$$E = |g^3| A v^2 K/2 \tag{22}$$

where K is the expansion or contraction loss coefficient as defined by Kays and London [10]. This result is valid for both expansions and contractions.

It is important to note that the derivation of (22) assumes that for an instantaneous Reynolds number and geometry, the rate of dissipation for unsteady conditions is approximately the same as that for steady conditions.

Bird, et al. [6] show that for steady, incompressible isothermal flow there is a relationship for E in terms of the steady-state friction factor Ff:

$$E = [2(gv)^2 \Delta x Ff/d] gA \quad \begin{matrix}\text{(parallel flow — shear}\\ \text{stress terms only)}\end{matrix} \tag{23}$$

If (23) can be considered to replace only those terms due to shear work (in accordance with the previous assumptions on the friction factor), then (23) may be modified to include the effects of the normal stresses which are again easily evaluated from the differential energy equation (again making the relevant Newtonian assumptions). The final result for parallel flow dissipation is then [5]:

$$E = V\mu \left\{ 2(gv)^2 Fr/d^2 + \frac{4}{3}\left[\frac{\partial}{\partial x}(gv)\right]^2 \right\} \qquad \text{(parallel flow)} \tag{24}$$

Note that (24) is always positive as would be expected.

A heat transfer coefficient is defined to evaluate the $\dot{Q}$. Again the quasi-steady assumption is used by assuming that the local heat transfer coefficient is solely dependent on the flow geometry and the local Reynolds number. In this work the Reynolds simple analogy [11] has been found to be satisfactory since Prandtl numbers are in the region of unity. The local heat transfer coefficient is given as follows:

$$h = kFr/(2d) \tag{25}$$

Transport properties are either evaluated from ideal gas results (specific heats at constant pressure or volume and gas conductivity) or empirical formulas (viscosity using the Sutherland formula). This approach has been found to work well, but it is quite possible to use empirical formulas for all the transport properties if additional accuracy is required.

Method of Solution

To complete the equation set, an equation of state must be specified. For this work the ideal gas equation was found to be adequate:

$$pV = mRT \tag{26}$$

Conduction in the machine's solid parts is included by writing the energy equation for each solid section within each cell [2, 8]. This procedure is standard and is not repeated here.

The complete set of equations for the working gas and the solid portions of the machine are applied simultaneously to each elemental cell as indicated in Fig. 5. The energy and continuity equations are applied to each consecutive cell whilst the momentum fluxes are not defined within the cells but only at the nodes. The momentum equation is however stepped along the machine by one cell length at a time. This arrangement staggers the application of the momentum equation so that the momentum fluxes can be determined at each node.

The resulting set of first order differential equations (in time) is solved simultaneously by using explicit techniques such as Runge Kutta [2, 5, 8]. This process has a warm up period which requires that the equations are cycled through a number of times until cyclic steady state has been reached. This would be equivalent to the thermal equilibrium state of a real machine. A technique has been devised to force the system to cyclic steady state at a much faster rate. This is done by altering the regenerator matrix temperatures in a weighted fashion until no net heat is transferred to the regenerator over a cycle [2, 8].

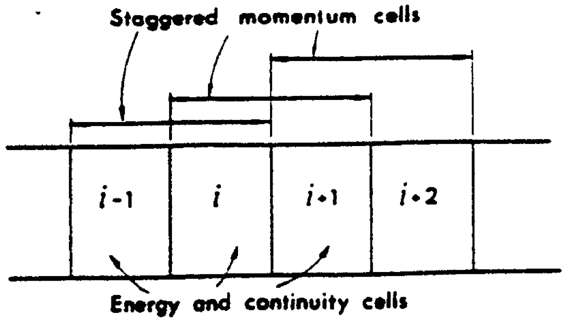

Fig. 5 Application of the continuity, energy and momentum equations

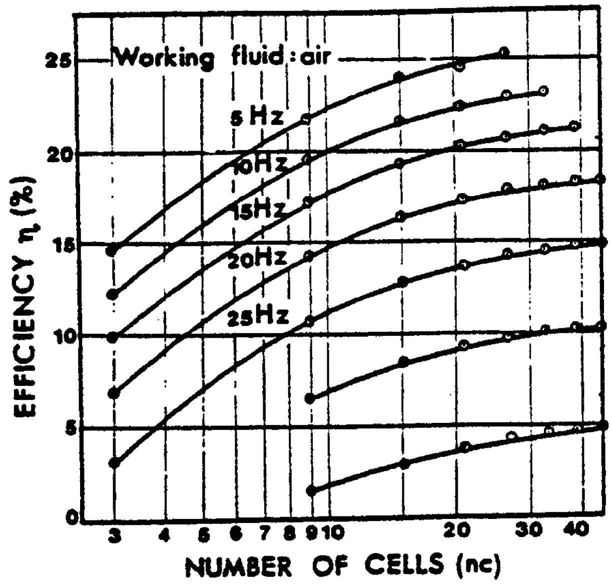

Fig. 6 Efficiency versus number of cells [2]

Results

1 Numerical Behavior. Fig. 6 shows thermal efficiency plotted against the number of cells, using air as a working fluid and at different operating frequencies. These curves were derived for a hypothetical engine previously described by Urieli, et al. [2].

It can be seen that the model is extremely well behaved. The difference in thermal efficiency for 39 cells and 45 cells is only 0.1.

2 Experimental Verification. A preliminary program to experimentally verify this approach has just been completed at the University of the Witwatersrand and has been presented in the literature [3]. Figs. 7 and 8 show an example comparison between experimental and simulation pressure-theta diagrams. Fig. 9 shows a similar comparison for the temperature distribution along the machine.

The discrepancy between the experimental and simulation expansion space pressure profiles is not fully understood. It is believed that this is probably due to not taking sufficient account of gas momentum in the working spaces [3, 5].

Table 1 gives a comparison between experimental and simulation values for a heat balance over the entire machine. There is an experimental error of approximately ± 7 percent.

A comparison with ideal analyses was also investigated which showed no inconsistancies in the simulation results [3, 5].

3 Performance of a Hypothetical Engine. Urieli [2, 8] has shown that it is possible to use this approach to generate performance curves for any hypothetical Stirling engine. A set of such curves is given in Fig. 10. These curves have been shown to be in agreement with the general trend of performance curves of real Stirling machines [2].

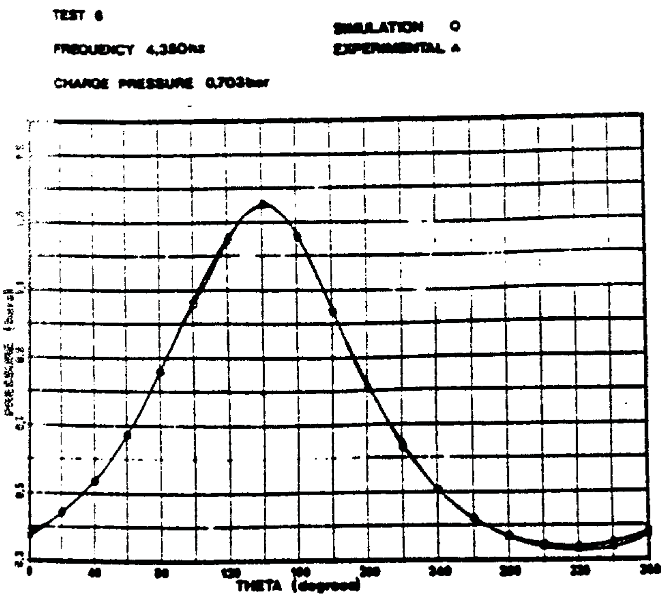

Fig. 7 Compression space pressure profiles [3]

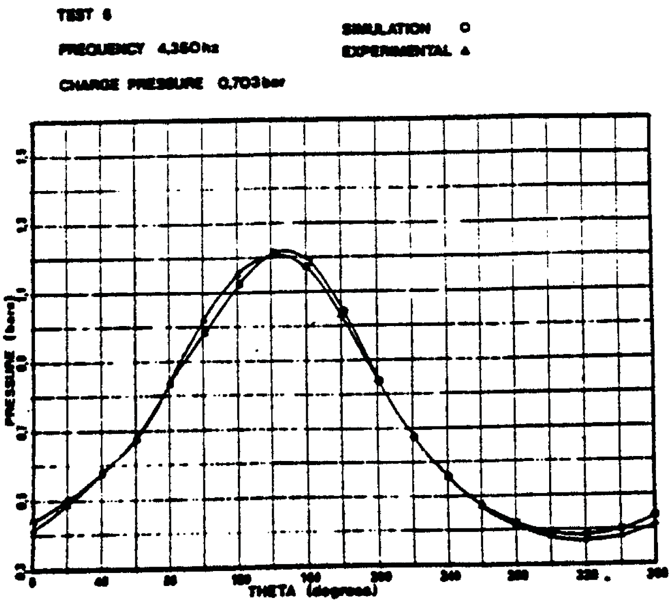

Fig. 8 Expansion space pressure profiles [3]

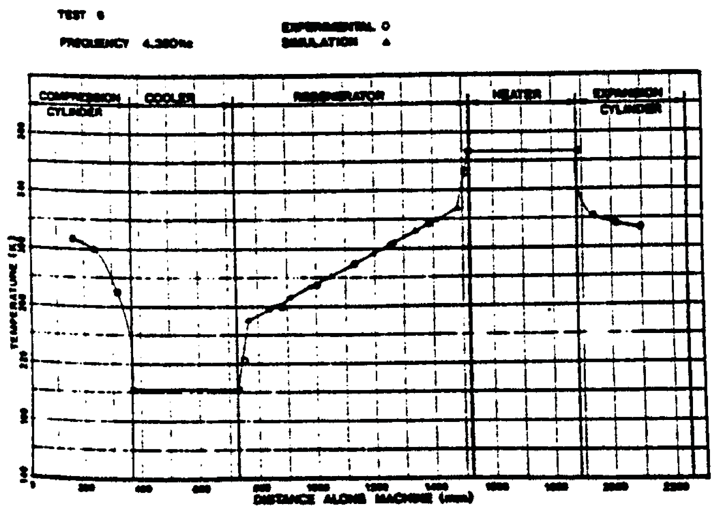

Fig. 9 Temperature distribution along machine [3]

Table 1 Overall energy balances

	Speed [Hz]	Q_k [J] Cooler	Q_h [J] Heater	W [J] Work	Energy Balance Error	Efficiency
Simulation	2,50	-46,628	49,047	2,886	-0,95%	5,88%
	3,35	-46,384	46,219	1,261	-3,09%	2,73%
	4,35	-44,690	42,288	-1,172	-2,91%	-2,77%
Exprmt.	2,50	-53,655	56,516	4,127	-2,24%	7,30%
	3,35	-53,895	51,387	2,592	-9,92%	5,04%
	4,35	-51,227	41,156	-0,860	-22,38%[1]	-2,09%

[1]This discrepancy has since been established as being due to a seal failure.

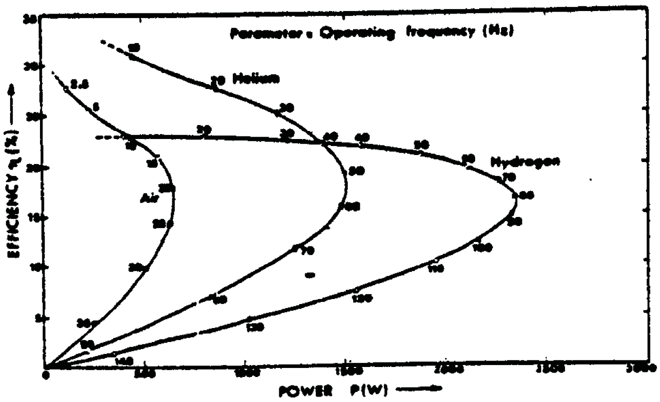

Fig. 10 Thermal efficiency versus power output

Conclusions

A more rigorous analysis has been presented to rationalise the approach first presented by Urieli, et al. [2]. The modelling of viscous dissipation and the inclusion of the normal stress terms have further improved the accuracy of the model.

Preliminary experimental comparisons are favourable and it would appear that the model can predict the performance of a real Stirling machine with an acceptable degree of accuracy.

References

1 Urieli, I., Rallis, C. J., "A New Regenerator Model for Stirling Cycle Machines," Report 67, School of Mechanical Engineering, University of the Witwatersrand, May 1976.

2 Urieli, I., Rallis, C. J., Berchowitz, D. M., "Computer Simulation of Stirling Cycle Machines," *Proceedings of the 12th IECEC*, Paper 779252, pp 1512–1521, 1977.

3 Berchowitz, D. M., Rallis, C. J., "A Computer and Experimental Simulation of Stirling Cycle Machines," *Proceedings of the 13th IECEC*, Paper 789111, 1978.

4 Finkelstein, T., "Generalized Thermodynamic Analysis of Stirling Engines," SAE paper 118B, Jan. 1960.

5 Berchowitz, D. M., "*A Computer and Experimental Simulation of Stirling Cycle Machines*," Dissertation MSc(Eng), University of the Witwatersrand, 1978.

6 Bird, R. B., Stewart, W. E., Lightfoot, E. N., *Transport Phenomena*, New York, John Wiley and Sons, 1960.

7 Kovtum, I. M., Nanmov, A. N., Nesterenko, V. B., "The Stirling Cycle in a Dissociating Gas," *International Chemical Engineering*, Vol. 7, No. 4, 1967.

