

**A Budget Workstation  
for  
Numerical Modeling**

**Gregory W. Smith**

**“The time has come,” the Walrus said,  
“To talk of many things.  
Of Unix, telnet, ftp,  
Of protocols and pings.”**

## **Workstation Software**

- **Multi-User and Multitasking**
- **Full 32-bit Environment**
- **Virtual Memory**
- **Support of Common PC Hardware**
- **Software Development Tools**

## **Hardware**

- **20 MHz 386 CPU**
- **Math Coprocessor**
- **8 Mb RAM**
- **Two 540 Mb IDE Hard Drives**
- **NEC CDR-37 SCSI CD-ROM Drive**

## **A Brief History of Unix**

- **AT&T Bell Laboratories**
  - **Dennis Ritchie and Ken Thompson**
  - **Developed on DEC Minicomputers**
  - **Rewritten in C, 1973**
- **Source Code Released for Academic Use Through 1979**
- **AT&T Release of System V, 1983**
- **Specific Releases (SunOS, Xenix, ULTRIX, etc.)**
- **IEEE POSIX Standard**

## **Low Cost Unix for the PC**

- **Coherent**
- **Minix**
- **Berkeley System Distribution**
  - **NetBSD**
  - **FreeBSD**
  - **386BSD**
- **Linux**

## **A Brief History of Linux**

- **Developed by Linus Torvalds**
- **Enhanced by Cooperative Development Over the Internet**
- **Source Code Released in August 1991 as a Freely Distributable MINIX Clone**
- **Version 0.02 Released With GNU C Compiler in October 1991**
- **Version 0.12 Released in January 1992**
- **Asymptotic Approach to Version 1.0 During 1992 and 1993**
- **Version 1.0 Released Early in 1994**

## **Linux Software Resources**

- **Internet FTP Archives**

- **nic.funet.fi**
- **sunsite.unc.edu**
- **tsx-11.mit.edu**

- **CD-ROM Vendors**

- **InfoMagic**
- **Walnut Creek**
- **Prime Time Freeware**

# **Linux Reference Resources**

- **Online Documentation**
- **Newsgroups and Mailing Lists**
  - **comp.os.linux.\* Newsgroups**
  - **Developer Mailing Lists**
- **Magazines**
  - **General Computing Magazines**
  - **Linux Journal**
- **Books**
  - **Generic Unix References**
  - **Linux Specific References**

# **Text Processing**

- **Editors**
  - **GNU EMACS**
  - **VI, SED, and EX**
- **Text Formatters**
  - **GROFF**
  - **GhostScript**
  - **T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X**

## **Programming Languages**

- **GNU C/C++ With GDB Debugger**
- **AT&T f2c Translator for Fortran**
- **Translators for Modula-2, Pascal, and Simula**
- **LISP**
- **AWK**
- **PERL**

# **Numerical Software**

- **GNUPlot**
- **Xfig and Transfig**
- **Linux Ports**
  - **Spice**
  - **LAPACK**
  - **FTNCHEK**
  - **TOOLPACK**
- **NetLib Source Code**

## **Numerical Examples**

- **Class Examples for Process Control**
  - **Nyquist Plot Using Complex Arithmetic**
  - **Adiabatic CSTR Modeled by Non-Linear and Linear ODE's**
- **Prepared on VAX-8800**
- **Conversions for Linux**
  - **NetLib Math Libraries**
  - **Output Through GNUPlot**

## **NetLib Math Libraries**

- **Network Archive of Free DOE Software**
  - **netlib.org**
  - **netlib.att.com**
- **Available Documentation**
  - **Online Manuals With Distribution**
  - **Journal Articles**
  - **Reference Books**
- **Available on CD-ROM**

```

subroutine lsoda (f, neq, y, t, tout, itol, rtol, atol, itask,
1         istate, iopt, rwork, lrw, iwork, liw, jac, jt)
    external f, jac
    integer neq, itol, itask, istate, iopt, lrw, iwork, liw, jt
    double precision y, t, tout, rtol, atol, rwork
    dimension neq(1), y(1), rtol(1), atol(1), rwork(lrw), iwork(liw)
-----
c this is the march 30, 1987 version of
c lsoda.. livermore solver for ordinary differential equations, with
c         automatic method switching for stiff and nonstiff problems.
c
c this version is in double precision.
c
c lsoda solves the initial value problem for stiff or nonstiff
c systems of first order ode-s,
c     dy/dt = f(t,y) ,   or, in component form,
c     dy(i)/dt = f(i) = f(i,t,y(1),y(2),...,y(neq)) (i = 1,...,neq).
c
c this a variant version of the lsode package.
c it switches automatically between stiff and nonstiff methods.
c this means that the user does not have to determine whether the
c problem is stiff or not, and the solver will automatically choose the
c appropriate method.  it always starts with the nonstiff method.
c
c authors..
c
c         linda r. petzold  and  alan c. hindmarsh,
c         computing and mathematics research division, 1-316
c         lawrence livermore national laboratory
c         livermore, ca 94550.
c
c references..
c 1. alan c. hindmarsh, odepack, a systematized collection of ode
c     solvers, in scientific computing, r. s. stepleman et al. (eds.),
c     north-holland, amsterdam, 1983, pp. 55-64.
c 2. linda r. petzold, automatic selection of methods for solving
c     stiff and nonstiff systems of ordinary differential equations,
c     siam j. sci. stat. comput. 4 (1983), pp. 136-148.
-----
c summary of usage.
c
c communication between the user and the lsoda package, for normal
c situations, is summarized here.  this summary describes only a subset
c of the full set of options available.  see the full description for
c details, including alternative treatment of the jacobian matrix,
c optional inputs and outputs, nonstandard options, and
c instructions for special situations.  see also the example
c problem (with program and output) following this summary.
c
c a. first provide a subroutine of the form..
c
c         subroutine f (neq, t, y, ydot)
c         dimension y(neq), ydot(neq)
c which supplies the vector function f by loading ydot(i) with f(i).
c
c b. write a main program which calls subroutine lsoda once for
c each point at which answers are desired.  this should also provide
c for possible use of logical unit 6 for output of error messages
c by lsoda.  on the first call to lsoda, supply arguments as follows..
c f         = name of subroutine for right-hand side vector f.
c           this name must be declared external in calling program.
c neq        = number of first order ode-s.
c y          = array of initial values, of length neq.
c t          = the initial value of the independent variable.
c tout       = first point where output is desired (.ne. t).
c itol       = 1 or 2 according as atol (below) is a scalar or array.
c rtol       = relative tolerance parameter (scalar).
c atol       = absolute tolerance parameter (scalar or array).
c           the estimated local error in y(i) will be controlled so as
c           to be less than

