(ymax - ymin);
vswe = aexp*(ymax - ymin);
thmaxe = asin(ymax/(yoke + crank));
thmaxc = pi - thmaxe;
thmine = pi + asin(ymin/(yoke - crank));
thminc = 3*pi - thmine;
alpha = 0.5*(thmaxc - thmaxe) + 0.5*(thminc - thmine);
phase = alpha*180/pi;
```

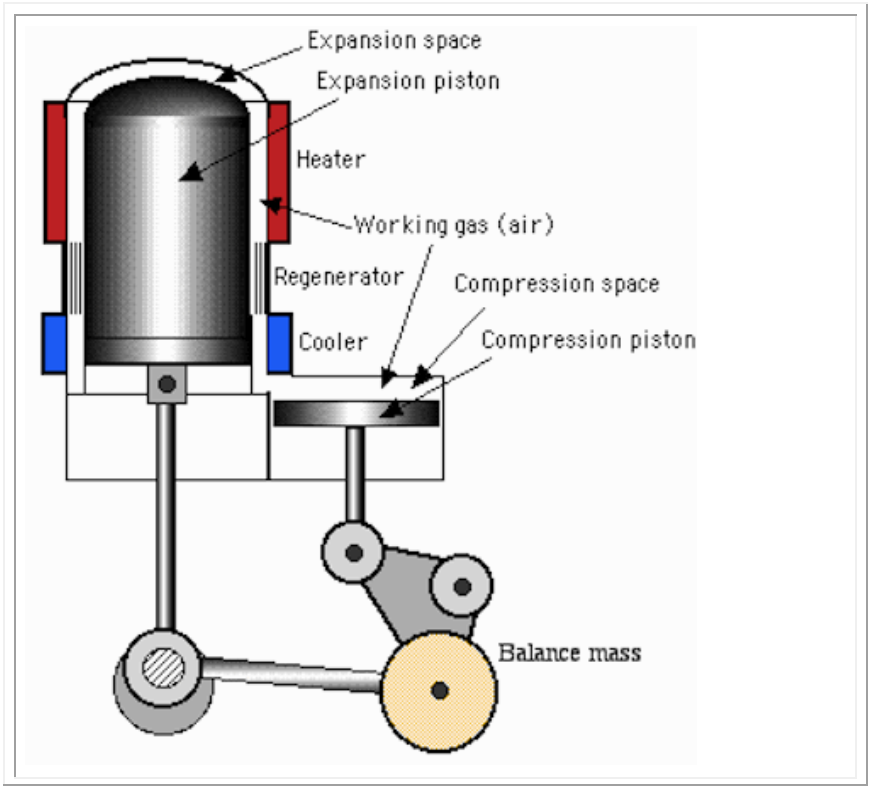
```
fprintf('\nRoss yoke drive engine data summary:\n');
fprintf('yoke height b1 %.1f [mm]\n', b1*1e3);
fprintf('yoke length b2 (1/2 yoke base) %.1f [mm]\n', b2*1e3);
fprintf('crank radius %.1f [mm]\n', crank*1e3);
fprintf('compression piston diameter %.1f [mm]\n', dcomp*1e3);
fprintf('expansion piston diameter %.1f [mm]\n', dexp*1e3);
fprintf('comp clearance, swept vols %.1f, %.1f [cm^3]\n', vclc*1e6, vswc*1e6);
fprintf('exp clearance, swept vols %.1f, %.1f [cm^3]\n', vcle*1e6, vswe*1e6);
fprintf('ymin = %.1f(cm), ymax = %.1f(cm)\n', ymin*1e2, ymax*1e2);
fprintf('alpha = %.1f(degrees)\n', phase);
%============================================================
```

The rest of the **define** function modules include the heat exchangers, regenerator, operating parameters, working gas parameters and Schmidt analysis, and all are independent of the engine type. The complete data file that defines the Ross D-90 Yoke drive engine is **ross90.dat**.

Appendix O: Ross Rocker-V Drive Engine

Andy Ross of Columbus, Ohio has been designing and building small air engines since the 1970's, including extremely innovative Alpha designs. Refer to Andy Ross' delightful book: **Making Stirling Engines** (Ross Experimental, 1993). More recently Andy Ross came up with the balanced Rocker-V mechanism design. He has published an article on the model Climax locomotive that he built using a small (20cc) Rocker-V engine, and allowed me to maintain a copy of this article "**A Class A Climax Locomotive**" (Appendix A.3). A number of these engines were built by students for a **Senior Design** (Appendix B) class at Ohio University in 2001.

The Rocker-V drive mechanism is similar to the standard V drive mechanism (as shown in the **Two Cylinder Stirling Engine** animation by **Matt Keveney**), however the addition of the rocker mechanism avoids the use of a heat exchanger section stretching across the V.



The Ross Rocker-V drive engine at Ohio University

```
%=============================================================
function rockerVdrive
% Ross rocker-V drive engine configuration
% Israel Urieli 4/14/02 & Martine Long 2/14/05
global vclc vcle % compression,expansion clearence vols [m^3]
global vswc vswe % compression, expansion swept volumes [m^3]
global alpha % phase angle advance of expansion space [radians]
global crank % crank radius [m]
global dcomp dexp % diameter of compression/expansion pistons [m]
global acomp aexp % area of compression/expansion pistons [m^2]
global conrodc conrode % length of comp/exp piston connecting rods [m]
global ycmax yemax % maximum comp/exp piston vertical displacement [m]
global new fid % new data file
fprintf('Ross rocker-V drive engine configuration\n');
if(strncmp(new,'y',1))
```

```
    vclc = input('enter compression space clearence volume [m^3]: ')
    vcle = input('enter expansion space clearence volume [m^3]: ');
    crank = input('enter crank radius [m]: ');
    conrodc = input('enter compression piston connecting rod length [m]: ');
    conrode = input('enter expansion piston connecting rod length [m]: ');
    dcomp = input('enter compression piston diameter [m]: ');
    dexp = input('enter expansion piston diameter [m]: ');
    phase = input('enter expansion phase angle advance [degrees]: ');

    fprintf(fid, '%.3e\n', vclc);
    fprintf(fid, '%.3e\n', vcle);
    fprintf(fid, '%.3e\n', crank);
    fprintf(fid, '%.3e\n', conrodc);
    fprintf(fid, '%.3e\n', conrode);
    fprintf(fid, '%.3e\n', dcomp);
    fprintf(fid, '%.3e\n', dexp);
    fprintf(fid, '%.1f\n', phase);
else
    vclc = fscanf(fid,'%e',1);
    vcle = fscanf(fid,'%e',1);
    crank = fscanf(fid,'%e',1);
    conrodc = fscanf(fid,'%e',1);
    conrode = fscanf(fid,'%e',1);
    dcomp = fscanf(fid,'%e',1);
    dexp = fscanf(fid,'%e',1);
    phase = fscanf(fid,'%f',1);
end
acomp = pi*dcomp^2/4.0;
aexp = pi*dexp^2/4.0;
ycmax = conrodc + crank;
ycmin = conrodc - crank;
yemax = conrode + crank;
yemin = conrode - crank;
vswc = acomp*(ycmax - ycmin);
vswe = aexp*(yemax - yemin);
fprintf('\nRoss rocker-V drive engine data summary:\n');
fprintf(' crank radius %.1f [mm]\n', crank*1e3);
fprintf(' compression piston connecting rod length %.1f [mm]\n', conrodc*1e3);
```

```
fprintf(' expansion piston connecting rod length %.1f [mm]\n', conrode*1e3);
fprintf(' compression piston diameter  %.1f [mm]\n', dcomp*1e3);
fprintf(' expansion piston diameter %.1f [mm]\n', dexp*1e3);
fprintf(' comp clearence,swept vols %.1f, %.1f [cm^3]\n', vclc*1e6,vswc*1e6);
fprintf(' exp clearence,swept vols %.1f, %.1f [cm^3]\n', vcle*1e6,vswe*1e6);
fprintf(' COMPRESSION ymin = %.1f(cm), ymax = %.1f(cm)\n',ycmin*1e2,ycmax*1e2)
fprintf(' EXPANSION ymin = %.1f(cm), ymax = %.1f(cm)\n',yemin*1e2,yemax*1e2)
fprintf(' expansion phase angle advance %.1f[degrees]\n', phase);
alpha = phase * pi/180;
%============================================================
```

The rest of the **define** function modules include the heat exchangers, regenerator, operating parameters, working gas parameters and Schmidt analysis, and all are independent of the engine type. The complete data file that defines the 20cc Rocker-V drive engine is **rockerV.dat**.

Appendix P: Heat Exchanger Functions (heatex)

The purpose of the heatex function is to select the heat exchanger types for the heater and cooler (or heat acceptor and heat rejector for refrigerators) as well as for the regenerator.

```
function heatex
% Specify heat exchanger geometric parameters
% Israel Urieli 3/31/02 (modified 12/01/03)
cooler;
regen;
heater;
%=====================================================
```

The purpose of the three heat exchanger functions is to specify values for the global variables subsequently used in the Schmidt analysis, Ideal Adiabatic or Simple simulations. Each global variable has the subscript h for heater, r for regenerator, and k for cooler (subscript c is used for compression space in the engine routines). For the heater and cooler the variables are defined as follows:

v = void volume (cu.m) - used only in the Schmidt and Ideal Adiabatic functions
The following variables are used only in the Simple simulation.
a = internal free flow area (sq.m)
awg = internal wetted area (sq.m) - subscript wg refers to wall/gas
d = hydraulic diameter (m) – refer to **Chapter 5a - Scaling Parameters**
l = effective length (m)

This learning resource includes three types of heat exchangers used in the heater and cooler: pipes (as used in the Ford-Philips 4-215 engine), annular slots (as used in the D-90 Ross Yoke-drive engine) and annulus (as used in the Ross RockerV engine).

```
function cooler
% Specify cooler geometric parameters
```

```
% Israel Urieli 4/15/02
global vk % cooler void volume [m^3]
global ak % cooler internal free flow area [m^2]
global awgk % cooler internal wetted area [m^2]
global dk % cooler hydraulic diameter [m]
global lk % cooler effective length [m]
global new fid % new data file
cooler_type = 'u';
while(strncmp(cooler_type, 'u',1))
    if (strncmp(new, 'y',1))
      fprintf('Available cooler types are:\n')
      fprintf('   p, for smooth pipes\n')
      fprintf('   a, for smooth annulus\n')
      fprintf('   s, for slots\n')
      cooler_type = input('enter cooler type ', 's');
      fprintf(fid, '%c\n', cooler_type(1));
   else
      fscanf(fid, '%c',1); % bypass the previous newline character
      cooler_type = fscanf(fid, '%c',1);
   end
   if (strncmp(cooler_type, 'p',1))
      [vk,ak,awgk,dk,lk] = pipes;
   elseif (strncmp(cooler_type, 'a',1))
      [vk,ak,awgk,dk,lk] = annulus;
   elseif (strncmp(cooler_type, 's',1))
      [vk,ak,awgk,dk,lk] = slots;
   else
      fprintf('cooler type is undefined\n')
      cooler_type = 'u';
   end
end
fprintf('cooler data summary:\n');
fprintf(' void volume(cc) %.2f\n', vk*1e6)
fprintf(' free flow area (cm^2) %.2f\n', ak*1e2)
fprintf(' wetted area (cm^2) %.2f\n', awgk*1e2)
fprintf(' hydraulic diameter(mm) %.2f\n', dk*1e3)
fprintf(' cooler length (cm) %.2f\n', lk*1e2)
%=======================================================
```

```
function heater
% Specify heater geometric parameters
% Israel Urieli 4/15/02
global vh % heater void volume [m^3]
global ah % heater internal free flow area [m^2]
global awgh % heater internal wetted area [m^2]
global dh % heater hydraulic diameter [m]
global lh % heater effective length [m]
global new fid % new data file
heater_type = 'u';
while (strncmp(heater_type, 'u',1))
   if (strncmp(new, 'y',1))
      fprintf('Available heater types are:\n')
      fprintf('   p, for smooth pipes\n')
      fprintf('   a, for smooth annulus\n')
      fprintf('   s, for slots\n')
      heater_type = input('enter heater type ', 's');
      fprintf(fid, '%c\n', heater_type(1));
   else
      fscanf(fid, '%c',1); % bypass the previous newline character
      heater_type = fscanf(fid, '%c',1);
   end
   if (strncmp(heater_type, 'p',1))
      [vh,ah,awgh,dh,lh] = pipes;
   elseif (strncmp(heater_type, 'a',1))
      [vh,ah,awgh,dh,lh] = annulus;
   elseif strncmp(heater_type, 's',1))
      [vh,ah,awgh,dh,lh] = slots;
   else
      fprintf('heater type is undefined\n')
      heater_type = 'u';
   end
end
fprintf('heater data summary:\n');
fprintf(' void volume(cc) %.2f\n', vh*1e6)
fprintf(' free flow area (cm^2) %.2f\n', ah*1e2)
fprintf(' wetted area (cm^2) %.2f\n', awgh*1e2)
fprintf(' hydraulic diameter(mm) %.2f\n', dh*1e3)
```