8 Urieli, I., "*A Computer Simulation of Stirling Cycle Machines*," Thesis PhD, University of the Witwatersrand, 1977.

9 Schock, A., Private communication with C. J. Rallis concerning equations used for Stirling machine simulation, 1977.

10 Kays, W. M., London, A. L., *Compact Heat Exchangers* 2nd Ed. McGraw-Hill, 1964.

11 Knudsen, J. G., Katz, D. L., *Fluid Dynamics and Heat Transfer*, Tokyo, McGraw-Hill, 1958.

APPENDIX E

COMPUTER PROGRAMME

E.1 Introduction

One of the aims of this work was to write a rapidly executing computer programme that would calculate the performance of either fixed or free dynamics Stirling engines with acceptable accuracy. The programme presented here automates the calculations of the linear analyses and operates fast enough to be practical when run on personal computers. Calculations for a single cycle are so rapid that multi-variable parametric plots and optimisations can be completed in minutes. Similar efforts using simulation methods are extremely costly and may take weeks to complete.

E.2 Programme Description

The programme is written in Microsoft Basic as implemented on the Apple Macintosh computer. This is one of the newer structured versions of this language which offers fairly powerful programming features not found on older versions. It is, however, an interpreted language and therefore does not have the speed of a compiled language. The translation of this code to a compileable language would greatly improve execution time.

The programme is divided into three major sections, namely:

1) The input section responsible for interfacing with the user, loading the data file and selecting the presentation of the calculations.

2) A set of subprogrammes that excercise the linear analysis in the various ways specified in the input section. For example, sensitivity plots, optimisation or just a single point solution. These programmes also handle the output.

3) The linear analysis.

The input section will not be listed here since it is long, machine specific and not particularly relevant. A table showing the location of the engine variables is given in Table E.1. This is the table constructed by the input section and loaded into the matrix DAT.

The subprogrammes that manipulate the linear analysis are:

a) GRAPH which simply presents the output in a phase plot of displacer and piston motions and a pV plot. This is very useful for laboratory work since the experimental data are generally taken in similar form.

b) PERF lists a performance summary of the nominal calculation point.

c) L.INV lists a loss inventory.

d) SPLOT produces sensitivity plots of sets of dependent variables against a selectable independent variable.

e) OPTIMIZE will optimise efficiency using up to four engine parameters selectable by the user. A Hooke and Jeeves algorithm is used, described fully in [KM73].

The linear analysis section includes subroutines that calculate one of two basic ideal cycles, namely, the isothermal (or Schmidt cycle) and the ideal adiabatic cycle. Either basic cycle may be selected by the user. Also included is a full linear dynamics analysis or a simple fixed sinusoidal dynamics model, again selectable by the user. Parasitic losses are calculated within the body of the linear analysis section.

Figure E.1 shows the output of GRAPH, PERF and L.INV. The pull-down menu system indicates that the adiabatic basic cycle has been selected. The interface and output of SPLOT is shown in Figures E.2 and E.3 and similarly for OPTIMIZE in Figures E.4 and E.5. Figure E.5 shows intermediate results from OPTIMIZE, which can be seen to agree with the sensitivity plot in Figure E.3. On completion, OPTIMIZE will create a new data file with the optimised parameters.

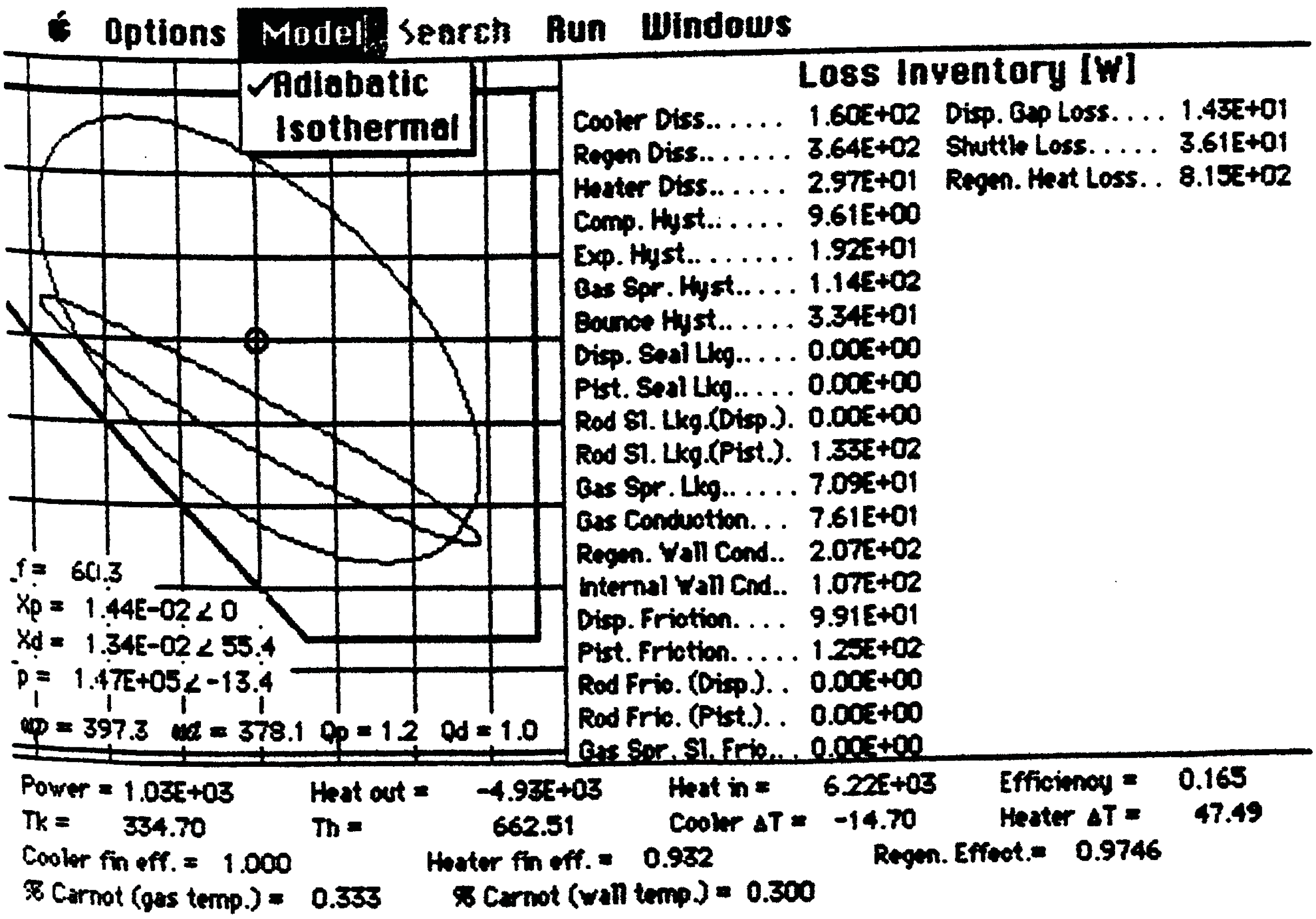

Figure E.1 Screen Output of GRAPH, PERF and L.INV

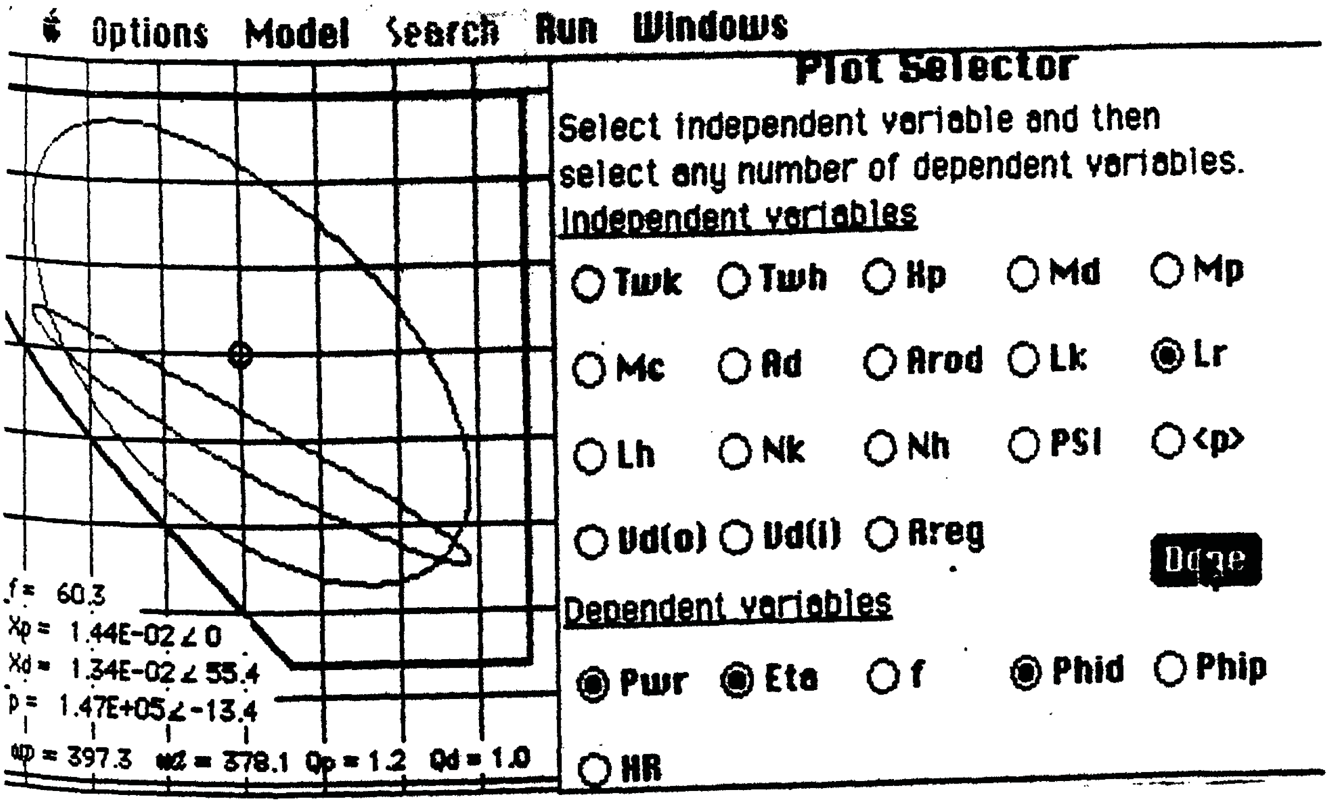

Figure E.2 Interface with SPLOT

The selection of the parameters show that power, efficiency and displacer/piston phase is to be plotted against variable L_r.