```

Apr 3 1995 12:05

**lsoda.f.comment**

Page 2

```

c          ewt(i) = rtol*abs(y(i)) + atol      if itol = 1, or
c          ewt(i) = rtol*abs(y(i)) + atol(i)   if itol = 2.
c      thus the local error test passes if, in each component,
c      either the absolute error is less than atol (or atol(i)),
c      or the relative error is less than rtol.
c      use rtol = 0.0 for pure absolute error control, and
c      use atol = 0.0 (or atol(i) = 0.0) for pure relative error
c      control.  caution.. actual (global) errors may exceed these
c      local tolerances, so choose them conservatively.
c  itask  = 1 for normal computation of output values of y at t = tout.
c  istate = integer flag (input and output).  set istate = 1.
c  iopt   = 0 to indicate no optional inputs used.
c  rwork  = real work array of length at least..
c          22 + neq * max(16, neq + 9).
c          see also paragraph e below.
c  lrw    = declared length of rwork (in user-s dimension).
c  iwork  = integer work array of length at least 20 + neq.
c  liw    = declared length of iwork (in user-s dimension).
c  jac    = name of subroutine for jacobian matrix.
c          use a dummy name.  see also paragraph e below.
c  jt     = jacobian type indicator.  set jt = 2.
c          see also paragraph e below.
c  note that the main program must declare arrays y, rwork, iwork,
c  and possibly atol.
c
c c. the output from the first call (or any call) is..
c      y = array of computed values of y(t) vector.
c      t = corresponding value of independent variable (normally tout).
c  istate = 2 if lsoda was successful, negative otherwise.
c          -1 means excess work done on this call (perhaps wrong jt).
c          -2 means excess accuracy requested (tolerances too small).
c          -3 means illegal input detected (see printed message).
c          -4 means repeated error test failures (check all inputs).
c          -5 means repeated convergence failures (perhaps bad jacobian
c            supplied or wrong choice of jt or tolerances).
c          -6 means error weight became zero during problem. (solution
c            component i vanished, and atol or atol(i) = 0.)
c          -7 means work space insufficient to finish (see messages).
c
c d. to continue the integration after a successful return, simply
c  reset tout and call lsoda again.  no other parameters need be reset.
c
c e. note.. if and when lsoda regards the problem as stiff, and
c  switches methods accordingly, it must make use of the neq by neq
c  jacobian matrix, j = df/dy.  for the sake of simplicity, the
c  inputs to lsoda recommended in paragraph b above cause lsoda to
c  treat j as a full matrix, and to approximate it internally by
c  difference quotients.  alternatively, j can be treated as a band
c  matrix (with great potential reduction in the size of the rwork
c  array).  also, in either the full or banded case, the user can supply
c  j in closed form, with a routine whose name is passed as the jac
c  argument.  these alternatives are described in the paragraphs on
c  rwork, jac, and jt in the full description of the call sequence below.
c
c-----

```

# **GNUPlot**

- **Interactive Plotting Program**
- **Generalized Graphics Device Support**
- **Output Redirection**
- **Plots Any Number of Data Files or Functions**
- **User Defined Labels and Ranges**
- **Load and Save Capability**

Apr 19 1995 10:59

ol2.f

Page 1

```

c***      Frequency response of the open loop
c***
c***      This program determines the frequency response of a process
c***      from the evaluation of the transfer function G(i*omega) where
c***      G(s) is the transfer function of the process.  The specific
c***      case considered here is the transfer function:
c***
c***      
$$G(s) = \frac{24}{(5s+1)(s+1)(10s+1)}$$

c***
c***      complex s, g
c***      parameter (omega0 = 0.001)
c***      parameter (cvtrad=57.29577951)
c***      open ( 10, file="ol2.dat" )
c***      do 10 i = 1, 501
c***
c***          Determine frequency
c***
c***          omega = omega0 * 10.0**((i-1)/100.0)
c***          s = cmplx ( 0.0, omega )
c***
c***          Evaluate the transfer functions at s = omega*i.
c***
c***          g = 24.0/(5.0*s + 1.0)/(s + 1.0)/(10.0*s + 1.0)
c***
c***          Print the results
c***
c***          if ( mod(i,10) .eq. 1 ) write(6,1000) omega, g
c***          write(10,1010) g
c***          continue
c***      10 stop
c***
c***      1000 format ( ' s = ', f10.6, ' i          g(s) = ( ',
c***      $          f13.6, ', ', f13.6, ' i)' )
c***      1010 format ( 2f13.6 )
c***      end