```
fprintf(' heater length (cm) %.2f\n', lh*1e2)
%=========================================================
```

The following functions are the three types of heat exchanger invoked by functions heater and cooler:

```
function [v,a,awg,d,len] = pipes
% homogeneous smooth pipes heat exchanger
% Israel Urieli 4/15/02
global new fid % new data file
fprintf('homogeneous bundle of smooth pipes\n')
if(strncmp(new,'y',1))
   d = input('enter pipe inside diameter [m] : ');
   len = input('enter heat exchanger length [m] : ');
   num = input('enter number of pipes in bundle : ');
   fprintf(fid, '%.3e\n', d);
   fprintf(fid, '%.3e\n', len);
   fprintf(fid, '%d\n', num);
else
   d = fscanf(fid,'%e',1);
   len = fscanf(fid,'%e',1);
   num = fscanf(fid,'%d',1);
end
a = num*pi*d*d/4;
v = a*len;
awg = num*pi*d*len;
%=========================================================
function [v,a,awg,d,len] = annulus
% annular gap heat exchanger
% Israel Urieli 12/01/03
% Modified 2/14/2010 wetted area
global new fid % new data file
fprintf(' annular gap heat exchanger\n')
if(strncmp(new,'y',1))
   dout = input('enter annular gap outer diameter [m] : ');
   din = input('enter annular gap inner diameter [m] : ');
```

```
   len = input('enter heat exchanger length [m] : ');
   fprintf(fid, '%.3e\n', dout);
   fprintf(fid, '%.3e\n', din);
   fprintf(fid, '%.3e\n', len);
else
   dout = fscanf(fid,'%e',1);
   din = fscanf(fid,'%e',1);
   len = fscanf(fid,'%e',1);
end
a = pi*(dout*dout - din*din)/4;
v = a*len;
awg = pi*dout*len;
d = dout - din;
%=========================================================
function [v,a,awg,d,len] = slots
% slots heat exchanger
% Israel Urieli 12/01/03
% Modified 2/14/2010 wetted area
global new fid % new data file
fprintf(' slots heat exchanger\n')
if(strncmp(new,'y',1))
   w = input('enter width of slot [m] : ');
   h = input('enter height of slot [m] : ');
   len = input('enter heat exchanger length [m] : ');
   num = input('enter number of slots : ');
   fprintf(fid, '%.3e\n', w);
   fprintf(fid, '%.3e\n', h);
   fprintf(fid, '%.3e\n', len);
   fprintf(fid, '%d\n', num);
else
   w = fscanf(fid,'%e',1);
   h = fscanf(fid,'%e',1);
   len = fscanf(fid,'%e',1);
   num = fscanf(fid,'%d',1);
end
a = num*w*h;
v = a*len;
awg = num*(w + 2*h)*len;
```

```
d = 4*v/awg;
%=======================================================
```

Appendix Q: Regenerator Functions

The regenerator connects the heater and cooler, hence in addition to evaluating the heat transfer from the regenerator matrix to the working gas, it is necessary to determine the conduction heat leakage through the regenerator housing. The housing can either be tubular or annular, and this learning resource includes two regenerator matrix types, either wire mesh matrix (as in the Ford-Philips 4-215 engine) or a wrapped foil matrix (as in the D-90 Ross Yoke-drive engine). We also consider the case of no regenerator matrix, in which the wetted area of the housing takes the role of the regenerator matrix (as in the Lehmann engine analyzed by Schmidt). The purpose of the regenerator functions is to specify values for the global variables subsequently used in the Schmidt analysis, Ideal Adiabatic or Simple simulations. Each global variable has the subscript r for heater and k for cooler (subscript c is used for compression space in the engine routines). The variables are defined as follows:

vr = void volume (cu.m) - used only in the Schmidt and Ideal Adiabatic functions
The following variables are used only in the Simple simulation.
lr = regenerator effective length (m)
cqwr = regenerator housing thermal conductance (W/K)
ar = regenerator internal free flow area (sq.m)
awgr0 = no matrix regenerator wetted area (sq.m)
awgr = regenerator wetted area (sq.m)
dr = regenerator hydraulic diameter (m) – refer to **Chapter 5a - Scaling Parameters**

The following five MATLAB functions in the **regen.m** function module specify all the above global variable values.

```
function regen
% Specifies regenerator geometric and thermal properties
% Israel Urieli 04/20/02 (modified 12/01/03)
% modified 2/12/2010 to include awgr0 (wetted area)
% modified 11/27/2010 to include 'no regenerator matrix'
global lr % regenerator effective length [m]
global awgr0 % no matrix regenerator wetted area [m^2]
global cqwr % regenerator housing thermal conductance [W/K]
```

```
global new fid % new data file
regen_type = 'u';
while(strncmp(regen_type,'u',1))
   if(strncmp(new,'y',1))
      fprintf('Available regenerator configurations are:\n')
      fprintf('   t, for tubular regenerator set\n')
      fprintf('   a, for annular regenerator\n')
      regen_type = input('enter regenerator configuration ','s');
      fprintf(fid, '%c\n', regen_type(1));
   else
      fscanf(fid, '%c',1); % bypass the previous newline character
      regen_type = fscanf(fid, '%c',1);
   end
   if(strncmp(regen_type,'t',1))
      printf('tubular regenerator housing\n')
      if(strncmp(new,'y',1))
         dout = input('enter tube housing external diameter [m] : ');
         domat = input('enter tube housing internal diameter [m] : ');
         lr = input('enter regenerator length [m] : ');
         num = input('enter number of tubes : ');
         fprintf(fid, '%.3e\n', dout);
         fprintf(fid, '%.3e\n', domat);
         fprintf(fid, '%.3e\n', lr);
         fprintf(fid, '%d\n', num);
      else
         dout = fscanf(fid, '%e',1);
         domat = fscanf(fid, '%e',1);
         lr = fscanf(fid, '%e',1);
         num = fscanf(fid, '%d',1);
      end
      dimat = 0;
      awgr0 = num*pi*domat*lr;
   elseif(strncmp(regen_type,'a',1))
      fprintf('annular regenerator housing\n')
      if(strncmp(new,'y',1))
         dout = input('enter housing external diameter [m] : ');
         domat = input('enter housing internal diameter [m] : ');
         dimat = input('enter matrix internal diameter [m] : ');
```

```
        lr = input('enter regenerator length [m] : ');
        fprintf(fid, '%.3e\n', dout);
        fprintf(fid, '%.3e\n', domat);
        fprintf(fid, '%.3e\n', dimat);
        fprintf(fid, '%.3e\n', lr);
      else
        dout = fscanf(fid, '%e',1);
        domat = fscanf(fid, '%e',1);
        dimat = fscanf(fid, '%e',1);
        lr = fscanf(fid, '%e',1);
      end
      num = 1;
      awgr0 = pi*(dimat + domat)*lr;
   else
      fprintf('regenerator configuration is undefined\n')
      regen_type = 'u';
   end
end
amat = num*pi*(domat*domat - dimat*dimat)/4; % regen matrix area
awr = num*pi*(dout*dout - domat*domat)/4; % regen housing wall area
%########temporary fix (4/20/02):
kwr = 25; % thermal conductivity [W/m/K]
% note that stainless steel thermal conductivity is temp dependent
%   25 W/m/K for normal engine conditions,
%    6 W/m/K for cryogenic coolers.
cqwr = kwr*awr/lr; % regen wall thermal conductance [W/K]
matrix(amat);
%=============================================================
function matrix(amat)
% Specifies regenerator matrix geometric and thermal properties
% Israel Urieli 03/31/02
% modified 11/27/10 for no regenerator matrix

global matrix_type % m)esh, f)oil or n)o matrix
global new fid % new data file

matrix_type = 'u';
while(strncmp(matrix_type,'u',1))
```

```
    if(strncmp(new,'y',1))
       fprintf('Available matrix types are:\n')
       fprintf('   m, for mesh matrix\n')
       fprintf('   f, for foil matrix\n')
       fprintf('   n, for no matrix\n')
       matrix_type = input('enter matrix type ','s');
       fprintf(fid, '%c\n', matrix_type(1));
    else
       fscanf(fid, '%c',1); % bypass the previous newline character
       matrix_type = fscanf(fid, '%c',1);
    end
    if(strncmp(matrix_type,'m',1))
       mesh(amat);
    elseif(strncmp(matrix_type,'f',1))
       foil(amat);
    elseif(strncmp(matrix_type,'n',1))
       nomatrix(amat);
    else
       fprintf('matrix configuration is undefined\n')
       matrix_type = 'u';
    end
end
%=============================================================
function mesh(amat)
% Specifies mesh matrix geometric and thermal properties
% Israel Urieli 03/31/02
global vr % regen void volume [m^3]
global ar % regen internal free flow area [m^2]
global awgr % regen internal wetted area [m^2]
global awgr0 % no matrix regenerator wetted area [m^2]
global lr % regenerator effective length [m]
global dr % regen hydraulic diameter [m]
global new fid % new data file
fprintf(' stacked wire mesh matrix\n')
if(strncmp(new,'y',1))
   porosity = input('enter matrix porosity : ');
   dwire = input('enter matrix wire diameter [m] : ');
   fprintf(fid, '%.3f\n', porosity);
```

```
   fprintf(fid, '%.3e\n', dwire);
else
   porosity = fscanf(fid,'%f',1);
   dwire = fscanf(fid,'%e',1);
end
ar = amat*porosity;
vr = ar*lr;
dr = dwire*porosity/(1 - porosity);
awgr = 4*vr/dr + awgr0;
fprintf(' matrix porosity: %.3f\n', porosity)
fprintf(' matrix wire diam %.2f(mm)\n', dwire*1e3)
fprintf(' hydraulic diam %.3f(mm)\n', dr*1e3)
fprintf(' total wetted area %.3e(sq.m)\n', awgr)
fprintf(' regenerator length %.1f(mm)\n', lr*1e3)
fprintf(' void volume %.2f(cc)\n', vr*1e6)
%============================================================
function foil(amat)
% Specifies foil matrix geometric and thermal properties
% Israel Urieli 03/31/02
global vr % regen void volume [m^3]
global ar % regen internal free flow area [m^2]
global awgr % regen internal wetted area [m^2]
global awgr0 % no matrix regenerator wetted area [m^2]
global lr % regenerator effective length [m]
global dr % regen hydraulic diameter [m]
global new fid % new data file
fprintf(' wrapped foil matrix\n')
if(strncmp(new,'y',1))
   fl = input('enter unrolled length of foil [m] : ');
   th = input('enter foil thickness [m] : ');
   fprintf(fid, '%.3f\n', fl);
   fprintf(fid, '%.3e\n', th);
else
   fl = fscanf(fid,'%f',1);
   th = fscanf(fid,'%e',1);
end
am = th*fl;
ar = amat - am;
```

```
vr = ar*lr;
awgr = 2*lr*fl + awgr0;
dr = 4*vr/awgr;
porosity = ar/amat;
fprintf(' unrolled foil length: %.3f(m)\n', fl)
fprintf(' foil thickness %.3f(mm)\n',th*1e3)
fprintf(' hydraulic diam %.3f(mm)\n', dr*1e3)
fprintf(' total wetted area %f(sq.m)\n', awgr)
fprintf(' void volume %.2f(cc)\n', vr*1e6)
fprintf(' porosity %.3f\n', porosity)
%=============================================================
function nomatrix(amat)
% Specifies conditions for no regenerator matrix
% Israel Urieli 11/27/10
global vr % regen void volume [m^3]
global ar % regen internal free flow area [m^2]
global awgr % regen internal wetted area [m^2]
global awgr0 % no matrix regenerator wetted area [m^2]
global lr % regenerator effective length [m]
global dr % regen hydraulic diameter [m]
fprintf(' no regenerator matrix\n')
ar = amat;
vr = ar*lr;
awgr = awgr0;
dr = 4*vr/awgr;
fprintf(' hydraulic diam %.3f(mm)\n', dr*1e3)
fprintf(' total wetted area %f(sq.m)\n', awgr)
fprintf(' void volume %.2f(cc)\n', vr*1e6)
%=============================================================
```

Appendix R: The Working Gas Function

The purpose of this function is to specify the values of the global variables relevant to the specific working gas choice - helium, hydrogen or air. The first 4 variables (rgas, cp, cv, gama) are required for the basic energy relations, and the last 4 variables (mu0, t0, t_suth, prandtl) are required in the Simple analysis for evaluation of heat transfer and flow friction of the working gas (refer to Chapter 5 - **Scaling Parameters** and **Pumping Loss**)