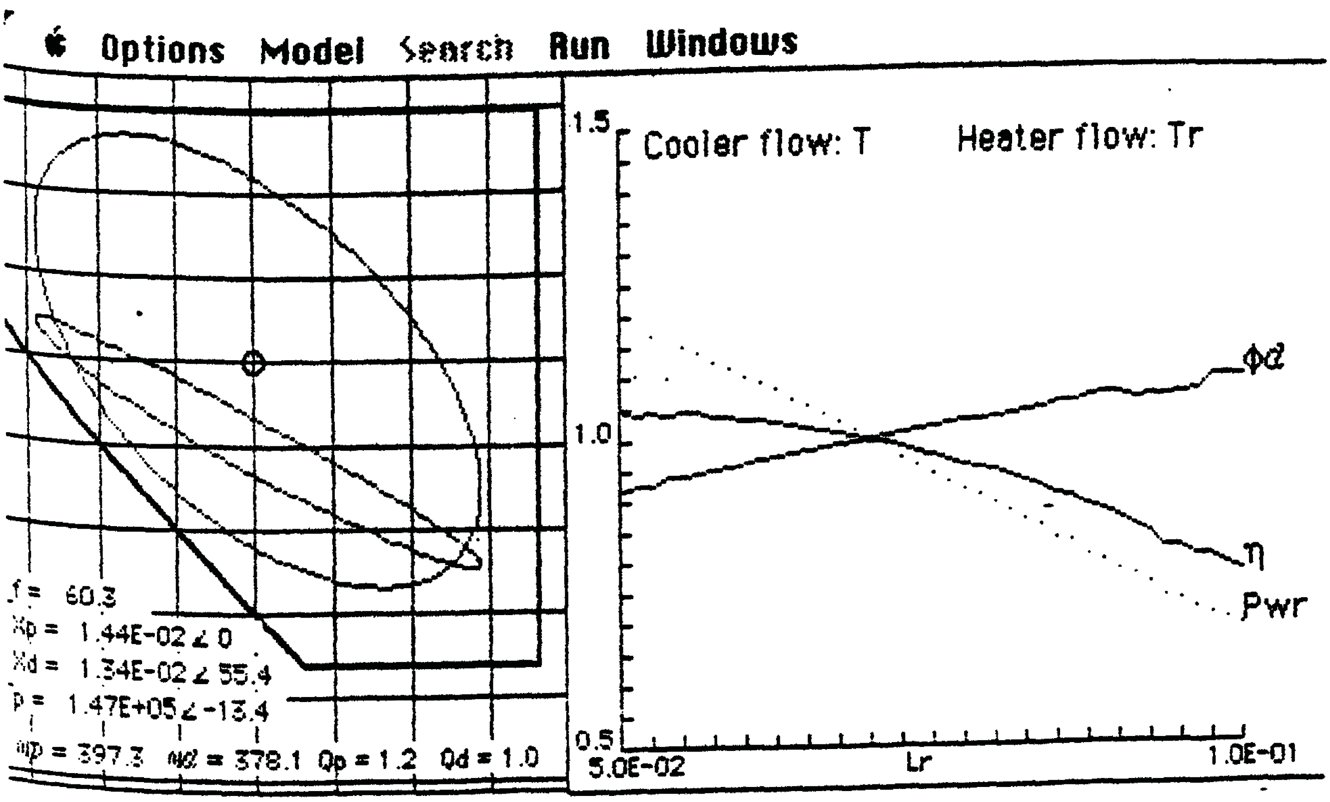

Figure E.3 Sceen Output from SPLOT

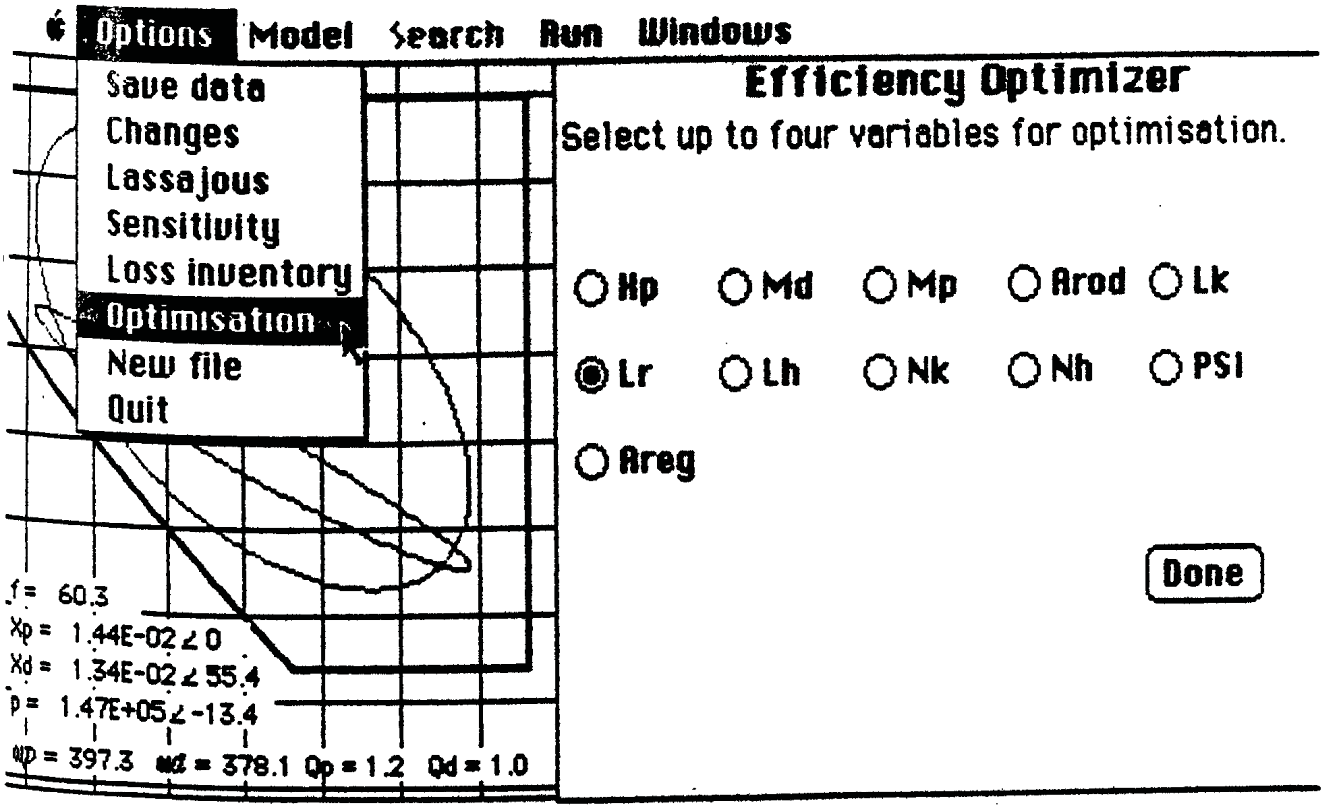

Figure E.4 Interface with OPTIMIZE

The selection shows that L_r will be optimised.

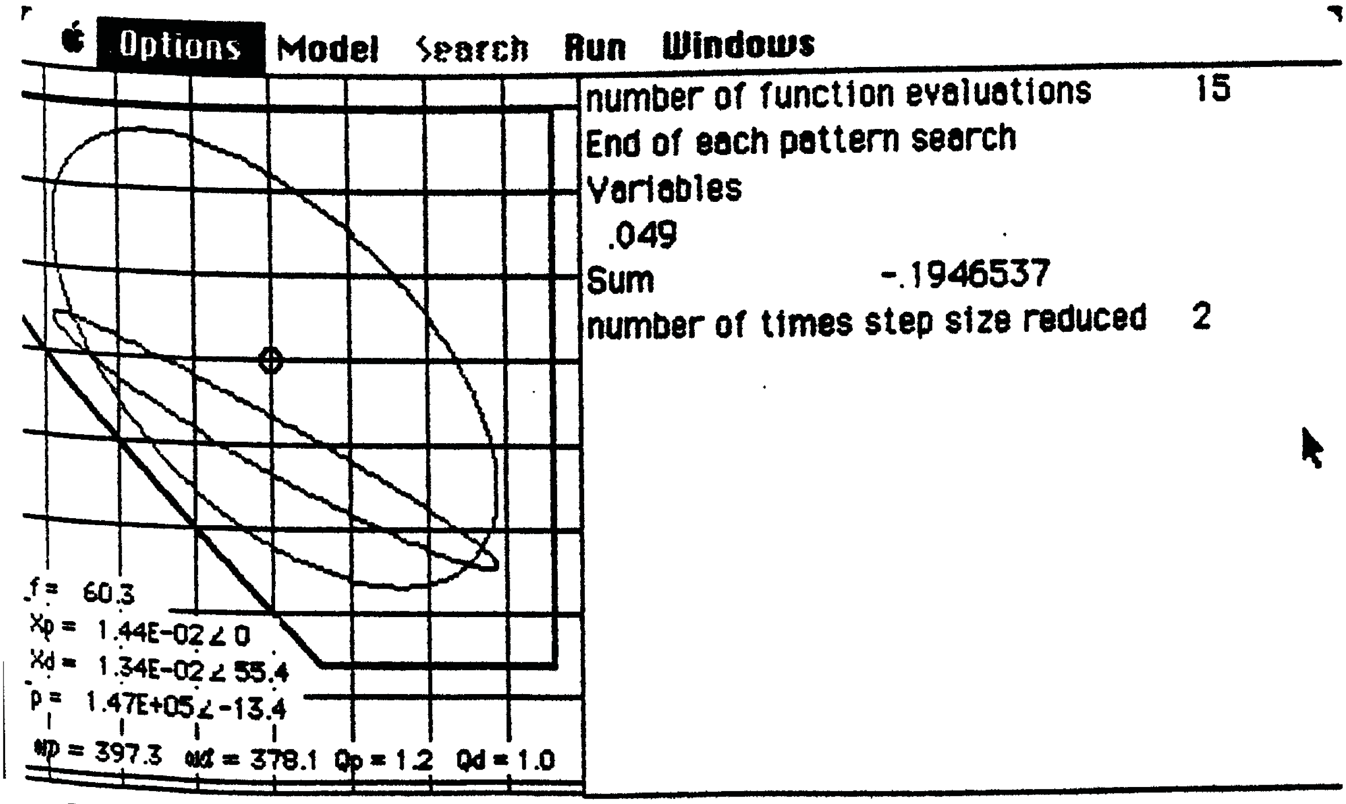

Figure E.5 Intermediate Output from OPTIMIZE

After fifteen evaluations the optimum length is near 49mm and the efficiency is about 19.5% (the negative of Sum).

Table E.1 Matrix of Input Variables for Computer Code

	1	2	3	4	5
1	$\langle p \rangle$	T_{wk}	T_{wh}	f	X_p
2	m_d	m_p	m_c	A	A_R
3	$c\ell c$	$c\ell e$	V_b	–	–
4	gas type	cooler type tubes = 0, fins = 1	heater type tubes = 0, fins = 1	X_d for fixed dynamics	Φ_d for fixed dynamics
5	d_k	L_k	n_k (tubes)	V_{clc}	$K_{d_{ext}}$ (external mech spr)
6	d_h	L_h	n_h (tubes)	V_{cle}	–
7	b_k	a_k	n_k (fins)	$\mathcal{H}_k$	t_k
8	b_h	a_h	n_h (fins)	$\mathcal{H}_h$	t_h
9	regenerator type	V_{cr}	L_r	ψ	d_m
10	displ. spr: sing=0, doub=1, mech=2	V_{gsi}	V_{gso} or V_{gs}	d_{pk}	–
11	–	–	–	–	–
12	μ_d	μ_p	μ_R	displacer seal type clear = 1, ring = 2	piston seal type clear = 1, ring = 2
13	rod seal type clear = 1, ring = 2	displacer clearance gap or ring width	piston clearance gap or ring width	rod clearance gap or ring width	displacer seal length or material
14	piston seal length or material	rod seal length or material	displ no of seals or eccentricity	pist no of seals or eccentricity	rod no of seals or eccentricity
15	displacer seal static force	piston seal static force	rod seal static force	displ leakage path comp/exp = 2	pist leakage path comp/bounce = 1
16	rod leakage path	gas spring seal type: clr=1, rng=2	g s seal clearance gap or ring width	gas spring seal length or material	gas spr no of seals or eccentricity
17	gas spring seal static force	–	μ_{gs}	t_r	regenerator I.D.
18	number of regenerators	regenerator wall material	k_{wr}	t_d	displacer dome material
19	k_{wd}	displacer type	h	stuffer O.D. thickness	stuffer O.D.
20	stuffer I.D. thickness	stuffer I.D.	stuffer insulation material	stuffer can material	displacer out stop
21	piston out stop	piston in stop	stuffer insulation conductivity	stuffer can conductivity	cooler fin thickness
22	cooler fin material	heater fin thickness	heater fin material	cooler fin conductivity	heater fin conductivity
23	cooler outer wall conductivity	heater outer wall conductivity	–	–	–

