Mathematical Modelling and Optimal Multivariable Control of Chemical Processes

A Thesis submitted for the degree of

Doctor of Philosphy

by

Mustafa Omer Nawari
Bsc. (Eng), Msc., M.I.E.E.E, A.M.B.C.S

Department of Chemical Engineering

University of Aston-in-Birmingham

1984

VOLUME (2)

VOLUME (1)

CHAPTER (1)

INTRODUCTION

CHAPTER (2)

LITERATURE REVIEW

CHAPTER 3

EXPERIMENTAL SET-UP

CHAPTER (6)

MULTIVARIABLE CLOSED-LOOP POLES ALLOCATION AND MINIMAL SENSITIVITY : A Synthesis Approach

CHAPTER (7)

COMPUTER-AIDED CONTROL SYSTEM DESIGN and SYNTHESIS PACKAGE

CHAPTER (10)

CONCLUSIONS AND FUTURE WORK

VOLUME (2)

APPENDIX A

APPENDIX B

APPENDIX C

APPENDIX D

APPENDIX E

APPENDIX F

APPENDIX (A)

MATHEMATICAL MODELLING OF THE DOUBLE EFFECT EVAPORATOR

A.1 INTRODUCTION

In the development of the mathematical model of the climbing film evaporator first a general development under which special cases of equipment can be treated will be considered. The generality is to cover the evaporation process where three distinct regimes can be addressed: liquid-phase only, into which heat exchangers can fall; liquid-vapour phase when vapours start to develop, and vapour-phase only, where all the liquid has evaporated and the vapour is getting superheated.

Now, if the middle region of the two phase is taken, then it will be easy to find special cases of all liquid or all vapour. The following general assumptions will be considered.

The temperature on the external side of the wall is constant. This kind of behaviour can be observed when a pure vapour is condensing on the outside of the tubes, when the shell fluid is perfectly mixed as in a continuous stirred tank reactor, or when the flow rate on one side of the exchanger is so high that for practical purposes its

temperature can be considered as being independent of the distance from the inlet, and this is almost the case in the double effect evaporator. This point also can be considered during designing very long tube/shell exchanger to allow for many distributed inlets to generate uniform temperature distribution. Indeed, the economical cost should be measured against the factors of safety, ease of operation, control, etc.

Fluid motion inside the tubes is plug flow, no axial or radial conduction in the fluid, no resistance of the wall to the heat transfer, and no significant accumulation of energy in the condensate film. Neglecting the radial temperature variation in the tube wall is reasonable for a thin wall made of a good thermal conductor, while the assumption of rectangular fluid velocity and temperature profiles should be quite accurate for turbulent flow conditions. We also assume an incompressible fluid and neglect axial heat conduction in both the wall and the fluid. Axial conduction in the tube wall should be small since the wall is thin (giving a high thermal resistance) and also since there is no intentional heat sink provided to drain off axial heat flow. In the fluid, axial conduction is relatively small (diffusion term) because of the poor conduction of most fluids and the existence of the "low resistance" heat flow path offered by the fluid transport "convection term".

Now consider the section of the tube, where liquid and vapour phases coexist, figure (A.1).

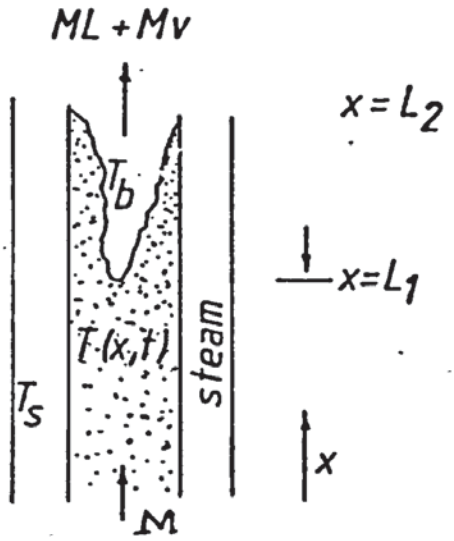


Figure (A.1.) A Section of the Two Regions Flow

The accumulated amount of heat inside an infinitestimal volume of length Δx,

$$dQ = \rho_v A_v L_v dx + \rho_L C_p T\, A_L\, dx + \rho_v C_{pv}\, T\, A_v\, dx$$

$$= \begin{matrix}\text{(latent heat of)} \\ \text{(evaporation)}\end{matrix} + \begin{matrix}\text{liquid-phase} \\ \text{heat}\end{matrix} + \begin{matrix}\text{vapour-phase} \\ \text{heat}\end{matrix}$$

where

$A_L\, dx$ = liquid volume

$A_v\, dx$ = vapour volume

note A_L and A_v are not constant, but functions of x.

The rate of change of Q per unit time Δt is given by

$$\frac{dQ}{dt} = \frac{\partial Q}{\partial x} \cdot \frac{dx}{dt} + \frac{\partial Q}{\partial t}$$

$$\frac{dQ}{dt} = \frac{C_p \partial \rho_L A_L T}{dx} + \frac{C_{pv} \partial \rho_u A_v T}{dx} + \frac{L_v \partial \rho A_v}{dx} \frac{dx}{dt}$$

$$+ \frac{C_p \partial \rho_L A_L T}{\partial t} + \frac{C_{pv} \partial \rho A_v T}{\partial t} + \frac{L_v \partial \rho_{Av}}{\partial t} \qquad (1.A)$$

This change of internal energy is due to the energy transferred from the neighbouring system. In this case it is the heat supplied by the steam on the shell side.

The heat transferred = $2\pi r\, U(T_s - T)\, dx$
where r is the tube radius, and T_s is the steam and external wall of the tube temperature.

Equation (1.A) now, is very general. Imposing the special conditions of each regime of the tube the solution can be found as follows:

A.2 Liquid Phase

The solution hold-up part can be modelled as

$L_v = 0$, there is no boiling.

ρ_L, A_L are constant and not functions either of time

$$\text{or space } \frac{\partial \rho_L A_L}{\partial t} = 0, \quad \frac{\partial \rho_L A_L}{\partial x} = 0$$

ρ_v, A_v are equal to zero, since there is no vapour there.

The heat transfer coefficient = U_L.

Imposing the above conditions in equation (1.A)

$$\frac{dQ}{dt} = C_p \rho_L A_L \frac{\partial T}{\partial x} \cdot \frac{dx}{dt} + C_p \rho A_L \frac{\partial T}{\partial t} = 2\pi r U_L (T_s - T) \quad (2.A)$$

$\frac{dx}{dt} = v(t)$ the liquid velocity which is, generally, in the form of equation (1.A), depends on the main mass flow-rate, liquid and vapour densities, and vapour mass rate. In equation (2.A), if T is considered to be composed of two components; steady state and varying component

$$T(x,t) = \bar{T}(x) + \tilde{T}(x,t)$$

where $\bar{T}(x)$ is the steady state component and hence independent of time. $T(x,t)$ is the time varying component and is a function of both the space and the time. Also, we take $v(t)$ to be composed of a steady-state component and a time varying one.

$$v(t) = \bar{v} + \tilde{v}(t)$$

where $\bar{v}$ is the constant in the range $0 \quad x \quad L_1$.

Rewriting equation (2.A), after inserting the above decompostion of T and v.

$$A\frac{\partial \tilde{T}}{\partial t} + B\bar{v}\frac{\partial \bar{T}}{\partial x} + B\tilde{v}\frac{\partial \tilde{T}}{\partial x} + B\bar{v}\frac{\partial \tilde{T}}{\partial x} + B\tilde{v}\frac{\partial \bar{T}}{\partial x} = C(\bar{T}_s - \bar{T}) + C(\tilde{T}_s - \tilde{T}) \quad (3.A)$$

where $A = C_p \rho_L A_L$, $B = A$, $C = 2\pi r U_L$

From Equation (3.A)

$$B\bar{v}\frac{\partial \bar{T}}{\partial x} = C(\bar{T}_s - \bar{T}) \qquad \text{steady-state component} \qquad (4.A)$$

$Bv = \rho_L A_L v = M$

$\bar{T}$ (o) to T_o inlet temperature

The only non linear term in the above expression is

$$B\tilde{v}\frac{\partial \tilde{T}}{\partial x}$$

The non-linear term will be taken to be considerably small compared to other terms in equation (3.A). Since it is formed of the multiplication of two varying components, which are usually of very small magnitudes. This assumption will linearise the model in equation (3.A). However, the plant can be regarded as linear only if the input signals are sufficiently small. The assumption of time invariance may also be valid only over restricted time intervals, for example while the plant is run steadily at one operating point.

After neglecting non-linear terms in equation (3.A) it becomes,

$$A\frac{\partial \tilde{T}}{\partial t} + Bv\frac{\partial \tilde{T}}{\partial x} + Bv\frac{\partial \tilde{T}}{\partial x} = C(\tilde{T}_s - \tilde{T}) \qquad (5.A)$$

This is a linear partial differential equation of hyperbolic type

$\frac{d\bar{T}}{dx}$, can be found from equation (4.A).

If the Laplace transform of equation (5.A) is taken w.r.t. inserting the initial condition $T(x,o^+) = 0$, it becomes

$$sA\overset{*}{T}(x,s) + B\overset{*}{v}(s)(\bar{T}_s - \bar{T}_o)k\bar{e}^{kx} + B\bar{v}\frac{d\overset{*}{T}(x,s)}{dx} = C(\overset{*}{T}_s - \overset{*}{T})$$

where $k(\bar{T}_s - \bar{T}_o)\bar{e}^{kx} = \frac{d\bar{T}}{dx}$ (from equation 4.A).

where T_o is the inlet temperature.

$$k = \frac{2\pi rU_L}{MCp}$$

$$\frac{d\overset{*}{T}(x,s)}{dx} + \frac{(sA + C)}{B\bar{v}}\overset{*}{T}(x,s) = \frac{C}{B\bar{v}} - \overset{*}{T}_s - \overset{*}{v}(s)\frac{k(T_s - T_o)\bar{e}^{kx}}{\bar{v}} \quad (6.$$

Now this is an ordinary first-order-differential equation in x. $0 < x \leq L_1$.

To solve equation (6.A), the integrating factor is

$$e^{\frac{(sA + C)}{B\bar{v}}x}$$

$$\overset{*}{T}(x,s).e^{\frac{(sA+C)}{B\bar{v}}} = \frac{C}{B\bar{v}} \overset{*}{T}_s \int e^{\frac{(sA+C)}{B\bar{v}}} dx - \frac{k}{\bar{v}} (\bar{T}_s - \bar{T}_o) \overset{*}{v}(s) \int e^{\frac{(sA+C-kBv)x}{B\bar{v}}} dx + C_1$$

where C_1 is the constant of integration, can be found from the boundary conditions, when $x = 0$, $\overset{*}{T}(x,s) = \overset{*}{T}(o,s) = \overset{*}{T}_o$ (disturbance at inlet).

$$\overset{*}{T} e^{\frac{(sA+C)}{B\bar{v}}} = \frac{C}{B\bar{v}} \cdot \frac{B\bar{v}}{(sA+C)} \overset{*}{T}_s e^{\frac{(sA+C)}{B\bar{v}} x} - \frac{K}{\bar{v}} (\bar{T}_s - \bar{T}_o) \cdot \frac{B\bar{v}}{(sA+C-k\bar{Bv})} \overset{*}{v}(s) \cdot e^{\frac{(sA+C-Bvk)^x}{B\bar{v}}} + C_1 \qquad (7.A)$$

$$\overset{*}{T}(o,s) = \frac{C}{(sA+C)} \overset{*}{T}_s - \frac{k(\bar{T}_s - \bar{T}_o)}{sA} \times B \overset{*}{v}(s) + C_1$$

Let $G_{Tv} = \dfrac{k(\bar{T}_s - \bar{T}_o)\, B}{A}$

$$C_1 = \overset{*}{T}(o,s) - \frac{C \overset{*}{T}_s}{sA+C} + \frac{G_{Tv}}{s} \cdot \overset{*}{v}(s)$$

Substitute C_1 into (7.A) above

$$\overset{*}{T}(x,s) = \frac{C}{(sA+C)}\overset{*}{T}_s - \frac{G_{Tv}}{s}\overset{*}{v}(s).\, e^{-kx} + \overset{*}{T}(o,s).e^{-\frac{(sA+C)x}{B\bar{v}}}$$

$$- \frac{C\,\overset{*}{T}_s}{(sA+C)} \cdot e^{-\frac{(sA+C)x}{B\bar{v}}} + \frac{G_{Tv}}{s}\overset{*}{v}(s) \cdot e^{-\frac{(SA+C)x}{B\bar{v}}}$$

$$\overset{*}{T}(x,s) = \frac{C\,\overset{*}{T}_s}{(sA+C)}\left(1 - e^{-\frac{(sA+C)x}{B\bar{v}}}\right) + \frac{G_{Tv}}{s}\overset{*}{v}(s)\left(e^{-\frac{(sA+C)x}{B\bar{v}}} - e^{-kx}\right)$$

$$+ \overset{*}{T}(o,s) \cdot e^{-\frac{(sA+C)x}{B\bar{v}}} \qquad (8.A)$$

The output temperature at $x = L_1$ is the boiling temperature, which has been assumed constant over that region ($x > L_1$).

$\overset{*}{T}(L_1,s) = \overset{*}{T}_b$ (the boiling temperature).

Substitute x = Ll in equation (8.A).

$$\overset{*}{T}(Ll,s) = \frac{C\overset{*}{T}_s}{sA+C}\left(1 - e^{-\frac{AsL_1}{B\bar{v}}} \cdot e^{-kL_1}\right) + \frac{G_{Tv}}{s}\overset{*}{v}(s)\left(e^{-\frac{AsLl}{B\bar{v}}} - 1\right)e^{-kL_1}$$

$$+ \overset{*}{T}(o,s)\, e^{-\frac{sAL_1}{B\bar{v}}} \cdot e^{-kL_1}$$

Recall that $\frac{A}{B} = 1$, and let $e^{-kL} = gss$

$$T_b^* = \frac{C\,T_s}{(sA+C)}\left(1 - e^{-L\frac{l_s}{\bar{v}}}\,gss\right) + \frac{G_{Tv}}{s}\,v^*(s)\left(e^{-L\frac{l}{\bar{v}}s} - 1\right)gss$$

$$+\ T^*(o,s)\,e^{-L\frac{l}{\bar{v}}s}\,g_{ss} \qquad (9.A)$$

Since ρ_L and A_L have been assumed constant w.r.t both time and space, we can write,

$$M^* = \rho_L A_L\, v^*(s)$$

Infeed flow perturbations.

Now equation (9.A) can be arranged

$$\boxed{T_b^* = Bl_e\, T_s^* + B2_e M^* + B3_e T_o^*} \qquad (10.A)$$

where

$$Bl_e = \frac{C\left(1 - e^{-\frac{Ll_s}{\bar{v}}}\,g_{ss}\right)}{(sA+C)} = \frac{k\bar{v}}{(s+k\bar{v})}\left(1 - e^{-\tau_s}\right)$$

$$B2_e = \frac{K(\bar{T}_s - \bar{T}_o)}{\rho_L A_L} = \frac{K(\bar{T}_s - \bar{T}_o)C_p}{A}\,g_{ss}\left(e^{-\tau s} - 1\right)$$

$$B3_e = g_{ss}\,e^{-\tau s}$$

where $\tau = \frac{L1}{v}$ pure time delay.

This is a transport time delay, taken from the inlet to the boiling region.

A.3 The Two-Phase Region

In this region we are no longer dealing with a single phase, since vapour starts to develop in the region x L1. The conditions which will be assumed here are $\rho_L A_L$, $\rho_v A_v$ both varying with time and space, the temperature T is constant w.r.t. x only, but varying with time. The vapour and liquid phase travel with equal velocities. The heat transfer coefficient is U_B.

Imposing these conditions in the general equation (1.A) we get:

$$\frac{dQ}{dt} = \left(C_p T_b \frac{\partial(\rho_L A_L)}{\partial x} + C_{pv} T_b \frac{\partial(\rho_v A_v)}{\partial x} + L_v \frac{(\rho_v A_v)}{\partial x}\right) \frac{dx}{dt}$$

$$+ \left(C_p T_b \frac{\partial}{\partial t}(\rho_L A_L) + C_{pv} T_b \frac{\partial}{\partial t}(\rho_v A_v) + L_v \frac{\partial}{\partial t}(\rho_v A_v)\right)$$

$$+ \left(C_p \rho_L A_L \frac{\partial T_b}{\partial t} + C_{pv} \rho_v A_v \frac{\partial T_v}{\partial t}\right) = 2\pi(T_s - T_v) \quad (11.A)$$

Now, $\frac{dx}{dt}$ = v the transport velocity of both the liquid phase and the vapour phase.

Or let $\rho_L A_L . v = m$ (liquid mass)

$\rho_v A_v . v - Mv$ (vapour mass)

Equation (11.A) can be rewritten as

$$C_p T_b \frac{\partial m}{\partial x} + C_{pv} T_b \frac{\partial Mv}{\partial x} + L_v \frac{Mv}{v} + C_p T_b \frac{\partial m}{\partial t} + C_{pv} T_b \frac{\partial Mv}{\partial t}$$

$$+ \frac{L_v}{v} \frac{\partial Mv}{\partial t} + \frac{(C_p m + C_{pv} Mv)}{v} \times \frac{\partial T_b}{\partial t} = 2\pi r U_b (T_s - T).$$

(12.A)

If the system is considered to have small perturbations about a stable operating condition of thermodynamics and hydrodynamics, then

$$m = \bar{m}(x) + \tilde{m}(x,t)$$

$$Mv = \overline{MV}(x) + \tilde{MV}(x,t)$$

$$T_b = \bar{T}_b + \tilde{T}(t)$$

Substitute these in equation (12.A).

$$C_p T_b \frac{\partial \tilde{m}}{\partial x} + C_{pv} T_b \frac{\partial \tilde{Mv}}{\partial x} + L_v \frac{\partial \tilde{Mv}}{\partial x} + \frac{C_p T}{\bar{v}} \frac{\partial \tilde{m}}{\partial t} + \frac{C_{pv} T_b}{v} \cdot \frac{\partial \tilde{Mv}}{\partial t}$$

$$+ \frac{L_v}{\bar{v}} \frac{\partial \tilde{Mv}}{\partial t} + \frac{(C_p \bar{m} - C_{pv} \overline{MV})}{v} \frac{\partial \tilde{T}}{\partial t} = 2\pi r U_b (\tilde{T}_s - \tilde{T}_b) \quad (13.A)a$$

$$C_pT_b \frac{\partial \bar{m}}{\partial x} + C_{pv}T_b \frac{\partial \bar{M}v}{\partial x} + L_v \frac{\partial \bar{M}v}{\partial x} = 2\pi r U_b (\bar{T}_s - \bar{T}_v) \quad (13.A)b$$

Where here we have considered that the perturbation in v is considerably less than its corresponding steady-state component and most of the effect is considered to come from the change in $\rho_L A_L$ and $\rho_v A_v$ w.r.t. time and space.