```

Jul 20 1998 23:33

ol2.scr

Page 1

s = 0.001000 i	g(s) = ( 23.995417, -0.383950 i)
s = 0.001259 i	g(s) = ( 23.992735, -0.483328 i)
s = 0.001585 i	g(s) = ( 23.988489, -0.608402 i)
s = 0.001995 i	g(s) = ( 23.981758, -0.765787 i)
s = 0.002512 i	g(s) = ( 23.971096, -0.963779 i)
s = 0.003162 i	g(s) = ( 23.954210, -1.212748 i)
s = 0.003981 i	g(s) = ( 23.927477, -1.525608 i)
s = 0.005012 i	g(s) = ( 23.885180, -1.918333 i)
s = 0.006310 i	g(s) = ( 23.818319, -2.410474 i)
s = 0.007943 i	g(s) = ( 23.712805, -3.025534 i)
s = 0.010000 i	g(s) = ( 23.546694, -3.790935 i)
s = 0.012589 i	g(s) = ( 23.286203, -4.736971 i)
s = 0.015849 i	g(s) = ( 22.880201, -5.893703 i)
s = 0.019953 i	g(s) = ( 22.253426, -7.283875 i)
s = 0.025119 i	g(s) = ( 21.300066, -8.909036 i)
s = 0.031623 i	g(s) = ( 19.882557, -10.725594 i)
s = 0.039811 i	g(s) = ( 17.845943, -12.610129 i)
s = 0.050119 i	g(s) = ( 15.063728, -14.323396 i)
s = 0.063096 i	g(s) = ( 11.526371, -15.503242 i)
s = 0.079433 i	g(s) = ( 7.450546, -15.736115 i)
s = 0.100000 i	g(s) = ( 3.326733, -14.732674 i)
s = 0.125893 i	g(s) = ( -0.197566, -12.532681 i)
s = 0.158489 i	g(s) = ( -2.590990, -9.569002 i)
s = 0.199526 i	g(s) = ( -3.687232, -6.491640 i)
s = 0.251189 i	g(s) = ( -3.716200, -3.866405 i)
s = 0.316228 i	g(s) = ( -3.116883, -1.971290 i)
s = 0.398107 i	g(s) = ( -2.304819, -0.796601 i)
s = 0.501187 i	g(s) = ( -1.546266, -0.173960 i)
s = 0.630957 i	g(s) = ( -0.955374, 0.094632 i)
s = 0.794328 i	g(s) = ( -0.547643, 0.169035 i)
s = 1.000000 i	g(s) = ( -0.292460, 0.155369 i)
s = 1.258925 i	g(s) = ( -0.146201, 0.114103 i)
s = 1.584893 i	g(s) = ( -0.068927, 0.073779 i)
s = 1.995262 i	g(s) = ( -0.030956, 0.043863 i)
s = 2.511887 i	g(s) = ( -0.013391, 0.024621 i)
s = 3.162278 i	g(s) = ( -0.005636, 0.013291 i)
s = 3.981072 i	g(s) = ( -0.002327, 0.006989 i)
s = 5.011873 i	g(s) = ( -0.000949, 0.003613 i)
s = 6.309574 i	g(s) = ( -0.000383, 0.001847 i)
s = 7.943283 i	g(s) = ( -0.000154, 0.000937 i)
s = 10.000000 i	g(s) = ( -0.000062, 0.000473 i)
s = 12.589254 i	g(s) = ( -0.000025, 0.000239 i)
s = 15.848932 i	g(s) = ( -0.000010, 0.000120 i)
s = 19.952623 i	g(s) = ( -0.000004, 0.000060 i)
s = 25.118866 i	g(s) = ( -0.000002, 0.000030 i)
s = 31.622778 i	g(s) = ( -0.000001, 0.000015 i)
s = 39.810722 i	g(s) = ( 0.000000, 0.000008 i)
s = 50.118725 i	g(s) = ( 0.000000, 0.000004 i)
s = 63.095737 i	g(s) = ( 0.000000, 0.000002 i)
s = 79.432823 i	g(s) = ( 0.000000, 0.000001 i)
s = 100.000008 i	g(s) = ( 0.000000, 0.000000 i)