E.3 Programme Listing

```
SUB     LINDYN(DAT(2),PPV,ETA,F,PHI*,BETSD,XR,KT$,HT$,NTST%,MTST%,ETYPE%)STATIC
SHARED    P1,P2,XP,XD,O,OP,OD,QP,QD,PI,TH,TK,QDK,QDH
SHARED   PMPK,PMPR,PMPH,HYSK,HYSH,GSH,E
SHARED    QDR,LD,LP,LRD,LRP,LG,QDC,DSP,PSP,GSP,RSP
SHARED   QCR,DTH,DTK,SHL,DGL,BH,QCIW,FINFK,FINFH
NMAX%=20: DTOL=.0005: TTOL=.001
PI=3.14159
P2=0: DTK1=0
TWK=DAT(1,2)
TWH=DAT(1,3)
IF NTST%=1 THEN GOTO STEST
TK=TWK
DTK=0
TH=TWH
DTH=0
TAU=TH/TK
TCM=TK:TEM=TH
PHI*=.8*PI/2!: E=1: PMPK=0: PMPR=0: PMPH=0: STFK=0: STFH=0: SNUK=0: SNUH=0
DG=0: LRP=0: LP=0: LD=0: GSP=0: RSP=0: PSP=0: DSP=0:LG=0: DS=0: PPV1=0
F=DAT(1,4):  O=F*2!*PI
STEST:
IF ETYPE%=0 THEN F=DAT(1,4): O=F*2!*PI
IF  DAT(4,2)=1  THEN  POA=2*(DAT(5,2)+DAT(21,5))/(DAT(5,2)*DAT(21,5))
IF  DAT(4,2)=1  THEN  FINK=SQR(POA/DAT(22,4))*DAT(7,1)
IF  DAT(4,3)=1  THEN  POA=2*(DAT(6,2)+DAT(22,2))/(DAT(6,2)*DAT(22,2))
IF  DAT(4,3)=1  THEN  FINH=SQR(POA/DAT(22,5))*DAT(8,1)
HLK=DAT(7,4):  HLH=DAT(8,4)
XP=DAT(1,5)
GOSUB GAS
   Z=GAM/(GAM-1)
   IF DAT(19,2)=1 THEN AP=DAT(2,4) ELSE AP=(DAT(2,4)-DAT(2,5))
  IF DAT(19,2)=1 THEN XRVF=(DAT(2,4)-DAT(2,5))/AP ELSE XRVF=1
  VSP=2!*DAT(1,5)*AP
  VMC=DAT(3,1)*AP
   IF DAT(4,2)=0 THEN AK=DAT(5,3)*PI*DAT(5,1)^2/4! ELSE AK=DAT(7,3)*DAT(7,1)*DAT(7,2)
  IF DAT(4,2)=0 THEN AWK=DAT(5,3)*PI*DAT(5,1)*DAT(5,2) ELSE
                    AWK=DAT(7,3)*2!*(DAT(7,1)+DAT(7,2))*DAT(5,2)
  VK=AK*DAT(5,2)+DAT(5,4)
   IF DAT(4,3)=0 THEN AH=DAT(6,3)*PI*DAT(6,1)^2/4! ELSE AH=DAT(8,3)*DAT(8,1)*DAT(8,2)
  IF DAT(4,3)=0 THEN AWH=DAT(6,3)*PI*DAT(6,1)*DAT(6,2) ELSE
                    AWH=DAT(8,3)*2!*(DAT(8,1)+DAT(8,2))*DAT(6,2)
  VH=AH*DAT(6,2)+DAT(6,4)
  DHK=4!*AK/AWK*DAT(5,2)
  DHH=4!*AH/AWH*DAT(6,2)
   DHR=DAT(9,5)*DAT(9,4)/(1!-DAT(9,4))
  ARR=DAT(9,2)*DAT(9,4)
  VR=ARR*DAT(9,3)
  AWR=4*(1-DAT(9,4))*VR/DAT(9,5)
  DD=SQR(4!/PI*DAT(2,4))
  DR=SQR(4/PI*DAT(2,5))
```

```
AHWK=2!*AP+PI*DD*DAT(3,1)
AHWH=2!*DAT(2,4)+PI*DD*DAT(3,2)
IF DAT(4,2)=1 THEN ARK=DAT(7,1)/DAT(7,2)
IF DAT(4,2)=1 THEN AGK=DAT(7,3)*(DAT(21,5)+DAT(7,2))*DAT(5,2)
IF DAT(4,3)=1 THEN ARH=DAT(8,1)/DAT(8,2)
IF DAT(4,3)=1 THEN AGH=DAT(8,3)*(DAT(22,2)+DAT(8,2))*DAT(6,2)
IF DAT(12,4)=1 THEN CLD=(1+DAT(14,3)^2)^.333*DAT(13,2)
IF DAT(12,5)=1 THEN CLP=(1+DAT(14,4)^2)^.333*DAT(13,3)
IF DAT(13,1)=1 THEN CLR=(1+DAT(14,5)^2)^.333*DAT(13,4)
IF DAT(16,2)=1 THEN CLGS=(1+DAT(16,5)^2)^.333*DAT(16,3)
IF DAT(4,3)=0 THEN LC=DAT(9,3)+DAT(5,2) ELSE LC=DAT(9,3)+DAT(6,2)
CN1=ARR*KR/DAT(9,3)+DAT(2,4)*10*KR/LC
CN2=PI*DD*DAT(18,4)*DAT(19,1)/LC
CN2=CN2+DAT(21,4)*PI*(DAT(19,5)*DAT(19,4)/DAT(9,3)+DAT(20,2)*DAT(20,1)/LC)
CN3=PI/4*(DAT(19,5)^2-DAT(20,2)^2)*DAT(21,3)
CN4=CN2+CN3
CN5=PI*DAT(17,5)*DAT(17,4)*DAT(18,1)*DAT(18,3)/DAT(9,3)
IF DAT(19,5)=0 OR DAT(20,2)=0 THEN SL=DAT(9,3)
IF DAT(19,5)>0 AND DAT(20,2)>0 THEN SL=1.15*DAT(9,3)
IF DAT(19,3)<=0 THEN G=.01*DD ELSE G=DAT(19,3)
GLO=2*SQR(KR*(TH+TK)*(GAM-1)/GAM/DAT(1,1)/O)
IF G>GLO THEN G=GLO
CN6=PI/2*DD*KR/(SL*G)
APV=PI*DD*DAT(19,3)*LC
DGLC=.5*CP/R*DAT(1,1)*APV/SL
APK=PI*DAT(10,4)*DAT(10,4)/4!
AW1=PI*DAT(10,4)*DAT(10,3)/APK+2*APK
IF DAT(10,1)=1 THEN AW2=PI*DAT(10,4)*DAT(10,2)/APK+2*APK
IF NTST%=0 THEN XR=.5
XR1=0: NC%=0
IF ETYPE%=0 THEN PHI#=DAT(4,5)*PI/180: XD=DAT(4,4): XR=XD/XP
WHILE(ABS(XR-XR1)>DTOL*XR)
  NC%=NC%+1
  IF ETYPE%=1 THEN XD=XR*XP
  TAU=TH/TK
  VSE=2!*XD*DAT(2,4)
  XRV=XR*XRVF
  VSC=VSP*SQR(1!-2!*XRV*COS(PHI#)+XRV*XRV)
  VCLC=VMC-.5*VSC
  VCLE=DAT(3,2)*DAT(2,4)-.5*VSE
  IF VCLC<0 THEN VCLC=0 'Piston displacer collision
  IF VCLE<0 THEN VCLE=0 'Collision on dome
  VCLE=VCLE+APV
  ANM=-XRV*SIN(PHI#)
  ADM=-1!+XRV*COS(PHI#)
  UCA=ATN(ANM/ADM)
  IF ANM<0 AND ADM>0 THEN UCA=UCA+2!*PI
  IF ADM<0 THEN UCA=UCA+PI
  alpha=UCA+PHI#-PI
  TR=(TH-TK)/LOG(TAU)
  IF MTST%=1 THEN GOSUB Adiabatic ELSE GOSUB Isothermal
    RHOK=DAT(1,1)/(R*TK): RHOR=DAT(1,1)/(R*TR): RHOH=DAT(1,1)/(R*TH)
  RP=P1/DAT(1,1)
```

```
        hf=2!*SQR(0/32!*(GAM-1)/(GAM*DAT(1,1)))*P1^2/(GAM*DAT(1,1))
      HYSK=hf*SQR(KK*TK)*AHWK
      HYSH=hf*SQR(KH*TH)*AHWH
      CLOSS=(CN1+CN4+CN6*XD*XD)*(TH-TK)
      FIG=BETAD#-PHI#: TAUA=TH/(TK+100)
       RO=RP*COS(FIG)+XD*(.5*LOG(TAUA)/SL-1/LC)
      RA=RP*SIN(FIG)
      CALL VECTOR(RO,RA,FIM,RR)
       DGL=DGLC*O*LOG(TAUA)*RR*XD*(1/PI-COS(FIM)/4)
      DGL=DGL+.5*PI*DD*P1*DAT(19,3)*XD*O*SIN(FIG)
      GSH=.5*DGS*(O*XD)^2
      QDK=QDK-HYSK-PMPK-.5*PMPR-GSH-QDR*(1-E)-LP-LRP-LG-CLOSS-DSP-PSP-GSP-DGL
      QDH=QDH-HYSH-PMPH-.5*PMPR+QDR*(1-E)+CLOSS
     GOSUB HEAT
       XREL=SQR((XD*SIN(PHI#))^2+(XD*COS(PHI#)-XP)^2)
      PW=PWI+PI*DDP*O*XREL*XREL-(HYSK+LP+LRP)/F
      SBD#=PW/PWI*SIN(BETAD#)
      IF ABS(SBD#)>1 THEN BETAD#=-PI/2! ELSE BETAD#=ATN(SBD#/SQR(-SBD#*SBD#+1!))
      BETSD=BETAD#: Phase angle modified for viscous losses
     PPV=F*PW
       DIF=KK*(TK+50!)/(Z*DAT(1,1))
       IF DAT(10,1)<>2 THEN FDG=2!*.5*KK*SQR(.5*O/DIF)*GAM*(GAM-1!)*(TK+50!)
       IF DAT(10,3)<>0 AND DAT(10,1)<>2 THEN DGS=FDG*AW1*(APK/(O*DAT(10,3)))^2
       IF DAT(10,1)=1 THEN DGS=DGS+FDG*AW2*(APK/(O*DAT(10,2)))^2
      IF DAT(10,1)=2 THEN DGS=0
      IF ETYPE%=0 THEN GOSUB Q
      IF ETYPE%=1 THEN GOSUB FREE
     IF ABS(PPV-PPV1)<.01*PPV THEN XR1=XR
     PPV1=PPV
       IF NC%>NMAX% THEN XR1=XR: NTST%=0:BEEP
  WEND
   IF DAT(3,4)=0 THEN AWP=2*AP+(DAT(3,3)/AP)*DD*PI ELSE AWP=DAT(3,4)
   BH=.5*FDG*AWP*(XP*AP/DAT(3,3))^2
  PPV=PPV-PSP-BH: 'Subtract mechanical losses
  QDC=CN1*(TH-TK)
  QCIW=CN4*(TH-TK)
  SHL=CN6*XD*XD*(TH-TK)
  QCR=CN5*(TWH-TWK)
  QDH=QDH+QCR
  QDK=QDK-QCR
  ETA=PPV/QDH
  IF ABS(ETA)>=1 THEN ETA=0
   IF NC%>=NMAX% THEN WINDOW 2,,(320,50)-(470,70),3: PRINT"NOT CONVERGING NOW";
   IF NC%>=NMAX% THEN WINDOW OUTPUT 4
   IF NC%>=NMAX% AND NTST%=0 THEN EXIT SUB
  IF NC%<NMAX% THEN WINDOW CLOSE 2
  EXIT SUB

'------------------------------Subroutines ------------------------------------------
FREE:
      P1OS=P1/SIN(PHI#)
      DPDXD=SIN(BETAD#)*P1OS/XD
      ANGP=PI-ABS(BETAD#)-ABS(PHI#)
```

```
 'For heat pump mode
 IF BETAD#>0 THEN ANGP=-ABS(BETAD#)+ABS(PHI#)
 DPDXP=SIN(ANGP)*P10S/XP
 GOSUB KDEXT
 KD=DAT(2,5)*DPDXD+KDE
 KPE=GAM*DAT(1,1)*AP*AP/DAT(3,3)
 KP=AP*ABS(DPDXP)+KPE
 ALFP=DAT(2,5)*DPDXP+DAT(2,2)/DAT(2,3)*KDE
 ALFT=AP*DPDXD
 OD=SQR(ABS(KD/DAT(2,1)))
 IF BETAD#>0 AND NC%>3 THEN GOTO MOTOR
 IF BETAD#<0 THEN  OP=SQR(KP/DAT(2,2))
   GOSUB Q 'Calc QD and required QP from (5.32) or QD (check (5.33))
 ODXQD=OD*QD: QDdOD=QD/OD: OD2=OD*OD
 IF KD<0 THEN ODXQD=-ODXQD :OD2=-OD2
 UP=ABS(ODXQD+OP*QP-2!*PI*DDP/DAT(2,1)*ALFT/DAT(2,2)*QDdOD*QP/OP)
 DN=ABS(QDdOD+QP/OP)
 O=SQR(UP/DN)
 F=O/(2*PI)
MOTOR:
 IF BETAD#>0 AND NC%>3 THEN F=DAT(1,4): O=2*PI*F: OP=0
 F1=1!-O^2/OD2
 F2=O/(2!*PI*ODXQD)
 FAC=SQR(F1^2+F2^2)
 XR1=XR
 XR=SQR(ALFP^2+(O*DDP)^2)/KD/FAC: XR=ABS(XR)
 UPP=-O*DDP/ALFP*F1+F2
 DNN=-F1-O*O*DDP/ALFP/(2!*PI*ODXQD)
 PHI#=ATN(UPP/DNN)
 IF  DNN<0 THEN PHI#=PHI#+PI
 IF UPP<0 THEN PHI#=PHI#+2*PI
RETURN
Isothermal:
 S=.5*(VSC/TK+VSE/TH)+(VK+VCLC)/TK+VR/TR+(VH+VCLE)/TH
 A=.5*SQR((VSE*COS(alpha)/TH+VSC/TK)^2+(VSE*SIN(alpha)/TH)^2)
 AOS=A/S
 beta#=ATN(VSE*SIN(alpha)/TH/(VSE*COS(alpha)/TH+VSC/TK))
 SARG=ABS(1!-AOS*AOS)
 P1=ABS(DAT(1,1)*2!/AOS*(SQR(SARG)-1!))
 P2=DAT(1,1)*SQR(SARG)*AOS*AOS*.5
   'Calculate pressure phase angle
 BETAD#=-(UCA-PI-beta#)
 PWI=-PI*AP*P1*XP*SIN(BETAD#):'Piston work ideal
 QDK=.5*O*VSC*DAT(1,1)/AOS*(SQR(SARG)-1!)*SIN(beta#)
 QDH=-TH/TK*QDK
 WTI=(QDH+QDK)/F
 FR=O*DAT(1,1)/R
 C1=(VK+VCLC+VSC)/AOS*(SQR(SARG)-1)
 OOP=C1*COS(beta#)+VSC/2
 ADJ=C1*SIN(beta#)
 CALL VECTOR(OOP,ADJ,PHWK,WK1)
 WK1=WK1*FR/TK
 C1=(VH+VCLE+VSE)/AOS*(SQR(SARG)-1)
```

```
OOP=C1*COS(beta*)+VSE/2*COS(alpha)
ADJ=C1*SIN(beta*)+VSE/2*SIN(alpha)
CALL VECTOR(-OOP,-ADJ,PHWH,WH1)
WH1=WH1*FR/TH
THK=PI-alpha+PHI*-PHWK
THH=PI-alpha+PHI*-PHWH
VRD=2!*(VK+VCLC+VR/(TAU-1!)*(TAU*LOG(TAU)/(TAU-1!)-1!))
C1=(VRD+VSC)/AOS*(SQR(SARG)-1!)
OOP=C1*COS(beta*)+VSC/2
ADJ=C1*SIN(beta*)
CALL VECTOR(OOP,ADJ,PHWR,WR)
THRR=PI-alpha+PHI*-PHWR
WR=WR*FR/TK
VF=(2*(VR/GAM+VK+VCLC+VH+VCLE)+VSC+VSE)*AOS
QR=VF^2+VSC^2+VSE^2+2*(VSC*VSE*COS(alpha)-VF*VSC*COS(beta*)-VF*VSE*COS(alpha-beta*))
QDR=F* DAT(1,1)*CP/R*SQR(ABS(QR))
RETURN
Adiabatic:
TCM1=0
VCM=VCLC+VSC/2: VEM=VCLE+VSE/2
WHILE ABS(TCM-TCM1)>TTOL*ABS(TCM)
IF TCM<=0 THEN TCM=TK
IF TEM<=0 THEN TEM=TH
DTC=TCA/TCM: DTE=TEA/TEM
TAUC=TCM/TK: TAUE=TEM/TH
AC1=VCM+DTC*VSC*COS(PHIC)/4
OOP=DTC*VCM*SIN(PHIC): ADJ=VSC/2+DTC*VCM*COS(PHIC)
CALL VECTOR(OOP,ADJ,PHICV,AC2)
AE1=VEM+DTE*VSE*COS(alpha-PHIE)/4
OOP=VSE*SIN(alpha)/2+DTE*VEM*SIN(PHIE)
ADJ=VSE*COS(alpha)/2+DTE*VEM*COS(PHIE)
CALL VECTOR(OOP,ADJ,PHIEV,AE2)