Now, if we can assume further $C_p \approx C_{pv}$ near boiling point equation (13.A) becomes

$$\bar{v} L_v \frac{\partial \tilde{M}v}{\partial x} + L_v \frac{\partial \tilde{M}v}{\partial t} + C_p \bar{M} \frac{\partial \tilde{T}_b}{\partial t} = 2\pi r U_b (\tilde{T}_s - \tilde{T}_v) \bar{v} \quad (14.A)a$$

and equation (13.A) becomes

$$L_v \frac{\partial \bar{M}v}{\partial x} = 2\pi r U_b (\bar{T}_s - \bar{T}_b) \quad (14.A)b$$

which confirms our assumption that the heat transferred is used to evaporate the liquid at a constant temperature $\bar{T}_v$, in the steady state condition.

Equation (14.A) above has been obtained after making substitutions from the following mass balance model in the region $x >= L_1$.

Accumulation of mass in a volume $A_L dx$ can be written as follows:

$$\frac{\partial}{\partial t}(A_L \rho_L dx) = m(x,t) - m(x + \Delta x, t) = am\, dx$$

multiply by $\bar{v}$

$$\frac{\partial m}{\partial t} = -\bar{v}\left(\frac{\partial m}{\partial x} - \bar{a}\, m\right)$$

where $\bar{a}$ is the rate of evaporation per unit length.

If the liquid mass input to the element dx is m then $\bar{a}$ m dx is the amount of liquid that evaporates.

Similarly, a mass balance equation can be written for the vapour stream.

$$\frac{\partial Mv}{\partial t} = -\bar{v}\left(\frac{\partial Mv}{\partial x} - \bar{a}\, m\right)$$

$\bar{a}m$, in the above two equations has appeared as a -ve quantity in the liquid stream mass balance and as +ve in the vapour mass stream. This is because an amount of $\bar{a}m$ dx goes from the liquid phase into the vapour phase. To consider the steady and unsteady states take

$$MV = \bar{M}\bar{V} + \tilde{M}\tilde{V}, \qquad m = \bar{m} + \tilde{m}$$

$$\frac{\partial \tilde{m}}{\partial t} = -\tilde{v}\left(\frac{\partial \tilde{m}}{\partial x} + a\, \tilde{m}\right) \qquad \text{(15.A)}$$

$$\frac{\partial \tilde{M}v}{\partial t} = -\bar{v}\left(\frac{\partial \tilde{M}v}{\partial x} - \bar{a}\,\tilde{m}\right)$$

$$0 = -\bar{v}\left(\frac{\partial \bar{m}}{\partial x} + \bar{a}\,\bar{m}\right)$$

$$0 = -\bar{v}\left(\frac{\partial \tilde{M}}{\partial x} - \bar{a}\,\bar{m}\right) \qquad \text{(15.A)b}$$

From equations (15.A)b

$$\frac{\partial \bar{m}}{\partial x} = \bar{a}\,\bar{m} = -\frac{\partial \bar{M}v}{\partial x}$$

This last condition indicates that

$\bar{m} + \bar{M}v$ = a constant = $\bar{M}$ (the total mass flow)

Also, we have from equation (15.A)a

$$\frac{\partial \tilde{m}}{\partial t} + \bar{v}\frac{\partial \tilde{m}}{\partial x} = -\frac{\partial \tilde{M}}{\partial t} - \bar{v}\frac{\partial \tilde{m}}{\partial x}$$

Making use of the last condition we arrive at equation (14.A) from equation (13.A).

In equation (14.A)a

$$L_v\frac{\partial \tilde{M}_v}{\partial x} + \frac{L_v}{v}\cdot\frac{\partial \tilde{M}v}{\partial t} + \frac{C_p\bar{M}}{v}\cdot\frac{\partial \tilde{T}}{\partial t} = 2\pi r U_b\,(\tilde{T}_s - \tilde{T}_b) \qquad \text{(14.A)}$$

If further we consider $\bar{v}$ is very large compared to C_pU and L; and that is usually the case,

$$L_v \frac{\partial \tilde{M}_v}{\partial x} \cong 2\pi U_b(\tilde{T}_s - \tilde{T}_b) \qquad \begin{matrix} x > L_1 \\ \text{or} \\ L_1 \lesssim x \lesssim L \end{matrix}$$

Integrating this equation w.r.t. x with the boundary conditions: at $x = 0$, $\tilde{M}_v = 0$, we get

$$\boxed{\tilde{M}_v = \frac{2\pi r U_B}{L_v} (\tilde{T}_s - \tilde{T}_b)(L - L_1)} \qquad (16.A)$$

A.4 The Two Phases Velocity, $\bar{v}$

Although in the analysis a value for v is not needed to reach a final solution. Nevertheless some interesting relations need to be observed.

The quantity $\bar{v}$ has been used in the above equations in the following sense.

$$(\rho_L A_L + \rho_v A_v)\,\bar{v} = M \quad \text{(total mass flow)} \qquad (17.A)$$

ρ_L, ρ_v and M usually can be known, but A_L and A_v are varying w.r.t. x.

Since the infinitestimal volume is the sum of liquid and vapour volumes

$$A_T\,dx = A_L\,dx + A_v\,dx$$

where A_T is the total cross-sectional area of the tube

$$A_T = A_L + A_v$$

$$A_L = A_T - A_v$$

$$\rho_L (A_T - A_v)\bar{v} + \bar{v}\rho A_v = M$$

where $\bar{v}\rho A_v$ = vapour mass

$$\rho_L(A_T - A_v)\bar{v} = M - Mv$$

$$M - Mv = \rho_l \bar{v} \left(A_T - \frac{Mv}{\frac{V}{\rho_v}}\right)$$

$$= \rho_L \left(\bar{v}A_T - \frac{Mv}{\rho_v}\right)$$

$$\frac{M - M_v}{\rho_L} + \frac{MV}{\frac{V}{\rho_v}} = \bar{v}A_T$$

$$\bar{v} = \frac{M - MV}{\rho_L A_T} + \frac{MV}{\rho A_T} \qquad (17.A)$$

Equation (17.A) is an interesting equation, it has a kinematic interpretation that the system velocity, $\bar{v}$, is the resultant velocity of the liquid phase if it were occupying the whole tubular volume plus the vapour phase velocity if vapour were to occupy the whole tubular volume. Also, the limiting conditions are very clear. For example, if there was only vapour i.e. M → Mv then

$$\bar{v} = \frac{MV}{\rho_v A_T}$$

or if there was no vapour i.e. Mv = 0 then

$$\bar{v} = \frac{M}{\rho_L A_T}$$ The usual expression for velocity of liquid flow.

Unsurprisingly, the velocity $\bar{v}$ could have been obtained from the famous work of Taylor and Rosenbrook but is in a different approach and formulation. A brief description of the work of Taylor and Rosenbrook can be given as follows:

If the equations of energy balance and mass balance are referred to a moving coordinate with a mean velocity $\bar{V}m$, which is the average of the stream velocities weighted by their repsective heat capacities, thus

$$\bar{V}m = \frac{V_1 A_L \rho_L C_p + V_v A_v \rho_v C_{pv}}{A_L \rho_L C_p + A_v \rho_v C_{pv}}$$

where V_1 is the velocity of the liquid stream, and V_2 is the velocity of vapour stream.

If we apply the condition of C_p C_{pv} at boiling point conditions then the expression for Vm becomes:

$$\bar{V}_m = \frac{V_L \rho_L A_L + V_v A_v \rho_v}{A_L \rho_L + A_v \rho_L}$$

$$\bar{V}_m(A_L \rho_L + A_v \rho_v) = V_L A_L \rho_L + V_v A_v \rho_v$$

$$= m + Mv = M$$

which is very similar to our expression in (17.A)a. But our equation in (17.A) shows in some explicit form that the mean velocity is a vector sum of the velocity of each phase if it was occupying the whole tube area A_T.

A.5 Tubes Wall

The internal tube that separates the fluid inside the tube from the steam on the shell, in some application may have considerable effect on the energy balance equations. The main assumptions we will invoke here are that the tube walls are thin and hence there is no temperature gradient across the tube thicnkness, the temperature of the wall is the same everywhere along the tube due to the well mixing of the steam inside the shell. As has been mentioned earlier the well mixing of the steam inside the shell will be similar to the dynamics of a well stirred tank.

So, to model this effect of the tube wall it suffices to consider the overall system of the shell/tube system. The accumulation of heat inside the wall,

= heat supplied by the steam

- heat taken by the fluid inside the tube

$$\frac{d\ W\ C_{pw}T_s}{dt} = S_cL_c - MC_p\ (T_s-T_o) - MvL_v \qquad (19.A)$$

where W is the mass of the tube wall, C_{pw} is its specific heat, S_c is the total rate of steam condensate.

$M\ C_p\ (T_s-T_o)$ is the total amount of heat removed by the liquid phase

MVL_v is the total amount of heat taken to evaporate Mv of fluid mass per second.

As usual, taking perturbating components and steady-state we can write equation (19.A) as follows:

$$W\,C_{pw}\,\frac{d\tilde{T}_s}{dt} = \tilde{S}_c L_c - \tilde{M}C_p\,(\bar{T}_b-\bar{T}_o) - \bar{M}C_p(\tilde{T}_b-\tilde{T}_o) - \tilde{M}_v L_v \qquad (20.A)a$$

$$\bar{S}\,L_c = \bar{M}C_p\,(\bar{T}_b-\bar{T}_o) + \bar{M}\,L_v \qquad (20.A)b$$

where equation (20.A)a shows the dynamic state and (20.A)b shows the steady-state, which complying well with the physical interpretation that the heat is supplied by the condensation of steam and removed through the transfer actions of $2\pi r U_L\,(\bar{T}_s-\bar{T})$ and $2\pi r U_b\,(\bar{T}_s-\bar{T}_b)$.

$$\bar{M}\,C_p\,(\bar{T}_b-\bar{T}_o) = \int_o^{L_1} 2\pi r\,U_L\,(\bar{T}_s-\bar{T}(x))\,dx$$

$$\bar{M}_v L_v = \int_{L_1}^{L} 2\pi r\,U_b\,(\bar{T}_s-\bar{T}_b)\,dx$$

Rearrange equation (20.A) and take the Laplace transform

$$\overset{*}{S}_c = \frac{\overset{*}{M}C_p}{L_c}\,(\bar{T}_b-\bar{T}_o) + \frac{\bar{M}C_p}{L_c}\,(\overset{*}{T}_b-\overset{*}{T}_o) + \frac{\overset{*}{M}_v L_v}{L} + \frac{WC_{pw}}{L_c}\,s\,\overset{*}{T}_s$$

A.6 Steam Chest

The mass balance on the steam shell can be considered in the following manner:

$$\frac{dV\rho}{dt} = S_S - S_C - S_O$$

where V is the total volume of the shell and assumes to be constant (the volume V is much greater than the hold-up of the condensate). S_S is the steam input flow rate, S_C is the rate of condensation of steam inside the shell due to the heat exchange, and S_O is the rate of drainage of the condensate from the shell.

Under steady-state, $\left(\frac{dV\rho}{dt} = 0\right)$

$$\bar{S}_S = \bar{S}_C + \bar{S}_O$$

and the dynamics state inside the shell will be

$$V \frac{d\tilde{\rho}}{dt} = \tilde{S}_S - \tilde{S}_C - \tilde{S}_O \qquad (21.A)$$

From the correlation equations in Appendix we can also have

$$\tilde{T}_S = 2.1 * \tilde{P}_S \qquad (22.A)$$

$$\tilde{\rho} = 37.091 \times 10^{-3} \tilde{P}_S \qquad (23.A)$$

An independent expression for S_O can be obtained by considering that the flow of drainage is affected by the steam pressure inside the shell, the atmospheric pressure outside the shell, and the friction factor of the drainage val

$$S_o = C_d \sqrt{P_s - P_{atm}}$$

$$S_o = \bar{S}_o + \tilde{S}_o = C_d \sqrt{\bar{P}_s + \tilde{P}_s - P_{atm}}$$

$$= C_d \sqrt{(\bar{P}_s - \bar{P}_{atm}) \quad (1 + \frac{\tilde{P}_s}{(\bar{P}_s - \bar{P}_{atm})}}$$

$$= C_d \sqrt{(\bar{P}_s - \bar{P}_{atm})} \cdot (1 + \frac{\tilde{P}_s}{(\bar{P}_s - \bar{P}_{atm})})^{\frac{1}{2}}$$

$$= \bar{S}_o (1 + \frac{\tilde{P}_s}{(\bar{P}_s - \bar{P}_{atm})})^{\frac{1}{2}}$$

Now assume, $\frac{\tilde{P}_s}{(\bar{P}_s - \bar{P}_{atm})} < 1$

then $\bar{S}_o + \tilde{S}_o \approx \bar{S}_o (1 + \frac{\tilde{P}_s}{2\,(\bar{P}_s - \bar{P}_{atm})})$

$$\tilde{S}_o = \frac{\bar{S}_o \tilde{P}_s}{2(\bar{P}_s - \bar{P}_{atm})} \qquad (24.A)$$

$$\overset{*}{S}_o = \frac{\bar{S}_o \overset{*}{T}_s}{2 \times 2.1 (\bar{P}_s - \bar{P}_a)}$$

Substitute equation (22.A) and (23.A) into equation (21.A) and take the Laplace transform, equations (21.A) becomes:

$$\frac{V \times 37.091 \times 10^{-3}}{2.1} \overset{*}{T}_s \cdot s = \overset{*}{S}_s - \overset{*}{S}_c - \overset{*}{S}_o$$

Let $V \times 37.091 \times 10^{-3} = a_p$

$$\boxed{\overset{*}{T_s} = \frac{2.1}{a_p S}\left(\overset{*}{S_s} - \overset{*}{S_c} - \overset{*}{S_o}\right)} \qquad (25.A)$$

So far, there are five independent equations, for T_s, M, T_v, S_o and S_c in three independent input variables to the process, S_s, M and T_o.

The equations are best put into a signal flow graph (Figure 4.3A) to facilitate and to give insight into the relationship between the variables.

Let $\dot{U}_B = \frac{2\pi r U_b}{L_c}(L - L_1)$ $\quad G_{FE} = \left(\frac{\overline{S}_o}{2(P_s - P_{atm})} + U_B\right)$

$$T_{FE} = \left(\frac{a_p}{2.1} + \frac{W C_{pw}}{L_c}\right) \Big/ \left(\frac{\overline{S}_o}{2(P_s - P_{atm})} + U_B\right)$$

$$\text{Loop 1} = \frac{-\overline{S}_o \times 2.1}{2a_p s(\overline{P}_s - \overline{P}_{atm}) \times 2.1}$$

$$\text{Loop 2} = \frac{-2.1 \times (U_B + WC_{pw}s')}{L_c a_p s}$$

$$\text{Loop 3} = \frac{2.1}{a_p s} \, Bl_e \, \frac{\overline{M}C_p + U_B}{L_c}$$

In loop 3 if we assume that $MC_p + U_B \quad L_c$ then we can ignore the effect of this loop in determining the characteristic equation of the system.

The characteristic equation Δ = 1 - loop 1 - loop 2

$$\Delta = 1 + \frac{\bar{S}_o}{2a_p S(P_s - P_a)} + 2.1 \frac{\frac{WC_{pw}s}{---}}{\frac{(U_B + L_c)}{---}}{a_p s}$$

$$= \frac{((2a_p C\bar{P}_s - \bar{P}_a) + 2.1\ (\bar{P}_s - \bar{P}_a)\ \frac{2WC_{pw}}{----}\ L_c + \bar{S}_o + 4.2\ (\bar{P}_s - \bar{P}_a)\ U_B))}{2a_p s\ (\bar{P}_s - \bar{P}_a)}$$

$$= \frac{2.1}{-----}\ \mu T_{FE}\ s + 1)\ G_{FE}$$
$$a_p S$$

Manipulating the different equations and using Mason's rule we get

$$\frac{\overset{*}{m}}{\overset{*}{S}} = \frac{U_B\ (s + Kv\ g_{ss}\ e^{-\tau s})}{(T_{FE}\ S + 1)\ (K_v + S)\ G_{FE}}$$

$$\frac{\overset{*}{M}_v}{\overset{*}{M}_T} = \frac{C_p\ (\bar{T}_v - \bar{T}_o)\ U_B\ (\beta_1 - 1)}{L_c\ G_{FE}\ (T_{FE}\ S' + 1)} - U_B\ B_{2e}$$

$$\frac{\overset{*}{M}}{\overset{*}{T}_o} = B3_e(-U_B) + (-\frac{\bar{M}\ C_p}{----})\ (-\frac{2.1}{-----})\ (U_B - U_B\ B1e)\ \frac{1}{\Delta}$$
$$L_c \qquad a_p s$$

$$= -U_B\ B_{3e} + \frac{\bar{M}C_p}{---} \times \frac{2.1}{---}\ U_B \cdot \frac{a_p s}{---}\ \frac{(1 - B1_e)}{---------------}$$
$$L_c \quad a_p s \quad 2.1 \quad G_{FE}\ (T_{FE}\ S + 1)$$

$$\frac{\overset{*}{T}_b}{\overset{*}{S}} = \frac{B1_e}{G_{FE}\,(T_{FE}\,s + 1)}$$

$$\frac{\overset{*}{T}_b}{\overset{*}{M}} = B2_e - \frac{C_p\,(\bar{T}_b - \bar{T}_o)\,B1_e}{L_c\,G_{FE}\,(T_{FE}S + 1)}$$

$$\frac{\overset{*}{T}_b}{\overset{*}{T}_o} = B3_e + \frac{\bar{M}}{L_c}\frac{C_p}{a_p S}\,(2.1)\,B1_e\,\frac{1}{\Delta}$$

$$= B3_e + \frac{\bar{M}C_p\,B1_e}{L_c\,G_{FE}\,(T_{FE}S + 1)}$$

$$\frac{\overset{*}{m}}{\overset{*}{M}} = 1 - \frac{\overset{*}{M}_v}{\overset{*}{M}} \qquad \frac{\overset{*}{M}_v}{\overset{*}{S}} = -\frac{\overset{*}{m}}{\overset{*}{S}}$$

Now it is clear that there are two dominant time constants in the dynamical model, T_{FE} and $1/K\bar{v}$.

$$T_{FE} = \left(\frac{\bar{a}_p}{2.1} + \frac{WC_{pw}}{L_c}\right) \Big/ \left(\frac{S_o}{2(P_s - P_a)} + U_B\right)$$

which includes the effects of the tube wall, heat transfer coefficients in the boiling region, and the conditions in the steam shell through a_p and P_s. Also, in U_B the subcooled length, L_1, is included. The effect of each of these in the time constant, T_{FE}, can be measured by substituting the corresponding value of each parameter of the above quantities.