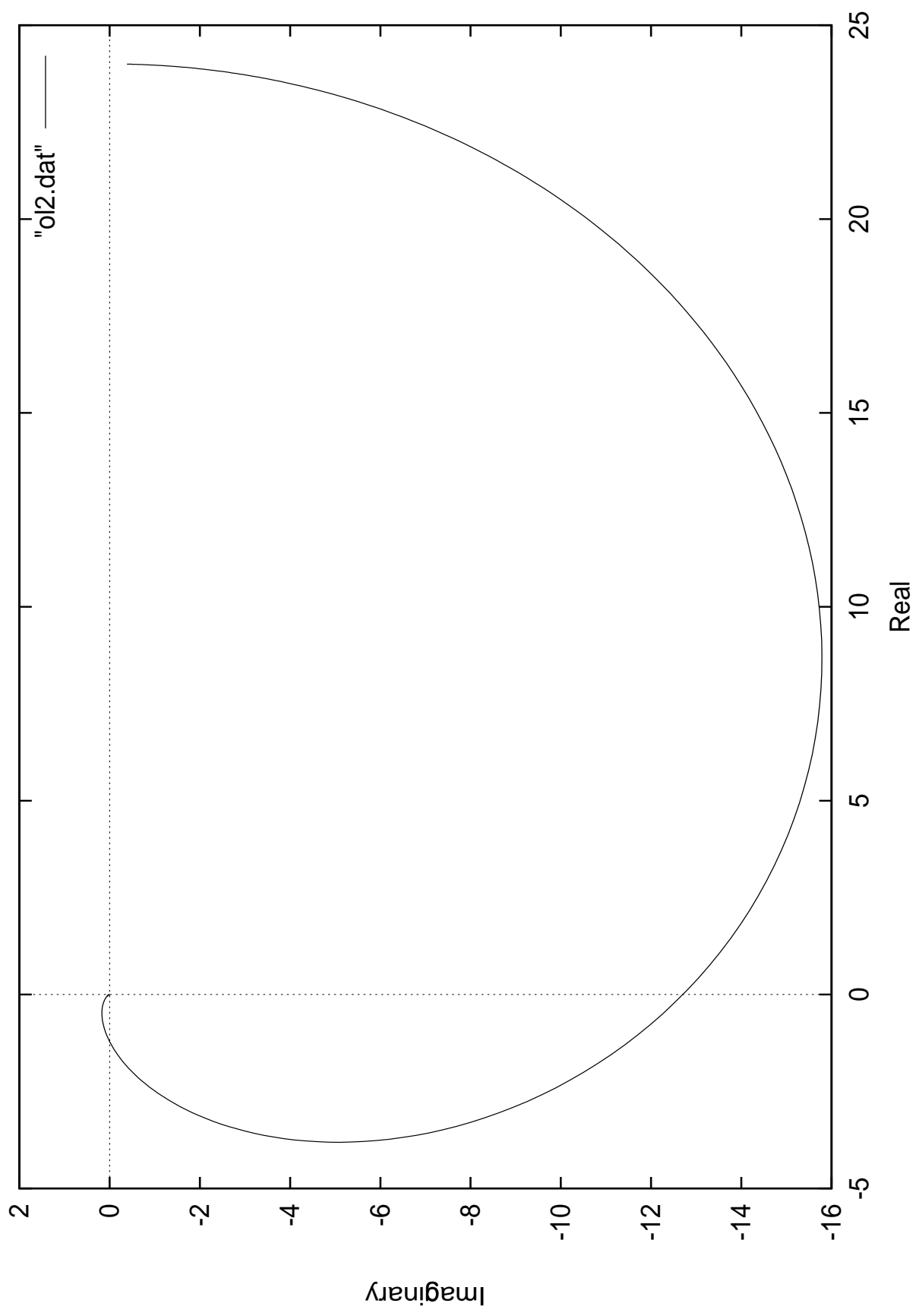
Apr 12 1995 09:18

ol2.gnuplot

Page 1

```
set terminal postscript
set output "ol2.ps"
set noclip points
set clip one
set noclip two
set border
set boxwidth
set dummy x,y
set format x "%g"
set format y "%g"
set format z "%g"
set nogrid
set key
set nolabel
set noarrow
set nologscale
set offsets 0, 0, 0, 0
set nopolar
set angles radians
set noparametric
set view 60, 30, 1, 1
set samples 100, 100
set isosamples 10, 10
set surface
set nocontour
set clabel
set nohidden3d
set cntrparam order 4
set cntrparam linear
set cntrparam levels auto 5
set cntrparam points 5
set size 1,1
set data style points
set function style lines
set xzeroaxis
set yzeroaxis
set tics in
set ticslevel 0.5
set xtics
set ytics
set ztics
set title "Nyquist Plot" 0,0
set notime
set rrange [-0 : 10]
set trange [-5 : 5]
set urange [-5 : 5]
set vrange [-5 : 5]
set xlabel "Real" 0,0
set xrange [-3.80958 : 23.9954]
set ylabel "Imaginary" 0,0
set yrange [-15.7877 : 0.170839]
set zlabel "" 0,0
set zrange [-10 : 10]
set autoscale r
set autoscale t
set autoscale xy
set autoscale z
set zero 1e-08
plot "ol2.dat" with line 1
```

Nyquist Plot



**HOMEWORK # 6**

Use the IMSL subroutine IVPRK to numerically model the adiabatic reactor (section 2.4.3.1 in the text by Gilbert) for the conditions listed below. Consider the cases below for the period from zero to fifteen hours:

1. The linear model (eq. 2.49 and 2.50) with a step input of  $\hat{C}_{A0} = 0.1$ .
2. The linear model (eq. 2.49 and 2.50) with a step input of  $\hat{C}_{A0} = -0.1$ .
3. The non-linear model (eq. 2.43 and 2.44) with a step input of  $\hat{C}_{A0} = 0.1$ .
4. The non-linear model (eq. 2.43 and 2.44) with a step input of  $\hat{C}_{A0} = -0.1$ .

The conditions for the reactor are as follows:

$$\begin{aligned}q &= 40.0 \text{ ft}^3/\text{hr} \\V &= 48.0 \text{ ft}^3 \\C_{A0} &= 0.50 \text{ lb} - \text{mol A}/\text{ft}^3 \\T^{ss} &= 600.0 \text{ }^\circ\text{R} \\C_A^{ss} &= 0.245 \text{ lb} - \text{mol A}/\text{ft}^3 \\\rho &= 50.0 \text{ lb}_m/\text{ft}^3 \\C_p &= 0.75 \text{ BTU}/\text{lb}_m^\circ\text{R} \\k_0 &= 7.08 \times 10^{10} \text{ hr}^{-1} \\E &= 30,000 \text{ BTU}/\text{lb} - \text{mol} \\\Delta H &= -5,000 \text{ BTU}/\text{lb} - \text{mol}\end{aligned}$$

## The Adiabatic Reactor

The adiabatic reactor is described by both mass and energy balances. The Arrhenius expression for the temperature dependence of the rate constant introduces a significant non-linear behavior that couples the differential equations describing the outlet concentration and temperature. In terms of the original engineering variables, the system of equations describing the tank take the form:

$$\frac{dC}{dt} = \frac{q}{V}C_i(t) - \frac{q}{V}C - r \quad (1)$$

$$\frac{dT}{dt} = \frac{q}{V}T_i(t) - \frac{q}{V}T - \frac{r\Delta H}{\rho c_p} \quad (2)$$

The Arrhenius kinetic expression gives

$$r = Ck_0 \exp\left(\frac{-E}{RT}\right) \quad (3)$$

Equations (1), (2), and (3) describe the non-linear model in terms of the original engineering variables.

These equations can be rewritten in terms of deviation variables to give:

$$\frac{d\hat{C}}{dt} = \frac{q}{V}\hat{C}_i(t) - \frac{q}{V}\hat{C} - \hat{r} \quad (4)$$

$$\frac{d\hat{T}}{dt} = \frac{q}{V}\hat{T}_i(t) - \frac{q}{V}\hat{T} - \frac{\hat{r}\Delta H}{\rho c_p} \quad (5)$$

The Arrhenius kinetic expression gives

$$\hat{r} = Ck_0 \exp\left(\frac{-E}{RT}\right) - C^{ss}k_0 \exp\left(\frac{-E}{RT^{ss}}\right) \quad (6)$$

Note that equations (4), (5), and (6) still describe the non-linear dynamics of the system since  $\hat{r}$  is not linear.

The Taylor series expansion gives the linear approximation for the reaction rate as

$$\hat{r} = k^{ss}\hat{C} + \alpha\hat{T} \quad (7)$$

where

$$k^{ss} = k_0 \exp\left(\frac{-E}{RT^{ss}}\right)$$
$$\alpha = \frac{k^{ss}E C^{ss}}{R(T^{ss})^2}$$

Substituting equation (7) into equations (4) and (5) gives the linear model for the reactor

$$\frac{d\hat{C}}{dt} = \frac{q}{V}\hat{C}_i(t) - \left(\frac{q}{V} + k^{ss}\right)\hat{C} - \alpha\hat{T} \quad (8)$$

$$\frac{d\hat{T}}{dt} = \frac{q}{V}\hat{T}_i(t) - \left( \frac{q}{V} + \frac{\alpha\Delta H}{\rho c_p} \right) \hat{T} - \frac{k^{ss}\Delta H}{\rho c_p}\hat{C} \quad (9)$$