```
function gas
% specifies the working gas properties (he, h2, air)
% Israel Urieli 4/20/02
global rgas % gas constant [J/kg.K]
global cp % specific heat capacity at constant pressure [J/kg.K]
global cv % specific heat capacity at constant volume [J/kg.K]
global gama % ratio: cp/cv
global mu0 % dynamic viscosity at reference temp t0 [kg.m/s]
global t0 t_suth % reference temperature [K], Sutherland constant [K]
global prandtl % Prandtl number
global new fid % new data file
gas_type = 'un';
while(strncmp(gas_type,'un',2))
  if(strncmp(new,'y',1))
    fprintf('Available gas types are:\n');
    fprintf('   hy)drogen)\n');
    fprintf('   he)lium\n');
    fprintf('   ai)r\n');
    gas_type = input('enter gas type: ','s');
    gas_type = [gas_type(1), gas_type(2)];
    fprintf(fid, '%s\n', gas_type);
  else
    fscanf(fid, '%c',1); % bypass the previous newline character
    gas_type = fscanf(fid, '%c',2);
  end
  if(strncmp(gas_type,'hy',2))
    fprintf('gas type is hydrogen\n')
    gama = 1.4;
```

```
      rgas = 4157.2;
      mu0 = 8.35e-6;
      t_suth = 84.4;
   elseif(strncmp(gas_type,'he',2))
      fprintf('gas type is helium\n')
      gama = 1.67;
      rgas = 2078.6;
      mu0 = 18.85e-6;
      t_suth = 80.0;
   elseif(strncmp(gas_type,'ai',2))
      fprintf('gas type is air\n')
      gama = 1.4;
      rgas = 287.0;
      mu0 = 17.08e-6;
      t_suth = 112.0;
   else
      fprintf('gas type is undefined\n')
      gas_type = 'un';
   end
end
cv = rgas/(gama - 1);
cp = gama*cv;
t0 = 273;
prandtl = 0.71;
```

Appendix S: Operating Conditions and Schmidt Analysis functions

The operating conditions finalize the specification of the required global variables, and are defined as shown in the MATLAB function below. Since the ideal regenerator has a linear temperature distribution, the formula for tr is developed in the section **Regenerator Mean Effective Temperature** and is used in the Schmidt analysis function below to evaluate the last required global variable - mgas - the mass of working gas.

```
function operat
% Determine operating parameters and do Schmidt analysis
% Israel Urieli 4/20/02
global pmean % mean (charge) pressure [Pa]
global tk tr th % cooler, regenerator, heater temperatures [K]
global freq omega % cycle frequency [herz], [rads/s]
global new fid % new data file
if(strncmp(new,'y',1))
   pmean = input('enter mean pressure (Pa) : ');
   tk = input('enter cold sink temperature (K) : ');
   th = input('enter hot source temperature (K) : ');
   freq = input('enter operating frequency (herz) : ');
   fprintf(fid, '%.1f\n', pmean);
   fprintf(fid, '%.1f\n', tk);
   fprintf(fid, '%.1f\n', th);
   fprintf(fid, '%.1f\n', freq);
else
   pmean = fscanf(fid,'%f',1);
   tk = fscanf(fid,'%f',1);
   th = fscanf(fid,'%f',1);
   freq = fscanf(fid,'%f',1);
end
tr = (th - tk)/log(th/tk);
omega = 2*pi*freq;
fprintf('operating parameters:\n');
fprintf(' mean pressure (kPa): %.3f\n',pmean*1e-3);
fprintf(' cold sink temperature (K): %.1f\n',tk);
```

```
fprintf(' hot source temperature (K): %.1f\n',th);
fprintf(' effective regenerator temperature (K): %.1f\n',tr);
fprintf(' operating frequency (herz): %.1f\n',freq);
Schmidt; % Do Schmidt analysis
%=============================================================
```

The Schmidt Analysis presented previously results in a closed form solution of the Ideal Isothermal Stirling cycle engine as summarized below:

$$Vc = Vclc + Vswc\ (1+\cos\theta)/2$$

$$Ve = Vcle + Vswe\ [1+\cos(\theta+\alpha)]/2$$

$$Wc = Qc = \pi\ Vswc\ p_{mean}\ \sin\beta\ (\sqrt{1-b^2}-1)/b$$

$$We = Qe = \pi\ Vswe\ p_{mean}\ \sin(\beta-\alpha)\ (\sqrt{1-b^2}-1)/b$$

$$W = Wc + We$$

$$\eta = W/Qe = 1 - Tk/Th$$

where

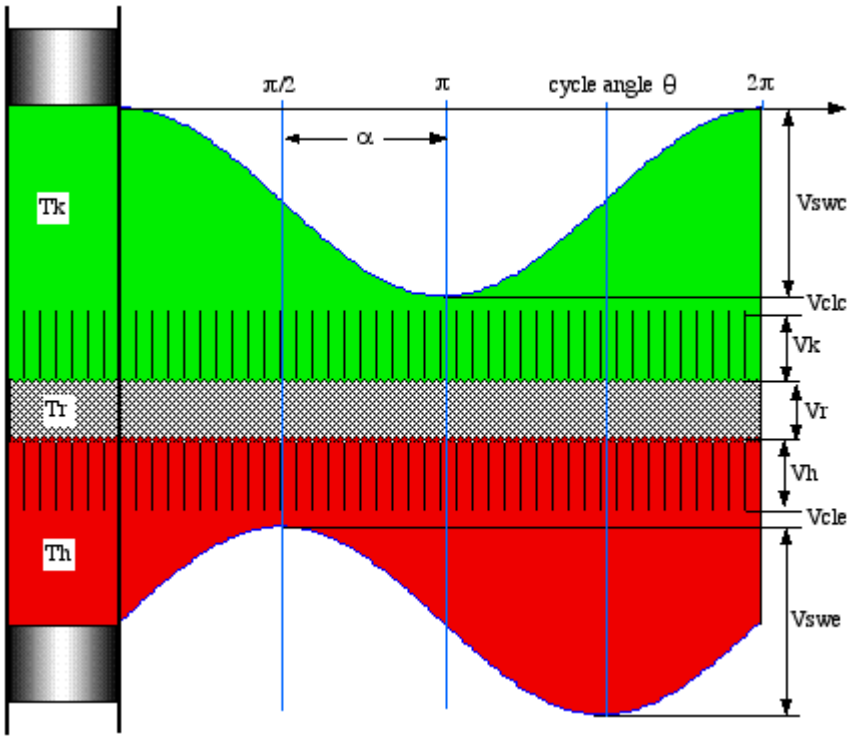
$$\tan\beta = \left(\frac{Vswe\ \sin\alpha/Th}{Vswe\ \cos\alpha/Th + Vswc/Tk}\right)$$

$$c = \frac{1}{2}\sqrt{\left(\frac{Vswe}{Th}\right)^2 + 2\frac{Vswe}{Th}\frac{Vswc}{Tk}\cos\alpha + \left(\frac{Vswc}{Tk}\right)^2}$$

$$s = \left[\frac{Vswc}{2\,Tk} + \frac{Vclc}{Tk} + \frac{Vk}{Tk} + \frac{Vr\ \ln(Th/Tk)}{(Th-Tk)} + \frac{Vh}{Th} + \frac{Vswe}{2\,Th} + \frac{Vcle}{Te}\right]$$

$$b = c/s$$

$$p_{mean} = M\,R \Big/ \left(s\sqrt{1-b^2}\right)$$

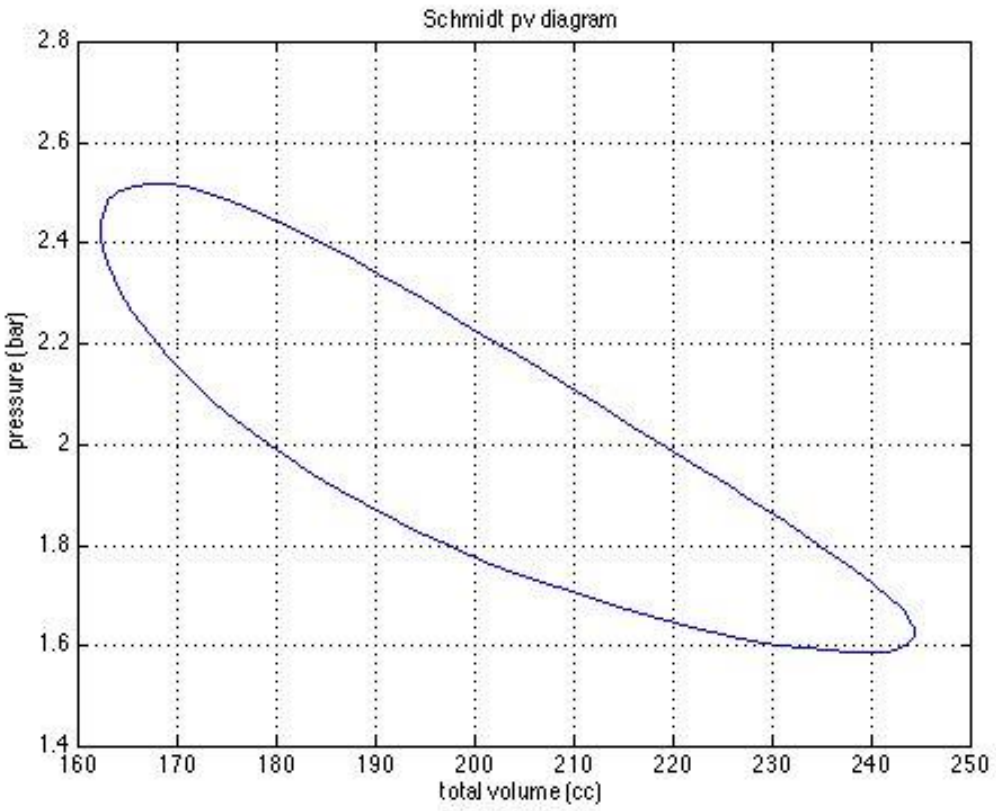


The following is the MATLAB function of the above equations. Notice that the last equation allows specification of the global variable - mgas - the mass of working gas as a function of the mean pressure - pmean.

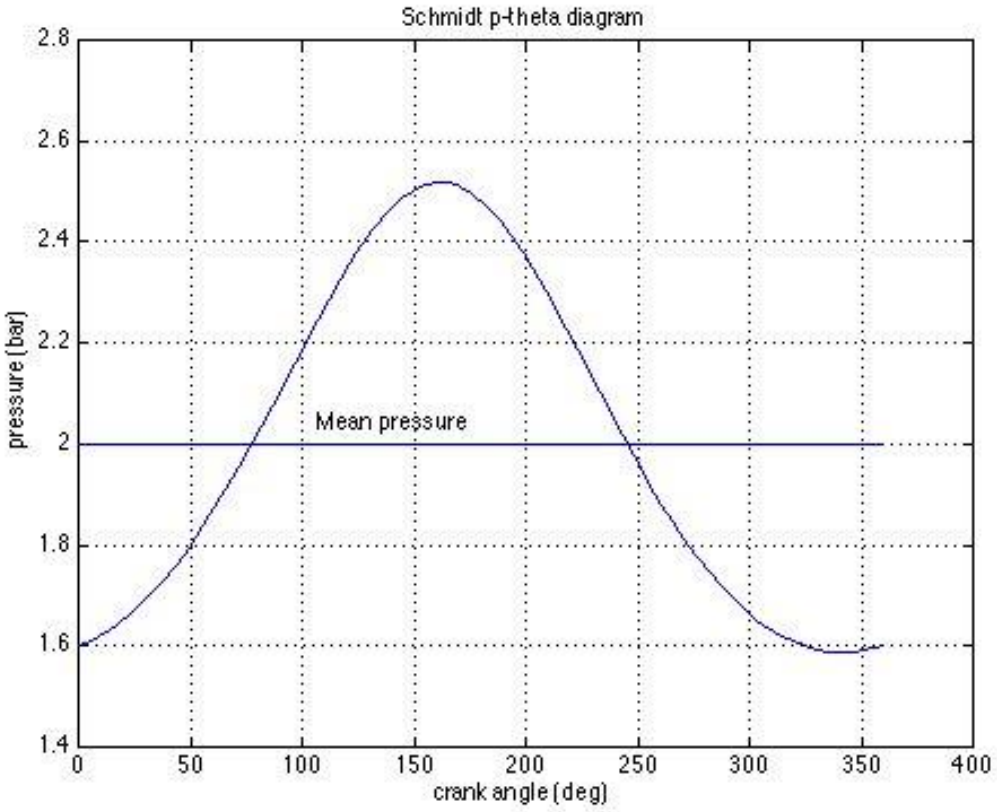
```
function Schmidt
% Schmidt anlysis
% Israel Urieli 3/31/02
global mgas % total mass of gas in engine [kg]
global pmean % mean (charge) pressure [Pa]
global tk tr th % cooler, regen, heater temperatures [K]
global freq omega % cycle frequency [herz], [rads/s]
global vclc vcle % compression,expansion clearence vols [m^3]
global vswc vswe % compression, expansion swept volumes [m^3]
global alpha % phase angle advance of expansion space [radians]
global vk vr vh % cooler, regenerator, heater volumes [m^3]
global rgas % gas constant [J/kg.K]
% Schmidt analysis
c = (((vswe/th)^2 + (vswc/tk)^2 + 2*(vswe/th)*(vswc/tk)*cos(alpha))^0.5)/2;
s = (vswc/2 + vclc + vk)/tk + vr/tr + (vswe/2 + vcle + vh)/th;
b = c/s;
sqrtb = (1 - b^2)^0.5;
bf = (1 - 1/sqrtb);
beta = atan(vswe*sin(alpha)/th/(vswe*cos(alpha)/th + vswc/tk));
fprintf(' pressure phase angle beta  %.1f(degrees)\n', beta*180/pi)
% total mass of working gas in engine
mgas = pmean*s*sqrtb/rgas;
fprintf(' total mass of gas:  %.3f(gm)\n', mgas*1e3)
% work output
wc = (pi*vswc*mgas*rgas*sin(beta)*bf/c);
we = (pi*vswe*mgas*rgas*sin(beta - alpha)*bf/c);
w = (wc + we);
power = w*freq;
eff = w/we; % qe = we
% Printout Schmidt analysis results
fprintf('====================  Schmidt analysis  ===============\n')
fprintf(' Work(joules) %.3e,  Power(watts) %.3e\n', w,power);
fprintf(' Qexp(joules) %.3e,  Qcom(joules) %.3e\n', we,wc);
```