The second time constant is:

$$K\bar{v} = \frac{2\pi r\, U_L\, \bar{v}}{M\, C_p} = \frac{2\pi r\, U_L}{A_T \rho_L C_L}$$

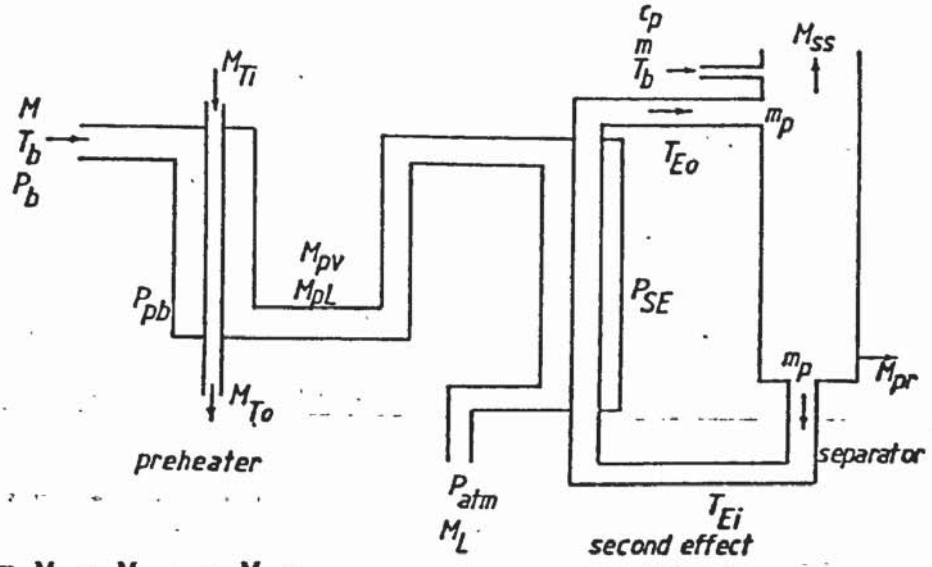
which includes the heat transfer coefficient in the subcooled length or the liquid-phase heat transfer coefficient, the density of the liquid inside the tube and its heat capacity, plus the tubes constant, e.g. the radius r.

APPENDIX A

MODELLING OF THE PREHEATER AND SECOND EFFECT

The preheater and the second effect were designed to operate as heat exchangers. Evaporation in these two units should not take place. The restriction is mainly due to the equipment manufacturers specifications. From a dynamic point of view, if the above limitations are not fulfilled the plant may exhibit unstable behaviour.

Now consider the schematic diagram below of the preheater, the second effect, and the second effect separator.



Mass Balance

Preheater

$$V_{ph} \frac{d\rho P_b}{dt} = M - M_{pv} - M_{pL}$$

$$M = C_1 \sqrt{P_b - P_{pb}} \qquad \tilde{\rho}_{pb} = 37.091 \times 10^{-3} \tilde{P}_{pb}$$

Second effect

$$V_{SE} \frac{d\rho_{SE}}{dt} = M_{pv} + M_{pL} - M_{pL} - M_{SL}$$

$$M_{pv} = C_2 \sqrt{P_{pv} - P_{SE}}$$

Separator

$$\frac{dAh\rho_{sp}C_{sp}}{dt} = mC_1 + m_p C_2 - m_p C_{sp}$$

$$A \frac{d\, h_{sp}\rho_{sp}}{dt} = m + mp - mp - Mss - M_{pro}$$

Energy Balance

$$\overset{*}{T}_o = \beta_{1p} \overset{*}{T}_{bp} + \beta_{2p} \overset{*}{M} + \beta_{3p} \overset{*}{T}_i$$

$$\tilde{M}_{PL}\lambda_v = \tilde{M}C_p (\bar{T}_o - \bar{T}_i) + \bar{M}C_p (\tilde{T}_o - \tilde{T}_c) + W_p C_{pp}\, d \frac{\tilde{T}_b}{dt} + \text{losses}$$

steady state $\bar{M}_{PL}\lambda_v = \bar{M}C_p (\bar{T}_o - \bar{T}_i) + \text{losses}$.

If for the moment the losses are neglected then

$$\bar{M}_{PL} = \bar{M}C_p (\bar{T}_o - \bar{T}_i)/\lambda$$

Second Effect

$$\overset{*}{T}_{EO} = \beta_{IE} \overset{*}{T}_{SE} + \beta_{2E} \overset{*}{M}_p + \beta_{3p} \overset{*}{T}_{Li}$$

$$\tilde{M}_{SL}\lambda_v = \tilde{m}E\, C_p (\bar{T}_{EO} - \bar{T}_E) + \bar{m}_p C_p (\tilde{T}_{EO} - \tilde{T}_{EI}) +$$

$$W_E C_E \frac{d\tilde{T}_{SE}}{dt} + \text{losses}$$

Steady state

$$\bar{M}_{SL}\lambda = \bar{M}_p C_p (\bar{T}_{EO} - \bar{T}_{Ei}) + \text{losses}$$

Separator

$$A \frac{d(h\rho_{sp}C_p T_{Ei})}{dt} = mC_p T_b + m_p C_p T_{EO} - m_p C_p T_{Ei} - M_{ss}\lambda - M_{pr} C_p T_{Ei}$$

$$A\rho_{sp}C_p T_{Ei} \frac{dh}{dt} + Ah\rho_{sp}C_p \frac{dT_{Ei}}{dt} = \bar{m}C_p\tilde{T}_b + \tilde{m}C_p\bar{T}_b + \bar{m}_pC_p(\tilde{T}_{EO}-\tilde{T}_{Ei}) - M_{ss}\lambda_v + \tilde{m}_pC_p (\bar{T}_{EO} - \bar{T}_{EC}) - \tilde{M}_{pr}C_pT_{Ei} - \bar{M}_{pr} C_p T_{Ei}$$

let $\rho_{sp} = \rho_{pr}$ outlet density (CSRT)

Preheater

$$M = C_1 \sqrt{P_b - P_{bp}}$$

$$\bar{M} + \tilde{M} = C_1 \sqrt{\bar{P}_b - \bar{P}_{bp} + \tilde{P}_b - \tilde{P}_{bp}}$$

$$= C_1 \sqrt{(\bar{P}_b - \bar{P}_{bp})\left(1 + \frac{(\tilde{P}_b - \tilde{P}_{bp})}{(\bar{P}_b - \bar{P}_{bp})}\right)}$$

$$= C_1 \sqrt{\bar{P}_b - \bar{P}_{pb}} \cdot \sqrt{1 + \frac{(\tilde{P}_b - \tilde{P}_{bp})}{(\bar{P}_b - \bar{P}_{pb})}}$$

$$\bar{M}\left(1 + \tfrac{1}{2} \frac{(\tilde{P}_b - \tilde{P}_{pb})}{(\bar{P}_b - \bar{P}_{pb})}\right)$$

(1) $$\tilde{M} \simeq \frac{\bar{M}}{2} \frac{(\tilde{P}_b - \tilde{P}_{pb})}{(\bar{P}_b - \bar{P}_{pb})}$$

(2) $$37.091 \times 10^{-3} V_{ph} \frac{d\tilde{P}_{pv}}{dt} = \tilde{M} - \tilde{M}_{pv} - \tilde{M}_{pL}$$

let $37.091 \times 10^{-3} V_{ph} = a_{ph}$

$$M_{pL} = \frac{MC_p (T_o - T_i)}{\lambda_v} + \frac{MC_p}{\lambda_v} (T_o - T_i) + \frac{W_p C_{pp}}{\lambda} \frac{dT_{pb}}{dt}$$

$$\tilde{P}_{pb} = \frac{\tilde{T}_{pb}}{2.1}$$

$$(3) \quad \tilde{M}_{pL} = \tilde{M}C_p \frac{(\bar{T}_o - \bar{T}_i)}{\lambda_v} + \frac{\bar{M}C_p}{\lambda_v} (\tilde{T}_o - \tilde{T}_i) + \frac{W_p C_p x 2.1}{\lambda_v} \frac{d \tilde{P}_{bp}}{dt}$$

Substitute equation (1) (3) into (2)

$$aph \frac{d \tilde{P}_{pb}}{dt} = \frac{\bar{M}}{2(P_b - P_{pb})} (\tilde{P}_b - \tilde{P}_{pb}) - \tilde{M}_{pv} - \frac{\tilde{M}C_p}{\lambda_v} (\bar{T}_o - \bar{T}_i) - \frac{\bar{M}C_p}{\lambda_v} (\tilde{T}_o - \tilde{T}_i)$$

$$\frac{-W_p C_{pp} 2.1}{\lambda_v} \cdot \frac{d\tilde{P}_{bp}}{dt}$$

$$(4) \quad -\frac{(aph + W_p C_p 2.1)}{\lambda_v} \frac{d\tilde{P}_{pb}}{dt} + \frac{\bar{M}}{2(\bar{P}_b - \bar{P}_{pb})} \tilde{P}_{pb} =$$

$$\frac{\bar{M}}{2(\bar{P}_b - \bar{P}_{pb})} \tilde{P}_b - \tilde{M}_{pv} - \frac{\tilde{M}C_p}{\lambda_v} (\bar{T}_o - \bar{T}_i) - \bar{M}C_p \frac{(\tilde{T}_o - \tilde{T}_i)}{\lambda_v}$$

Laplace equation 4 above.

$$ap (s + T_{pp}) \overset{*}{P}_{pb} = \frac{\bar{M}}{2(P_b - P_{pb})} \overset{*}{P}_b - \overset{*}{M}_{pv} - \frac{\overset{*}{M}C_p}{\lambda_v} (\bar{T}_o - \bar{T}_i)$$

$$- \frac{\bar{M}C_p}{\lambda_v} (\overset{*}{T}_o - \overset{*}{T}_i)$$

$$\alpha_{ph} = \frac{(aph + 2.1\ W_p C_{pp})}{\lambda_v}$$

$$T_{pp} = \frac{M/2(\bar{P}_b - \bar{P}_{pb})}{(aph + \frac{2.1\ W_p\ C_{pp}}{\lambda_v})}$$ an inverse time constant

Second Effect

(1) $$\tilde{M}_{pv} = \bar{M}_{pv}\ \frac{(\tilde{P}_{pb} - \tilde{P}_{SE})}{(\bar{P}_{pb} - \bar{P}_{SE})}$$

$$37.091 \times 10^{-3}\ V_{SE}\ \frac{d\ \tilde{\rho}_{SE}}{dt} = \tilde{M}_{pv} - \tilde{M}_{SL}$$

(The effect of the liquid water M_{pL} is neglected when considering the density of vapour inside the SE shell).

M_{SL} is the condensate formed in the SE shell.

$$a_{SE} = 37.091 \times 10^{-3} \times V_{SE}$$

(2) $$a_{SE}\ \frac{d\ \tilde{P}_{SE}}{dt} = \tilde{M}_{pv} - \tilde{M}_{SL}$$

(3) $$\tilde{M}_{SL} = \tilde{m}_p\ \frac{C_p}{\lambda_v}\ (\bar{T}_{EO} - \bar{T}_{Ei}) + \frac{\bar{m}_p C_p}{\lambda_v}(\tilde{T}_{EO} - \tilde{T}_{Ei}) + \frac{W_E C_E}{\lambda_v}\ \frac{d\tilde{T}_{SE}}{dt}$$

$$\tilde{T}_{SE} = 2.1\ \tilde{P}_{SE}$$

Substitute (1) and (3) into (2)

$$a_{SE}\frac{d\tilde{P}_{SE}}{dt} = \frac{\bar{M}_{pv}}{2(P_{pb}-P_{SE})}(\tilde{P}_{pb} - \tilde{P}_{SE}) - \tilde{m}_p C_p \frac{(\bar{T}_{EO}-\bar{T}_{Ei})}{\lambda_v}$$

$$- \frac{\bar{m}_p C_p}{L_v}(\tilde{T}_{EO}-\tilde{T}_{Ei}) \quad - \frac{2.1\ W_E\ C_{pE}}{L_v} \cdot \frac{d\tilde{P}_{SE}}{dt}$$

$$(4)\quad \frac{(a_{SE} + 2.1W_E C_E)}{\lambda_v}\frac{d\tilde{P}_{SE}}{dt} + \frac{\bar{M}_{pv}}{2(\bar{P}_{pb}-\bar{P}_{SE})}\tilde{P}_{SE} =$$

$$\frac{\bar{M}_{pv}}{2(P_{pb}-P_{SE})}\tilde{P}_{pb} - \tilde{m}_p C_p (\bar{T}_{EO}-\bar{T}_{Ei}) - \frac{\bar{m}_p C_p}{\lambda_v}(\tilde{T}_{EO}-\tilde{T}_{Ei})$$

$$\alpha_{SE} = (a_{SE} + \frac{2.1\ W_E C_E)}{\lambda_v}$$

$$T_{SE} = \frac{\bar{M}_{pv}/2(\bar{P}_{pb} - \bar{P}_{SE})}{\alpha_{SE}} \qquad \text{an inverse time constant}$$

Laplace equation (4)

$$\alpha_{SE}\ (s + T_{SE})\ \overset{*}{P}_{SE} = \frac{\bar{M}_{pv}}{2(\bar{P}_{pb} - \bar{P}_{SE})} \cdot \overset{*}{P}_{pb} - \frac{\overset{*}{\bar{m}}_p C_p}{\lambda_v}(\bar{T}_{EO}-\bar{T}_{Ei})$$

$$- \frac{\bar{m}_p C_p}{\lambda_v}(\overset{*}{T}_{EO} - \overset{*}{T}_{Ei})$$

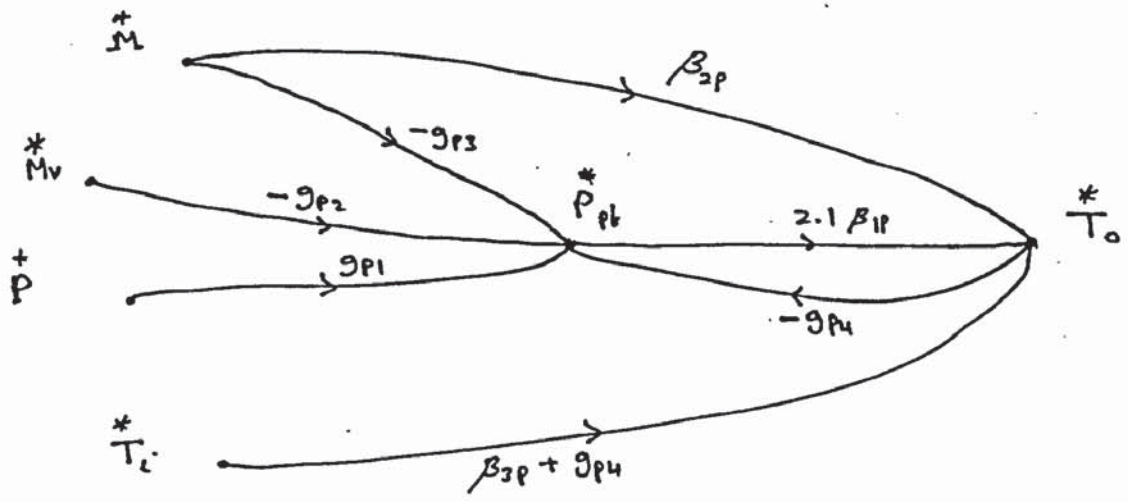
The Signal Flow Graphs

Preheater

$$\overset{*}{P}_{pb} = g_{p1}\frac{\overset{*}{P}_b}{(s+T_{pp})} - \frac{g_{p2}\overset{*}{M}_v}{(s+T_{pp})} - \frac{g_{p3}\overset{*}{M}}{(s+T_{pp})} - g_{p4}\frac{(\overset{*}{T}_o-\overset{*}{T}_i)}{(s+T_{pp})}$$

$$g_{p1}\frac{\bar{M}}{2\alpha_p(P_b-P_{pb})}, \quad g_{p2} = \frac{1}{\alpha p}, \quad g_{p3} = \frac{C_p(\bar{T}_o-\bar{T}_i)}{\lambda\alpha_p}, \quad g_{p4}\frac{\bar{M}C_p}{\alpha_p\lambda_v}$$

$$\overset{*}{T}_o = \beta_{1p}\,2.1\,\overset{*}{P}_{pb} + \beta_{2p}\overset{*}{M} + \beta_{3p}\overset{*}{T}_i$$