Note that equations (8) and (9) can be rearranged into the standard linear forms

$$\tau_1 \frac{d\hat{C}}{dt} + \hat{C} = K_1 \hat{C}_i(t) - K_2 \hat{T} \quad (10)$$

$$\tau_2 \frac{d\hat{T}}{dt} + \hat{T} = K_3 \hat{T}_i(t) - K_4 \hat{C} \quad (11)$$

where the constants are defined by

$$\tau_1 = \frac{V}{q + k^{ss}V}$$

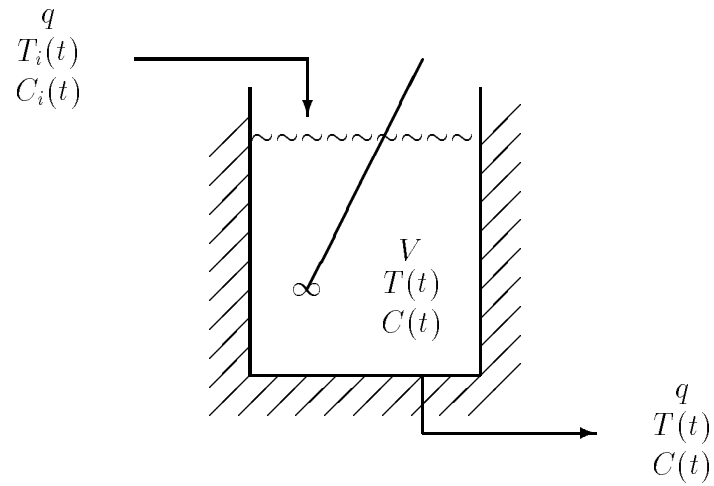
$$\tau_2 = \frac{V\rho c_p}{q\rho c_p + \alpha V\Delta H}$$

$$K_1 = \frac{q}{q + k^{ss}V}$$

$$K_2 = \frac{\alpha V}{q + k^{ss}V}$$

$$K_3 = \frac{q\rho c_p}{q\rho c_p + \alpha V\Delta H}$$

$$K_4 = \frac{k^{ss}V\Delta H}{q\rho c_p + \alpha V\Delta H}$$



Apr 4 1995 14:27

adiabati.f

Page 1

```

c***      Chem E 462 Homework Problem Number 6
c***
c***      Dynamic model of an adiabatic stirred tank reactor
c***      undergoing an exothermic reaction.  This program
c***      integrates both the non-linear and the linear models
c***      simultaneously.
c***
c***      Variables used:
c***      y(1)  -- Integrated concentration for linear model
c***      y(2)  -- Integrated temperature for linear model
c***      y(3)  -- Integrated concentration for non-linear model
c***      y(4)  -- Integrated temperature for non-linear model
c***
c***      Common blocks:
c***      /lin/      -- Parameters for the linear model
c***      tau1      -- Time constant for concentration
c***      tau2      -- Time constant for temperature
c***      ak1       -- Gain for temperature in concentration
c***                equation
c***      ak2       -- Gain for inlet concentration in the
c***                concentration equation
c***      ak3       -- Gain for concentration in temperature
c***                equation
c***      /nonlin/   -- Parameters for non-linear model
c***      r         -- Ideal gas constant, BTU/(lb-mol deg-R)
c***      q         -- Flow rate, ft^3/hr
c***      v         -- Vessel volume, ft^3
c***      cp        -- Specific heat, BTU/(lb-m deg-R)
c***      rho       -- Density, lb-m/ft^3
c***      tss       -- Initial steady state temperature, deg-R
c***      css       -- Initial steady state concentration,
c***                lb-mol/ft^3
c***      e         -- Activation energy, BTU/lb-mol
c***      deltah    -- Heat of reaction, BTU/lb-mol
c***      ak0       -- Pre-exponential term in Arrhenius rate
c***                expression, hr^(-1)
c***      akss      -- Initial steady state rate coefficient,
c***                hr^(-1)
c***      /step/    -- Parameter for concentration inlet function
c***      cstep     -- Step input for the inlet concentration
c***
c***      implicit double precision (a-h,o-z)
c***      parameter (neq=4)
c***      parameter (nplot=1500)
c***      parameter (dt=0.01)
c***      parameter (lrw=100)
c***      parameter (liw=30)
c***      dimension y(neq), rwork(lrw), iwork(liw)
c***      common /lin/  tau1, tau2, ak1, ak2, ak3
c***      common /nonlin/ r, q, v, cp, rho, tss, css, e, deltah,
c***      $              ak0, akss
c***      common /step/ cstep
c***      external tank, dummy
c***      open ( 10, file="adiabatic.out" )
c***
c***      Set control parameters
c***
c***      istate = 1
c***      itask = 1
c***      iopt = 0
c***      jt = 2
c***      tol = 1.0e-8
c***
c***      Set the parameters for this case
c***
c***      r = 1.987
c***      q = 40.0
c***      v = 48.0

```

Apr 4 1995 14:27

adiabati.f

Page 2

```

      cp = 0.75
      rho = 50.0
      tss = 600.0
      css = 0.245
      e = 30000.0
      deltah = -5000.0
      ak0 = 7.08e10
      akss = ak0*exp(-e/r/tss)
      alpha = akss*e*css/r/tss/tss
      tau1 = v/(q + akss*v)
      tau2 = rho*cp*v/(rho*cp*q + alpha*deltah*v)
      ak1 = alpha*v/(q + akss*v)
      ak2 = q/(q + akss*v)
      ak3 = akss*deltah*v/(rho*cp*q + alpha*deltah*v)
      t = 0.0
      cstep = -0.3
c***
c***      Initial conditions and numerical error tolerance
c***
      y(1) = 0.0
      y(2) = 0.0
      y(3) = 0.0
      y(4) = 0.0
      write(6,1000) cstep
c***
c***      Loop through printing the concentration every 0.25 hour
c***
      do 20 i = 1, nplot
         tend = real(i) * dt
         call lsoda ( tank, neq, y, t, tend, 1, tol, tol,
           $          itask, istate, iopt, rwork, lrw,
           $          iwork, liw, dummy, jt )
         if ( istate .le. 0 ) then
            write(6,1010) istate
            stop
         endif
c***
c***      Convert deviation variables to process variables for output.
c***
         time = tend
         concl = y(1) + css
         templ = y(2) + tss
         concnl = y(3) + css
         tempnl = y(4) + tss
         write(10,1020) time, concl, templ, concnl, tempnl
         if ( mod(i,25) .eq. 0 )
           $       write(6,1020) time, concl, templ, concnl, tempnl
      20    continue
c***
      stop
c***
1000    format ( '1Input step:', f10.2, '//, 16x,
           $      '---- Linear ----', 8x, '-- Non-Linear --', /,
           $      4x, 'Time', 2(8x, 'Conc', 8x, 'Temp') )
1010    format ( 1x, 'Error return from LSODA, ISTATE=', i5 )
1020    format ( 1x, f7.2, 1p, 4e12.3 )
      end
c=====
      subroutine tank ( neq, t, y, dydt )
      implicit double precision (a-h,o-z)
      dimension y(neq), dydt(neq)
      common /lin/ tau1, tau2, ak1, ak2, ak3
      common /nonlin/ r, q, v, cp, rho, tss, css, e, deltah,
           $          ak0, akss
c***
c***      Evaluate derivatives for linear model
c***
      dydt(1) = (ak2*cin(t) - y(1) - ak1*y(2)) / tau1