```
fprintf(' indicated efficiency %.3f\n', eff);
fprintf('=========================================================\n')
% Plot Schmidt analysis pv and p-theta diagrams
fprintf('Do you want Schmidt analysis plots\n');
choice = input('y)es or n)o: ','s');
if(strncmp(choice,'y',1))
   plotpv
end
% Plot Allan Organ's particle mass distribution in Natural Coordinates
fprintf('Do you want particle mass distribution plot\n');
choice = input('y)es or n)o: ','s');
if(strncmp(choice,'y',1))
   plotmass
end
```

Function **plotmass** is invoked to do a particle mass distribution plot (refer: **plotmass function**).
Function **plotpv** is invoked to do Schmidt analysis pV and p-theta diagrams, typically as follows:



D-90 Ross Yoke-drive engine pV diagram



D-90 Ross Yoke-drive engine p-theta diagram

```
function plotpv
% plot pv and p-theta diagrams of Schmidt analysis
% Israel Urieli 1/6/03
global vclc vcle % compression,expansion clearence vols [m^3]
global vswc vswe % compression, expansion swept volumes [m^3]
global alpha % phase angle advance of expansion space [radians]
global vk % cooler void volume [m^3]
global vh % heater void volume [m^3]
global vr % regen void volume [m^3]
global mgas % total mass of gas in engine [kg]
global rgas % gas constant [J/kg.K]
global pmean % mean (charge) pressure [Pa]
global tk tr th % cooler, regenerator, heater temperatures [K]
theta = 0:5:360;
```

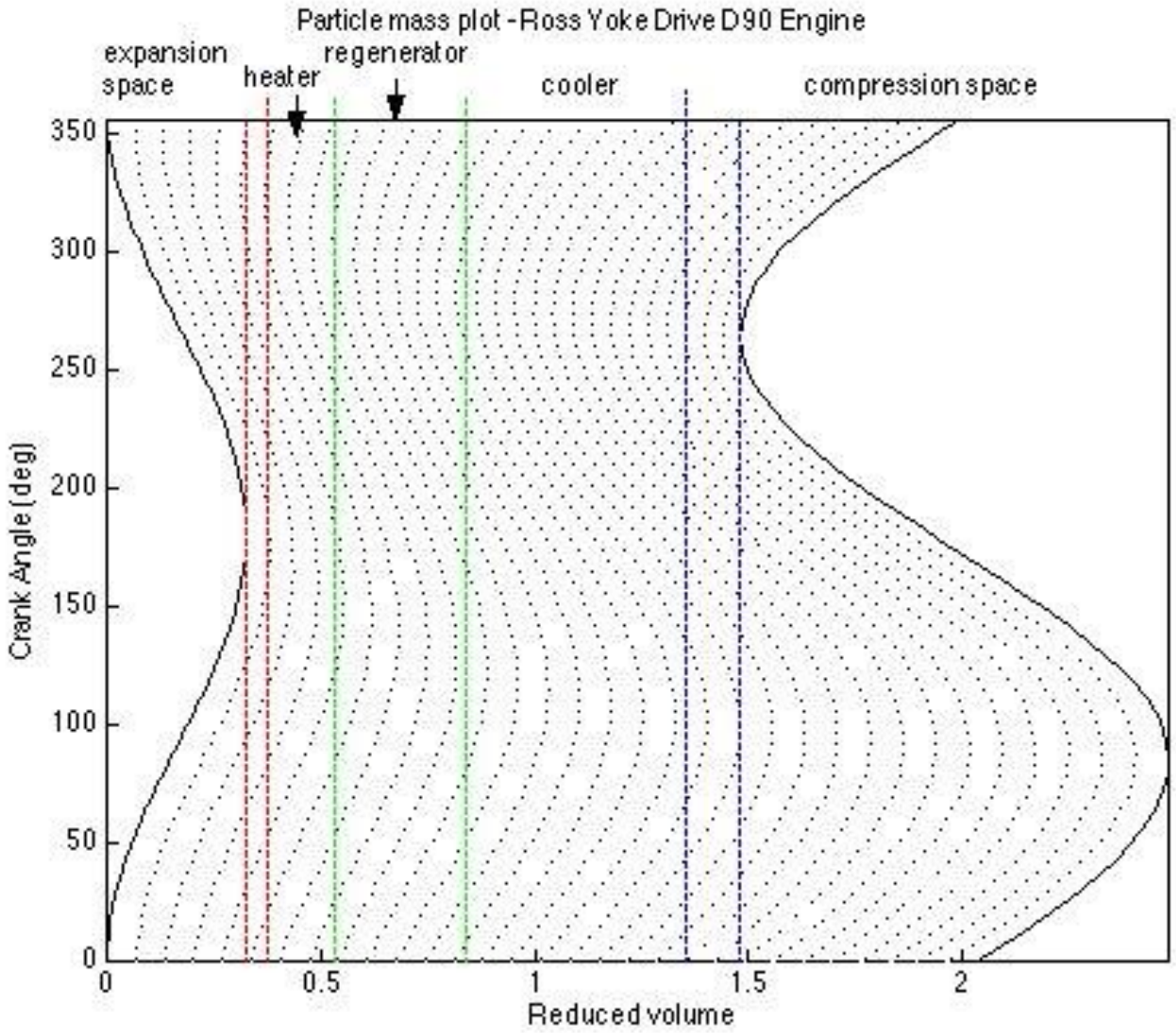
```
vc = vclc + 0.5*vswc*(1 + cos(theta*pi/180));
ve = vcle + 0.5*vswe*(1 + cos(theta*pi/180 + alpha));
p = mgas*rgas./(vc/tk + vk/tk + vr/tr + vh/th + ve/th)*1e-5; % [bar]
vtot = (vc + vk + vr + vh + ve)*1e6; % [cc]
figure
plot(vtot,p)
grid on
xlabel('total volume (cc)')
ylabel('pressure (bar)')
title('Schmidt pv diagram')
figure
plot(theta,p)
grid on
hold on
x = [0,360];
y = [pmean*1e-5, pmean*1e-5];
plot(x,y)
xlabel('crank angle (deg)')
ylabel('pressure (bar)')
title('Schmidt p-theta diagram')
```

Appendix T: Plotmass Function

The particle trajectory function 'plotmass'

The function **plotmass** allows one to do a particle trajectory plot using Natural Coordinates, as defined by Allan Organ: **'Natural' coordinates for analysis of the practical Stirling cycle** (ImechE, 1992). Refer also to Chapter 9 of his book: **The Regenerator and the Stirling Engine** (John Wiley, 1997).

Essentially, for an ideal gas the mass in any volume is inversely proportional to the temperature. Thus for the particle mass trajectory plot we normalize the volumes with respect to the local temperature. In the typical particle trajectory plot shown below the reduced volumes of the expansion and heater spaces are much smaller than those of the cooler and compression spaces. The regenerator is normalized with respect to the regenerator mean effective temperature.



This plot shows particles of equal mass flowing through the engine over a cycle. In this design it appears as though the cooler volume can be reduced, since some mass particles never leave the cooler. This information is not intuitively obvious, and enables a better understanding of the process, however at this stage it should not be used to influence the design of the machine.

The following MATLAB function **plotmass** was developed by students of the Stirling Cycle Machine Analysis course, and is invoked by the function **Schmidt**.

```
function plotmass
% Kyle Wilson 10-2-02
% Stirling Cycle Machine Analysis ME 589
```

```
% Particle Trajectory Map
% Equations from Organ's "'Natural' coordinates for analysis of the practical
% Stirling cycle" and Oegik  Soegihardjo's 1993 project on the same topic
% Modified by Israel Urieli (11/27/2010) to obtain correct phase advance
% angle alpha subsequent to error determined by Zack Alexy (March 2010)

% Global values from define program
global vclc vcle % compression,expansion clearance vols [m^3]
global vswc vswe % compression, expansion swept volumes [m^3]
global alpha % phase angle advance of expansion space [radians]
global vk % cooler void volume [m^3]
global vh % heater void volume [m^3]
global vr % regen void volume [m^3]
global pmean % mean (charge) pressure [Pa]
global tk tr th % cooler, regenerator, heater temperatures [K]

NT = th/tk; % Temperature ratio
Vref = vswe;     % Reference volume (m^3)
%% Fixed reduced volumes (dimensionless)
vswe_r = (vswe/Vref)/NT;     % Reduced expansion swept volume
vcle_r = (vcle/Vref)/NT;       % Reduced expansion clearance volume
vh_r = (vh/Vref)/NT;                % Reduced heater void volume
vr_r = (vr/Vref)*log(NT)/(NT-1);   % Reduced regenerator void volume
vk_r = (vk/Vref);                      % Reduced cooler void volume
vswc_r = (vswc/Vref);          % Reduced compression swept volume
vclc_r = (vclc/Vref);          % Reduced compression clearance volume
%% Phase domain
angi = 0;
angf = 2*pi;
dang = 0.1;
ang = [angi:dang:angf];
n = size(ang);
%% Reduced volume variations
for i = 1:n(2)
   deg(i) = ang(i)*180/pi;
   Ve(i) = (vswe/2)*(1- cos(ang(i)));  % Expansion volume vs phase
   Vc(i) = (vswc/2)*(1+ cos(ang(i) - alpha));  % Compression volume vs phase
   ve(i) = (Ve(i)/Vref)/NT;    % Reduced expansion vs phase
   vc(i) = Vc(i)/Vref;   % Reduced compression vs phase
```