Second Effect Shell

$$\overset{*}{P}_{SE} = g_{s1}\frac{\overset{*}{P}_{pb}}{(s+T_{SE})} - \frac{g_{s2}\overset{*}{m}}{(s+T_{SE})} - g_{s3}\frac{(\overset{*}{T}_{EO}-\overset{*}{T}_{Ei})}{(s+T_{SE})}$$

$$g_{s1}\frac{\bar{M}_{pv}}{2\alpha_{SE}(P_{pb}-P_{SE})}, \quad g_{s2}, = \frac{C_p\,(\bar{T}_{EO}-\bar{T}_{El})}{\alpha_{SE}\,\lambda_v}, \quad g_{s3}, = \frac{\bar{m}_pC_p}{\alpha_{SE}\lambda_v}$$

$$\overset{*}{T}_{EO} = \beta_{s1}\,2.1\,\overset{*}{P}_{SE} + \beta_{S2}\overset{*}{m} + \beta_{s3}\overset{*}{T}_{Ei}$$

$$\overset{*}{M}_{SL}\,\lambda_v = \overset{*}{m}_E\,C_p\,(\bar{T}_{EO}-\bar{T}_{Ei}) + \bar{M}_E\,C_p\,(\overset{*}{T}_{E\,o}-\overset{*}{T}_{Ei}) + sW_E\,C_{pE}\overset{*}{T}_{SE}$$

$$+ \text{losses}$$

Separators

Mass balance

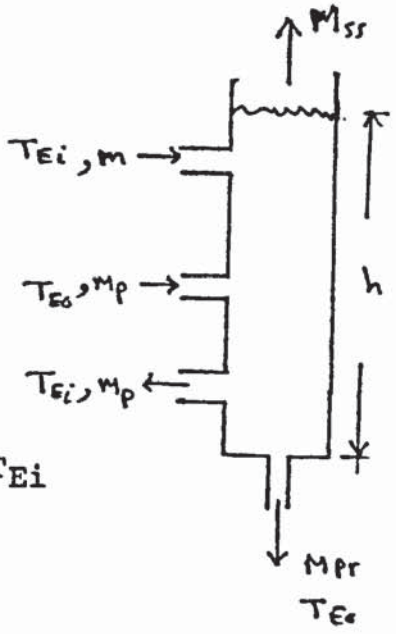
$$\frac{Adh}{dt} = M + \cancel{M_{EO}} - M_{ss} - \cancel{M_{Ei}} - M_{pr}$$

neglect the dynamic effect of the pump

$$\frac{Adhp}{dt} = m - M_{ss} - m_{pr}$$

Mass balance

$$\frac{Ad(\rho h)}{dt} = m + \cancel{m_p} - \cancel{m_p} - M_{pr} - M_{ss}$$



Heat balance

$$C_p A \frac{d(\rho h T_{EO})}{dt} = mC_p T_{Ei} + m_p C_p T_{EO} - m_p C_p T_{Ei}$$

$$- M_{pr} T_{Ei} C_p - M_{ss} \lambda_v$$

Evaluation of L_1, K, U_L and U

Now if we assume that the rate of condensation is equal along the tube then

$$S_c = Ls$$

then for the subcooled length:

$$\text{the heat transferred} = 2\pi\tau\, U_L \int_0^L (\bar{T}_s - \bar{T}(x))dx$$

$$= MC_p\,(\bar{T}_s - \bar{T}_o)\,(1 - g_{ss})$$

$$= L_c\,\bar{s}\,.L_1 = L_c\,\frac{\bar{S}_c}{L}\,.\,L_1$$

$$L_c\,\frac{\bar{S}_c}{L}\,.\,L_1 = MC_p\,(\bar{T}_s - \bar{T}_o)\,(1 - g_{ss})$$

$$= MC_p\,(\bar{T}_b - \bar{T}_o)$$

$$\text{Therefore } L_1 = \frac{\bar{M}C_p\,(\bar{T}_b - \bar{T}_o)\,.L}{L\,\bar{S}_c}$$

$$\text{or, } \frac{L_1}{L} = \frac{\bar{M}C_p\,(\bar{T}_b - \bar{T}_o)}{L\,\bar{S}_c}$$

$$= \frac{\text{amount of heat taken by the subcooled region}}{\text{total amount of heat delivered by the steam}}$$

From the steady-state solution of equation (4.A)

$$MC_p\,\frac{d\bar{T}}{dx} = 2\pi\tau\, U_L\,(\bar{T}_s - \bar{T}(x))$$

$$\text{or,} \quad \frac{d\bar{T}}{dx} = K\,(\bar{T}_s - \bar{T}(x)) \qquad 0 \leq x \leq L_1$$

$$\bar{T}(o) = T_o$$

$$\bar{T}(L_1) = T_b$$

$$K\,L_1 = -\ln \frac{(\bar{T}_s - \bar{T}_b)}{\bar{T}_s - \bar{T}_o}$$

$$\text{Therefore } K = -\frac{1}{L1} \ln \frac{(\bar{T}_s - \bar{T}_b)}{\bar{T}_s - \bar{T}_o}$$

$$K = \frac{2\pi\tau\, U_L}{M\, C_p}$$

$$U_L = \frac{K\, \bar{M}.C_p}{2\pi\tau}$$

Also for the steady-state equation of (14.A)b

$$\bar{M}\, L_v = 2\pi\tau\, U_B\, (L - L_1)\, (\bar{T}_s - \bar{T}_b)$$

$$\text{Therefore } U_b = \frac{M\, L_v}{2\pi\tau\, (L - L_1)\, (\bar{T}_s - \bar{T}_b)}$$

Hence knowing the subcooled length, L_1, we can evaluate U_L and U_B.

Summary of the Parameters of the Climbing Film Evaporator

PARAMETERS r, L, ρ, C_p, $\bar{P}a$

$$\bar{T}_s = 71.204 + 2.1 * \bar{P}_s - 0.015 * P_s^{-2}$$

$$\bar{L}_c = 2501.64 - 2.407 \cdot T_s$$

$$\bar{L}_v = 2501.64 - 2.407 \cdot T_b$$

$$\bar{v} = \bar{M}/\rho.\pi\tau^2 \qquad \text{liquid velocity}$$

$$L_1 = \bar{M}.C_p.(\bar{T}_b - \bar{T}_o) \cdot L.S.L_c \qquad \text{subcool length}$$

$$\tau = L1/\bar{v}$$

$$U_L = \frac{L_c\, L_n \dfrac{\bar{T}_s - \bar{T}_o}{\bar{T}_s - \bar{T}_b}}{2\pi r(T_b - T_o) \cdot L} \cdot S$$

$$K = 2\pi\tau U_L/MC_p \qquad \text{(not a fn of } r\text{)}$$

$$K_v = 2\, U_L/\rho r C_p$$

$$g_{ss} = e^{-KL1} \qquad \text{(not a fn of } r\text{)}$$

$$U = (M - m)\, L_v/2\pi\tau(T_s - T_b)\,(L - L_1)$$

$$U_B = 2\pi\tau U_b\,(L - L_1)/L_c$$

$$G_{FE} = So/2\,(\bar{P}_s - \bar{P}a) + U_B$$

$$T_{FE} = 16.942 \times 10^{-4}/(G_{FE})! \qquad \text{The effect of the tube metal is included in } 16.942 \times 10^{-4}$$

$$T_b = (sL_c - (M - m)L_v/M.C_p + T_o$$

DATA P_s, T_b, T_o, s, M, m.

P_s in Absolute units of Pressure

APPENDIX (B)

Parameter Evaluations for Simulation

$$\frac{\overset{*}{Mv}}{\overset{*}{S}} = \frac{a_1 s + a_2 e^{-\tau s}}{b_1 s^2 + b_2 s + b3} \cdot \frac{1}{(T_2 S + 1)}$$

T_2 is the time constant of the orifice plate.

$a_1 = U_B$, $a_2 a = UB \cdot Kv \cdot gss$

$b_1 = G_{FE} T_{FE}$ $b_2 = G_{FE} (1 + T_{FE} Kv)$ $b_3 = Kv \cdot G_{FE}$

$$\frac{\overset{*}{MV}}{\overset{*}{S}} = \frac{a_1 s + a_2 e^{-\tau s}}{b_1 T_2 S^3 + (b_2 T_2 + b_1) S^2 + (b_3 T_2 + b_2) S + b_3}$$

$$\frac{\overset{*}{T_b}}{\overset{*}{S}} = \frac{a_3 - a_4 e^{-\tau s}}{b_1 T_1 S^3 + (b_2 T_1 + b_1) S^2 + (B_3 T_1 + b_2) S + b_3}$$

$a_3 = Kv$, $a_4 = Kv.Gss$.

PADE APPROX

Expand $e^{-\tau s}$ $\quad \dfrac{2-S\tau}{2+S\tau}$

$$\frac{M^*}{S^4} = \frac{a_1\tau\, S^2 + (2a_1 - a_2\tau)S + 2a_2}{b_1T_2\tau s^4+(b_2T_2+b_1+2b_1T_2)S^3+(b_3T_2+b_2+2b_2T_2+2b_1)+S^2+(b_3\tau+2b_3T_2+2b_2)S+2b_3}$$

$$\frac{T_b}{S^*} \quad \frac{\tau(a_3 - a_4)s + 2(a_3 - a_4)}{b_1T_1\tau S^4+(b_2T_1+b_1+2b_1T_1)S^3+(b_3T_1+b_2+2b_2T_1+2b_1)S^2+(b_3\tau+2b_3T_1+2b_3)S+2b_3}$$

$$\frac{\overset{*}{T}_b}{\overset{*}{M}} \quad \frac{(a_5 S^2 + a_6 S + a_7)e^{-\tau s} - (a_5 S^2 + a_a S + a7)}{b_4 S^3 + b_5 S^2 + b_6 S} \cdot \frac{1}{(T_1 S + 1)}$$

$a_5 = g_1 \cdot T_{FE}$

$a_6 = g_1 \cdot (1 + Kv\ T_{FE} + g_2 \cdot g_{ss}/g_1)$

$a_7 = g_1 \cdot Kv$

$a_8 = g_1 \cdot T_{FE} = a_5$

$a_9 = g_1\ (1 + Kv \cdot T_{FE} + g_2/g_1)$

$a_{10} = g_1 \cdot Kv = a_7$

$b_4 = M\ L_c\ G_{FE}\ T_{FE}$

$b_5 = M \cdot G_{FE}\ (1 + T_{FE} \cdot Kv) L_c$

$b_6 = M\ L_c \cdot Kv \cdot G_{FE}$

$g_1 = Kv\ (T_s - T_o)\ g_{ss}\ L_c \cdot G_{FE}$

$g_2 = C_p\ (T_b - T_o)\ Kv \cdot M$

$$\frac{\overset{*}{T}_b}{\overset{*}{M}} \quad \frac{(a_5 S^2 + a_6 S + a_7)e^{-\tau s} - (a_5 S^2 + a_a S + a_7)}{b_4 T_1 S^4 + (b_5 T_1 + b_4) S^3 + (b_6 T_1 + b_5) S^2 + b_6 + 0.0}$$

PADE' APPROX

$$e^{-\tau s} = \frac{2 - \tau s}{2 + \tau s}$$

$$\frac{\overset{*}{T}b}{\overset{*}{M}} = \frac{-2\tau a_5 S^3 - (\tau a_6 + \tau a_9) S^2 + s(a_6 - a_9 - a_7\tau)S}{\mu b_4 T_1 \tau S^5 + \tau(b_5 T_1 b_4 + \frac{2b_4 T_1}{\tau})S^4 + (\tau b_6 T_1 + \tau b_5 + 2b_5 T_1 + 2b_4)S^3 + (\tau b_6 + 2b_6 T_1 + 2b_5)S}$$

divide by s

for vapour and liquid product.

for vapour $\overset{*}{MV}$

$$\frac{\overset{*}{M}v}{\overset{*}{M}} = \frac{(a_{11}S^2 + a_{12} S + a_{13}) - (a_{14} S^2 + a_{15} + S + a_{16})e^{-\tau s}}{b_4 S^3 + b_5 S^2 + b_6 S}$$

$$a_{11} = g_3 (T_{FE} - g_4/g_3)$$

$$a_{12} = g_s (T_{FE} \cdot Kv + 1)$$

$$a_{13} = g_3 \cdot Kv$$

$$a_{14} = g_3 \cdot T_{FE}$$

$$a_{15} = g_3 \cdot (T_{FE} \cdot Kv + 1 + g_4 g_{ss} Kv/g_3)$$

$$a_{16} = g_3 \cdot Kv = a_{13}$$

$$g_3 = UB \cdot Kv (T_s - T_o) g_{ss} L_c G_{FE}$$

$$g_4 = M C_p \cdot U_B (T_b - T_o)$$

Now $\overset{*}{m} = \overset{*}{M} - \overset{*}{Mv}$ (water - vapour)

divide by $\overset{*}{M}$

$$\frac{\overset{*}{m}}{\overset{*}{M}} = 1 - \frac{\overset{*}{MV}}{\overset{*}{M}} \text{ , now substitute for } \overset{*}{M}/\overset{*}{M}$$

$$\frac{\overset{*}{m}}{\overset{*}{M}} = \frac{b_4 S^3 + b_5 S^2 + b_6 S - (a_{11} S^2 + a_{12} S + a_{13}) + (a_{14} S^2 + a_{15} S + a_{16}) e^{-\tau s}}{b_4 S^3 + b_5 S^2 + b_6 S} \cdot \frac{1}{(T_2 S + 1)}$$

APPENDIX C

THERMODYNAMICS CORRELATIONS

Correlation between pressure and temperature of saturated water vapour is obtained using data in µPerryß.

$$T = 71.204 + 2.1\,P - 0.015\,P^2 \qquad \text{for } 10 \leq P \leq 50 \text{ psia}$$

$$T = 24\,\mathrm{Ln}P + 32.87 \qquad 0 \leq P \leq 10 \text{ psia}$$

where P is in Psia and T in ^{o}C.

For vapour mass density

$$\rho = (41.193 + 37.091P)10^{-3}$$

P is in Psia

ρ is in kg/m^3

Latent heat of vaporisation at temp $T^{o}C$

$$L_v = 2501.64 - 2.407 \cdot T$$

Also we have assumed that the latent heat of vaporisation is equal to the condensation heat.

APPENDIX D

FLOW THROUGH ORIFICE PLATES

$$W = K\ Y\ A_2 \sqrt{2g_c\ (P_1 - P_2)\ \rho_1}$$

A_2 = cross sectional area of throat sq.ft.

C = coefficient of discharge $0.6 \leqslant C \leqslant 0.7$

g_c = 32.17

$K = C / \sqrt{1 - \beta^4}$

P_1 = Upstream pressure lb force/sq ft

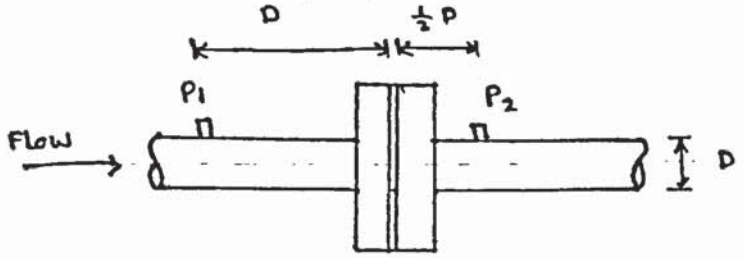
P_2 = Downstream pressure lb force/sq ft

W = lb/s

$\beta = \frac{D1}{D} = \frac{\text{Throat Dia}}{\text{Pipe Dia}} < 1$

Y = 1 for liquids $0.9 \leqslant Y < 1$
expansion factor for gases

ρ = lb/cu ft density upstream



In our special case we take

$Y \approx 0.92$ or 0.95 $W = 0.0217 \sqrt{\Delta p}$

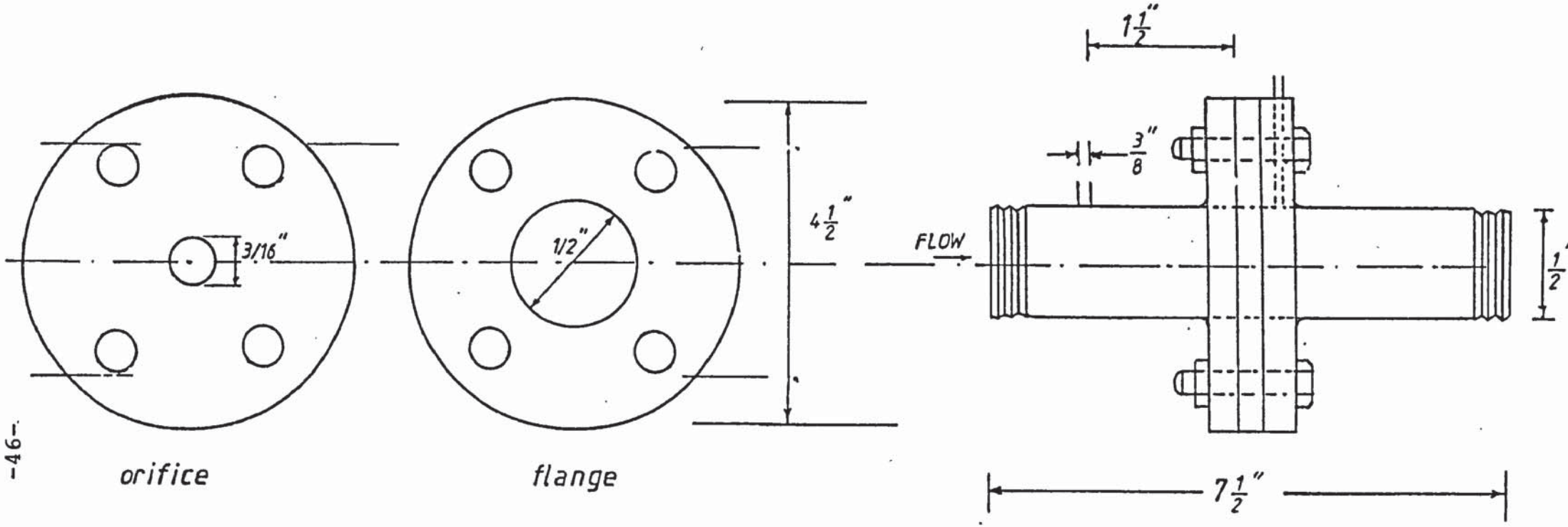
$C \approx 0.62$ 0.92

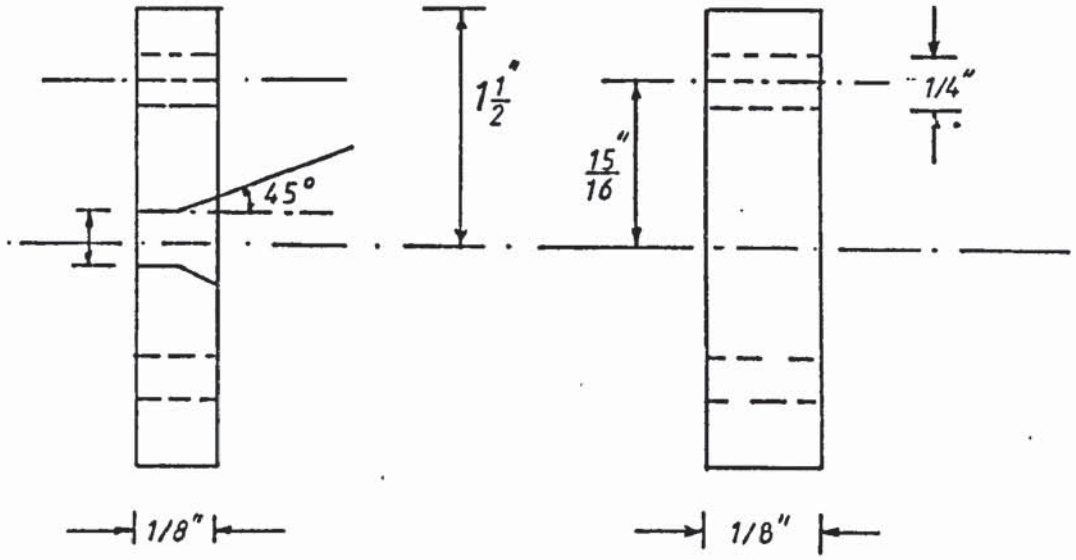
$$W = 0.92 \times 0.63 \times \frac{0.352 \times 2.73}{12 \times 2.2} \sqrt{\Delta p} \quad \text{kg/s}$$