OOP=(AC2/TCM*SIN(PHICV)+AE2/TEM*SIN(PHIEV))
ADJ=AC2/TCM*COS(PHICV)+AE2/TEM*COS(PHIEV)
CALL VECTOR(OOP,ADJ,BETA1,A1)
S=AC1/TCM+VK/TK+VR/TR+VH/TH+AE1/TEM
AOS=A1/S
CON3=(1-TAUE)/2*VSE/VEM+TAUE*(1-TAUE^2)/96*(VSE/VEM)^3
CON5=(1-TAUC)/2*VSC/VCM+TAUC*(1-TAUC^2)/96*(VSC/VCM)^3
OOP=VMC*AOS*SIN(BETA1)-VMC*DTC*SIN(PHIC)
ADJ=VSC/2-VMC*AOS*COS(BETA1)+VMC*DTC*COS(PHIC)
CALL VECTOR(OOP,ADJ,THC,AMC)
OOP=VME*AOS*SIN(BETA1)-VME*DTE*SIN(PHIE)-VSE/2*SIN(alpha)
ADJ=VSE/2*COS(alpha)-VME*AOS*COS(BETA1)+VME*DTE*COS(PHIE)
CALL VECTOR(OOP,ADJ,THE,AME)
CC(1)=(TCM*PI/4*AOS/Z*VSC*SIN(BETA1)/AMC+TK)/(1+PI/4*SIN(BETA1+THC)*AOS/Z)
CC(2)=(TEM*PI/4*AOS/Z*VSE*SIN(BETA1-alpha)/AME+TH)/(1+PI/4*AOS/Z*SIN(BETA1+THE))
BC1=CC(1): BC2=CC(1)*AOS/Z
BE1=CC(2): BE2=CC(2)*AOS/Z
BC3=CC(1)
BE3=CC(2)
OOP=CON5*SIN(alpha)-AOS*(1-TAUC/GAM)*SIN(BETA1)
ADJ=CON5*COS(alpha)-AOS*(1-TAUC/GAM)*COS(BETA1)
```

```
CALL VECTOR(OOP,ADJ,ALFC,BC4)
BC4=BC3*BC4
OOP=CON3*SIN(alpha)-AOS*(1-TAUE/GAM)*SIN(BETA1)
ADJ=CON3*COS(alpha)-AOS*(1-TAUE/GAM)*COS(BETA1)
CALL VECTOR(OOP,ADJ,ALFE,BE4)
BE4=BE3*BE4
TCM1=TCM
TCM=.5*(BC1+BC3)+BC2*SIN(BETA1+THC)/PI+BC4*SIN(THC+ALFC)/PI
TEM=.5*(BE1+BE3)+BE2*SIN(BETA1+THE)/PI+BE4*SIN(THE+ALFE)/PI
OOP=.5*(BC2*SIN(BETA1)-BC4*SIN(ALFC))
ADJ=.5*(BC2*COS(BETA1)-BC4*COS(ALFC))
CALL VECTOR(OOP,ADJ,PHIC,TCA)
OOP=.5*(BE2*SIN(BETA1)-BE4*SIN(ALFE))
ADJ=.5*(BE2*COS(BETA1)-BE4*COS(ALFE))
CALL VECTOR(OOP,ADJ,PHIE,TEA)
WEND
WC=-DAT(1,1)*AOS*VSC/2*PI*SIN(BETA1)
WE=-DAT(1,1)*AOS*VSE/2*PI*SIN(BETA1-alpha)
WTI=WC+WE
QDK=F*WC
QDH=F*WE
beta*=BETA1
BETAD*=-(UCA-PI-beta*)
P1=DAT(1,1)*AOS
PWI=-PI*AP*P1*XP*SIN(BETAD*):Piston work ideal
VF=(2*(VR/GAM+VK+VCLC+VH+VCLE)+VSC+VSE)*AOS
QR=VF^2+VSC^2+VSE^2+2*(VSC*VSE*COS(alpha)-VF*VSC*COS(beta*)-VF*VSE*COS(alpha-beta*))
QDR=F*DAT(1,1)*CP/R*SQR(ABS(QR))
FR=O*DAT(1,1)/R
PHWK=THC
PHWH=THE-PI
WK1=AMC*FR/TCM
WH1=AME*FR/TEM
THK=PI/2-alpha+PHI*-PHWK
THH=PI/2-alpha+PHI*-PHWH
OOP=.5*(WK1*SIN(THK)+WH1*SIN(THH))
ADJ=.5*(WK1*COS(THK)+WH1*COS(THH))
CALL VECTOR(OOP,ADJ,THRR,WR)
RETURN
HEAT:
'Peak RE -- for half peak divide by 2
REK=DHK*WK1/(AK*MUK)
REH=DHH*WH1/(AH*MUH)
IF REK<2000 THEN KT$="L" ELSE KT$="Tr"
IF REH<2000 THEN HT$="L" ELSE HT$="Tr"
IF REK>4000 THEN KT$="T"
IF REH>4000 THEN HT$="T"
REP=WR*DHR/(ARR*MUR)
'CL=33.3+.458*REP: Ergun equation
CL=40+1.047*REP: 'Modified Ergun
FD=MUR*CL*DAT(9,3)/(DHR*DHR): 'No multiplier included
CALL TFRIC(0,PRK,REK,DAT(5,2),DHK,DAT(4,2),ARK,NUK,TFK,TT)
HTK=NUK*KK/DHK: X=SQR(HTK)*FINK
```

```
 IF DAT(21,5)=0 OR DAT(4,2)=0 THEN FINEF=1 ELSE GOSUB FIN.E
FINFK=FINEF
  DTK=QDK*(1/(FINEF*AWK*HTK)+DAT(7,5)/DAT(23,1)/AQK): TK=TWK-DTK
   CALL TFRIC(0,PRH,REH,DAT(6,2),DHH,DAT(4,3),ARH,NUH,TFH,TT)
 HTH=NUH*KH/DHH: X=SQR(HTH)*FINH
 IF DAT(22,2)=0 OR DAT(4,3)=0 THEN FINEF=1 ELSE GOSUB FIN.E
FINFH=FINEF
  DTH=QDH*(1/(FINEF*AWH*HTH)+DAT(8,5)/DAT(23,2)/AQH): TH=TWH-DTH
IF TK<TWK THEN TK=TWK
IF TH<TWK THEN TH=TWK
THW=ABS(PHWH-PHWK)
 SR=SQR(2*(1-COS(THW)))
 THR=ATN(SIN(THW)/(COS(THW)-1))
GO=WR/DAT(9,2)
 RERH=GO*DHR/(4*DAT(9,4)*MUR*.91)
IF RERH<50 THEN J=.91/RERH^.51*.91 ELSE J=.61/RERH^.41*.91
H=J*CP*GO/PRR^.666
NTU=H*AWR/(CP*WR)
E=NTU*SR/(PI+NTU*SR)
PMPR=FD*(WR/RHOR)^2/ARR
 FK=8*(TFK*DAT(5,2)/DHK+HLK/4!)/(RHOK*AK)^2*WK1^3
 FH=8*(TFH*DAT(6,2)/DHH+HLH/4!)/(RHOH*AH)^2*WH1^3
 IF REK<2000 THEN PMPK=FK/16! ELSE PMPK=FK*(TT/16-TT/(3*PI)+.333)/PI
 IF REH<2000 THEN PMPH=FH/16! ELSE PMPH=FH*(TT/16-TT/(3*PI)+.333)/PI
RETURN
FIN.E: Fin effectiveness
  FINEF=(EXP(X)-1/EXP(X))/(EXP(X)+1/EXP(X))/X
RETURN
RETURN
GAS:
 IF DAT(4,1)=0 OR DAT(4,1)>4 PRINT "ERROR IN GAS TYPE"
 ON DAT(4,1) GOSUB HE,H2,AIR,ARGON
 CP=GAM*R/(GAM-1)
  MUK=MU0*(273+TSU)/(TK+TSU)*(TK/273)^1.5
  MUH=MU0*(273+TSU)/(TH+TSU)*(TH/273)^1.5
 MUR=(MUK+MUH)/2!
 KK=KT1*TK+KT2: KH=KT1*TH+KT2: KR=(KK+KH)/2
  PRH=CP*MUH/KH: PRK=CP*MUK/KK: PRR=CP*MUR/KR
RETURN
   HE:
      GAM=1.67: R=2076!: TSU=80: MU0=1.885E-05
     KT1=.000253: KT2=.066
   RETURN
   H2:
      GAM=1.41: R=4121.735: TSU=84.4: MU0=8.35E-06
     KT1=.00041: KT2=.069
   RETURN
    AIR:
      GAM=1.4: R=287!: TSU=112: MU0=1.708E-05
     KT1=.0000596: KT2=.0086
    RETURN
    ARGON:
     GAM=1.67: R=208.195: TSU=133: MU0=.0000212
```

```
KT1=.0000535: KT2=.00175
RETURN

KDEXT:
IF DAT(10,1)<>2 THEN KD1=GAM*DAT(1,1)*APK*APK/DAT(10,3) ELSE KDE=DAT(5,5): RETURN
IF DAT(10,1)=0 THEN KDE=KD1:RETURN
KDE=KD1+KD1*DAT(10,3)/DAT(10,2):Rod area not accounted for
RETURN
Q:
DDK=2*RHOK*DAT(2,4)*PMPK/WK1*COS(THK)
DDH=2*RHOH*DAT(2,4)*PMPH/WH1*COS(THH)
DDR=2*RHOR*DAT(2,4)*PMPR/WR*COS(THRR)
DDV=(DDK+DDH+DDR)/(O*XD*SIN(PHI#))
DDPK=-2*RHOK*DAT(2,4)*PMPK/WK1*SIN(THK)
DDPH=-2*RHOH*DAT(2,4)*PMPH/WH1*SIN(THH)
DDPR=-2*RHOR*DAT(2,4)*PMPR/WR*SIN(THRR)
DDP=(DDPK+DDPH+DDPR)/(O*XP)-DDV*XR*COS(PHI#)
IF DAT(13,2)>0 OR DAT(13,4)>0 THEN GOSUB D.SEAL
IF DAT(16,3)>0 THEN GOSUB GS.SEAL
DDGL=2*(DGL+HYSH)/(O*XD)^2
DID=DDV+DGS+DS+DG+DDGL
IF DAT(13,3)>0 OR DAT(13,4)>0 THEN GOSUB P.SEAL
IF ETYPE%=0 THEN RETURN
QD=OD*DAT(2,1)/(2!*PI*DID)
IF NC%>5 AND BETAD#>0 THEN QP=1000 : RETURN
DP=2!*PPV/(XP*O)^2
QP=OP*DAT(2,2)/(2!*PI*ABS(DP))
FQP=OD*OP/(O*O)*(1!-ALFP*ALFT/(KD*KP))-(OD^2+OP^2-O^2)/(OD*OP)
QPT=(1!/FQP)/(4!*PI*PI*QD)
RETURN
D.SEAL:
PA=P1:PHIA=-PHI#+BETAD#
IF DAT(15,4)=1 THEN PB=0:PHIB=PI
IF DAT(15,4)=2 THEN YB=O*XD*DDV: YC=-O*XP*DDP: YA=SQR(YB^2+YC^2-2*YB*YC*COS(PHI#)
IF DAT(15,4)=2 THEN YX=YB/YA*SIN(PHI#): PHIB=ATN(YX/SQR(ABS(1-YX*YX)))-PI/2: PB=YA/DAT(2,4)
IF DAT(12,4)=1 THEN GOTO CLR.D
DPF=2*ABS(PA*COS(PHIA)-PB*COS(PHIB))+PI*ABS(PA*SIN(PHIA)-PB*SIN(PHIB))
DSP=.5*O*XD*(DAT(12,1)*DAT(13,2)*DD*.667*DPF+4!*DAT(15,1)*PI)
DS=2*DSP/(O*XD)^2: RETURN
CLR.D:
For clearance loss
RETURN
P.SEAL:
Mean pressures on each side of seal are assumed equal
PA=P1:PHIA=BETAD#:PB=GAM*DAT(1,1)*AP*XP/DAT(3,3):PHIB=PI
SC1=MUK*(O*XP)^2/2: SC2=PA^2-2*PA*PB*COS(PHIA-PHIB)+PB^2
IF DAT(12,5)=1 THEN GOTO CLR.P
DPF=2*ABS(PA*COS(PHIA)-PB*COS(PHIB))+PI*ABS(PA*SIN(PHIA)-PB*SIN(PHIB))
PSP=.5*O*XP*(DAT(12,2)*DAT(13,3)*DD*.667*DPF+4*DAT(15,2)*PI)
LP=0: GOTO R.SEAL
CLR.P:
LP=PI*DD*(DAT(14,1)*SC1/CLP+CLP^3*SC2/(24*MUK*DAT(14,1))): PSP=0
R.SEAL:
```

```
    IF DAT(13,1)=1 THEN GOTO CLR.R
     RSP=.5*O*XP*(DAT(12,3)*DAT(13,4)*DR*.667*DPF+4*DAT(15,3)*PI)
    LRP=0: RETURN
    CLR.R:
      LRP=PI*DR*(DAT(14,2)*SC1/CLR+CLR^3*SC2/(24*MUK*DAT(14,2))): RSP=0
  RETURN
  GS.SEAL:
    IF DAT(10,1)=2 THEN RETURN
     IF DAT(10,1)=0 THEN PA=0 ELSE PA=GAM*DAT(1,1)*APK*XD/DAT(10,2)
     PB=GAM*DAT(1,1)*APK*XD/DAT(10,3)
    PHIA=PI: PHIB=0
    IF DAT(16,2)=1 THEN GOTO CLR.GS
     DPF=2*ABS(PA*COS(PHIA)-PB*COS(PHIB))+PI*ABS(PA*SIN(PHIA)-PB*SIN(PHIB))
      GSP=.5*O*XD*(DAT(17,3)*DAT(16,3)*DAT(10,4)*.667*DPF+4*DAT(17,1)*PI)
     DG=2*GSP/(O*XD)^2: LG=0: RETURN
    CLR.GS:
     SC1=MUR*(O*XD)^2/2: SC2=PA^2-2*PA*PB*COS(PHIA-PHIB)+PB^2
      LG=PI*DAT(10,4)*(DAT(16,4)*SC1/CLGS+CLGS^3*SC2/(24*MUR*DAT(16,4)))
     DG=2*LG/(O*XD)^2: GSP=0
    RETURN
END SUB
'-----------------------Subprograms-----------------------------------------

   SUB TFRIC(O,PR,RE,RL,DH,NT,AR,RNU,FRIC,TT)STATIC: 'Friction and heat transfer
    PI=3.14159
  'Knudsen & Katz pp231,388 and Trikha
      TF=40!/(1!+6.4E+07/O^2)+8.1/(1!+40000!/O^2)+1!/(1!+700!/O^2)
    PE=RE*PR
    LD=RL/DH
    LOD=LD/PE
    IF NT=1 THEN RNU=.43*AR+2.51
    IF NT=1 AND RNU>7.6 THEN RNU=7.6
     'IF NT=1 THEN RNU=(1!+(.003+.039/AR)/LOD)*RNU
     IF NT=1 THEN FRIC=(1!+.5*TF)*(12.4+1.95*AR-.106*AR*AR)/RE
      IF NT=0 THEN RNU=(3.66+.104/(LOD)/(1!+.016/LOD^.8))
     IF NT=0 THEN FRIC=(1!+.5*TF)*3.435*SQR(LOD*PR)*DH/RL
    IF RE<2000 THEN EXIT SUB
    RT=2000/RE
    ARG=-RT*RT+1
     IF ARG=0 THEN TT=PI/2 ELSE TT=ATN(RT/SQR(ARG))
    T=1!+101.3/(RE^.75)*TF
    IF RE<3000 THEN FRID=T*.0007343*RE^.3142 ELSE FRID=T*.0791/RE^.25
     FRIC=((FRIC*TT+FRID*(PI/2-TT))*2/PI+FRID)/2
    DF=(1/LD)^.054: 'Factor for entrance effect [Knudsen & Katz pp401]
     'RNUT=FRIC/2!*RE*PR/(1+5*SQR(FRIC/2)*(PR-1)): 'Prandtl analogy for Nu
    RNUT=.036*RE^.8*PR^.33*DF: 'Correlation equation
     'RNUT=RNUT*(1+3.1/LD): 'Factor for entrance configuration
     RNU=((RNU*TT+RNUT*(PI/2-TT))*2/PI+RNUT)/2
  END SUB
   SUB VECTOR(OP,ADJ,ANGLE,SIDE)STATIC: 'Phasors
  PI=3.14159265#
    SIDE=SQR(OP^2+ADJ^2)
    ANGLE=ATN(OP/ADJ)
```

```
  IF OP<0 AND ADJ>0 THEN ANGLE=ANGLE+2*PI
  IF ADJ<0 THEN ANGLE=ANGLE+PI
END SUB
'----------------End sub programs for linear dynamics-----------------------------------
  SUB GRAPH(DOS,PIOS,PIS,CLC,CLE,P1,P2,BETAD,XP,XR,PHI,O,OP,OD,QP,QD)STATIC
   WINDOW 1,,(0,20)-(220,275),3
  XD=XP*XR
  PI=3.14159