```
   vt(i) = vswe_r + vcle_r + vh_r + vr_r + vk_r + vclc_r + vc(i); % Total volume vs phase
end
figure
step = 30;
for m = 1:step-1
   for i = 1:n(2)
      v(i) = ve(i) + (m/step)*(vt(i)-ve(i)); % Reduced volume segments
   end
   hold on
   plot(v,deg, 'k:')
end
hold on
plot(ve,deg,'k')
plot(vt,deg, 'k')
%% Vertical lines
L1 = vswe_r;   % Boundary of reduced expansion swept volume
L2 = L1 + vcle_r;  % Boundary of reduced expansion clearance volume
L3 = L2 + vh_r;   % Boundary of reduced heater void volume
L4 = L3 + vr_r;   % Boundary of reduced regenerator void volume
L5 = L4 + vk_r;   % Boundary of reduced cooler void volume
L6 = min(vt);     % Boundary of reduced expansion swept volume
point1 = [L1;L1];   % Preparing for plot
point2 = [L2;L2];
point3 = [L3;L3];
point4 = [L4;L4];
point5 = [L5;L5];
point6 = [L6;L6];
point = [0;deg(n(2))];
plot(point1, point, 'r--', point2, point, 'r--', point3, point, 'g--')
plot(point4, point, 'g--', point5, point, 'b--', point6, point, 'b--')
axis([0 max(vt) 0 deg(n(2))])
xlabel('Reduced volume')
ylabel('Crank Angle (deg)')
title('Particle mass plot')
hold off
```

Appendix U: An Approach to Solving Ordinary Differential Equations

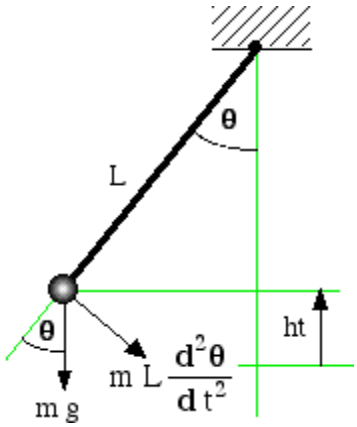
In many real life applications where we wish to evaluate a function, we are only able to define the relationship that the derivative of that function must satisfy. One example is the dynamics of a vehicle in which we wish to determine the velocity or acceleration, however the unknowns are specified only in terms of ordinary differential equations (ODEs) and initial conditions. In this note we wish to show one approach to the numerical solution of ODEs using MATLAB and the Classical fourth order Runge-Kutta method. This is a very personal approach and is somewhat different to that normally used, however I will try to justify it further on. The simplest non-trivial example that I could think of to develop this approach is that of the large angle pendulum

Consider a simple pendulum having length L, mass m and instantaneous angular displacement θ (theta [radians]), as shown below:

From a force balance we obtain:

$$m\,g\,\sin(\theta) + m\,L\,\frac{d^2\theta}{dt^2} = 0$$

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L}\sin(\theta)$$



For **small initial angles θ** we make the assumption that **sin(**θ**)**=θ leading to the well known analytical solution:

If a pendulum's swinging quite free
Then it's always a marvel to me
That each tick plus each tock
Of the grandfather clock
Is 2 pi root L over g

$$T = 2\pi\sqrt{\frac{L}{g}}$$

(Refer: [Top 150 Limericks])

However for large angles θ we resort to a numerical method to solve the differential equation above. We first reduce the second order differential equation to a set of two first order differential equations by introducing ω (omega - angular velocity [radians/s]), leading to:

$$\frac{d\theta}{dt} = \omega$$
$$\frac{d\omega}{dt} = -\frac{g}{L}\sin(\theta)$$

Fortunately we can always reduce a n-th order ODE into a n-vector set of first order ODEs. Thus we need only concentrate on developing numerical techniques for solving a vector set of first order ODEs. Processing vectors by computer simply requires looping sequentially through the indices of the vector, and MATLAB is particularly geared to this type of analysis. MATLAB provides five separate routines for solving ODEs of the form dy = f(x,y) where x is the independent variable and y and dy are the solution and derivative vectors. They are extremely simple to use, however they do not provide the versatility that I am looking for, including choosing the step size dx and overloading the y-vector. For example consider the case that we wish to determine the height (ht) of the pendulum mass as shown in the above diagram. This does not require a differential equation since the height is related to the angle θ as follows:

$$ht = L\,(1 - \cos(\theta))$$

Consider now the practical aspects of writing a function for evaluating this set of derivatives. One problem is that vectors are usually assigned generic names such as y for the set of dependent variables and dy for their derivatives, however we would like to retain their identity in terms of θ (theta), ω (omega) and height. This makes the programs much more readable, and can be easily done by means of indices, as shown in the MATLAB function below:

```
function [y,dy] = dpend(t,y)
% dpend.m - The pendulum differential equations
% g [m/s^2] acceleration due to gravity
% length [m] pendulum length
% Izzi Urieli - Jan 21, 2002 (updated 12/08/07)
global g length
THETA = 1; % index of angle [radians]
OMEGA = 2; % index of angular velocity [rads/sec]
HEIGHT = 3; % index of pendulum mass height [m]
```

```
dy(THETA) = y(OMEGA);
dy(OMEGA) = -(g/length)*sin(y(THETA));
y(HEIGHT) = length*(1 - cos(y(THETA)));
```

Notice the use of upper case characters for the indices as is the convention in programming, however all other variables (including the global variables) begin with a lower case letter. Notice also both y and dy are returned (instead of only dy as is normally done). This is because the y-vector has been "overloaded" with the height variable which is evaluated as a function of θ in this function. This option is not available in MATLAB using their conventional ODE solving routines, however I find that I need to use it extensively.

Consider now a typical main function for this case study:

```
% pend_rk4.m - Solve the large angle pendulum problem using rk4
% Izzi Urieli - December 8, 2007
clear all
THETA = 1; % index of angle [radians]
OMEGA = 2; % index of angular velocity [rads/sec]
HEIGHT = 3; % index of pendulum mass height [m]
global g length
g = 9.807; % [m/s/s} accel. due to gravity
length = input('Enter pendulum length [m]: ');
angle0 = input('Enter initial pendulum angle [degrees]: ');
period = 2*pi*sqrt(length/g);
fprintf('small angle period is %.3f seconds\n',period);
tfinal = 1.5*period;
%######################using rk4:
n = input('Enter the number of solution points: ');
dt = tfinal/(n - 1);
t = 0;
y(THETA) = angle0*pi/180; % initial pendulum angle [radians]
y(OMEGA) = 0;
y(HEIGHT) = length*(1 - cos(y(THETA))); % initial pend. height [m]
time(1) = t;
angle(1) = angle0;
height(1) = y(HEIGHT)*100; % cm (for scaling on the plot)
for i = 2:n
    [t, y, dy] = rk4('dpend',2,t,dt,y);
```

```
    time(i) = t;
    angle(i) = y(THETA)*180/pi;
  height(i) = y(HEIGHT)*100;
end
%##############################plotting:
plot(time,angle)
grid on
hold on
axis([0 tfinal -90 90])
x0 = [0,tfinal];
y0 = [0,0];
plot(x0,y0,'k-')
x1 = [period,period];
y1 = [-90,90];
plot(x1,y1,'r-')
plot(time,height, 'g-')
xlabel('time [seconds]')
ylabel('pendulum angle[degrees], height[cm] (green)')
title('pendulum angle and height vs time')
```

As with many MATLAB functions, this one is very short and mostly self-documenting. Notice that after specifying the system parameters we use the for loop to build the time, angle, and height arrays in order to plot them. The most interesting aspect of this program is the rk4 function call, to integrate one step dt of the differential equation set given in function dpend, as follows:

```
[t, y, dy] = rk4('dpend',2,t,dt,y);
```

Thus the input arguments are the string constant representing the derivative function dpend, the number of differential equations in the set (2), the independent variable and increment (t,dt), and the dependent variable vector (y). The output arguments are the solution variables and derivatives [t,y,dy] integrated over one time step dt.

One of the most widely used of all the single step Runge-Kutta methods (which include the Euler and Modified Euler methods) is the so called 'Classical' fourth order method with Runge's coefficients. Being a fourth order method (equivalent in accuracy to including the first five terms of the Taylor series expansion of the solution) it requires four evaluations of the derivatives over each increment dx, denoted respectively by

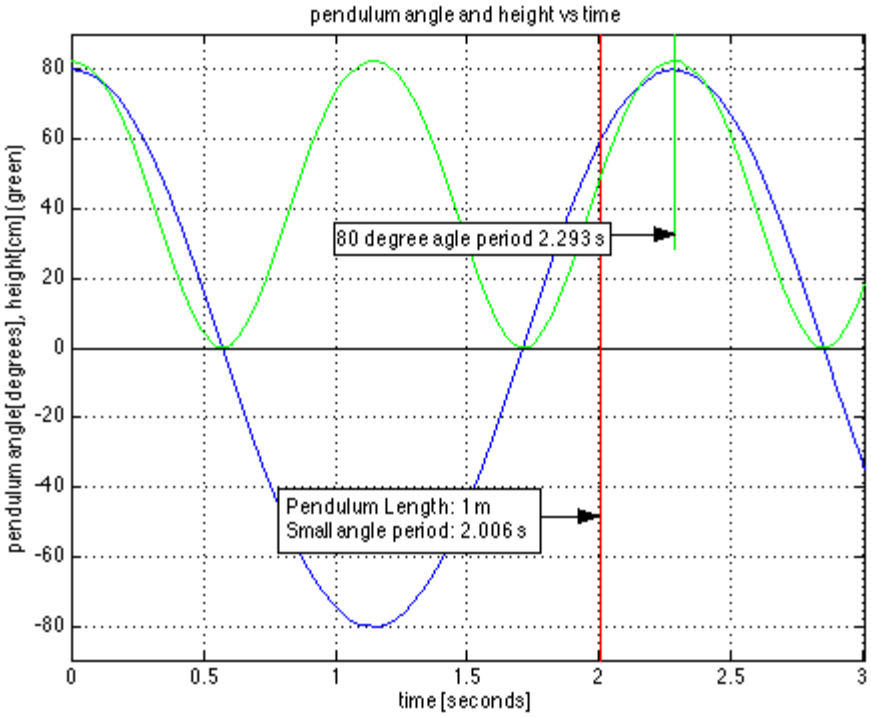
dy1, dy2, dy3, dy4. These are then weighted and summed in a very specific manner to obtain the final derivative dy, as follows:

```
function [x, y, dy] = rk4(deriv,n,x,dx,y)
% rk4.m - Classical fourth order Runge-Kutta method
%Integrates n first order differential equations
%dy(x,y) over interval x to x+dx
%Izzi Urieli - Jan 21, 2002
x0 = x;
y0 = y;
[y,dy1] = feval(deriv,x0,y);
for i = 1:n
      y(i) = y0(i) + 0.5*dx*dy1(i);
end
xm = x0 + 0.5*dx;
[y,dy2] = feval(deriv,xm,y);
for i = 1:n
      y(i) = y0(i) + 0.5*dx*dy2(i);
end
[y,dy3] = feval(deriv,xm,y);
for i = 1:n
      y(i) = y0(i) + dx*dy3(i);
end
x = x0 + dx;
[y,dy4] = feval(deriv,x,y);
for i = 1:n
      dy(i) = (dy1(i) + 2*(dy2(i) + dy3(i)) + dy4(i))/6;
      y(i) = y0(i) + dx*dy(i);
end
```

The most interesting aspect of this function is the use of the MATLAB function feval to evaluate the function argument which is passed as a string. This is a fundamental aspect of MATLAB programming, and understanding this is essential to developing MATLAB programs.

The three MATLAB function mfiles shown above **pend_rk4.m**, **dpend.m**, and **rk4.m** can be downloaded and executed as required.

The plotting capabilities of MATLAB are extremely simple to use and very versatile. A typical output plot (with labels added) for this pendulum case study follows:



Notice that the small angle period is about 2..0 s, however at 80 degrees the period is seen to be significantly increased (2.293 s). This could only have been determined by numerically solving the nonlinear differential equation set. Notice also the height plot which was added in order to illustrate the method of "overloading" the y-vector.

I asked **Dr. John Cotton** (our MATLAB guru) to comment on this approach - his response was:

"Matlab a few years back transitioned to using function handles in place of feval (which may be phased out in the future). If for some reason you had a nasty function or just wanted to use the Matlab solvers (ODExxx) you could still do what you are doing here. The options allow controlling step sizes (or error tolerances) for the canned solvers. (This is beyond what many students are shown or educated to use however, which I tend to drone on about to them.) There are ways of getting the heights out as a separate variable using extended argument lists so you don't need to overload the y vector, although I would have calculated the height vector from the angles right before plotting in your main function. (Then again "toMAYto, toMAHto".)"

Oh well! Obviously the Large Angle Pendulum problem presented above is a contrived example. This technical note serves mainly as an introduction to solving significantly more complex problems, such as the **Adiabatic Analysis of Stirling Cycle Machines** - (Then again "poTAYto, poTAHto".)

Appendix V: Function Set “simple”

Appendix V.1: The main program 'sea' (stirling engine analysis)

We first invoke the **define** function (Chapter 3C) set to specify all the global variables needed to simulate a specific engine system and do a Schmidt analysis, and then invoke either the **ideal adiabatic** or **simple** analysis functions.