Δp in psi

$$W = 0.021 \sqrt{\Delta p} \quad \text{kg/s and } \Delta p \text{ in psi}$$

$$W_{max} = 0.0245 \sqrt{\Delta p}$$





Orifice plate to measure the liquid flow-rate from the cyclone

APPENDIX E

DYNSIM

PPLOT1

ST5112

STEADY

PARAM2

RS4212

RS5121

```
**********
**********   Pdn =   7          2299-MNI          Terminal: 56          23 MAR 84 15:46:
**********
```

LISTING OF 2299MNI *DYNSIM LAST UPDATED ON 18-OCT-82 AT 19:02:46

```
      INTEGER   DT
      REAL BIAS,YES,CL,FIN
      REAL XC(1600),YC(1600)
      REAL D1(5,1600)
      REAL A(8),U,K(8),K2(8),T,YO,A2(8),Y(8)
111     WRITE(3,50)
50          FORMAT(23H INPUT NO. OF STATES  N)
      READ(3 ,*) N
      WRITE(3,100)
100     FORMAT(22H INPUT TIME DELAY  DT )
       READ (3 ,*) DT
      WRITE(3,200)
200   FORMAT(31H INPUT INPUT SIGNAL MAGNIT.   U)
      READ(3 ,*) U
      WRITE(3,250)
250   FORMAT(25H INPUT STEADY-STATE  BIAS)
      READ(3 ,*) BIAS
      WRITE(3,300)
300   FORMAT(32H INPUT INTEGRATION STEP SIZE  ST)
      READ(3 ,*)   ST
      WRITE(3,400)
400   FORMAT(31H INPUT FEEDBACK GAINS  A MATRIX)
      READ(3 ,*)(A(L),L=1,N)
      WRITE(3,500)
500     FORMAT(31H INPUT OUTPUT GAINS  MATRIX A2 )
      READ(3 ,*)(A2(L),L=1,N)
      WRITE(3,600)
600     FORMAT(29H INPUT DELAY FEEDBACK GAINS K)
      READ(3 ,*)(K(L),L=1,N)
      WRITE(3,700)
700     FORMAT(28H INPUT DELAY OUTPUT GAINS K2)
     READ(3 ,*)(K2(L),L=1,N)
      WRITE(3,800)
800     FORMAT('INPUT NH HORIZN. SCALE &NV VERT.  SCALE')
      READ(3 ,*) NH
      DO 10 L=1,N
         Y(L)=0.0
10     CONTINUE
      DO 30 LL=1,5
        DO 80 L2=1,1600
         D1(LL,L2)=0.0
80      CONTINUE
30    CONTINUE
      DT=DT/ST
       YO=0.0
      T=0.0
      DO 60 L=1,NH
            IL=L
        CALL STEINT(U,ST,N,Y,A,A2,K,K2,YO,IL,DT,D1)
         T=T+ST


            XC(L)= T
          YC(L)=YO+BIAS
60      CONTINUE
       WRITE (40,*) (YC(L),L=1,NH)
       WRITE (3,1000)
1000   FORMAT ( ' END OF CALCULATION ' )
     STOP
     END
      SUBROUTINE STEINT(S,H,M,X,AA,AA2,KK,KK2,XO,IE,TD,DD)
       INTEGER B,M,TD
       REAL S,XO,X1,X(M),H,AA(M)
       REAL AA2(M),KK(M),KK2(M)
     REAL    DD(5,1600)
      X1=0.0
      B=IE+TD
      DO 40 I=1,M
          X1=X1+AA(I)*X(I)+KK(I)*DD(I,IE)
40     CONTINUE
      X1=X1+S
      X(M)=X(M)+X1*H-.5*H*X1*H
      DD(M,B)=X(M)
      XO1=0.0
      IF(M.EQ.1)  GOTO  70
      DO 20 J=1,M-1
        X(M-J)=X(M-J)+H*X(M-J+1)-.5*H*X(M-J+1)*H
        DD(M-J,B)=X(M-J)
        XO1=XO1+AA2(J)*X(J)+KK2(J)*DD(J,IE)
20      CONTINUE
70    XO=XO1+ AA2(M)*X(M)+KK2(M)*DD(M,IE)
      RETURN
      END
```

```
**********
**********
**********
```

```
**********
**********   Pón =   7          2299-MNI          Terminal: 56          23 MAR 84 15:47:
**********
```

LISTING OF 2299MNI *PPLOT1 LAST UPDATED ON 17-MAR-84 AT 11:35:54

```
 1          DIMENSION  TEM(122),TIM(122)
 2          INTEGER    DT
 3          REAL BIAS,CL,FIN
 4          REAL XG(1600),YG(1600)
 5          REAL D1(5,1600)
 6          REAL GK(8),GK2(8)
 7          REAL A(8),A2(8),Y(8)
 8          REAL U, YO
 9            REAL T
10            DIMENSION IRR(16),IR2(16)
11   4      TIM(1)=0.0
12            READ(25,2000) IRR(1)
13  *         READ (25,1010) IR2
14            KPIC=0
15          DO 10 I=1,121
16             READ(25,1000) TEM(I)
17            TIM(I+1)=10.+TIM(I)
18    10    CONTINUE
19   1000    FORMAT(G12.6)
20  *1010   FORMAT (4A4)
21   2000     FORMAT(A3)
22  *       CALL APDS4
23          CALL T4010
24           E=TEM(3)-TEM(118)
25          CALL PICCLE
26           E=1.2*E
27           IF (E) 1,2,2
28    1      VBEG = TEM(3)+E
29           VEND = TEM(3)-E
30           GO TO 3
31    2      VBEG = TEM(118)-E
32           VEND = TEM(118)+E
33    3      CALL SHIFT2(40.,40.)
34           CALL AXIPOS(1,0.0,0.0,120.0,1)
35           CALL AXIPOS(1,0.0,0.0,60.0,2)
36           CALL AXISCA(3,30,0.0,1200.0,1)
37           CALL AXISCA(3,10,VBEG,VEND, 2)
38           CALL AXIDRA(1,1,1)
39           CALL AXIDRA(-1,-1,2)
40           CALL MOVTO2(50.0,-15.)
41           CALL CHAANG(0.0)
42           CALL CHASIZ(2.5,3.0)
43            CALL CHAMOD
44           CALL CHAHOL(14H TIME IN SEC*.)
45           CALL MOVTO2(-30.0,30.)
46           CALL CHAANG(90.)
47           CALL CHAARR(IR2,4,4)
48              CALL CHAANG(0.0)
49           CALL MOVTO2(0.0,0.0)
50           CALL GRAPOL(TIM,TEM,121)


51           CALL MOVTO2(60.0,20.0)
52           CALL LINTO2(61.5,20.0)
53           CALL CHAHOL(16H    EXPERIMENT*.)
54           CALL MOVTO2(60.0,75.0)
55           CALL CHAARR(IRR,1,3)
56             WRITE (3,3000)
57   3000      FORMAT(28H TO CHANGE VARIABLE ? 1 OR 0)
58             READ (3,*) CV
59             IF (CV.EQ.1) GOTO 4
60   1500    FORMAT(2I3)
61   111      WRITE(3,50)
62   50              FORMAT(23H INPUT NO. OF STATES  N)
63           READ(3,*) N
64           WRITE(3,100)
65   100       FORMAT(22H INPUT TIME DELAY  DT )
66           READ(3 ,*)  DT
67           WRITE(3,200)
68   200     FORMAT(31H INPUT INPUT SIGNAL MAGNIT.    U)
69           READ(3,*) U
70           WRITE(3,250)
71   250     FORMAT(25H INPUT STEADY-STATE  BIAS)
72           READ(3,*) BIAS
73           WRITE(3,300)
74   300     FORMAT(32H INPUT INTEGRATION STEP SIZE  ST)
75           READ(3,*)   ST
76           WRITE(3,400)
77   400     FORMAT(31H INPUT FEEDBACK GAINS  A MATRIX)
78           READ(3,*)(A(L),L=1,N)
79           WRITE(3,500)
80   500       FORMAT(31H INPUT OUTPUT GAINS  MATRIX A2 )
81           READ(3,*)(A2(L),L=1,N)
82           WRITE(3,600)
83   600       FORMAT(30H INPUT DELAY FEEDBACK GAINK GK)
84           READ(3,*)(GK(L),L=1,N)
85           WRITE(3,700)
```

```
      CALL MOVT02(60.0,20.0)
      CALL LINT02(61.5,20.0)
      CALL CHAHOL(16H    EXPERIMENT*.)
      CALL MOVT02(60.0,75.0)
      CALL CHAARR(IRR,1,3)
        WRITE (3,3000)
3000    FORMAT(28H TO CHANGE VARIABLE ? 1 OR 0)
        READ (3,*) CV
        IF (CV.EQ.1) GOTO 4
1500  FORMAT(2I3)
111    WRITE(3,50)
50           FORMAT(23H INPUT NO. OF STATES  N)
      READ(3,*) N
      WRITE(3,100)
100     FORMAT(22H INPUT TIME DELAY  DT )
      READ(3 ,*)  DT
      WRITE(3,200)
200   FORMAT(31H INPUT INPUT SIGNAL MAGNIT.    U)
      READ(3,*) U
      WRITE(3,250)
250   FORMAT(25H INPUT STEADY-STATE  BIAS)
      READ(3,*) BIAS
      WRITE(3,300)
300   FORMAT(32H INPUT INTEGRATION STEP SIZE  ST)
      READ(3,*)   ST
      WRITE(3,400)
400   FORMAT(31H INPUT FEEDBACK GAINS  A MATRIX)
      READ(3,*)(A(L),L=1,N)
      WRITE(3,500)
500     FORMAT(31H INPUT OUTPUT GAINS  MATRIX A2 )
      READ(3,*)(A2(L),L=1,N)
      WRITE(3,600)
600     FORMAT(30H INPUT DELAY FEEDBACK GAINK GK)
      READ(3,*)(GK(L),L=1,N)
      WRITE(3,700)
700     FORMAT(29H INPUT DELAY OUTPUT GAINS GK2)
     READ(3,*)(GK2(L),L=1,N)
      WRITE(3,800)
800     FORMAT ('   INPUT NH  HORIZONTAL  SCALE ... ')
      READ(3,*) NH
       WRITE (3,810)
810    FORMAT ('  INPUT  START OF DATA PLOTTING  LSR ...')
       READ (3,*) LSR
      DO 120 L=1,N
          Y(L)=0.0
120     CONTINUE
      DO 30 LL=1,5
        DO 80 L2=1,1600
         D1(LL,L2)=0.0
80      CONTINUE
30    CONTINUE
      DT=DT/ST
       YO=0.0
      T=0.0
        FOR LT=1 , LSR
            YG(LT) =BIAS
        ENDFOR
      DO 60 L=1,NH
            IL=L
         CALL STEINT(U,ST,N,Y,A,A2,GK,GK2,YO,IL,DT,D1)
          T=T+ST
              XG(L)= T
          YG(L+LSR)=YO+BIAS
60      CONTINUE
       KPIC=KPIC+1
      CALL PICBEG (KPIC)
      CALL PENSEL(5,.7,3)
      CALL GRAPOL(XG,YG,1200)
      CALL MOVT02(60.0,10.0)
      CALL LINT02(61.5,10.0)
      CALL CHAHOL(11H    MODEL*.)
      CALL PENEND(5,.7,3)
      CALL PICEND
      WRITE(3,350)
350   FORMAT(23H IS GRAPH O.K.  ? 1 OR 0)
      READ(3,*) YES
        IF (YES.EQ.0) GOTO 111
       JJ = LSR
     FOR II=1,121
        WRITE (40,*) YG(JJ)
       JJ = JJ + 11
     END FOR !II!
        WRITE(3,450)
450     FORMAT(26H TO CLEAR PICTURE ? 1 OR 0)
      READ(3,*) CL
        IF (CL.EQ.0) GOTO 111
     CALL PICCLE
     WRITE(3,550)
550   FORMAT(18H FINISHED ? 1 OR 0)
      READ(3,*) FIN
      IF (FIN.EQ.0) GOTO 4
     CALL DEVEND
     STOP
     END
```

```
      DO 120 L=1,N
          Y(L)=0.0
  120   CONTINUE
      DO 30 LL=1,5
        DO 80 L2=1,1600
         D1(LL,L2)=0.0
   80   CONTINUE
   30 CONTINUE
      DT=DT/ST
       YO=0.0
      T=0.0
        FOR LT=1 , LSR
            YC(LT) =BIAS
        ENDFOR
      DO 60 L=1,NH
            IL=L
        CALL STEINT(U,ST,N,Y,A,A2,CK,CK2,YO,IL,DT,D1)
          T=T+ST
              XC(L)= T
          YC(L+LSR)=YO+BIAS
   60   CONTINUE
       KPIC=KPIC+1
      CALL PICBEG (KPIC)
      CALL PENSEL(5,.7,3)
      CALL CRAPOL(XC,YC,1200)
      CALL MOVTO2(60.0,10.0)
      CALL LINTO2(61.5,10.0)
      CALL CHAHOL(11H     MODEL*.)
      CALL PENEND(5,.7,3)
      CALL PICEND
      WRITE(3,350)
  350 FORMAT(23H IS GRAPH O.K.  ? 1 OR 0)
      READ(3,*) YES
        IF (YES.EQ.0) GOTO 111
       JJ = LSR
     FOR II=1,121
        WRITE (40,*) YC(JJ)
       JJ = JJ + 11
     END FOR !II!
        WRITE(3,450)
  450   FORMAT(26H TO CLEAR PICTURE ? 1 OR 0)
      READ(3,*) CL
        IF (CL.EQ.0) GOTO 111
     CALL PICCLE
     WRITE(3,550)
  550 FORMAT(18H FINISHED ? 1 OR 0)
      READ(3,*) FIN
      IF (FIN.EQ.0) GOTO 4
     CALL DEVEND
     STOP
     END
      SUBROUTINE STEINT(S,H,M,X,AA,AA2,KK,KK2,XO,IE,TD,DD)
       INTEGER B,M,TD
       REAL S,XO,X1,X(M),H,AA(M)
       REAL AA2(M),KK(M),KK2(M)
     REAL    DD(5,1600)
      X1=0.0
      B=IE+TD
      DO 40 I=1,M
          X1=X1+AA(I)*X(I)+KK(I)*DD(I,IE)
   40  CONTINUE
      X1=X1+S
      X(M)=X(M)+X1*H-.5*H*X1*H
      DD(M,B)=X(M)
      XO1=0.0
      IF(M.EQ.1)  GOTO  70
      DO 20 J=1,M-1
        X(M-J)=X(M-J)+H*X(M-J+1)-.5*H*X(M-J+1)*H
        DD(M-J,B)=X(M-J)
        XO1=XO1+AA2(J)*X(J)+KK2(J)*DD(J,IE)
   20   CONTINUE
   70 XO=XO1+ AA2(M)*X(M)+KK2(M)*DD(M,IE)
      END
```

```
**********
**********   Pdn =   7          2299-MNI          Terminal: 56          23 MAR 84 15:46:4
**********
```

LISTING OF 2299MNI *ST5112 LAST UPDATED ON 27-APR-82 AT 16:36:02

1	97.6838
2	80.7427
3	100.779
4	104.772
5	105.309
6	99.9412
7	13.5221
8	31.2500
9	30.5956
10	12.8897
11	0.03178
12	0.04186
13	0.05090
14	0.49465
15	0.03955
16	19.9169

```
**********
**********   No. of pages    1   2299-MNI          Terminal: 56          23 MAR 84 15:46:5
**********
```

```
**********
**********   Pdn =   7          2299-MNI          Terminal: 56          23 MAR 84 15:46
**********
```