  BDEG=BETAD*180!/PI
  PHDEG=PHI*180!/PI
  F=O/(2!*PI)
  PRINT "Piston amp.",XP
  PRINT "Displacer amp.",XD
  PRINT "Pressure amp.",P1
  PRINT "What output scales would"
   INPUT "you like (mm/mm)?",OS
   INPUT "and (bar/mm)?",OSP
  CLS
  CIRCLE (100,100),4
 'Lines per centimetre is LPC
 'Columns per centimetre is CPC
 LPC=30
 CPC=30
 IYL=255/LPC
 IXL=220/CPC
 OSY=100 MOD LPC
 OSX=100 MOD CPC
 FOR IY=0 TO IYL
  LINE (0,OSY+IY*LPC)-(220,OSY+IY*LPC)
 NEXT IY
 FOR IX=0 TO IXL
  LINE (OSX+IX*CPC,0)-(OSX+IX*CPC,280)
 NEXT IX
 XSCALE=CPC*100!/OS
 YSCALE=LPC*100!/OS
 PYSC=LPC/1E+07/OSP
 'If not specified, set limits - piston 16% overstroke to mechanical limits
 'displacer 50% "out" overstroke to mechanical limits
 IF DOS=0 THEN DOS=1.5*XD
 IF PIOS=0 THEN PIOS=1.08*XP
 IF PIS=0 THEN PIS=1.08*XP
 TLX=100-PIS*XSCALE
 TRX=100+PIOS*XSCALE
 TRY=100-CLE*YSCALE
 TLY=TRY
 BRX=TRX
 BRY=100+DOS*YSCALE
 CALL PENSIZE (2,2)
 CALL MOVETO(TLX,TLY)
 CALL LINETO(TRX,TRY)
 CALL MOVETO(TRX,TRY)
 CALL LINETO(BRX,BRY)
 BPX=BRX-(PIOS-DOS+CLC)*XSCALE
```

```
CALL MOVETO(BRX,BRY)
CALL LINETO(BPX,BRY)
BDY=TLY+(CLE-PIS+CLC)*YSCALE
CALL MOVETO(TLX,TLY)
CALL LINETO(TLX,BDY)
CALL LINETO(BPX,BRY)
CALL PENNORMAL
FOR I=1 TO 181
THETA=I/90!*3.14159
XPA=XP*SIN(THETA)
XDA=XD*SIN(THETA+PHI)
XPP=100-XPA*XSCALE
XDP=100-XDA*YSCALE
IF I=1 THEN LINE (XPP,XDP)-(XPP,XDP) ELSE LINE -(XPP,XDP)
NEXT I
PI2=2!*3.14159
FOR I=1 TO 181
THETA=I/90!*3.14159
P=P1*SIN(THETA+BETAD)-P2*SIN(2!*(THETA+BETAD))
PP=70+2*LPC-P*PYSC
XPA=XP*SIN(THETA)
XPP=100-XPA*XSCALE
IF I=1 THEN LINE (XPP,PP)-(XPP,PP) ELSE LINE -(XPP,PP)
NEXT I
CALL TEXTSIZE(9)
CALL TEXTFONT(142)
CALL MOVETO(3,246): PRINT"Σp =
CALL TEXTMODE(1): TEXTFONT(1)
CALL MOVETO(30,246): PRINT USING "###.#";OP;
CALL TEXTFONT(142)
CALL MOVETO(65,246): PRINT"Σd =";
CALL TEXTFONT(1)
CALL MOVETO(92,246): PRINT USING "###.#";OD;
CALL MOVETO(124,246): PRINT"Qp =";
CALL MOVETO(141,246): PRINT USING "##.#";QP;
CALL MOVETO(171,246): PRINT"Qd =";
CALL MOVETO(188,246): PRINT USING "##.#";QD;
CALL TEXTMODE(0)
CALL MOVETO(3,228): PRINT"p =
CALL TEXTMODE(1)
CALL MOVETO(20,228): PRINT USING "##.##^^^^";P1
CALL TEXTFONT(142)
CALL MOVETO(70,228): PRINT"□"
CALL TEXTFONT(1): CALL MOVETO(80,228): PRINT USING "###.#";BDEG
CALL TEXTMODE(0)
CALL MOVETO(3,215): PRINT"Xd =
CALL TEXTMODE(1)
CALL MOVETO(25,215): PRINT USING "##.##^^^^";XD
CALL TEXTFONT(142)
CALL MOVETO(75,215): PRINT"□"
CALL TEXTFONT(1): CALL MOVETO(80,215): PRINT USING "###.#";PHDEG
CALL TEXTMODE(0)
CALL MOVETO(3,202): PRINT"Xp =
```

```
CALL TEXTMODE(1)
 CALL MOVETO(25,202): PRINT USING "##.##^^^^";XP
CALL TEXTFONT(142)
  CALL MOVETO(75,202): PRINT"□";: CALL TEXTFONT(1): CALL MOVE(4,0): PRINT"0"
CALL TEXTMODE(0)
CALL TEXTFONT(1)
 CALL MOVETO(3,189): PRINT"f =
CALL TEXTMODE(1)
 CALL MOVETO(20,189): PRINT USING"###.#";F
 CALL TEXTMODE(0): CALL TEXTSIZE(12)
END SUB
SUB PERF(PI,O,PWR,TH,TK,QDK,QDH,ETA,FK,FH,DTK,DTH,HYSH,GSH,E)STATIC
DIM A$(20)
 WINDOW 3,,(0,275)-(520,350),3
 TWK=TK+DTK: TWH=TH+DTH
 CEG=1-TK/TH: CEW=1-TWK/TWH
 CG=ETA/CEG: CW=ETA/CEW
  LET A$(1)="Power =" : LET A$(2)="Efficiency =" : LET A$(3)="Heat out =" : LET A$(4)="Heat in ="
  LET A$(5)="Tk =": LET A$(6)="Th =" : LET A$(7)="Cooler ΔT =": LET A$(8)="Heater ΔT ="
  LET A$(9)="Cooler fin eff. =": LET A$(10)="Heater fin eff. =": LET A$(11)="Regen. Effect.="
  LET A$(12)="% Carnot (gas temp.) =": LET A$(13)="% Carnot (wall temp.) ="
CALL TEXTSIZE(9)
 CALL MOVETO(40,13): PRINT USING "##.##^^^^";PWR
 CALL MOVETO (185,13): PRINT USING "##.##^^^^";QDK
 CALL MOVETO(315,13): PRINT USING "##.##^^^^";QDH
 CALL MOVETO(460,13): PRINT USING "#.###";ETA
CALL MOVETO (4,13)
CALL TEXTMODE(1)
  PRINT A$(1);PTAB(120);A$(3);PTAB(260);A$(4);PTAB(390);A$(2)
 CALL MOVETO(40,26): PRINT USING "####.##";TK
 CALL MOVETO (185,26): PRINT USING "####.##";TH
 CALL MOVETO(325,26): PRINT USING "###.##";DTK
 CALL MOVETO(460,26): PRINT USING "###.##";DTH
CALL MOVETO (4,26)
  PRINT A$(5);PTAB(120);A$(6);PTAB(260);A$(7);PTAB(390);A$(8)
 CALL MOVETO(85,39): PRINT USING "#.###";FK
 CALL MOVETO(250,39): PRINT USING "#.###";FH
 CALL MOVETO(420,39): PRINT USING "#.####";E
CALL MOVETO(4,39)
 PRINT A$(9);PTAB(165);A$(10);PTAB(340);A$(11);
 CALL MOVETO(120,52): PRINT USING "#.###";CG;
 CALL MOVETO(290,52): PRINT USING "#.###";CW;
 CALL MOVETO(4,52): PRINT A$(12);PTAB(175);A$(13)
CALL TEXTMODE(0): CALL TEXTSIZE(12)
ERASE A$
END SUB
SUB L.INV STATIC
   DIM A$(30),V(30)
    SHARED  PMPK,PMPR,PMPH,HYSK,HYSH,GSH,E
    SHARED QDR,LD,LP,LRD,LRP,LG,QDC,DSP,PSP,GSP,RSP
   SHARED QCR,SHL,DGL,BH,QCIW
    WINDOW 4,,(220,20)-(520,275),3
   QRL=(1-E)*QDR
```

```
CALL MOVE (90,0)
CALL TEXTSIZE(12): CALL TEXTFACE(1)
PRINT "Loss Inventory [W]"
CALL TEXTSIZE(9): CALL TEXTFACE(0)
  A$(1)="Cooler Diss.": A$(2)="Regen Diss.": A$(3)="Heater Diss."
  A$(4)="Comp. Hyst.": A$(5)="Exp. Hyst.": A$(6)="Gas Spr. Hyst."
  A$(7)="Bounce Hyst.": A$(8)="Disp. Seal Lkg.": A$(9)="Pist. Seal Lkg."
  A$(10)="Rod Sl. Lkg.(Disp.)": A$(11)="Rod Sl. Lkg.(Pist.)"
  A$(12)="Gas Spr. Lkg.": A$(13)="Gas Conduction": A$(14)="Regen. Wall Cond."
  A$(15)="Internal Wall Cnd.": A$(16)="Disp. Friction": A$(17)="Pist. Friction"
 A$(18)="Rod Fric. (Disp.)"
  A$(19)="Rod Fric. (Pist.)": A$(20)="Gas Spr. Sl. Fric.": A$(21)="Disp. Gap Loss"
  A$(22)="Shuttle Loss": A$(23)="Regen. Heat Loss"
  V(1)=PMPK: V(2)=PMPR: V(3)=PMPH: V(4)=HYSK: V(5)=HYSH
  V(6)=GSH: V(7)=BH: V(8)=LD: V(9)=LP: V(10)=LRD: V(11)=LRP
  V(12)=LG: V(13)=QDC: V(14)=QCR: V(15)=QCIW: V(16)=DSP: V(17)=PSP
V(18)=RDSP
  V(19)=RPSP: V(20)=GSP: V(21)=DGL: V(22)=SHL: V(23)=QRL
IX%=5
FOR I%=1 TO 23
 II%=I%* 12+14
  IF I%>20 THEN IX%=150: II%=II%-240
GOSUB P.LINE
NEXT I%
ERASE A$: ERASE V
EXIT SUB
P.LINE:
     CALL MOVETO(IX%,II%)
     PRINT USING "&";A$(I%);"..........
     CALL MOVETO(IX%+85,II%)
     PRINT USING "##.##^^^^";V(I%);
RETURN
END SUB
SUB SPLOT(TDAT(2),DAT(2),PWR0,ET0,F0,PHD0,PHP0,XR0)STATIC
    SHARED MTST%,ETYPE%
    DIM TITLE$(20),NBP%(7)
   FOR I%=1 TO 25
   FOR J%=1 TO 5
     TDAT(I%,J%)=DAT(I%,J%)
   NEXT J%: NEXT I%
    IF TDAT(4,2)=1 THEN CIRC=(TDAT(7,2)+TDAT(21,5))* TDAT(7,3)
    IF TDAT(4,3)=1 THEN CIRH=(TDAT(8,2)+TDAT(22,2))* TDAT(8,3)
   WINDOW 4,,(220,20)-(520,275),3
  CALL MOVE (90,0)
  CALL TEXTSIZE(14)
  CALL TEXTFACE(1)
  PRINT "Plot Selector"
  CALL TEXTSIZE(12)
  CALL TEXTFACE(0)
  PRINT "Select independent variable and then"
  PRINT "select any number of dependent variables."
  CALL TEXTFACE(4)
  PRINT "Independent variables"
```

```
CALL TEXTFACE(0)
    TITLE$(1)="Twk": TITLE$(2)="Twh": TITLE$(3)="Xp": TITLE$(4)="Md"
    TITLE$(5)="Mp": TITLE$(6)="Mc": TITLE$(7)="Ad": TITLE$(8)="Arod"
    TITLE$(9)="Lk": TITLE$(10)="Lr": TITLE$(11)="Lh": TITLE$(12)="Nk"
    TITLE$(13)="Nh": TITLE$(14)="PSI": TITLE$(15)="<p>":TITLE$(18)="Areg"
  IF DAT(10,1)=0 THEN TITLE$(16)="Vd" ELSE TITLE$(16)="Vd(o)": TITLE$(17)="Vd(i)"
 NJ%=0
 FOR J%=1 TO 4
 FOR I%=1 TO 5
  X1=5+(I%-1)*55: X2=X1+55
  Y1=70+(J%-1)*30: Y2=Y1+15
  NJ%=NJ%+1
   BUTTON NJ%,1,TITLE$(NJ%),(X1,Y1)-(X2,Y2),3
 NEXT I%: NEXT J%
IF DAT(10,1)=0 THEN BUTTON CLOSE 17: BUTTON CLOSE 19: BUTTON CLOSE 20
ACT=DIALOG(0)
WHILE ACT <> 1: ACT=DIALOG(0): WEND
 INB%=DIALOG(1): BUTTON INB%,2
CALL MOVE(0,120)
CALL TEXTFACE(4)
PRINT "Dependent variables"
CALL TEXTFACE(0)
  BUTTON 19,1,"Pwr",(5,210)-(60,225),3
  BUTTON 20,1,"Eta",(60,210)-(115,225),3
  BUTTON 21,1,"T",(115,210)-(170,225),3
  BUTTON 22,1,"Phid",(170,210)-(225,225),3
  BUTTON 23,1,"Phip",(225,210)-(280,225),3
  BUTTON 24,1,"XR",(5,240)-(60,255),3
  BUTTON 26,1,"Done",(225,170)-(270,190),1
K%=0
WHILE (DIALOG(1)<>26 OR K%>6)
ACT=DIALOG(0)
WHILE ACT <> 1: ACT=DIALOG(0): WEND
 K%=K%+1
 NBP%(K%)=DIALOG(1)
  IF NBP%(K%) < 19 THEN K%=K%-1: BEEP
  IF NBP%(K%)>=19 THEN NS%=2 ELSE NS%=1
 BUTTON NBP%(K%),NS%
WEND
FOR I%=1 TO 26
BUTTON CLOSE I%
NEXT I%
CLS
CALL TEXTSIZE(9)
PRINT "Note that finned heater and cooler outer diameters remain"
PRINT "constant when changing Nk and Nh thus the flow gaps"
PRINT "change accordingly."
 CALL TEXTSIZE(12): PRINT " "
PRINT"Dependent variables are normalized"
 INPUT "Upper bound of independent variable";UB
 INPUT "Lower bound of independent variable";LB!
PRINT "Number of increments? - "
 INPUT " suggest >= 10  ",INC%
```

```
CLS
 LINE (20,240)-(260,240)
 LINE (20,20)-(20,240)
 FOR I%=1 TO 20
   IX%=20+I%*12: IY%=240-I%*11
  LINE (IX%,240)-(IX%,236)
  LINE (20,IY%)-(24,IY%)
 NEXT I%
 CALL TEXTSIZE(9): CALL TEXTMODE(1)