```

Apr 4 1995 14:27

adiabati.f

Page 3

```

dydt(2) = -(ak3*y(1) + y(2)) / tau2
c***
c***      Evaluate derivatives for non-linear model.  First convert
c***      deviation variables into process variables to evaluate the
c***      reaction rate.
c***
      temp = tss + y(4)
      ca = css + y(3)
      ak = ak0*exp(-e/r/temp)
      rhat = ak*ca - akss*css
      dydt(3) = q/v*(cin(t) - y(3)) - rhat
      dydt(4) = -q/v*y(4) - rhat*deltah/rho/cp
      return
      end
c=====
      subroutine dummy ( neq, t, y, ml, mu, pd, nrowpd )
      implicit double precision (a-h,o-z)
      return
      end
c=====
      function cin ( time )
c***
c***      Step in concentration input at time 0.50
c***
      implicit double precision (a-h,o-z)
      common /step/ cstep
      if ( time .ge. 0.50 ) then
         cin = cstep
      else
         cin = 0.0
      endif
      return
      end

```

Jul 21 1998 14:27

adiabatic.scr

Page 1

1Input step: -0.30

Time	---- Linear ----		-- Non-Linear --	
	Conc	Temp	Conc	Temp
0.25	2.450E-01	6.000E+02	2.450E-01	6.000E+02
0.50	2.450E-01	6.000E+02	2.450E-01	6.000E+02
0.75	1.944E-01	5.992E+02	1.943E-01	5.992E+02
1.00	1.637E-01	5.972E+02	1.625E-01	5.974E+02
1.25	1.477E-01	5.944E+02	1.434E-01	5.950E+02
1.50	1.428E-01	5.910E+02	1.325E-01	5.924E+02
1.75	1.457E-01	5.874E+02	1.268E-01	5.899E+02
2.00	1.541E-01	5.836E+02	1.242E-01	5.876E+02
2.25	1.664E-01	5.798E+02	1.236E-01	5.855E+02
2.50	1.810E-01	5.761E+02	1.242E-01	5.837E+02
2.75	1.970E-01	5.725E+02	1.254E-01	5.821E+02
3.00	2.137E-01	5.692E+02	1.271E-01	5.807E+02
3.25	2.303E-01	5.660E+02	1.288E-01	5.795E+02
3.50	2.466E-01	5.631E+02	1.306E-01	5.785E+02
3.75	2.623E-01	5.604E+02	1.323E-01	5.777E+02
4.00	2.772E-01	5.579E+02	1.339E-01	5.770E+02
4.25	2.911E-01	5.556E+02	1.354E-01	5.764E+02
4.50	3.041E-01	5.535E+02	1.367E-01	5.759E+02
4.75	3.161E-01	5.517E+02	1.379E-01	5.754E+02
5.00	3.271E-01	5.500E+02	1.390E-01	5.751E+02
5.25	3.371E-01	5.485E+02	1.399E-01	5.748E+02
5.50	3.462E-01	5.471E+02	1.407E-01	5.745E+02
5.75	3.545E-01	5.459E+02	1.414E-01	5.743E+02
6.00	3.620E-01	5.448E+02	1.420E-01	5.741E+02
6.25	3.687E-01	5.438E+02	1.425E-01	5.740E+02
6.50	3.747E-01	5.430E+02	1.429E-01	5.739E+02
6.75	3.801E-01	5.422E+02	1.433E-01	5.738E+02
7.00	3.849E-01	5.415E+02	1.436E-01	5.737E+02
7.25	3.892E-01	5.409E+02	1.439E-01	5.736E+02
7.50	3.931E-01	5.404E+02	1.442E-01	5.736E+02
7.75	3.965E-01	5.399E+02	1.443E-01	5.735E+02
8.00	3.995E-01	5.395E+02	1.445E-01	5.735E+02
8.25	4.022E-01	5.391E+02	1.447E-01	5.734E+02
8.50	4.046E-01	5.388E+02	1.448E-01	5.734E+02
8.75	4.067E-01	5.385E+02	1.449E-01	5.734E+02
9.00	4.086E-01	5.382E+02	1.450E-01	5.734E+02
9.25	4.103E-01	5.380E+02	1.450E-01	5.734E+02
9.50	4.118E-01	5.378E+02	1.451E-01	5.733E+02
9.75	4.130E-01	5.376E+02	1.451E-01	5.733E+02
10.00	4.142E-01	5.375E+02	1.452E-01	5.733E+02
10.25	4.152E-01	5.373E+02	1.452E-01	5.733E+02
10.50	4.161E-01	5.372E+02	1.452E-01	5.733E+02
10.75	4.169E-01	5.371E+02	1.453E-01	5.733E+02
11.00	4.176E-01	5.370E+02	1.453E-01	5.733E+02
11.25	4.182E-01	5.369E+02	1.453E-01	5.733E+02
11.50	4.187E-01	5.368E+02	1.453E-01	5.733E+02
11.75	4.192E-01	5.368E+02	1.453E-01	5.733E+02
12.00	4.196E-01	5.367E+02	1.453E-01	5.733E+02
12.25	4.200E-01	5.367E+02	1.453E-01	5.733E+02
12.50	4.203E-01	5.366E+02	1.453E-01	5.733E+02
12.75	4.206E-01	5.366E+02	1.454E-01	5.733E+02
13.00	4.208E-01	5.366E+02	1.454E-01	5.733E+02
13.25	4.210E-01	5.365E+02	1.454E-01	5.733E+02
13.50	4.212E-01	5.365E+02	1.454E-01	5.733E+02
13.75	4.214E-01	5.365E+02	1.454E-01	5.733E+02
14.00	4.216E-01	5.365E+02	1.454E-01	5.733E+02
14.25	4.217E-01	5.364E+02	1.454E-01	5.733E+02
14.50	4.218E-01	5.364E+02	1.454E-01	5.733E+02
14.75	4.219E-01	5.364E+02	1.454E-01	5.733E+02
15.00	4.220E-01	5.364E+02	1.454E-01	5.733E+02