LISTING OF 2299MNI *STEADY LAST UPDATED ON 13-JAN-83 AT 12:05:27

```
      REAL  TS,PS,LC,LV,TB,VF,M
      REAL  WD,PI,RF,L1,CP,TO,LF
      REAL  S,TDF,ULF,KF,KVF,GSF
      REAL  UB1,ML,UB,GFE,TFE,TEI
      REAL  TBP,TEO,TI,TCSO,TCO
      REAL  TCI,PB,PE,MV,TBC
C      PREHEATER PARAMETERS
      REAL  LVP,PPB,MPL,WP,ALPH,CPP
      REAL  TPP,GP1,GP2,GP3,GP4,VP
      REAL  LP,RP,TDP,ULP,KP,KVP,GSP
C        SECOND EFFECT PARAMETERS
      REAL  LVE,TEB,ME,MPV,ALPHE,RE
      REAL  TSE,GE1,GE2,GE3,VE,ULE
      REAL  TDE,LE,KE,KVE,GSE
       DATA WD,CP,PI,LF/985.22,4.1868,3.142,2.59/
       DATA RF/0.0412/
       DATA WP,CPP,LP,RP/4.9102,.38494,1.416,.04275/
       DATA RE,LE/.0945,1.065/
C        CALCULATION OF PROCESS CONSTANTS
       READ(20,100) TB
       READ(20,100) TO
       READ(20,100) TEI
       READ(20,100) TBP
       READ(20,100) TBC
       READ(20,100) TEO
       READ(20,100) TI
       READ(20,100) TCSO
       READ(20,100) TCO
       READ(20,100) TCI
       READ(20,100) S
       READ(20,100) ML
       READ(20,100) M
       READ(20,100) PB
       READ(20,100) PE
       READ(20,100) PS
 100    FORMAT(G12.6)
C      DATA FORMAT IN STEADY-STATE DATA FILES IS IN
C      THE ABOVE FORMAT.
C  THIS IS TO CHANGE PRESS.  INTO ABSOLUT
       PS=PS+14.69
       PB = PB+14.69
       PE = PE+14.69
       TS=71.204+2.1*PS-0.015*PS**2.
       LC=2501.64-2.407*TS
       LV=2501.64-2.407*TB
       VF=M/(WD*PI*RF**2.)
       L1=M*CP*(TB-TO)*LF/(S*LC)
       TDF=L1/VF
       ULF=LC*ALOG((TS-TO)/(TS-TB))*S/(2.
     1 *PI*RF*LF*(TB-TO))


       KF=2.*PI*RF*ULF/(M*CP)
       KVF=2.*ULF/(WD*RF*CP)
       GSF=EXP(-KF*L1)
       UB1=(M-ML)*LV/(2.*PI*RF*(TS-TB)*(LF-L1))
       UB=2.*PI*RF*UB1*(LF-L1)/LC
       GFE=S/(2.*(PS-14.69))+UB
       TFE =16.942E-04/GFE
       MV=M-ML
C      THE FOLLOWING CAL. IS FOR THE PREHEATER
C      AVERAGE RADII IS USED.
       LVP=2501.64-2.407*TBP
       MPL=M*CP*(TO-TI)/LVP
       PB=(2.1-SQRT(2.1**2.-4.*(TBC-71.204)*.015))/(2.*.015)
      PPB=(2.1-SQRT(2.1**2.-4.*(TBP-71.204)*.015))/(2.*.015)
       ALPH=2.885E-04+3.9693/LVP
      TPP=MV/(2.*(PB-PPB)*ALPH)
      GP1=MV/(2.*ALPH*(PB-PPB))
      GP2=1./ALPH
      GP3=CP*(TO-TI)/(LVP*ALPH)
      GP4=M*CP/(LVP*ALPH)
       VP=M/(WD*PI*RP**2.)
       TDP=LP/VP
       ULP=M*CP*ALOG((TBP-TI)/(TBP-TO))/(LP*2.*PI*RP)
       KP=2.*PI*RP*ULP/(M*CP)
      KVP=2.*ULP/(WD*RP*CP)
      GSP=EXP(-KP*LP)
C
C      THE FOLLOWING CAL.  IS FOR THE SEC.  EFFECT
C
       TEB=71.204+2.1*PE-.015*PE**2.
       LVE=2501.64-2.407*TEB
       MPV=MV-MPL
       ME =MPV*LVE/(CP*(TEO-TEI))
       ALPHE=5.6001E-04+1.2085/LVE
       TSE=MPV/(2.*(PPB-PE)*ALPHE)
```

```
      PB = PB+14.69
      PE = PE+14.69
      TS=71.204+2.1*PS-0.015*PS**2.
      LC=2501.64-2.407*TS
      LV=2501.64-2.407*TB
      VF=M/(WD*PI*RF**2.)
      L1=M*CP*(TB-TO)*LF/(S*LC)
      TDF=L1/VF
      ULF=LC*ALOG((TS-TO)/(TS-TB))*S/(2.
     1*PI*RF*LF*(TB-TO))


      KF=2.*PI*RF*ULF/(M*CP)
      KVF=2.*ULF/(WD*RF*CP)
      GSF=EXP(-KF*L1)
      UB1=(M-ML)*LV/(2.*PI*RF*(TS-TB)*(LF-L1))
      UB=2.*PI*RF*UB1*(LF-L1)/LC
      GFE=S/(2.*(PS-14.69))+UB
      TFE =16.942E-04/GFE
      MV=M-ML
C     THE FOLLOWING CAL. IS FOR THE PREHEATER
C     AVERAGE RADII IS USED.
      LVP=2501.64-2.407*TBP
      MPL=M*CP*(TO-TI)/LVP
      PB=(2.1-SQRT(2.1**2.-4.*(TBC-71.204)*.015))/(2.*.015)
     PPB=(2.1-SQRT(2.1**2.-4.*(TBP-71.204)*.015))/(2.*.015)
      ALPH=2.885E-04+3.9693/LVP
     TPP=MV/(2.*(PB-PPB)*ALPH)
     GP1=MV/(2.*ALPH*(PB-PPB))
     GP2=1./ALPH
     GP3=CP*(TO-TI)/(LVP*ALPH)
     GP4=M*CP/(LVP*ALPH)
      VP=M/(WD*PI*RP**2.)
      TDP=LP/VP
      ULP=M*CP*ALOG((TBP-TI)/(TBP-TO))/(LP*2.*PI*RP)
      KP=2.*PI*RP*ULP/(M*CP)
     KVP=2.*ULP/(WD*RP*CP)
     GSP=EXP(-KP*LP)
C
C     THE FOLLOWING CAL. IS FOR THE SEC. EFFECT
C
      TEB=71.204+2.1*PE-.015*PE**2.
      LVE=2501.64-2.407*TEB
      MPV=MV-MPL
      ME =MPV*LVE/(CP*(TEO-TEI))
      ALPHE=5.6001E-04+1.2085/LVE
      TSE=MPV/(2.*(PPB-PE)*ALPHE)
      GE1 =MPV/(2.*ALPHE*(PPB-PE))
      GE2=CP*(TEO-TEI)/(ALPHE*LVE)
      GE3=ME*CP/(ALPHE*LVE)
      VE=ME/(WD*PI*RE**2.)
      TDE=LE/VE
C     ULE=ME*CP*ALOG((TEB-TEI)/(TEB-TEO))/(LE*2.*PI*RE)
      KE=2.*PI*RE*ULE/(ME*CP)
      KVE=2.*ULE/(WD*RE*CP)
      GSE=EXP(-KE*LE)
       WRITE (3,991)
 991   FORMAT (' INPUT 1 FOR FE PARAM ELSE PREHEATER ')
       READ (3,*) NU
       IF (NU.EQ.1) THEN
      WRITE(25,100) UB,KVF,GSF,TDF
      WRITE(25,100) GFE,TFE,LC,LV
      WRITE(25,100) TS,TB,TO,M,ML,S
       ELSE
       WRITE (3,990)
 990   FORMAT ('    PREHEATER   PARAMETER ..... ')
       WRITE (25,*) GSP,KVP,TDP,GP3,GP4
      WRITE (3,*) MV,MPV,TPP,ALPH,PPB,PB,ML
       END IF
      STOP
       END
```

LISTING OF 2299MNI *PARAM2 LAST UPDATED ON 18-OCT-82 AT 19:29:45

```
C   PROGRAM TO READ DATA FROM RSXXXX FILES .
C   THESE FILES (RSXXXX) CONTAIN THE RESULTS
C   FROM THE STEADY-STATE CALCULATION BY
C   PROGRAM "STEADY" . THE LAST INTEGER IDENTIFY
C   THE RESPONSE TYPE . STEAM OR FEED PERTURBATION.
C   ICHS .
C  ***********************************************
C  ***********************************************
      REAL  UB, KV, GSS, TD, GFE, TFE, LC, LV
      REAL  TS, TB, TO, M, ML, S, T2, T1
      REAL A(6), A2(6), K1(6), K2(6)
      INTEGER  ICHS, IPA, ITP
      READ (30,*) UB, KV, GSS, TD
      READ (30,*) GFE, TFE, LC, LV
      READ (30,*) TS, TB, TO, M, ML, S
       READ (30,*) ICHS
      WRITE (3,200)
 200  FORMAT ('INPUT  INSTRUMENT TIME CONSTANT  T1 & T2 ')
      READ (3,*)  T1,T2
        CP=4.1868
       G1 = KV*(TS-TO)*GSS*LC*GFE
       G2 = CP*(TB-TO)*KV*M
       G3 = UB*KV*(TS-TO)*GSS*LC*GFE
       G4 = M*CP*UB*(TB-TO)
C ***********
       A1 = UB
       A2A = UB*KV*GSS
       A3 = KV
       A4 = KV*GSS
       A5 = G1*TFE      !     = AB
       A6 = G1*(1. +KV*TFE+G2*GSS/G1)
       A7 = G1*KV    !    = A10
       A9 = G1*(1. +KV*TFE +G2/G1)
      A11 = G3*(TFE-G4/G3)
      A12 = G3*(TFE*KV + 1. )
      A13 = G3*KV     !    = A16
      A14 = G3*TFE
      A15 = G3* (TFE *KV+1.  +G4*GSS*KV/G3)
C    ***********
       B1 = GFE*TFE
       B2 = GFE *(1. +TFE*KV)
       B3 = KV*GFE
       B4 = M*LC*GFE*TFE
       B5 = M*LC*GFE*(1. +TFE*KV)
       B6 = M*LC*KV*GFE
C   CHOOSE SYSTEM , 1=STEAM   2=FEED  FROM RSXXXX FILES
C      READ (30,*)    ICHS
       IF (ICHS.EQ.2)  GOTO 4
       WRITE ( 3,800)
 800   FORMAT( 'NOW THE SYSTEM IS IN STEAM ')


       WRITE (3,400)
 400   FORMAT( 'INPUT 1=PURE TIME DELAY, 2=PADE APPROX. ')
       READ (3,*) IPA
        IF (IPA.EQ.1) GO TO 1
C  NOW THE CALCULATION IS IN PADE APPROX. *****
       WRITE (3,500)
 500  FORMAT ('INPUT 1=TEMP.  TB ,  2=PRODUCT  ML ')
       DO 10  L= 1,4
          K1(L)=0.0
          K2(L)=0.0
  10   CONTINUE
       READ (3,*) ITP
      IF(ITP.EQ.1) THEN
        TT=T1
          ELSE
            TT=T2
        ENDIF
          PQ=B1*TT*TD
          A(1) = -1.*B3/PQ
        A(2) = -(B3*TD+1.*B3*TT+1.*B2)/PQ
        A(3) = -(B3*TT+B2+1.*B2*TT+1.*B1)/PQ
          A(4) = -(B2*TT+B1+1.*B1*TT)/PQ
        IF (ITP.EQ.2) GO TO 2
         A2(1) = 1.*(A3-A4)/PQ
         A2(2) = TD*(A3+A4)/PQ
         A2(3) = 0.0
         A2(4) = 0.0
        GO TO 2000
   2     A2(1) = 1.*A2A/PQ
         A2(2) = (1.*A1-A2A*TD)/PQ
         A2(3) = A1*TD/PQ
         A2(4) = 0.0
        GO TO 2000
C  ** CALCULATION NOW IS IN TIME DELAY *****
   1   DO 20  L = 1,  4
```

```
71            A(3) = -(B3*T1+B2+1 *B2*T1+A. *B1)/PQ
72              A(4) = -(B2*TT+B1+1. *B1*TT)/PQ
73            IF (ITP.EQ.2) GO TO 2
74             A2(1) = 1. *(A3-A4)/PQ
75             A2(2) = TD*(A3+A4)/PQ
76             A2(3) = 0.0
77             A2(4) = 0.0
78            GO TO 2000
79    2        A2(1) = 1. *A2A/PQ
80             A2(2) = (1. *A1-A2A*TD)/PQ
81             A2(3) = A1*TD/PQ
82             A2(4) = 0.0
83            GO TO 2000
84  C  ** CALCULATION NOW IS IN TIME DELAY *****
85    1   DO 20  L = 1,  4
86              K1(L) = 0.0
87    20  CONTINUE
88             WRITE (3,500)
89             READ (3,*) ITP
90          IF(ITP.EQ.1) THEN
91               TT=T1
92             ELSE
93               TT=T2
94           ENDIF
95              PQ = B1*TT
96           A(1) = 0.0
97          A(2) =  -B3/PQ
98          A(3) = -(B3*TT+B2)/PQ
99           A(4) = -(B2*TT+B1)/PQ
100          IF  (ITP.EQ.2)  GO TO 3
101           A2(1) = A3/PQ
102           A2(2) = 0.0
103           A2(3) = 0.0
104           K2(1) = -A4/PQ
105           K2(2) = 0.0
106           K2(3) = 0.0
107         GO TO 2000
108   3      A2(1) = M*B3/PQ
109          A2(2) = M*B2/PQ
110          A2(3) = (M*B1/10. -A1)/PQ
111          A2(4) = 0.0
112          K2(1) = 0.0
113          K2(2) = -A2A/PQ
114          K2(3) = 0.0
115          K2(4) = 0.0
116        GO TO 2000
117   4     WRITE (3,700)
118   700  FORMAT('NOW THE SYSTEM IS FEED RESPONSE  M ')
119           WRITE (3,400)
120           READ (3,*) IPA
121          IF (IPA.EQ.1)  GO TO 5
122 C  NOW THE CALCULATION  IS IN PADE APPROX.
123           DO 30 L = 1, 5
124             K1(L) = 0.0
125             K2(L) = 0.0
126   30   CONTINUE
127           WRITE (3,500)
128           READ  (3,*) ITP
129          IF(ITP.EQ.1) THEN
130              TT=T1
131            ELSE
132              TT=T2
133           ENDIF
134             PQ = B4*TT*TD
135           A(1) = 0.0
136           A(2) = -1. *B6/PQ
137           A(3) = -(TD*B6+1. *B6*TT+1. *B5)/PQ
138           A(4) = -(TD*B6*TT+TD*B5+1. *B5*TT+1. *B4)/PQ
139           A(5) = -(B5*TT+B4+1. *B4*TT/TD)/PQ
140         IF (ITP.EQ.2) GO TO 6
141           A2(1) = 0.0
142           A2(2) = 1. *(A6-A9-A7*TD)/PQ
143           A2(3) = -TD*(A6+A9)/PQ
144           A2(4) = -1. *TD*A5/PQ
145           A2(5) = 0.0
146              GO TO 2000
147    6      A2(1) = 0.0
148           A2(2) = 1. *(B6+A15-A12-TD*A13)/PQ
149           A2(3) = (TD*(B6-A12-A15)+1. *(B5+A14-A11))/PQ
150           A2(4) = TD*(B5-A11-A14+1. *B4/TD)/PQ
151           A2(5) = TD*B4/PQ
152           GO TO 2000
153 C  *** CALCULATION NOW IS IN PURE TIME DELAY *****
154   5     DO 40 L = 1,4
155           K1(L) = 0.0
156   40     CONTINUE
157           WRITE ( 3,500)
158           READ (3,* ) ITP
159           IF (ITP.EQ.1) THEN
160             TT = T1
161              ELSE
162                  TT = T2
163           ENDIF
164            PQ = B4*TT
165            A(1) = 0.0
166           A(2) = B6/PQ
              A(3) = (B4*TT+35)/PQ
```

```
155             K1(L) = 0.0
156     40      CONTINUE
157             WRITE ( 3,500)
158             READ (3,* ) ITP
159             IF (ITP.EQ.1) THEN
160               TT = T1
161                ELSE
162                     TT = T2
163             ENDIF
164              PG = B4*TT
165              A(1) = 0.0
166             A(2) = B6/PG
167              A(3) = (B6*TT+B5)/PG
168              A(4) = (B5*TT+B4)/PG
169             IF (ITP.EQ.2) GO TO 7
170              A2(1) = -A7/PG
171              A2(2) = -A9/PG
172              A2(3) = -A5/PG
173              A2(4) = 0.0
174              K2(1) = A7/PG
175              K2(2) = A6/PG
176              K2(3) = A5/PG
177              K2(4) = 0.0
178             GO TO 2000
179     7        A2(1) = -A13/PG
180              A2(2) = (B6-A12)/PG
181              A2(3) = (B5-A11) /PG
182              A2(4) = B4/PG
183              K2(1) = A13/PG
184              K2(2) = A15/PG
185              K2(3) = A14/PG
186              K2(4) = 0.0
187     2000     WRITE (3,900)
188     900     FORMAT (' END OF CALCULATION ')
189              WRITE (40,*) ( A(I) , I=1,L)
190              WRITE (40,*) ( A2(I) , I=1,L)
191              WRITE (40,*) (K2(I) , I=1,L)
192              WRITE (40,*) ICHS, IPA, ITP
193              WRITE (40,*) T1, T2
194             END
```

****
********** No. of pages 4 2299-MNI Terminal: 56 23 MAR 84 15:50:01
****

```
**********
**********  Pdn =   7          2299-MNI          Terminal: 56          23 MAR 84 15:50:
**********
```

LISTING OF 2299MNI *RS4212 LAST UPDATED ON 14-DEC-82 AT 20:08:41

1	0.916368E-03	u_8
2	0.193214E-01	k_V
3	0.644437	ζ_s
4	22.7405	T_b
5	0.135931E-02	ζ_{FE}
6	1.24618	T_{FE}
7	2189.68	L_C
8	2269.36	L_V
9	129.604	T_s
10	96.5000	T_b
11	78.2353	T_o
12	0.715000E-01	M
13	0.422300E-01	m
14	0.209000E-01	Σ
15	1	

```
**********
**********  No. of pages     1   2299-MNI          Terminal: 56          23 MAR 84 15:50:
**********
```

```
**********************************************************************************************************
**********************************************************************************************************
***************
***************   Pdn =   7          2299-MNI          Terminal: 56              23 MAR 84 15:50:07
***************
**********************************************************************************************************
**********************************************************************************************************

 LISTING OF 2299MNI *RS5212   LAST UPDATED ON 14-OCT-82 AT 16:06:25

    1  0.549904E-03        uB
    2  0.200939E-01        Kv
    3  0.657688            Gss
    4   20.8534            nf
    5  0.977799E-03        GFe
    6   1.73267            TFE
    7   2182.10            Lc
    8   2267 71            Lv
    9   132 755            Ts
   10   97.1883            Tb
   11   78.6765            To
   12  0.684000E-01        M1
   13  0.495800E-01        M
   14  0.231800E-01        δ
   15   2

**********************************************************************************************************
**********************************************************************************************************
***************
***************   No. of pages     1   2299-MNI          Terminal: 56              23 MAR 84 15:50:10
***************
**********************************************************************************************************
**********************************************************************************************************
```

APPENDIX F

THE M6800 EXECUTIVE

As with the HADIOS Executive, the M6800 can operate either in the ON-line or OFF-line mode. The user HADIOS parameter and storage values take the same significance with arrays used are A (dimensioned 11) and B (dimensioned 86). But note the following differences:

1 - Array A must be an integer array. For array B, only the elements corresponding to the channels scanned (B(A(3)) - B(A(4)) and H316 data transferred (B(57)-B(57+M-1)) must remain in integer format.

2 - The control outputs must be converted into integer format first (using the INT function) before conveying them to the M6800 Executive.

3 - The counter contents used by the M6800 (potentially B(49), B(52), B(55)) are scanned as multiples of the RTC clock resolution (20 ms). The user must convert these to the required units through program.

As in the HADIOS Executive Package, there are basically four subroutines. However, a call to a house-keeping sub-

routine (CALL SUBO) must be made first. This prints out the high SD BASIC address (in hexidecimal) and the high address for the entire program segment (SD BASIC and M6800 Executive). This allows the programmer to examine data locations which by default (the data storage area can be assigned elsewhere via the DATA ORIGIN statement (begin at the top of the SD-BASIC Program segment.

SUBO also requests the user to specify which mode he wants the M6800 Executive to operate in. The M6800 may not independently access a process plant but can be programmed to engage in asynchronous data transfer operations as part of a distributed computing activity between the two processors.

CALL SUBI (A(0), B(0))

This is the basic scanning subroutine. It is similar to call (1, A(0), B(0)) in the H316 except that SUBI is called once every scan.