  CALL MOVETO(2,250): PRINT USING "##.#^^^^";LB!;
  CALL MOVETO(240,250): PRINT USING "##.#^^^^";UB;
  CALL MOVETO(130,250): PRINT TITLE$(INB%);
  CALL MOVETO(3,240): PRINT USING "#.#";.5;
  CALL MOVETO(3,130): PRINT USING "#.#";1!;
  CALL MOVETO(3,20): PRINT USING "#.#";1.5;
 CALL TEXTSIZE(12): CALL TEXTMODE(0)
  D=(UB-LB!)/INC%: DD=D
   X=20: PP=130: PE=130: PF=130: PPH=130: PB=130: PX=130: ITST%=0
 FOR I%=0 TO INC%
  IF INB%=1 THEN TDAT(1,2)=LB!+I%*DD
  IF INB%=2 THEN TDAT(1,3)=LB!+I%*DD
  IF INB%=3 THEN TDAT(1,5)=LB!+I%*DD
  IF INB%=4 THEN TDAT(2,1)=LB!+I%*DD
  IF INB%=5 THEN TDAT(2,2)=LB!+I%*DD
  IF INB%=6 THEN TDAT(2,3)=LB!+I%*DD
  IF INB%=7 THEN TDAT(2,4)=LB!+I%*DD
  IF INB%=8 THEN TDAT(2,5)=LB!+I%*DD
  IF INB%=9 THEN TDAT(5,2)=LB!+I%*DD
  IF INB%=10 THEN TDAT(9,3)=LB!+I%*DD
  IF INB%=11 THEN TDAT(6,2)=LB!+I%*DD
  IF INB%=12 AND TDAT(4,2)=0 THEN TDAT(5,3)=LB!+I%*DD
  IF INB%=12 AND TDAT(4,2)=1 THEN TDAT(7,3)=LB!+I%*DD
   IF INB%=12 AND TDAT(4,2)=1 THEN TDAT(7,2)=CIRC/DAT(7,3)-DAT(21,5)
  IF INB%=13 AND TDAT(4,3)=0 THEN TDAT(6,3)=LB!+I%*DD
  IF INB%=13 AND TDAT(4,3)=1 THEN TDAT(8,3)=LB!+I%*DD
   IF INB%=13 AND TDAT(4,3)=1 THEN TDAT(8,2)=CIRH/DAT(8,3)-DAT(22,2)
 IF INB%=14 THEN TDAT(9,4)=LB!+I%*DD
 IF INB%=15 THEN TDAT(1,1)=LB!+I%*DD
 IF INB%=16 THEN TDAT(10,3)=LB!+I%*DD
 IF INB%=17 THEN TDAT(10,2)=LB!+I%*DD
 IF INB%=18 THEN TDAT(9,2)=LB!+I%*DD
IF  INB%=7  OR  INB%=18  THEN  TDAT(17,4)=(TDAT(2,4)+TDAT(9,2))/(DAT(2,4)+DAT(9,2))*DAT(17,4)
    X1=X
     PP1=PP: PE1=PE: PF1=PF: PPH1=PPH: PB1=PB: PX1=PX
    F=F0: PHIL#=PHD0: BETAD=PHP0: XR=XR0
     IF I%=0 THEN NTST%=0
INIT:     CALL  LINDYN(TDAT(),PPV,ETA,F,PHIL#,BETAD,XR,KT$,HT$,NTST%,MTST%,ETYPE%)
     IF NTST%=0 THEN NTST%=1: GOTO INIT
    WINDOW OUTPUT 4
    PHI=PHIL#
    PP=240-PPV/PWR0*110
    PE=240-ETA/ET0*110
    PF=240-F/F0*110
```

```
PPH=240-PHI/PHD0*110
PB=240-BETAD/PHP0*110
PX=240-XR/XR0*110
X=20+240*I%/INC%
CALL MOVETO(30,30): PRINT USING "&";"Cooler flow: ";KT$;
CALL MOVETO(150,30): PRINT USING "&";"Heater flow: ";HT$;
FOR K%=1 TO 6
IF NBP%(K%)=19 AND PP1<240 THEN PSET(X,PP),33
IF NBP%(K%)=20 AND PE1<240 THEN LINE(X1,PE1)-(X,PE)
IF NBP%(K%)=21 AND PF1<240 THEN LINE(X1,PF1)-(X,PF)
IF NBP%(K%)=22 AND PPH1<240 THEN LINE(X1,PPH1)-(X,PPH)
IF NBP%(K%)=23 AND PB1<240 THEN LINE(X1,PB1)-(X,PB)
IF NBP%(K%)=24 AND PX1<240 THEN LINE(X1,PX1)-(X,PX)
NEXT K%
NEXT I%
FOR K%=1 TO 6
CALL TEXTMODE(1)
IF NBP%(K%)=19 THEN CALL MOVETO(X,PP): PRINT "Pwr"
CALL TEXTFONT(142)
IF NBP%(K%)=20 THEN CALL MOVETO(X,PE): PRINT ""
IF NBP%(K%)=21 THEN CALL MOVETO(X,PF): PRINT "f"
IF NBP%(K%)=22 THEN CALL MOVETO(X,PPH): PRINT "fd"
IF NBP%(K%)=23 THEN CALL MOVETO(X,PB): PRINT "fp"
CALL TEXTFONT(1)
IF NBP%(K%)=24 THEN CALL MOVETO(X,PX): PRINT "XR"
CALL TEXTMODE(0)
NEXT K%
WINDOW CLOSE 2
ERASE TITLE$,NBP%
END SUB
SUB   OPTIMIZE(TDAT(2),DAT(2),NBP%(1),PWR0,ET0,F0,PHIG,BETAD,XR0)STATIC
SHARED MTST%,ETYPE%
DIM TITLE$(15)
READ iprint%, itmax%, nkat%
DATA 1.,2001,20!
READ halpha,hbeta,epsy
DATA 1.,0.5,0.001
FOR I%=1 TO 25
FOR J%=1 TO 5
TDAT(I%,J%)=DAT(I%,J%)
NEXT J%: NEXT I%
WINDOW 4,,(220,20)-(520,275),3
CALL MOVE (70,0)
CALL TEXTSIZE(14)
CALL TEXTFACE(1)
PRINT"Efficiency Optimizer"
CALL TEXTSIZE(12): CALL TEXTFACE(0)
PRINT"Select up to four variables for optimisation."
TITLE$(1)="Xp": TITLE$(2)="Md": TITLE$(3)="Mp": TITLE$(4)="Arod"
TITLE$(5)="Lk": TITLE$(6)="Lr": TITLE$(7)="Lh": TITLE$(8)="Nk"
TITLE$(9)="Nh": TITLE$(10)="PSI": TITLE$(11)="Areg"
NJ%=0
FOR J%=1 TO 2
```

```
FOR I%=1 TO 5
 X1=5+(I%-1)*55: X2=X1+55
 Y1=70+(J%-1)*30: Y2=Y1+15
 NJ%=NJ%+1
  BUTTON NJ%,1,TITLE$(NJ%),(X1,Y1)-(X2,Y2),3
NEXT I%: NEXT J%
 BUTTON 11,1,TITLE$(11),(5,130)-(60,145),3
 BUTTON 20,1,"Done",(225,170)-(270,190),1
K%=0
WHILE (K%<4 AND DIALOG(1)<>20)
ACT=DIALOG(0)
WHILE ACT<>1: ACT=DIALOG(0): WEND
 IF DIALOG(1)<>20 THEN K%=K%+1: NBP%(K%)=DIALOG(1): BUTTON NBP%(K%),2
WEND
FOR I%=1 TO 10
BUTTON CLOSE I%
NEXT I%: BUTTON CLOSE 11: BUTTON CLOSE 20
CLS
NTST%=0: 'New start for each run
hqd=0!
CALL    hooke    (MTST%,NTST%,ETYPE%,F0,PHIG,BETAD,XR0,TDAT(),NBP%(),K%,Itmax%,nkat%,epsy,halpha,hbeta,
hqd,Iprint%,ET0,X)
 ERASE TITLE$
OUTO:
   MENU OFF
      INFILE$=FILES$(0,"Save Opt file as...")
      IF INFILE$="" THEN GOTO OUTO
    OPEN INFILE$ FOR OUTPUT AS #1
    FOR I%=1 TO 25
       WRITE#1,TDAT(I%,1),TDAT(I%,2),TDAT(I%,3),TDAT(I%,4),TDAT(I%,5)
    NEXT I%
    CLOSE #1: NAME INFILE$ AS INFILE$, "DATA"
END SUB


SUB      hooke(MTST%,NTST%,ETYPE%,F,PHIG,BETAD,XR,TDAT(2),NBP%(1),NSTAGE%,maxk%,nkat%,epsy,
halpha,hbeta,hqd,Iprint%,ET0,X)  STATIC
DIM  RK(4),EPS(4),hq(4),qq(4),w(4)
PHIL#=PHIG
PRINT "halpha, hbeta, maxk%, nkat%"
PRINT  halpha,hbeta,maxk%,nkat%
PRINT  "NSTAGE%",NSTAGE%
PRINT "Initial step sizes"
FOR I%=1 TO NSTAGE%
  IF NBP%(I%)=1 THEN RK(I%)=TDAT(1,5)
  IF NBP%(I%)=2 THEN RK(I%)=TDAT(2,1)
  IF NBP%(I%)=3 THEN RK(I%)=TDAT(2,2)
  IF NBP%(I%)=4 THEN RK(I%)=TDAT(2,5)
  IF NBP%(I%)=5 THEN RK(I%)=TDAT(5,2)
  IF NBP%(I%)=6 THEN RK(I%)=TDAT(9,3)
  IF NBP%(I%)=7 THEN RK(I%)=TDAT(6,2)
  IF NBP%(I%)=8 THEN RK(I%)=TDAT(5,3)
  IF NBP%(I%)=9 THEN RK(I%)=TDAT(6,3)
```

```
     IF NBP%(I%)=10 THEN RK(I%)=TDAT(9,4)
     IF NBP%(I%)=11 THEN RK(I%)=TDAT(9,2)
  EPS(I%)=RK(I%)*.05
  PRINT    "I,eps(I%)",I%,EPS(I%)
  NEXT I%
  PRINT "epsy",epsy
  kflag%=0
  FOR I%=1 TO NSTAGE%
  hq(I%)=RK(I%)
  w(I%)=0!
  NEXT I%
  kat% =0
  kk1%=0
  70 : kcount%=0
     wbest=w(NSTAGE%)
  FOR I%=1 TO NSTAGE%
    GOSUB DAT.TRNS
  NEXT I%
      CALL LINDYN(TDAT(),PPV,ETA,F,PHIL*,BETAD,XR,KT$,HT$,NTST%,MTST%,ETYPE%)
    WINDOW OUTPUT 4
    SUM=-ETA
     kk1%=kk1%+1
    bo=SUM
    IF kk1% = 1 THEN hqd=SUM: GOTO H1
    IF bo > hqd THEN kflag%=1 ELSE hqd=bo

' Establishing the search pattern
H1:  FOR I%=1 TO NSTAGE%
       qq(I%)=RK(I%)
       tsrk=RK(I%)
        RK(I%)=RK(I%)+EPS(I%)
     GOSUB DAT.TRNS
         CALL LINDYN(TDAT(),PPV,ETA,F,PHIL*,BETAD,XR,KT$,HT$,NTST%,MTST%,ETYPE%)
      WINDOW OUTPUT 4
       SUM=-ETA
       kk1%=kk1%+1
       w(I%)=SUM
      IF w(I%) < hqd THEN GOTO 58
        RK(I%)=RK(I%)-2!*EPS(I%)
      GOSUB DAT.TRNS
         CALL LINDYN(TDAT(),PPV,ETA,F,PHIL*,BETAD,XR,KT$,HT$,NTST%,MTST%,ETYPE%)
      WINDOW OUTPUT 4
      SUM=-ETA
       kk1%=kk1%+1
       w(I%)=SUM
      IF w(I%) < hqd THEN GOTO 58
      RK(I%)=tsrk
       IF I%=1 THEN w(I%)=bo ELSE w(I%)=w(I%-1)
      kcount%=1+kcount%
     GOTO 55
58 :   hqd=w(I%)
      qq(I%)=RK(I%)
55 :   PRINT "I=",I%
```

```
        NEXT I%
        IF Iprint%=0 THEN GOTO 65 ELSE GOTO 60
60 :    CALL MOVETO(0,0): CLS
        PRINT"number of function evaluations",kk1%
'
' Record responses and location
'
PRINT "End of each pattern search"
PRINT"Variables"
FOR I%=1 TO NSTAGE%
PRINT RK(I%)
NEXT I%
PRINT"Sum",hqd
         'ETA=-hqd
         'PE=240-ETA/ETO*110: X=X+1
         'PSET (X,PE)
'
' Test to determine termination
'
65 : IF kk1%>maxk% THEN GOTO 94
      IF kat% >= nkat% THEN GOTO 94
      hhf=w(NSTAGE%)-wbest
      IF ABS(hhf)<=epsy THEN GOTO 94
'
' If all axes fail to reduce step size
'
   IF kcount% >=NSTAGE% THEN GOTO 28
   FOR I%=1 TO NSTAGE%
     PRINT "hq(i%)",I%,hq(I%)
    PRINT "RK(I%)",RK(I%)
   PRINT "Halpha",halpha
      RK(I%)=RK(I%)+halpha*(RK(I%)-hq(I%))
   NEXT I%
      FOR I%=1 TO NSTAGE%
       hq(I%)=qq(I%)
      NEXT I%
      GOTO 70
'
' Reduce step size
'
28 : kat%=kat%+1
      IF kflag%=1 THEN GOTO 202
      GOTO 204
202 : kflag%=0
      FOR I%=1 TO NSTAGE%
       RK(I%)=hq(I%)
      NEXT I%
204 : FOR I%=1 TO NSTAGE%
       EPS(I%)=EPS(I%)*hbeta
      NEXT I%
      IF Iprint%=0 THEN GOTO 70
      PRINT "number of times step size reduced",kat%
      GOTO 70
```

```
94 :  PRINT "The final eps are"
      FOR I%=1 TO NSTAGE%
      PRINT EPS(I%)
      NEXT I%
      PRINT "The final rk are"
      FOR I%=1 TO NSTAGE%
      PRINT RK(I%)
      NEXT I%
      PRINT "The minimum response is",hqd
      PRINT "Final x's"
      FOR I%=1 TO NSTAGE%
       PRINT I%,RK(I%)
      NEXT I%
      PRINT "Number of function evaluations",kk1%
     EXIT SUB
   DAT.TRNS:
    IF NBP%(I%)=1 THEN TDAT(1,5)=RK(I%)
    IF NBP%(I%)=2 THEN TDAT(2,1)=RK(I%)
    IF NBP%(I%)=3 THEN TDAT(2,2)=RK(I%)
    IF NBP%(I%)=4 THEN TDAT(2,5)=RK(I%)
    IF NBP%(I%)=5 THEN TDAT(5,2)=RK(I%)
    IF NBP%(I%)=6 THEN TDAT(9,3)=RK(I%)
    IF NBP%(I%)=7 THEN TDAT(6,2)=RK(I%)
    IF NBP%(I%)=8 THEN TDAT(5,3)=RK(I%)
    IF NBP%(I%)=9 THEN TDAT(6,3)=RK(I%)
    IF NBP%(I%)=10 THEN TDAT(9,4)=RK(I%)
    IF NBP%(I%)=11 THEN TDAT(9,2)=RK(I%)
  RETURN
END SUB
```