Apr 4 1995 21:02

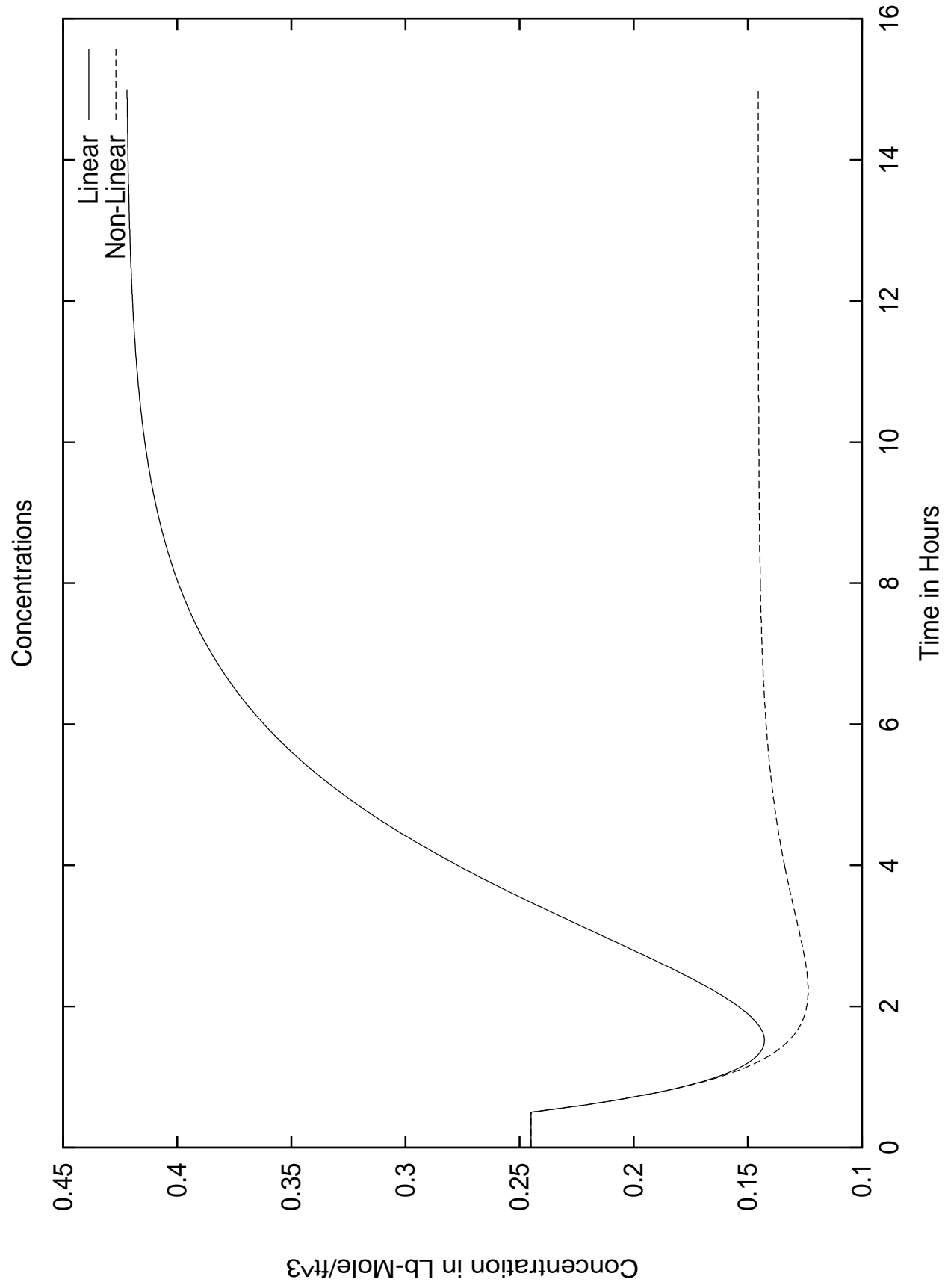
**concentr.gnuplot**

Page 1

```

set terminal postscript landscape monochrome dashed "Helvetica" 14
set output 'concentr.ps'
set noclip points
set clip one
set noclip two
set border
set boxwidth
set dummy x,y
set format x "%g"
set format y "%g"
set format z "%g"
set nogrid
set key
set nolabel
set noarrow
set nologscale
set offsets 0, 0, 0, 0
set nopolar
set angles radians
set noparametric
set view 60, 30, 1, 1
set samples 100, 100
set isosamples 10, 10
set surface
set nocontour
set clabel
set nohidden3d
set cntrparam order 4
set cntrparam linear
set cntrparam levels auto 5
set cntrparam points 5
set size 1,1
set data style points
set function style lines
set xzeroaxis
set yzeroaxis
set tics in
set ticslevel 0.5
set xtics
set ytics
set ztics
set title "Concentrations" 0,0
set notime
set rrange [-0 : 10]
set trange [-5 : 5]
set urange [-5 : 5]
set vrange [-5 : 5]
set xlabel "Time in Hours" 0,0
set xrange [0.01 : 15]
set ylabel "Concentration in Lb-Mole/ft^3" 0,0
set yrange [0.1236 : 0.422]
set zlabel "" 0,0
set zrange [-10 : 10]
set autoscale r
set autoscale t
set autoscale xy
set autoscale z
set zero 1e-08
plot "adiabatic.out" using 1:2 title "Linear" with l 1, "adiabatic.out" using 1:4 ti
tle "Non-Linear" with l 2