CALL SUB2

This subroutine call is similar to Call(2) in the H316, to exit the scanning mode if the number of scans required is reached.

CALL SUB3 (N, U)

This subroutine is similar to CALL (3, N, U) in the H316 except that in the M6800, N and U must represent integer values.

U = 0 to 32767 corresponding to 0 - 10 volts.

N = output channel number of DGOA, (0 - 15).

N = 1 steam valve

N = 2 infeed valve

CALL SUB4 (M, D(0))

This subroutine transfers M consecutive data words (two bytes each) located in integer array D, which is dimensioned 29. The first CB2 output in the sequence transfers to code byte (254) which prepares the HADIOS Executive for the subsequent data transfer. Like all data transfers to the H316, the CBN2 interrupt is used to inform the H316 that a data byte is available and the H316 acknowledges receipt via the CB1.

APPENDIX F

ERROR HANDLING IN M6800 EXECUTIVE

As far as the M6800 user is concerned, the error conditions may encompass the following:

1 - Initialisation errors detected by the M6800 Executive.

2 - SD BASIC Run Time Package (RTP) errors.

3 - FORTRAN errors in the library routine used by the M6800 interrupt response of the HADIOS Executive.

4 - A user interference in the NMI button.

Run Time (SD-BASIC) errors are channelled to the M6800 Executive SUB99 (EN, LN) where EN and LN are assigned the error and line (last number encountered) numbers respectively, and the SD-BASIC "ON ERROR GOTO" facility. Run-time errors in the HADIOS Executive which concerns the M6800 include the specifying of a counter currently being used by the H316 user and FORTRAN library routine errors. These errors are only passed on the M6800 Executive for recognition at the next scanning interrupt.

User interference must be via the NMI button once the M6800-H316 protocol has started. SD-BASIC can recover from input errors by responding with a "INPUT ERROR?" message, and ignores superfluous input data before the carriage return key is pressed. Each CTRL/0 (the CONTROL and 0 keys together) can also be used to "delete" a previous character input.

APPENDIX F

HADIOS SUBROUTINE

The user communicates with HADIOS by calls from his BASIC Program as follows:

CALL (1, A(0), B(0))

Subroutine 1 is the basic scanning routine. Arrays A (dimensioned 13) and B (dimensioned 86) contain the user's parameters and variables.

A - Array

A(0) Scanning interval in second

A(1) Devices required

=1 Analogue input

=2 Counter 1

=4 Counter 2

=8 Counter 3

The required set of devices is selected by setting A(1) equal to the sum of the appropriate values above.

A(2) Number of scans required, including the initial scan at time zero.

A(3) First analogue channel to be scanned.

A(4) Last analogue channel to be scanned. The computer will scan all the channels between A(3) and A(4) inclusively.

A(5) A delay element in the analogue inputs sampling code. The specified analogue channels (A(3) - A(4)) are scanned as many times as possible over a period of one H316 clock resolution (20 ms) and the cumulative sum for each channel is divided by that ensemble number. The smaller A(5) is, the smaller the delay after each multiplexing operation and therefore the larger the ensemble number. This averages out the noise signals giving the effect of a simple filter.

A(5)=33 is a typical value and zero is not recommended.

A(6) Counter 1 scan type.

=0 Non-interrupt mode

=1 Counter interrupts enabled.

A(7) Counter 1 preset value (0 - 255)

A(8) Counter 2 scan type

=0 Non-interrupt mode

=1 Counter interrupt enabled.

A(9) Counter 2 preset value (0 - 255)

A(10) Counter 3 type

=0 Non-interrupt mode

=1 Interrupts enabled.

A(11) Counter 3 preset value (0 - 255)

A(12) A Program flag in the HADIOS executive

=0 Normal, the H316 scans at its own clock frequency.

=1 The HADIOS devices come under exclusive control of the M6800 user. The H316 clock is not interrupt enabled and only the following elements of Array A need to be specified:

A(1) = integral multiple of M6800 sampling intervals.

A(2) = total number of scans required. The Executive (H316) also stores values of Analogue and/or Counter Inputs, (specified by the M6800 user) into the appropriate B array elements. The H316 BASIC Program can therefore access these values at the interval specified by A(1) and subject them to further mathematical treatment and/or graphics. In fact, the treated data can also be communicated to the M6800 via another call to subroutine 4.

A(13) A program flag in the HADIOS executive.

=0 Normal on-line with the plant only.

=1 The H316 is in OFF-line mode. The H316 is not clock interrupt driven for a plant scan but may communicate with the M6800 which may or may not be in OFF-line mode. If the M6800 is in OFF-line, then the H316 has to interrupt the M6800 through subroutine 4 for a H316 to M6800 data transfer. The subroutine of A(1) A(2) is the same as when A(12) = 1.

B - Array

B(0) - B(47) Analogue inputs channel readings (each averaged over the ensemble number as returned in A(5))

B(48) Counter 1 interrupt time (sec).

B(49) Counter 1 contents at scanning time.

B(50) Number of Counter interrupts during the last scanning interval.

B(51) Counter 2 interrupt time (sec).

B(52) Counter 2 contents at scanning time.

B(53) Number of Counter 2 interrupts during the last scanning interval.

B(54) Counter 3 interrupt time (sec).

B(55) Counter 3 contents at scanning time

B(56) Number of counter 3 interrupts during the last scanning interval.

B(56) - B(86) Storage area for data transferred from the M6800 in the range (0 - 32767).

CALL (2) This call exists from the scanning mode if the number of specified scans in A(2) has finished.

CALL (3, N, U)

This subroutine converts digital output signals into analogue outputs via DGOA and extra hardware (a 12-bit DAC and a zero-order hold).

N — Analogue output channels (0 - 15)
N = 1 steam input valve
N = 2 infeed valve

U — Digital equivalent of the analogue output
For 0 U 32767 the corresponding analogue output is 0 to 10 volts. The DGOA can be accessed by both computers.

CALL (4, I, M, D(0)

This subroutine is used to transfer data stored in the D array to the M6800.
If I = 0 then the M6800 must be in the ON-line mode and the subroutine merely places the data in a temporary buffer to be picked up by the M6800 at its next process scan, else I = 1, which causes the routine to immediately interrupt the M6800 to initiate the data transfer.

M — Number of N316 computer words to be transferred including D(0) in one transfer operation. (M 30).

D(0) — Array locations D(0) - D(M-1) are consecutively occupied the M data words.

APPENDIX F

ERROR HANDLING AND SENSE SWITCH USAGE IN H316

The error message has the format ERROR XY line NNN, where NNN is the BASIC line number and XY is one of the following:

BK A program BREAK has occurred in the course of program execution. Sense switch 1 (S.S.1) has been set.

CE The H316 user has specified a counter which is already under M6800 command.

GE Relevant only when A(12)=1. The H316 was still executing the interscan BASIC processing when the next relevant M6800 scanning interrupt occurs. It is therefore similar to error TF.

NC The HADIOS controller has interrupted the CPU but the interrupting device was not a counter.

NM An Alarm Input interrupt has occurred but it was not CA2 or CB2 from the M6800.

NR An unidentified interrupt has occurred.

RI A real integer conversion error has occurred in a library routine, i.e. the conversion of a real number outside the range (-32768 to 32767) has been attempted.

TF The H316 clock interrupt frequency is too small and BASIC code was still being executed when the next scanning interrupt arrives.

UI Program execution has been terminated via a user interference i.e. S.S.2. has been set and detected.

Sense Switch Usage

S.S.1 Normally used to terminate a BASIC Program.

S.S.2. If set, program returns to the BASIC Command mode.

S.S.3 The INPUT statement, in the BASIC program, readings data from cassette or paper tape.

S.S.4 The PRINT statement, in the BASIC program, writes data to cassette or paper tape.

APPENDIX (G)

SUBROUTINE GRAPH

Subroutine GRAPH

```
      SUBROUTINE GRAPH(XN,A)
C-----Subroutine Graph for use with Tektronix Graphics Routines
C     This is a steering routine ... N and C-array elements
C     must be defined at the BASIC level.Note that C(0) in
C     BASIC corresponds to A(1) in this routine.
C
      DIMENSION A(8)
      IX=IFIX(XN+.2)
      GOTO(10,15,20,30,40,50),IX
10    CALL INITT(0)
      RETURN
15    CALL VWINDO(A(2),A(3),A(4),A(5))
      RETURN
20    IF=IFIX(A(1)+.1)
      GOTO (1,2,3,4,5,6),IF
    1 CALL MOVEA(A(6),A(7))
      RETURN
    2 CALL POINTA(A(6),A(7))
      RETURN
    3 CALL DRAWA(A(6),A(7))
      RETURN
    4 CALL MOVER(A(6),A(7))
      RETURN
    5 CALL POINTR(A(6),A(7))
      RETURN
    6 CALL DRAWR(A(6),A(7))
      RETURN
30    IC=IFIX(A(8)+.2)
      CALL VCURSR(IC,A(6),A(7))
      A(8)=FLOAT(IC)
      RETURN
40    IC=IFIX(A(8)+.2)
      CALL ANCHO(IC)
      RETURN
50    IX=A(6)
      IY=A(7)
      CALL FINITT(IX,IY)
      RETURN
      END
```

CALL (5, N, C(Ø))

This call is used in the main BASIC program by specifying parameter N and elements of the C array (dimensioned as 7 cell array C(7) as follows :

N = 1	enter graphic mode.
= 2	set the windows.
= 3	perform graphic functions (draw, move,..., etc).
= 4	to invoke the cursor facilities.
= 5	output alphanumeric characters to screen.
= 6	exit graphic mode.

For N = 3

C(Ø) = 1	dark move.
= 2	move and draw the point.
= 3	draw a line connecting the initial and final point.

The x and y coordinate values are conveyed through C(5) and C(6), respectively.

C(1), C(2) =	minimum value and range of x coordinate.
C(3), C(4) =	minimum value and range of y coordinate.
C(5), C(6) =	values of x and y to be plotted.

C(7) = to output text on the screen or to invoke the cursor facility.

The graphic package was originally written for the Tektronix 4010 terminal. Using the more advanced Newbury 8510 multirange-terminal, two control operations are needed to enter and leave its graphic input mode.

C(7) = 29 , N = 5 , CALL (5,N,C(0)) - enter Newbury 8510.

C(7) = 24 , N = 5 , CALL (5,N,C(0)) - leave graphics mode to return to standard mode.

APPENDIX (H)

PROGRAMS AND DATA FILES TO FIND THE WEINER -HOPF CONTROLLER

THE LIBRARY APA

```
**********
**********  Pdn =  7          2299-MNI          Terminal: 56          23 MAR 84 15:42:01
**********
```

LISTING OF 2299MNI *PLPR LAST UPDATED ON 11-AUG-83 AT 13:40:46

```
 1  4 0
 2  0 4
 3  5 5
 4  5 4
 5  1 0
 6  -1.619
 7  .8198
 8  -.1279
 9  1.396E-4
10  0.0
11  0.0
12  1.0
13  -2.337
14  1.77
15  -.433
16  4.32E-4
17  0.0
18  48.159
19  -18.667
20  -.307
21  .4538
22  4.996E-3
23  0.0
24  -22.396
25  31.9699
26  -6.307
27  -3.32
28  5.325E-3
29  -.764
30  3.174
31  -3.882
32  1.311
33  .101
34  8.6595E-4
35  0.0
36  .447
37  -.797
38  .3506
39  -6.992E-4
40          0         7         0         0         1         0         0         0
41          1         1
42    0.1656290852
43          1         2
44     0.000000000      -8.817770757       24.04972189      -22.51565824
45     7.159590577       0.6677608903      -0.5467052518      8.7216316388E-04
46          2         1
47    7.2759576142E-12
48          2         2
49    -30.14072240
50          3         1
```

```
-8846.066022        -38881.91137
     3         2
 0.000000000
     4         1
 0.000000000
     4         2
4.3418485195E-12
     4         0         2         2         7         9         9         5
     8        10        10         6         0         0         0         0
     1         1
 0.1656093362        -18.69074210         1.589414671        0.6068195501
7.0350254900E-03
     1         2
 0.000000000
     1         3
 0.3937202716        -0.5117520763        0.1638397658
     1         4
-2.5849436923E-05    2.9174037879E-03    -2.4850774634E-04
     2         1
 1.1923715186E-04     114.4657872        -193.1151925         104.5124085
-16.41115961         -1.691644903         0.4324302367        4.9210423613E-03
     2         2
-30.14072240          87.23408395        -137.9168737         130.1761605
-73.06492599          21.98724214        -1.663194943        -0.9886451917
 0.2975773007        -2.6035046410E-02
     2         3
 9.124896500         -25.18719182         41.63017329        -40.72353906
 22.84593913         -6.457609644         0.2643535383        0.3433411508
-8.4378171284E-02     6.1811786827E-03
     2         4
 1.5606822362E-04    -1.7866753437E-02     3.0145525732E-02    -1.6889109828E-02
 3.4828798533E-03    -1.9435843720E-04
     3         1
-1.4496626923E+11    -1.3916746094E+17    -3.7690418160E+17     9.0492062799E+17
-5.3855014157E+17     8.9756110562E+16     8.5142168290E+15    -2.3168422488E+15
-2.6297505164E+13
     3         2
 0.000000000         6.4726513868E+16    -2.9849074186E+17     5.7874509057E+17
-6.0681507647E+17     3.6371891115E+17    -1.1547528345E+17     1.0089918456E+16
 4.9214176560E+15    -1.5585668711E+15     1.3912840353E+14
     3         3
 2.8900888130E+15    -2.0432801615E+16     8.3983845258E+16    -1.7295559703E+17
 1.8984612385E+17    -1.1423504410E+17     3.4187351024E+16    -1.8301081505E+15
-1.7321905908E+15     4.4339255857E+14    -3.3031534051E+13
     3         4
-1.8974681738E+11     2.0888281340E+13     5.8827199849E+13    -1.4056067026E+14
 8.6019067105E+13    -1.8375824226E+13     1.0386299549E+12
     4         1
-6.6694249245E-23
     4         2
 4.3418485195E-12
     4         3
-6.0120792922E-13
     4         4
-5.5858966619E-26
```

```
**********
**********  Pdn =  7          2299-MNI          Terminal: 56          23 MAR 84 15:36:56
**********
```

LISTING OF 2299MNI *COPRIME LAST UPDATED ON 27-JAN-84 AT 15:55:22

```
  1  * THIS PROGRAM FIND THE LEFT COPRIME & RIGHT
  2  * COPRIME OF MATRIX P .  THE DATA IS KEPT IN
  3  * FILE PLPR .
  4  ***********************************************
  5          REAL PL(4,2,0:50),PR(2,4,0:50),VL(4,4,0:60)
  6          REAL VR(4,4,0:60)
  7         REAL IC(2,2,0:35),PR1(2,4,0:40)
  8         INTEGER DIC(2,2),DPR1(2,4)
  9          INTEGER DPR(2,4),DPL(4,2),DVL(4,4),DVR(4,4)
 10  *
 11  *
 12          CALL READP(4,2,DPL,PL)
 13          CALL READP(2,4,DPR1,PR1)
 14           CALL READP (2,2,DIC,IC)
 15          CALL CREDUCE(PL,4,2,DPL,VL,DVL)
 16            CALL POLPOL (IC,PR1,PR,DIC,DPR1,DPR,2,2,4)
 17          CALL RREDUCE(PR,2,4,DPR,VR,DVR)
 18          CALL WRITP (4,2,DPL,PL)
 19          WRITE (40,105)
 20   105    FORMAT (' LEFT COPRIME UNIMODULAR MATRIX ')
 21          CALL WRITP(4,4,DVL,VL)
 22          WRITE (40,102)
 23   102    FORMAT ('*******NOW RIGHT COPRIME *******')
 24          CALL WRITP (2,4,DPR,PR)
 25          WRITE (40,110)
 26   110    FORMAT (' RIGHT COPRIME UNIMODULAR MATRIX ')
 27          CALL WRITP(4,4,DVR,VR)
 28          STOP
 29          END
```

```
**********
**********  No. of pages    1   2299-MNI          Terminal: 56          23 MAR 84 15:37:02
**********
```

```
**********
**********   Pdn =   7          2299-MNI          Terminal: 56          23 MAR 84 15:39:03
**********
```

LISTING OF 2299MNI *PLP LAST UPDATED ON 11-AUG-83 AT 13:23:39

```
    1  3 0
    2  0 3
    3  4 4
    4  4 3
    5  1.0
    6  -1.619
    7  .8198
    8  -.1279
    9  0.0
   10  0.0
   11  1.0
   12  -2.337
   13  1.77
   14  -.433
   15  0.0
   16  48.159
   17  -18.667
   18  -.307
   19  .4538
   20  0.0
   21  -22.396
   22  31.9699
   23  -6.307
   24  -3.32
   25  -.764
   26  3.174
   27  -3.882
   28  1.311
   29  .101
   30  0.0
   31  .447
   32  -.797
   33  .3506
   34          0          6          0          0          0          0          0          0
   35          1          1
   36     5550.997628
   37          1          2
   38     0.000000000      -3574.594127       4894.680666       -714.3624142
   39    -589.6508366      -24.93839621      -0.1384586449
   40          2          1
   41   -3.0775979541E-09
   42          2          2
   43    1.5156103070E-03
   44          3          1
   45    9.5705820438E-11
   46          3          2
   47     0.000000000
   48          4          1
   49   -1.1517326181E-08
   50          4          2
```

```
-7.9414333498E-05
   4   0   1   3  12  11  10  10
  13  12  11  11   6   5   4   4
   1   1
 5842.516774    188.9715198   -175.9280772   -14.82128874
-0.3118595115
   1   2
 0.000000000
   1   3
 167.2243133    6.432989312
   1   4
 381.5695625   -7.337224785   -11.11130130   -0.3949191240
   2   1
-5.2516535288E-02   3.430052776   -748.3758740   2141.020305
 -2365.995900   1172.058301   -159.0938015   -76.73752522
 27.80603047   -0.5694931353   -0.7067456088   4.6964909665E-02
 3.2839166015E-03
   2   2
 1.5156103070E-03   -1.418080895   330.8370508   -985.4107733
 1106.123398   -508.5681870   9.439286317   66.22800705
 -13.23874613   -2.287765766   0.5471984911   2.7224404563E-02
   2   3
-6.4848596761E-02   14.83466197   -58.42445685   95.40660514
 -81.02256924   35.77872933   -5.709451213   -1.359444991
 0.6330752425   -5.2065648761E-02   -3.6198755042E-03
   2   4
-6.8738916497E-02   0.2275586937   -48.88749695   142.3501478
 -162.1745894   86.14048028   -16.68416835   -2.844829600
 1.579333420   -0.1336495444   -9.1411114844E-03
   3   1
-0.6907871174   22.69383239   -9869.966233   38523.56108
 -60775.85306   48186.66790   -18325.97209   1194.105696
 1428.586191   -392.6113777   -1.408703767   10.40636500
-0.6072799897   -4.5483060938E-02
   3   2
 0.000000000   -8.470399318   4360.642099   -17542.73656
 28198.83928   -22009.64124   7167.957376   740.4073896
 -1091.412952   153.2673212   38.88381337   -7.220733211
-0.3770647681
   3   3
-0.3962566544   195.1052621   -973.3435834   2064.006726
 -2387.152386   1592.805667   -570.6447445   61.19557589
 27.15593766   -9.453105748   0.6735073884   5.0136175225E-02
   3   4
-0.9041716197   2.433291299   -644.7241852   2549.189956
 -4104.782157   3379.225066   -1412.525740   193.6598202
 60.17568713   -23.63214735   1.730839892   0.1266066655
   4   1
-1.1517326181E-08   6.571192524   -10.80692582   5.456419137
-0.6343317239   -0.1505078015   2.4919503227E-02
   4   2
-7.9414333498E-05   -2.903594884   5.122855351   -2.520275937
-4.9396032049E-02   0.2065882663
   4   3
-0.1296396364   0.3552278872   -0.3627836069   0.1635878979
-2.7468876437E-02
   4   4
 0.000000000   0.4291587251   -0.7279238061   0.3966863445
-6.9365938554E-02
```

```
**********
**********   Pdn =   7          2299-MNI          Terminal: 56          23 MAR 84 15:39:45
**********
```