```
% sea (stirling engine analysis) - main program
%Israel Urieli 7/20/02
clc;
clear all;
% define a specific engine
define;
choice = 'x';
while(~strncmp(choice,'q',1))
        fprintf('Choose simulation:\n');
        choice = input('a)diabatic, s)imple q)uit: ','s');
        if(strncmp(choice,'a',1))
                [var,dvar] = adiabatic;
        elseif(strncmp(choice,'s',1))
                [var,dvar] = simple;
        end
end
fprintf('quitting simulation...\n');
```

Appendix V.2: The Heater Heat Transfer Performance Function 'hotsim'

```
function  tgh = hotsim(var,twh,qrloss)
% evaluate heater average heat transfer performance
% Israel Urieli, 7/22/2002
% Modified 2/6/2010 to include regenerator qrloss
% Arguments:
%   var(22,37) array of variable values every 10 degrees (0 - 360)
%   twh - heater wall temperature [K]
%   qrloss - heat loss due to imperfect regenerator [J]
% Returned values:
%   tgh - heater average gas temperature [K]

% Row indices of the var, array:
 TC = 1;   % Compression space temperature (K)
 TE = 2;  % Expansion space temperature (K)
 QK = 3;   % Heat transferred to the cooler (J)
 QR = 4;  % Heat transferred to the regenerator (J)
 QH = 5;  % Heat transferred to the heater (J)
 WC = 6;   % Work done by the compression space (J)
 WE = 7;  % Work done by the expansion space (J)
 W  = 8;  % Total work done (WC + WE) (J)
 P  = 9;  % Pressure (Pa)
 VC = 10; % Compression space volume (m^3)
 VE = 11; % Expansion space volume (m^3)
 MC = 12; % Mass of gas in the compression space (kg)
 MK = 13; % Mass of gas in the cooler (kg)
 MR = 14; % Mass of gas in the regenerator (kg)
 MH = 15; % Mass of gas in the heater (kg)
 ME = 16; % Mass of gas in the expansion space (kg)
 TCK = 17; % Conditional temperature compression space / cooler (K)
 THE = 18; % Conditional temeprature heater / expansion space (K)
 GACK = 19; % Conditional mass flow compression space / cooler (kg/rad)
 GAKR = 20; % Conditional mass flow cooler / regenerator (kg/rad)
 GARH = 21; % Conditional mass flow regenerator / heater (kg/rad)
 GAHE = 22; % Conditional mass flow heater / expansion space (kg/rad)

global  th % heater temperature [K]
global  freq omega % cycle frequency [herz], [rads/s]
```

```
global  ah % heater internal free flow area [m^2]
global  awgh % heater internal wetted area [m^2]
global  dh % heater hydraulic diameter [m]

% Calculating the Reynolds number over the cycle
for(i = 1:1:37)
        gah(i) = (var(GARH,i) + var(GAHE,i))*omega/2;
        gh = gah(i)/ah;
        [mu,kgas,re(i)] = reynum(th,gh,dh);
end

% Average and maximum Reynolds number
sumre=0;
remax=re(1);
for  (i=1:1:36)
        sumre=sumre + re(i);
        if (re(i) > remax)
                remax = re(i);
        end
end
reavg = sumre/36;

[ht,fr] = pipefr(dh,mu,reavg); % Heat transfer coefficient
tgh = twh - (var(QH,37)+qrloss)*freq/(ht*awgh); % Heater gas temperature [K]

fprintf('============ Heater Simple analysis ============\n')
fprintf(' Average Reynolds number : %.1f\n',reavg)
fprintf('  Maximum Reynolds number : %.1f\n',remax)
fprintf(' Heat transfer coefficient [W/m^2*K] : %.2f\n',ht)
fprintf('heater wall/gas temperatures: Twh = %.1f[K], Tgh = %.1f[K]\n',twh,tgh);
```

Appendix V.3: The Cooler Heat Transfer Performance Function 'kolsim'

```
function  tgk = kolsim(var,twk,qrloss)
% evaluate cooler average heat transfer performance
% Israel Urieli, 7/22/2002
% Modified 2/6/2010 to include regenerator qrloss
% Arguments:
%   var(22,37) array of variable values every 10 degrees (0 - 360)
%   twk - cooler wall temperature [K]
%   qrloss - heat loss due to imperfect regenerator [J]
% Returned values:
%   tgk - cooler average gas temperature [K]

% Row indices of the var, array:
 TC = 1;   % Compression space temperature (K)
 TE = 2;  % Expansion space temperature (K)
 QK = 3;   % Heat transferred to the cooler (J)
 QR = 4;  % Heat transferred to the regenerator (J)
 QH = 5;  % Heat transferred to the heater (J)
 WC = 6;   % Work done by the compression space (J)
 WE = 7;  % Work done by the expansion space (J)
 W  = 8;  % Total work done (WC + WE) (J)
 P  = 9;  % Pressure (Pa)
 VC = 10; % Compression space volume (m^3)
 VE = 11; % Expansion space volume (m^3)
 MC = 12; % Mass of gas in the compression space (kg)
 MK = 13; % Mass of gas in the cooler (kg)
 MR = 14; % Mass of gas in the regenerator (kg)
 MH = 15; % Mass of gas in the heater (kg)
 ME = 16; % Mass of gas in the expansion space (kg)
 TCK = 17; % Conditional temperature compression space / cooler (K)
 THE = 18; % Conditional temeprature heater / expansion space (K)
 GACK = 19; % Conditional mass flow compression space / cooler (kg/rad)
 GAKR = 20; % Conditional mass flow cooler / regenerator (kg/rad)
 GARH = 21; % Conditional mass flow regenerator / heater (kg/rad)
 GAHE = 22; % Conditional mass flow heater / expansion space (kg/rad)

global  tk % cooler temperature [K]
global  freq omega % cycle frequency [herz], [rads/s]
```

```
global  ak % cooler internal free flow area [m^2]
global  awgk % cooler internal wetted area [m^2]
global  dk % cooler hydraulic diameter [m]

% Calculating the Reynolds number over the cycle
for(i = 1:1:37)
        ak(i) = (var(GACK,i) + var(GAKR,i))*omega/2;
        gk = gak(i)/ak;
        [mu,kgas,re(i)] = reynum(tk,gk,dk);
end

% Average and maximum Reynolds number
sumre=0;
remax=re(1);
for  (i=1:1:36)
        sumre=sumre + re(i);
        if (re(i) > remax)
                remax = re(i);
        end
end
reavg = sumre/36;

[ht,fr] = pipefr(dk,mu,reavg); % Heat transfer coefficient
tgk = twk - (var(QK,37)-qrloss)*freq/(ht*awgk); % Heater gas temperature [K]

fprintf('============ Cooler Simple analysis ============\n')
fprintf(' Average Reynolds number : %.1f\n',reavg)
fprintf(' Maximum Reynolds number : %.1f\n',remax)
fprintf(' Heat transfer coefficient [W/m^2*K] : %.2f\n',ht)
fprintf('cooler wall/gas temperatures: Twk = %.1f[K], Tk = %.1f[K]\n',twk,tgk);
```

Appendix V.4: The Regenerator Enthalpy Loss Function 'regsim'

```
function  qrloss = regsim(var)
% Evaluate the effectiveness and performance of the regenerator
% Israel Urieli, 7/23/2002 - Modified 2/15/2010
% modified 11/27/2010 to include 'no regenerator matrix'
% Arguments:
%   var(22,37) array of variable values every 10 degrees (0 - 360)
% Returned value:
%   qrloss - regenerator net enthalpy loss [J]

% Row indices of the var, array
 TC = 1;   % Compression space temperature (K)
 TE = 2;  % Expansion space temperature (K)
 QK = 3;   % Heat transferred to the cooler (J)
 QR = 4;  % Heat transferred to the regenerator (J)
 QH = 5;  % Heat transferred to the heater (J)
 WC = 6;   % Work done by the compression space (J)
 WE = 7;  % Work done by the expansion space (J)
 W  = 8;  % Total work done (WC + WE) (J)
 P  = 9;  % Pressure (Pa)
 VC = 10; % Compression space volume (m^3)
 VE = 11; % Expansion space volume (m^3)
 MC = 12; % Mass of gas in the compression space (kg)
 MK = 13; % Mass of gas in the cooler (kg)
 MR = 14; % Mass of gas in the regenerator (kg)
 MH = 15; % Mass of gas in the heater (kg)
 ME = 16; % Mass of gas in the expansion space (kg)
 TCK = 17; % Conditional temperature compression space / cooler (K)
 THE = 18; % Conditional temeprature heater / expansion space (K)
 GACK = 19; % Conditional mass flow compression space / cooler (kg/rad)
 GAKR = 20; % Conditional mass flow cooler / regenerator (kg/rad)
 GARH = 21; % Conditional mass flow regenerator / heater (kg/rad)
 GAHE = 22; % Conditional mass flow heater / expansion space (kg/rad)

global  matrix_type % m)esh or f)oil
global  ar % regen internal free flow area [m^2]
global  awgr % regen internal wetted area [m^2]
global  dr % regen hydraulic diameter [m]
global  tr % regen temperature [K]
```

```
global  freq omega % cycle frequency [herz], [rads/s]

% Reynolds number over the cycle
for(i = 1:1:37)
      gar(i) = (var(GAKR,i) + var(GARH,i))*omega/2;
      gr = gar(i)/ar;
      [mu,kgas,re(i)] = reynum(tr,gr,dr);
end

% average and maximum Reynolds number
sumre = 0;
remax = re(1);
for(i = 1:1:36)
          sumre = sumre + re(i);
          if (re(i) > remax)
                remax = re(i);
          end
end
reavg = sumre/36;

% Stanton number, number of transfer units, regenerator effectiveness
if (strncmp(matrix_type,'m',1))
          [st,fr] = matrixfr(reavg);
elseif (strncmp(matrix_type,'f',1))
          [st,ht,fr] = foilfr(dr,mu,reavg);
elseif (strncmp(matrix_type,'n',1))
          [st,ht,fr] = foilfr(dr,mu,reavg);
end
ntu = st*awgr/(2*ar);
effect = ntu/(ntu + 1);

% Calculate qrloss
for  (i=1:1:37)
          qreg(i) = var(QR,i);
end
qrmin = min(qreg);
qrmax = max(qreg);
qrloss = (1 - effect)*(qrmax - qrmin);

% Regenerator simple analysis results:
fprintf('============ Regenerator Simple analysis ============\n')
fprintf('Average Reynolds number: %.1f\n', reavg);
```

```
fprintf('Maximum Reynolds number: %.1f\n', remax);
fprintf('Stanton number(Average Re): %.3f\n',st);
fprintf('Number of transfer units: %.1f\n',ntu);
fprintf('Regenerator effectiveness : %.3f\n',effect);
fprintf('Regenerator net enthalpy loss: %.1f[W]\n', qrloss*freq);
```

Appendix V.5: The Pressure Drop Available Work Loss Function 'worksim'

```
function  dwork = worksim(var,dvar);
% Evaluate the pressure drop available work loss [J]
% Israel Urieli, 7/23/2002
% Arguments:
%   var(22,37) array of variable values every 10 degrees (0 - 360)
%   dvar(16,37) array of derivatives every 10 degrees (0 - 360)
% Returned value:
%   dwork - pressure drop available work loss [J]