```



Apr 4 1995 21:06

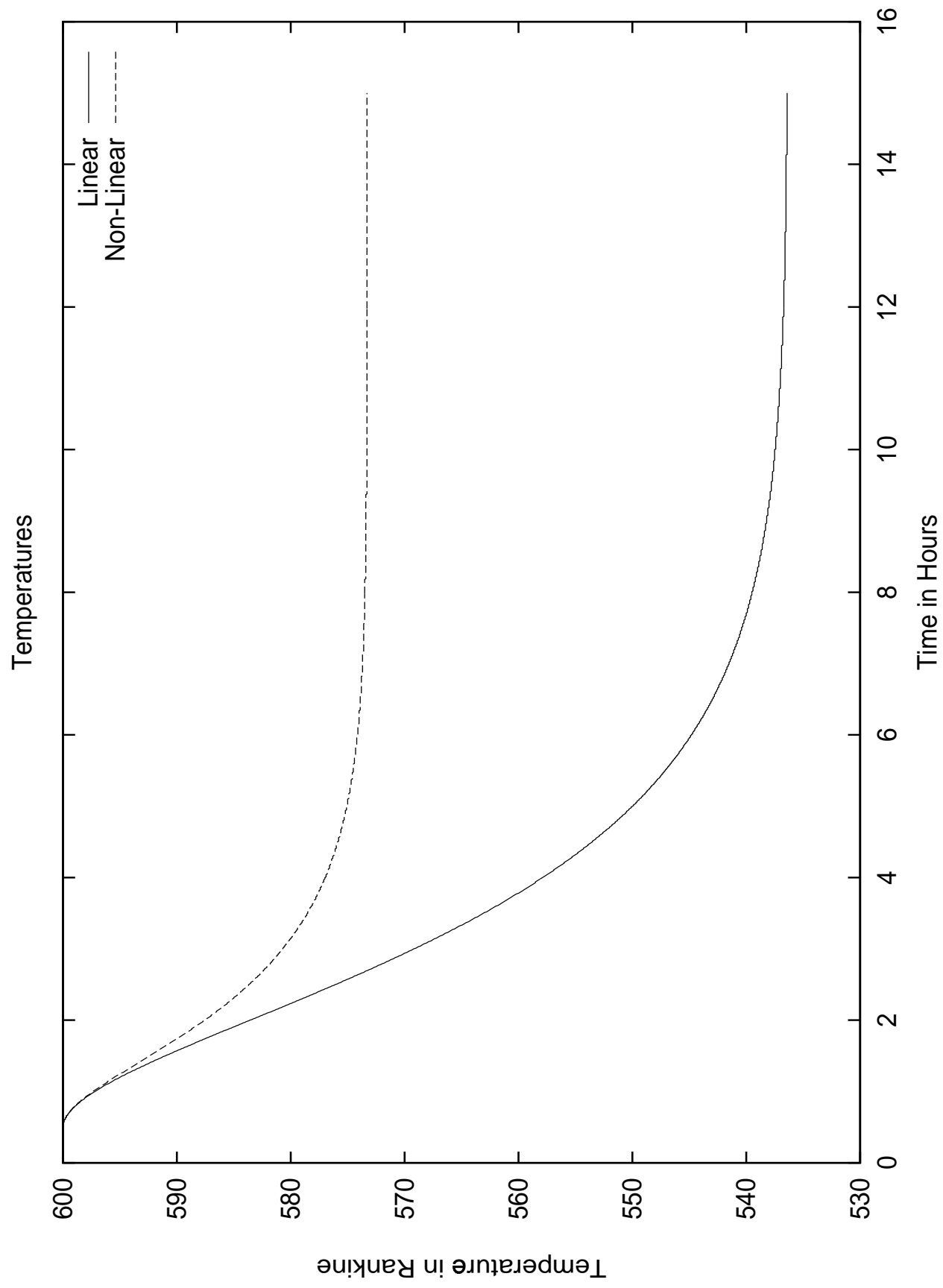
temps.gnuplot

Page 1

```

set terminal postscript landscape monochrome dashed "Helvetica" 14
set output 'temps.ps'
set noclip points
set clip one
set noclip two
set border
set boxwidth
set dummy x,y
set format x "%g"
set format y "%g"
set format z "%g"
set nogrid
set key
set nolabel
set noarrow
set nologscale
set offsets 0, 0, 0, 0
set nopolar
set angles radians
set noparametric
set view 60, 30, 1, 1
set samples 100, 100
set isosamples 10, 10
set surface
set nocontour
set clabel
set nohidden3d
set cntrparam order 4
set cntrparam linear
set cntrparam levels auto 5
set cntrparam points 5
set size 1,1
set data style points
set function style lines
set xzeroaxis
set yzeroaxis
set tics in
set ticslevel 0.5
set xtics
set ytics
set ztics
set title "Temperatures" 0,0
set notime
set rrange [-0 : 10]
set trange [-5 : 5]
set urange [-5 : 5]
set vrange [-5 : 5]
set xlabel "Time in Hours" 0,0
set xrange [0.01 : 15]
set ylabel "Temperature in Rankine" 0,0
set yrange [536.4 : 600]
set zlabel "" 0,0
set zrange [-10 : 10]
set autoscale r
set autoscale t
set autoscale xy
set autoscale z
set zero 1e-08
plot "adiabatic.out" using 1:3 title "Linear" with l 1, "adiabatic.out" using 1:5 ti
tle "Non-Linear" with l 2

```



TANSTAAFL

## **Pitfalls**

- **Setup and System Administration**
  - **Scattered Documentation**
  - **Un\*x  $\neq$  Un\*x**
- **Continuous Updates**
- **Limited DOS Compatibility**
- **Lack of Commercial Applications**
- **FORTTRAN is a Translator**

## **Conclusions**

- **“Free” Unix for the PC Works for Numerical Modeling**
  - **FORTRAN Codes Need Little Modification**
  - **Unix Tools Aid in Analyzing Results**
- **Workstation Performance Has a Price**
  - **Administration of a Multi-User System**
  - **Compatibility With DOS Applications**

## References

- **Stefan Strobel and Thomas Uhl.** *Linux: Unleashing the Workstation in Your PC.* **Springer-Verlag, 1994.**
- **Matt Welsh and Lar Kaufman.** *Running Linux.* **O'Reilly & Associates, Inc., 1995.**
- **Olaf Kirch.** *Linux Network Administrators Guide.* **O'Reilly & Associates, Inc., 1995.**