APPENDIX F

REFERENCES

An70 Anvoner S. *Solution of Problems in Mechanics of Machines*, London: Pitman, 1970, vol 1.

BB81 Berchowitz D M and Berggren R W. *Appendix Gap Losses in Reciprocating Machines*, Latham, New York: Mechanical Technology Inc, 1981. (Report No 81ASE187ER16, prepared for NASA Lewis Research Center).

Be78 Berchowitz D M. *A Computer and Experimental Simulation of Stirling Cycle Machines*, MSc Dissertation, University of the Witwatersrand, Johannesburg, South Africa, 1978.

Be81.1 Berchowitz D M. *Double Acting Stirling Engine Phase Control*, U S Patent 4 395 880, 2 Aug 1983, Filed March 1981.

Be81.2 Berchowitz D M. *Independently Variable Phase and Stroke Control for a Double Acting Stirling Engine*, U S Patent 4 387 566, 14 June 1983, Filed March 1981.

Be82 Berchowitz D M. The design, development and performance of a duplex Stirling natural gas liquefier, *Proc 17th IECEC*, Los Angelese, California, Aug 1982, paper 789111, pp 1730-1738,

Be83 Berchowitz D M. The development of a 1kW electrical output free piston Stirling engine alternator unit, *Proc 18th IECEC*, Orlando, Florida, Aug 1983, paper 839146, pp 897-901.

Be84 Berchowitz D M. Stirling engines in developing countries, *Proc Small Engines and their Fuels in Developing Countries*, I Mech E and Inst of Energy, Univ of Reading, Sept 1984.

Bea69 Beale W T. *Free-Piston Stirling Engines - Some Model Tests and Simulations*, S A E Paper 690 230, 1969.

Bea71 Beale W T. *Stirling Cycle Type Thermal Device*, U S Patent 3 552 120, 5 Jan 1971.

Bej77 Bejan A. The concept of irreversibility in heat exchanger design: counterflow heat exchangers for gas-to-gas applications, *J Heat Transfer, Trans ASME*, vol 99, Aug 1977, pp 374-380.

Ben73 Benson G M. Thermal oscillators, *Proc 8th IECEC*, Philadelphia, Aug 1973, paper 739076, pp182-189.

Ben77.1 Benson G M. *Thermal Oscillators*, U S Patent 4 044 558, 1977.

Ben77.2 Benson G M. Thermal oscillators, *Proc 12th IECEC*, Washington D C, Aug 1977, paper 779247, pp 1478-1487.

Ber69 Bergles A E. Survey and evaluation of techniques to augment convective heat and mass transfer, *Prog in Heat and Mass Transfer*, vol 1, 1969, pp331-424.

BK77 Belaire R C and Kitzner E W. Ford Motor Company, private communication to I Urieli, quoted in *Stirling Cycle Engine Analysis* [UB84], 1977.

BM82 Berggren R W and Moynihan T. Effects of displacer seal clearance on free piston Stirling engine performance, *Proc 17th IECEC*, Los Angelese, Aug 1982, paper 829313, pp1885-1891.

BR71 Beale W T, Rauch J, Lewis R and Mulej D. *Free-Cylinder Stirling Engines for Solar-Powered Water Pumps*, ASME paper 71-WA/Sol-11, 1971.

BR75 Beale W T and Rankin C F. A 100W Stirling electric generator for solar or solid fuel heat sources, *Proc 10th IECEC*, Newark, Delaware, Aug 1975, paper 759152, pp 1020-1022.