% Row indices of the var, dvar arrays:
 TC = 1;   % Compression space temperature (K)
 TE = 2;  % Expansion space temperature (K)
 QK = 3;   % Heat transferred to the cooler (J)
 QR = 4;  % Heat transferred to the regenerator (J)
 QH = 5;  % Heat transferred to the heater (J)
 WC = 6;   % Work done by the compression space (J)
 WE = 7;  % Work done by the expansion space (J)
 W  = 8;  % Total work done (WC + WE) (J)
 P  = 9;  % Pressure (Pa)
 VC = 10; % Compression space volume (m^3)
 VE = 11; % Expansion space volume (m^3)
 MC = 12; % Mass of gas in the compression space (kg)
 MK = 13; % Mass of gas in the cooler (kg)
 MR = 14; % Mass of gas in the regenerator (kg)
 MH = 15; % Mass of gas in the heater (kg)
 ME = 16; % Mass of gas in the expansion space (kg)
 TCK = 17; % Conditional temperature compression space / cooler (K)
 THE = 18; % Conditional temeprature heater / expansion space (K)
 GACK = 19; % Conditional mass flow compression space / cooler (kg/rad)
 GAKR = 20; % Conditional mass flow cooler / regenerator (kg/rad)
 GARH = 21; % Conditional mass flow regenerator / heater (kg/rad)
 GAHE = 22; % Conditional mass flow heater / expansion space (kg/rad)
% Size of var(ROWV,COL), dvar(ROWD,COL)
 ROWV = 22; % number of rows in the var matrix
 ROWD = 16; % number of rows in the dvar matrix
 COL = 37; % number of columns in the matrices (every 10 degrees)
```

```
%==========================================================
===========

global  tk tr th % cooler, regenerator, heater temperatures [K]
global  freq omega % cycle frequency [herz], [rads/s]
global  vh % heater void volume [m^3]
global  ah % heater internal free flow area [m^2]
global  dh % heater hydraulic diameter [m]
global  lh % heater effective length [m]
global  vk % cooler void volume [m^3]
global  ak % cooler internal free flow area [m^2]
global  dk % cooler hydraulic diameter [m]
global  lk % cooler effective length [m]
global  vr % regen void volume [m^3]
global  ar % regen internal free flow area [m^2]
global  lr % regenerator effective length [m]
global  dr % regen hydraulic diameter [m]
global  matrix_type % m)esh or f)oil

dtheta = 2*pi/36;
dwork = 0; % initialise pumping work loss

for(i = 1:1:36)
        gk = (var(GACK,i) + var(GAKR,i))*omega/(2*ak);
        [mu,kgas,re(i)] = reynum(tk,gk,dk);
        [ht,fr] = pipefr(dk,mu,re(i));
        dpkol(i) = 2*fr*mu*vk*gk*lk/(var(MK,i)*dk^2);

        gr = (var(GAKR,i) + var(GARH,i))*omega/(2*ar);
        [mu,kgas,re(i)] = reynum(tr,gr,dr);
        if(strncmp(matrix_type,'m',1))
                [st,fr] = matrixfr(re(i));
        elseif  (strncmp(matrix_type,'f',1))
                [st,ht,fr] = foilfr(dr,mu,re(i));
        end
        dpreg(i) = 2*fr*mu*vr*gr*lr/(var(MR,i)*dr^2);

        gh = (var(GARH,i) + var(GAHE,i))*omega/(2*ah);
        [mu,kgas,re(i)] = reynum(th,gh,dh);

        [ht,fr] = pipefr(dh,mu,re(i));
        dphot(i) = 2*fr*mu*vh*gh*lh./(var(MH,i)*dh^2);
        dp(i) = dpkol(i) + dpreg(i) + dphot(i);
```

```
        dwork=dwork+dtheta*dp(i)*dvar(VE,i); % pumping work [J]
        pcom(i) = var(P,i);
        pexp(i) = pcom(i) + dp(i);
end

dpkol(COL) = dpkol(1);
dpreg(COL) = dpreg(1);
dphot(COL) = dphot(1);
dp(COL) = dp(1);
pcom(COL) = pcom(1);
pexp(COL) = pexp(1);

choice = 'x';
while(~strncmp(choice,'q',1))
    fprintf('Choose pumping loss plot type:\n');
    fprintf('   h - for heat exchanger pressure drop plot\n');
    fprintf('   p - for working space pressure plot\n');
    fprintf('   q - to quit\n');
    choice = input('h)x_pdrop, p)ressure, q)uit: ','s');
    if(strncmp(choice,'h',1))
        figure;
        x = 0:10:360;
        plot(x,dpkol,'b-',x,dphot,'r-',x,dpreg,'g-');
        grid on
        xlabel('Crank angle (degrees)');
        ylabel('Heat exchanger pressure drop [Pa]');
        title('Heat exchanger pressure drop vs crank angle');
    elseif(strncmp(choice,'p',1))
        figure
        x = 0:10:360;
        pcombar = pcom*1e-5;
        pexpbar = pexp*1e-5;
        plot(x,pcombar,'b-',x,pexpbar,'r-');
        grid on
        xlabel('Crank angle (degrees)');
        ylabel('Working space pressure [bar]');
        title('Working space pressure vs crank angle');
    end
end
fprintf('quitting pressure plots...\n');
```

Appendix W: The Ideal Adiabatic Model Function 'adiabatic'

The purpose of the adiabatic function set is to determine the numerical solution of the ideal adiabatic model equation set (refer to the **Equation Summary and Method of Solution**). Recall in the equation set that apart from the constants, there are 22 variables and 16 derivatives to be solved over a complete cycle (θ = [0, 2π]) as follows:
Tc, Te, Qk, Qr, Qh, Wc, We - seven derivatives to be integrated numerically
W, p, Vc, Ve, mc, mk, mr, mh, me - nine analytical variables and derivatives
Tck, The, mck', mkr', mrh', mhe' - six conditional and mass flow variables (derivatives undefined)

Function **adiabatic** invokes the function set **adiab** to fill in the solution matrices **var** and **dvar** over a complete cycle, and then invokes function **plotadiab** (shown below) to do the required plots of the simulation results.

```
function  [var,dvar] = adiabatic
% ideal adiabatic simulation and temperature/energy vs theta plots
% Israel Urieli, 7/20/2002
% Returned values:
%   var(22,37) array of variable values every 10 degrees (0 - 360)
%   dvar(16,37) array of derivatives every 10 degrees (0 - 360)
% Row indices of the var, dvar arrays:
 TC = 1;   % Compression space temperature (K)
 TE = 2;  % Expansion space temperature (K)
 QK = 3;   % Heat transferred to the cooler (J)
 QR = 4;  % Heat transferred to the regenerator (J)
 QH = 5;  % Heat transferred to the heater (J)
 WC = 6;   % Work done by the compression space (J)
 WE = 7;  % Work done by the expansion space (J)
 W  = 8;  % Total work done (WC + WE) (J)
 P  = 9;  % Pressure (Pa)
 VC = 10; % Compression space volume (m^3)
 VE = 11; % Expansion space volume (m^3)
 MC = 12; % Mass of gas in the compression space (kg)
 MK = 13; % Mass of gas in the cooler (kg)
 MR = 14; % Mass of gas in the regenerator (kg)
```

```
 MH = 15; % Mass of gas in the heater (kg)
 ME = 16; % Mass of gas in the expansion space (kg)
 TCK = 17; % Conditional temperature compression space / cooler (K)
 THE = 18; % Conditional temeprature heater / expansion space (K)
 GACK = 19; % Conditional mass flow compression space / cooler (kg/rad)
 GAKR = 20; % Conditional mass flow cooler / regenerator (kg/rad)
 GARH = 21; % Conditional mass flow regenerator / heater (kg/rad)
 GAHE = 22; % Conditional mass flow heater / expansion space (kg/rad)
% Size of var(ROWV,COL), dvar(ROWD,COL)
 ROWV = 22; % number of rows in the var matrix
 ROWD = 16; % number of rows in the dvar matrix
 COL = 37; % number of columns in the matrices (every 10 degrees)
%=========================================================
global  freq % cycle frequency [herz]
global  tk tr th % cooler, regenerator, heater temperatures [K]
global  vk % cooler void volume [m^3]
global  vr % regen void volume [m^3]
global  vh % heater void volume [m^3]
% do ideal adiabatic analysis:
[var,dvar] = adiab;
% Print out ideal adiabatic analysis results
eff = var(W,COL)/var(QH,COL); % engine thermal efficency
Qkpower = var(QK,COL)*freq;   % Heat transferred to the cooler (W)
Qrpower = var(QR,COL)*freq;   % Heat transferred to the regenerator (W)
Qhpower = var(QH,COL)*freq;   % Heat transferred to the heater (W)
Wpower = var(W,COL)*freq;     % Total power output (W)
fprintf('========== ideal adiabatic analysis results ============\n')
fprintf(' Heat transferred to the cooler: %.2f[W]\n', Qkpower);
fprintf(' Net heat transferred to the regenerator: %.2f[W]\n', Qrpower);
fprintf(' Heat transferred to the heater: %.2f[W]\n', Qhpower);
fprintf(' Total power output: %.2f[W]\n', Wpower);
fprintf(' Thermal efficiency : %.1f[%%]\n', eff*100);
fprintf('=========================================================\n')
% Various plots of the ideal adiabatic simulation results
plotadiab(var,dvar);
```

Function **plotadiab** is invoked both by function **adiabatic** (above) and by function **simple**.

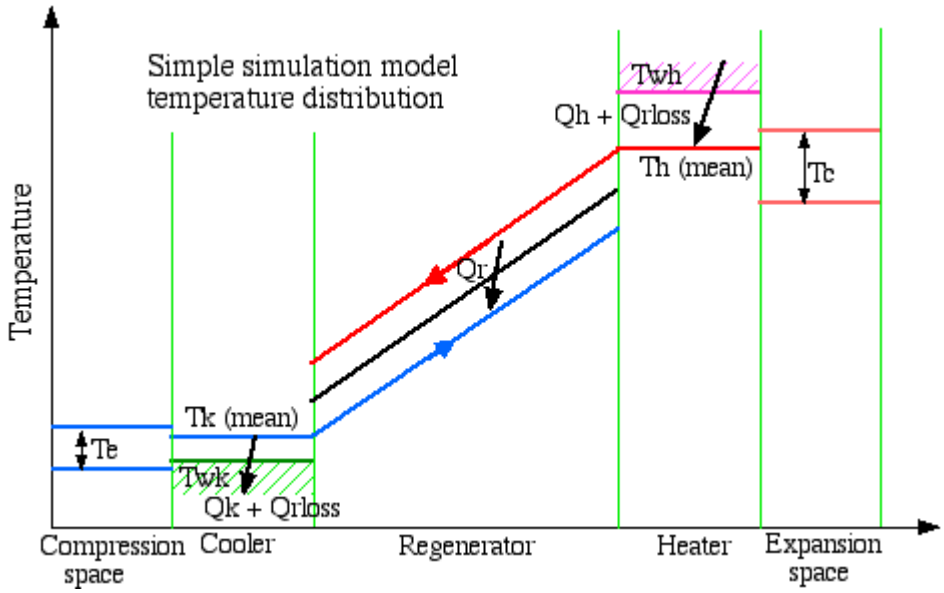
```
function  plotadiab(var,dvar)
% various plots of ideal adiabatic simulation results
% Israel Urieli, 7/21/2002 (corrected temp plots 12/3/2003)
% Arguments:
%   var(22,37) array of variable values every 10 degrees (0 - 360)
%   dvar(16,37) array of derivatives every 10 degrees (0 - 360)
% Row indices of the var, dvar arrays:
  TC = 1;   % Compression space temperature (K)
  TE = 2;  % Expansion space temperature (K)
  QK = 3;   % Heat transferred to the cooler (J)
  QR = 4;  % Heat transferred to the regenerator (J)
  QH = 5;  % Heat transferred to the heater (J)
  WC = 6;   % Work done by the compression space (J)
  WE = 7;  % Work done by the expansion space (J)
  W  = 8;  % Total work done (WC + WE) (J)
  P  = 9;  % Pressure (Pa)
  VC = 10; % Compression space volume (m^3)
  VE = 11; % Expansion space volume (m^3)
  MC = 12; % Mass of gas in the compression space (kg)
  MK = 13; % Mass of gas in the cooler (kg)
  MR = 14; % Mass of gas in the regenerator (kg)
  MH = 15; % Mass of gas in the heater (kg)
  ME = 16; % Mass of gas in the expansion space (kg)
  TCK = 17; % Conditional temperature compression space / cooler (K)
  THE = 18; % Conditional temeprature heater / expansion space (K)
  GACK = 19; % Conditional mass flow compression space / cooler (kg/rad)
  GAKR = 20; % Conditional mass flow cooler / regenerator (kg/rad)
  GARH = 21; % Conditional mass flow regenerator / heater (kg/rad)
  GAHE = 22; % Conditional mass flow heater / expansion space (kg/rad)
% Size of var(ROWV,COL), dvar(ROWD,COL)
  ROWV = 22; % number of rows in the var matrix
  ROWD = 16; % number of rows in the dvar matrix
  COL = 37; % number of columns in the matrices (every 10 degrees)
%=============================================================
global  tk tr th % cooler, regenerator, heater temperatures [K]
global  vk % cooler void volume [m^3]
global  vr % regen void volume [m^3]
global  vh % heater void volume [m^3]
choice = 'x';
while(~strncmp(choice,'q',1))
```

```
fprintf('Choose plot type:\n');
fprintf('   p - for a PV diagram\n');
fprintf('   t - for a temperature vs crank angle plot\n');
fprintf('   e - for an energy vs crank angle plot\n');
fprintf('   q - to quit\n');
choice = input('p)vdiagram, t)emperature, e)nergy, q)uit: ','s');
if (strncmp(choice,'p',1))
        figure
        vol = (var(VC,:) + vk + vr + vh + var(VE,:))*1e6; % cubic centimeters
        pres = (var(P,:))*1e-5; % bar
        plot(vol,pres, 'k')
        grid on
        xlabel('Volume (cc)')
        ylabel('Pressure (bar [1bar = 100kPa])')
        title('P-v diagram')
elseif (strncmp(choice,'t',1))
        figure
        x = 0:10:360;
        Tcomp = var(TC,:);
        Texp = var(TE,:);
        plot(x,Tcomp,'b-',x,Texp,'r-');
        hold on
        x = [0,360];
        y = [tk,tk];
        plot(x,y,'b-')
        y = [tr,tr];
        plot(x,y,'g-')
        y = [th,th];
        plot(x,y,'r-')
        hold off
        grid on
        xlabel('Crank angle (degrees)');
        ylabel('Temperature (K)');
        title('Temperature vs crank angle');
elseif (strncmp(choice,'e',1))
        figure
        x = 0:10:360;
        Qkol = var(QK,:); % [J]
        Qreg = var(QR,:); % [J]
        Qhot = var(QH,:); % [J]
```

```
            Work = var(W,:); % [J]
            Wcom = var(WC,:); % [J]
            Wexp = var(WE,:); % [J]
            plot(x,Qkol,'b-',x,Qreg,'g-',x,Qhot,'r-',x,Work,'k-.',x,Wcom,'b--',x,Wexp,'r--');
            grid on
            xlabel('Crank angle (degrees)');
            ylabel('Energy [Joules]');
            title('Energy vs crank angle');
        end
end
fprintf('quitting ideal adiabatic plots...\n');
```

Appendix X: The Simple Analysis Function 'simple'

The purpose of the **simple** function set is to evaluate the heat transfer and flow-friction effects of the three heat exchangers on the performance of the engine. Recall that the non-ideal heater and regenerator result in the mean effective temperature of the gas in the heater space (Th) being *lower* than that of the heater wall (Twh). Similarly the non-ideal cooler and regenerator result in the mean effective temperature of the gas in the cooler space (Tk) being *higher* than that of the cooler wall (Twk). The functions **hotsim**, **kolsim** and **regsim** (evaluating Qrloss) iteratively determines these temperature differences using the convective heat transfer equations, the values of Qh and Qk being evaluated by the Ideal Adiabatic analysis function **adiab**.



Finally function **worksim** is invoked to evaluate the reduction in power due to the flow friction pressure drop loss through the heat exchangers.

```
function  [var,dvar] = simple
% simple analysis - including heat transfer and pressure drop effects
% Israel Urieli, 7/22/2002 (modified 12/3/2003 for temp plots)
% Modified 2/7/2010 to include qrloss in hotsim & kolsim
% Modified 2/15/2010 logical reorganization
% Returned values:
```

```
%   var(22,37) array of variable values every 10 degrees (0 - 360)
%   dvar(16,37) array of derivatives every 10 degrees (0 - 360)
% Row indices of the var, dvar arrays:
 TC = 1;  % Compression space temperature (K)
 TE = 2;  % Expansion space temperature (K)
 QK = 3;  % Heat transferred to the cooler (J)
 QR = 4;  % Heat transferred to the regenerator (J)
 QH = 5;  % Heat transferred to the heater (J)
 WC = 6;  % Work done by the compression space (J)
 WE = 7;  % Work done by the expansion space (J)
 W  = 8;  % Total work done (WC + WE) (J)
 P  = 9;  % Pressure (Pa)
 VC = 10; % Compression space volume (m^3)
 VE = 11; % Expansion space volume (m^3)
 MC = 12; % Mass of gas in the compression space (kg)
 MK = 13; % Mass of gas in the cooler (kg)
 MR = 14; % Mass of gas in the regenerator (kg)
 MH = 15; % Mass of gas in the heater (kg)
 ME = 16; % Mass of gas in the expansion space (kg)
 TCK = 17; % Conditional temperature compression space / cooler (K)
 THE = 18; % Conditional temeprature heater / expansion space (K)
 GACK = 19; % Conditional mass flow compression space / cooler (kg/rad)
 GAKR = 20; % Conditional mass flow cooler / regenerator (kg/rad)
 GARH = 21; % Conditional mass flow regenerator / heater (kg/rad)
 GAHE = 22; % Conditional mass flow heater / expansion space (kg/rad)
% Size of var(ROWV,COL), dvar(ROWD,COL)
 ROWV = 22; % number of rows in the var matrix
 ROWD = 16; % number of rows in the dvar matrix
 COL = 37; % number of columns in the matrices (every 10 degrees)
%==========================================================
global  freq % cycle frequency [herz]
global  tk tr th % cooler, regenerator, heater temperatures [K]
global  cqwr % regenerator housing thermal conductance [W/K]
twk = tk; % Cooler wall temp - equal to initial cooler gas temp
twh = th; % Heater wall temp - equal to initial heater gas temp
epsilon = 1; % allowable temperature error bound for cyclic convergence
terror = 10*epsilon; % Initial temperature error (to enter loop)
while  (terror>epsilon)
        [var,dvar] = adiab;
        qrloss = regsim(var); % included 2/7/2010
```

```
        tgh = hotsim(var,twh,qrloss); % new heater gas temperature
        tgk = kolsim(var,twk,qrloss); % new cooler gas temperature
        terror = abs(th - tgh) + abs(tk - tgk);
        th = tgh;
        tk = tgk;
        tr = (th-tk)/log(th/tk);
end
fprintf('===== converged heater and cooler mean temperatures =====\n');
fprintf('heater wall/gas temperatures: Twh = %.1f[K], Th = %.1f[K]\n',twh,th);
fprintf('cooler wall/gas temperatures: Twk = %.1f[K], Tk = %.1f[K]\n',twk,tk);
% Print out ideal adiabatic analysis results
eff = var(W,COL)/var(QH,COL); % engine thermal efficency
Qkpower = var(QK,COL)*freq;   % Heat transferred to the cooler (W)
Qrpower = var(QR,COL)*freq;   % Heat transferred to the regenerator (W)
Qhpower = var(QH,COL)*freq;   % Heat transferred to the heater (W)
Wpower = var(W,COL)*freq;     % Total power output (W)
fprintf('========== ideal adiabatic analysis results ==========\n');
fprintf(' Heat transferred to the cooler: %.2f[W]\n', Qkpower);
fprintf(' Net heat transferred to the regenerator: %.2f[W]\n', Qrpower);
fprintf(' Heat transferred to the heater: %.2f[W]\n', Qhpower);
fprintf(' Total power output: %.2f[W]\n',  Wpower);
fprintf(' Ideal Adiabatic Thermal efficiency: %.1f[%%]\n', eff*100);
fprintf('============ Regenerator analysis results============\n');
fprintf(' Regenerator net enthalpy loss: %.1f[W]\n', qrloss*freq);
qwrl = cqwr*(twh - twk)/freq;
fprintf(' Regenerator wall heat leakage: %.1f[W]\n', qwrl*freq);
% Temperature plot of the simple simulation results
                    figure
                    x = 0:10:360;
                    Tcomp = var(TC,:);
                    Texp = var(TE,:);
                    plot(x,Tcomp,'b-',x,Texp,'r-');
                    hold on
                    x = [0,360];
                    y = [twk,twk];
                    plot(x,y,'b--')
                    y = [tk,tk];
                    plot(x,y,'b-')
                    y = [tr,tr];
                    plot(x,y,'g-')
```

```
                    y = [th,th];
                    plot(x,y,'r-')
                    y = [twh,twh];
                    plot(x,y,'r--')
                    hold off
                    grid on
                    xlabel('Crank angle (degrees)');
                    ylabel('Temperature (K)');
                    title('Simple Simulation - Wall and Gas Temps vs crank angle');
% Various plots of the ideal adiabatic simulation results
plotadiab(var,dvar);
fprintf('========= pressure drop simple analysis ============\n');
dwork = worksim(var,dvar);
fprintf(' Pressure drop available work loss: %.1f[W]\n', dwork*freq)
actWpower = Wpower - dwork*freq;
actQhpower = Qhpower + qrloss*freq + qwrl*freq;
acteff = actWpower/actQhpower;
fprintf(' Actual power from simple analysis: %.1f[W]\n', actWpower);
fprintf(' Actual heat power in from simple analysis: %.1f[W]\n', actQhpower);
fprintf(' Actual efficiency from simple analysis: %.1f[%%]\n', acteff*100);
```

Appendix Y: Reynolds Number, Flow Friction & Heat Transfer Coefficients

The following four functions **reynum** (Reynolds Number), **pipefr** (tubular), **foilfr** (wrapped foil) and **matrixfr** (wire mesh) are used in the **simple** function set for heat transfer and flow friction evaluation.

```
function  [mu,kgas,re] = reynum(t,g,d)
% evaluate dynamic viscosity, thermal conductivity, Reynolds number
% Israel Urieli, 7/22/2002 (mu units correction 2/13/2011)
% Arguments:
%   t - gas temperature [K]
%   g - mass flux [kg/m^2.s]
%   d - hydraulic diameter [m]
% Returned values:
%   mu - gas dynamic viscosity [kg/m.s]
%   kgas - gas thermal conductivity [W/m.K]
%   re - Reynolds number

global  cp % specific heat capacity at constant pressure [J/kg.K]
global  mu0 % dynamic viscosity at reference temp t0 [kg.m/s]
global  t0 t_suth % reference temperature [K], Sutherland constant [K]
global  prandtl % Prandtl number

mu = mu0*(t0 + t_suth)/(t + t_suth)*(t/t0)^1.5;
kgas = cp*mu/prandtl;
re = abs(g)*d/mu;
if (re < 1)
        re = 1;
end
```

```
function  [ht,fr]=pipefr(d,mu,re);
% evaluate heat transfer coefficient, Reynolds friction factor
% Israel Urieli, 7/22/2002 (corrected header 2/20/2011)
% Arguments:
%   d - hydraulic diameter [m]
%   mu - gas dynamic viscosity [kg.m/s]
%   re - Reynolds number
% Returned values:
%   ht - heat transfer coefficient [W/m^2.K]
%   fr - Reynolds friction factor ( = re*fanning friction factor)

global  cp % specific heat capacity at constant pressure [J/kg.K]
global  prandtl % Prandtl number

% Personal communication with Allan Organ, because of oscillating
% flow, we assume that flow is always turbulent. Use the Blasius
% relation for all Reynolds numbers:
fr=0.0791*re^0.75;
% From Reynolds simple analogy:
ht=fr*mu*cp/(2*d*prandtl);
```

```
function  [st,ht,fr] = foilfr(d,mu,re)
% evaluate regenerator wrapped foil stanton number, friction factor
% Israel Urieli, 7/22/2002
% Arguments:
%   d - hydraulic diameter [m]
%   mu - gas dynamic viscosity [kg.m/s]
%   re - Reynolds number
% Returned values:
%   st - Stanton number
%   ht - heat transfer coefficient [W/m^2.K]
%   fr - Reynolds friction factor ( = re*fanning friction factor)

global  cp % specific heat capacity at constant pressure [J/kg.K]
global  prandtl % Prandtl number

if (re < 2000) % normally laminar flow
        fr = 24;
else
        fr = 0.0791*re^0.75;
end
% From Reynolds simple analogy:
st=fr/(2*re*prandtl);
ht=st*re*cp*mu/d;
```

```
function  [st,fr] = matrixfr(re)
% evaluate regenerator mesh matrix stanton number, friction factor
% Israel Urieli, 7/22/2002
% Arguments:
%   re - Reynolds number
% Returned values:
%   st - Stanton number
%   fr - Reynolds friction factor ( = re*fanning friction factor)

global  prandtl % Prandtl number
% equations taken from Kays & London (1955 edition)
st = 0.46*re^(-0.4)/prandtl;
fr = 54 + 1.43*re^0.78;
```