LISTING OF 2299MNI *PLPRNEC LAST UPDATED ON 27-JAN-84 AT 16:36:33

```
  1  3 0
  2  0 3
  3  4 4
  4  3 3
  5  1.0
  6  -1.619
  7  .8198
  8  -.1279
  9  0.0
 10  0.0
 11  1.0
 12  -2.337
 13  1.77
 14  -.433
 15  .4107
 16  2.836
 17  .9397
 18  -1.4105
 19  .2599
 20  -0.0244
 21  -2.1581
 22  3.0901
 23  -.5718
 24  -.3344
 25  -1.73
 26  5.701
 27  -6.2318
 28  2.225
 29  -.402
 30  1.388
 31  -1.5134
 32  .5268
 33  4 0 5 4
 34  0 3 4 4
 35  1.0 -3.104 3.562 -1.7899 .3318
 36  0.0
 37  4.107 22.2608 -31.357 -18.0126 29.766 -6.764
 38  -.244 -21.3936 47.475 -29.449 1.047 2.568
 39  0.0
 40  1.0 -2.404 2.0897 -.76989 9.975E-2
 41  -1.73 7.0625 -10.718 7.129 -1.775 1.857E-2
 42  -.402 1.41633 -1.6744 .7912 -.1217
 43   0 0
 44   0 0
 45  1.0
 46  0.0
 47  0.0
 48  1.0
 49          0         6         0         0         0         0         0         0
 50          1         1


 51     93062.55924
 52          1         2
 53    -2779.201484      -54630.77137       83545.13940      -19063.30201
 54    -6622.131146      -418.9724455       4.412801644
 55          2         1
 56    5.5197723080E-09
 57          2         2
 58    4.7623500983E-03
 59          3         1
 60   -7.4017182885E-08
 61          3         2
 62    4.5474735089E-13
 63          4         1
 64    2.6127623168E-10
 65          4         2
 66    4.4485491195E-05
 67  LEFT COPRIME UNIMODULAR MATRIX
 68          3         0         1         3        11         9         9         7
 69         12        10        10         8         5         3         3         0
 70          1         1
 71     90467.89949       40958.25694      -16274.89180       145.7230025
 72          1         2
 73     0.000000000
 74          1         3
 75     28223.31691      -258.8007613
 76          1         4
 77     5200.379481       2354.409467      -935.5319826       8.376616636
 78          2         1
 79   -1.2020654887E-03   0.2898551515      -2.731811201       8.152834363
 80    -4.059247702      -5.071272212       6.398577575      -2.489320535
 81    0.2363517221       6.1549968964E-02 -7.2292752504E-03 -1.0542458513E-03
 82          2         2
```

```
          1           4
    5200.379481        2354.409467       -935.5319826        8.376616636
          2           1
-1.2020654887E-03   0.2898551515       -2.731811201        8.152834363
   -4.059247702       -5.071272212        6.398577575       -2.489320535
   0.2363517221    6.1549968964E-02   -7.2292752504E-03   -1.0542458513E-03
          2           2
-2.4217492210E-02   -0.4426614717       3.341597831       -9.272146757
    10.92909348       -5.706907903       0.9744153734       0.1301479652
-3.0366416152E-02   -3.4142485711E-03
          2           3
   -0.2395016282        1.812406980       -5.969291913        8.821376941
   -6.407090136        2.188692385       -0.1999490847       -5.4633866185E-02
   6.5881748412E-03    9.6514399047E-04
          2           4
-5.7552244371E-02   2.8205872275E-02    0.1020485285       -0.1212697981
   4.8647820680E-02   -5.6471977366E-03   -9.0627308441E-04   1.3275616960E-04
          3           1
   7.5755130418E-02       -2.884327023        34.46731430       -127.4427131
    151.0625789        4.794461592       -137.5682800        110.6236461
   -34.78404251        2.355751527       0.8756741594       -8.1393905892E-02
-1.3602867683E-02
          3           2
   0.3248118447        4.453754070       -43.04162022        147.6454078
   -242.8463943        205.3538822       -84.62088489        11.10560491
    2.021624925       -0.3533256278       -4.4053833833E-02
          3           3
    2.563611077       -23.25947071        90.58333916       -176.4686526
    186.0516105       -107.3443700        30.49467355       -1.964018857
   -0.7792087153    7.4126312177E-02    1.2453192001E-02
          3           4
   0.6523873876       -1.062334130       -0.7865093805        2.683868564
   -2.113167588       0.6913630998       -6.2637109198E-02   -1.3190203004E-02
    1.7305899812E-03
          4           1
```

```
-1.3784434946E-03   -7.7797823480E-03    8.3117034640E-03    4.9767877851E-03
-7.2579504312E-03    1.8579566640E-03
          4           2
   8.6382416831E-05    6.4562763213E-03   -1.1092778602E-02    6.0171219775E-03
          4           3
   3.0225540985E-03   -8.1466174596E-03    6.7393866548E-03   -1.7009274452E-03
          4           4
  -7.9237259110E-05
*******NOW RIGHT COPRIME *******
          0           0           0           0           8           1           0           0
          1           1
   -86.30056122
          1           2
    0.000000000
          1           3
   7.2759576142E-12
          1           4
    0.000000000
          2           1
   -199.3084406        1037.266008       -2174.242862        2330.582237
   -1300.251041        304.2465875        18.44411743       -17.42168162
    1.356615702
          2           2
    7.760957510       -9.263610269
          2           3
   9.2212954768E-05
          2           4
   6.9884390541E-07
 RIGHT COPRIME UNIMODULAR MATRIX
          4           6          17           7           0           7          17           8
          2           4          15           5           4           6          17           7
          1           1
   -405.4733241        137.6496101        349.4912909       -223.0383468
    35.17519886
          1           2
    4.448907932        25.94291846       -24.05441882       -33.47587060
    24.22275226        5.928122480       -3.012105999
          1           3
   -2.374448093        14.48007848       -50.41857305        158.5211351
   -308.0251421        294.4376251        2.855773237       -346.5604611
    417.7096541       -233.4293160        42.43592408        26.17699739
   -19.29802172        4.747144653       -0.1109297925       -0.1784662313
   3.2844931293E-02   -1.6959364584E-03
          1           4
   1.9349737094E-02    8.1460160407E-02   -0.2621934773        0.1615898815
   0.1476651324       -0.2563405349        0.1290240939       -2.0554583549E-02
          2           1
    0.000000000
          2           2
    5.175253248        2.393159072        44.97355929       -33.15564110
   -9.571583900        7.778864277        0.8683819118       -0.1508904837
          2           3
    1.540088622       -10.24459762        42.40286996       -150.1004256
    403.5614677       -741.5986526        901.3246357       -696.5995984
    293.3439636       -5.911384703       -65.34907804        33.60024309
   -5.210466864       -1.314142918        0.6117956841       -4.8891611517E-02
-1.0841291197E-02    1.9670855825E-03
          2           4
```

```
157          2          1
158      0.000000000
159          2          2
160      5.175253248          2.393159072          44.97555929         -33.15564110
161     -9.571583900          7.778864277         0.8683819118         -0.1508904837
162          2          3
163      1.540088622         -10.24459762          42.40286996          -150.1004256
164      403.5614677         -741.5986526          901.3246357          -696.5995984
165      293.3439636         -5.911384703         -65.34907804          33.60024309
166     -5.210466864         -1.314142918         0.6117956841         -4.8891611517E-02
167     -1.0841291197E-02    1.9670855825E-03
168          2          4
169     -1.1245363884E-02    -1.2211526910E-02    0.2025535047         -0.4575145623
170      0.4471536656         -0.1926132202        1.5158779104E-02     1.0362747988E-02



171     -1.0296752662E-03
172          3          1
173      85.34841945         -135.8167924          39.22444681
174          3          2
175     -1.167012276         -4.461502727          10.80219005         -3.056991728
176     -3.358849286
177          3          3
178      0.6416793402         -3.500468247          13.83621785         -48.24586615
179      118.4197028         -188.8439584          193.4358706         -121.1728347
180      35.54902449          7.401218001         -10.94400283          3.884237444
181     -0.3765898581        -0.1134319233         3.1182714579E-02     -1.8911668340E-03
182          3          4
183     -5.0757132046E-03    -1.1174674552E-02     7.1787989515E-02     -0.1121867434
184      7.7905705050E-02    -2.2920756539E-02
185          4          1
186      128.4967039         -43.62191083         -110.7556928          70.68206636
187     -11.14721202
188          4          2
189     -1.409883144         -8.221452097          7.622976281          10.60868564
190     -7.676322063         -1.878654287         0.9545527895
191          4          3
192      1.069380972         -4.588815701          15.97792028         -50.23621071
193      97.61484445         -93.30888635         -0.9050100795          109.8268968
194     -132.3744635          73.97502107         -13.44817537         -8.295632984
195      6.115648143         -1.304395985         3.5154255182E-02     5.6556920297E-02
196     -1.0408737536E-02    5.3745149853E-04
197          4          4
198     -6.1320370275E-03    -2.5815168310E-02     8.3090540363E-02     -5.1208713176E-02
199     -4.6795884356E-02     8.1235711053E-02    -4.0888437769E-02     6.5138594328E-03
```

LISTING OF 2299MNI *PNEC LAST UPDATED ON 8-FEB-84 AT 10:45:40

```
  1  3 0
  2  0 3
  3  3 3
  4  3 2
  5  1.0 -1.618 .818 -.1269
  6  0.0
  7  0.0
  8  1.0 -2.335 1.7668 -.4309
  9    48.1592 -19.1963 -.09567 .45465
 10    -22.396 31.9333 -6.2569 -3.3302
 11  -.764 3.2239 -4.09146 1.57657
 12    .447 -.7961 .3491
 13  4 0 5 5
 14  0 4 4 4
 15  1.0 -3.104 3.5621 -1.7899 .3318
 16  0.0
 17  0.0 48.1592 -90.7608 44.7903 -1.0 -1.1887
 18  0.0 -22.396 49.13346 -30.7817 1.475 2.5576
 19  0.0
 20  1.0 -2.405 2.0914 -.7707 .09987
 21  -.764 3.8257 -6.63017 4.7978 -1.24115
 22  0.0 .447 -.8273 .477 -.0808
 23  0 0
 24  0 0
 25  1.0
 26  0.0
 27  0.0
 28  1.0
 29          0          4          0          0          0          0          0          0
 30          1          1
 31      10.82372354
 32          1          2
 33     0.8563355912        -9.493688912         12.03142379        -2.199798145
 34     -1.211338943
 35          2          1
 36    -9.5984216385E-08
 37          2          2
 38    -0.4026234300
 39          3          1
 40     1.8868422215E-08
 41          3          2
 42      0.000000000
 43          4          1
 44     5.7828389086E-10
 45          4          2
 46      0.000000000
 47   LEFT COPRIME UNIMODULAR MATRIX
 48          2          0          1          2          5          4          4          5
 49         12         11         11         12          6          5          5          6
 50          1          1



 51      12.46724671        -0.8458724677        -0.3357822513
 52          1          2
 53      0.000000000
 54          1          3
 55    -1.8207246831E-02    0.3609103470
 56          1          4
 57      1.003503560       -6.8085283968E-02    -2.7027513959E-02
 58          2          1
 59      22788.40315         -35583.06779         18178.42576        -2785.877972
 60     -265.3897968          72.32355109
 61          2          2
 62     -10766.16137         17745.16784        -8178.618161        -164.8940821
 63      605.4958775
 64          2          3
 65     -444.0901035         1159.241380        -1128.044566         484.9968962
 66     -77.73584763
 67          2          4
 68      1834.265754        -2864.123573         1463.203175        -224.2386412
 69     -21.36154133          5.821408902
 70          3          1
 71     -4478.131881         63884.20843        -239307.7872         429789.6331
 72     -435340.4024         257022.3759        -80300.86290         5381.504803
 73      4656.555405        -1401.703169         77.60017522         21.17302725
 74     -2.551122468
 75          3          2
 76      2115.578498        -30364.15149         115310.0878        -207759.1227
 77      205573.5382        -110783.4119         23921.86130         5325.064443
 78     -3874.402378          451.3942975         104.7047876        -21.35810693
 79          3          3
 80      87.26653371        -1336.478590         5978.491976        -13394.76776
 81      17721.08215        -14669.59131         7546.876413        -2166.312935
 82      165.8634139          94.62746029        -29.80331725         2.742034434
 83          3          4
```

```
 79          3          3
 80    87.26653371          -1336.478590          5978.491976          -13394.76776
 81    17721.08215          -14669.59131          7546.876413          -2166.312935
 82    165.8634139          94.62746029          -29.80331725          2.742034434
 83          3          4
 84    -360.5306683          5142.116145          -19262.16926          34594.28027
 85    -35041.06831          20688.03764          -6463.512246          433.1637413
 86    374.8116994          -112.8247602          6.246130673          1.704242221
 87   -0.2053428907
 88          4          1
 89    -137.2951401          375.3101173          -370.0854403          153.9173154
 90    -20.07304304          -2.407350822          0.5585082468
 91          4          2
 92    64.86133963          -182.9433632          178.9709469          -61.28349351
 93    -4.271535043          4.675855042
 94          4          3
 95    2.675545653          -10.12030517          15.16352951          -11.24852444
 96    4.130234584          -0.6003039303
 97          4          4
 98    -11.05104961          30.20915905          -29.78861857          12.38898833
 99    -1.615703180          -0.1937866376          4.4954995035E-02
100  *******NOW RIGHT COPRIME *******
101          0          0          0          0          8          1          0          0
102          1          1
103   -0.5208355682
104          1          2
105    0.000000000
106          1          3
107   7.2759576142E-12
108          1          4
109    0.000000000
110          2          1
```

```
111    -210.8944501          1369.313534          -3385.328373          4264.056414
112    -2949.096123          1076.390674          -165.7450421          2.412055811
113   -0.6018376153
114          2          2
115    0.4080378711          -0.3225883842
116          2          3
117   7.6126307249E-06
118          2          4
119   1.0566498077E-07
120  RIGHT COPRIME UNIMODULAR MATRIX
121          5          8          19          9          0          5          16          6
122          2          5          16          6          4          6          17          7
123          1          1
124   -0.5208355682          -13295.22189          10002.33023          1398.066645
125    -451.6753842          -57.41484659
126          1          2
127    0.000000000          1934641.164          -4966869.110          4332877.504
128    -1298568.135          -80041.37004          81487.30725          -2205.567569
129    -1321.792830
130          1          3
131   7.2759576142E-12          1919811998.          -2.1835431839E+10          1.1068707109E+11
132   -3.3337123198E+11          6.6784156647E+11          -9.4043575628E+11          9.5574405004E+11
133   -7.0530150271E+11          3.7237976062E+11          -1.3384598576E+11          2.8171551582E+10
134    -1103442610.          -1106849036.          273092269.3          -14072517.63
135    -2971821.751          349124.2437          -10162.37773          1570.976692
136          1          4
137    0.000000000          -10429.82787          34748.22226          -43824.30066
138    24853.98535          -4919.234148          -769.0940315          347.6581222
139    -1.962562611          -5.446458419
140          2          1
141    0.000000000
142          2          2
143    -30690.83784          181159.3876          -384478.3249          387441.5227
144    -190895.7342          37503.37732
145          2          3
146    -30455995.23          447977838.3          -2963816452.          1.1744357079E+10
147   -3.1282003562E+10          5.9482279772E+10          -8.3541929817E+10          8.8282397586E+10
148   -7.0699927530E+10          4.2788617126E+10          -1.9306482930E+10          6316808586.
149    -1425998226.          203197124.6          -15565621.22          589597.7658
150    -44573.49917
151          2          4
152    165.4593202          -1103.101622          2819.199910          -3672.928080
153    2625.554412          -988.7600550          154.5330581
154          3          1
155    276.0398561          -410.0298408          151.2994525
156          3          2
157    -40171.78782          132573.2545          -169647.2706          104788.7139
158    -31029.29590          3483.184999
159          3          3
160    -39863868.13          482614275.7          -2634687797.          8654181497.
161   -1.9190355976E+10          3.0477216412E+10          -3.5819905913E+10          3.1693392253E+10
162   -2.1235467900E+10          1.0731818744E+10          -4031725128.          1094808412.
163    -204834141.9          24398788.44          -1656083.791          70566.59939
164    -4139.833657
165          3          4
166    216.5697909          -880.2359133          1460.834371          -1263.940010
167    599.0569011          -146.6335645          14.35249294
168          4          1
169   1.1315685413E-02          288.8503478          -265.6845144          14.11969920
170    7.448485337
```

```
166      216.5697909          -880.2359133          1460.834371          -1263.940010
167      599.0569011          -146.6335645          14.35249294
168            4          1
169     1.1315685413E-02       288.8503478          -265.6845144          14.11969920
170      7.448485337
```

```
171            4          2
172     -2.8814142026E-07      -42032.05062          114949.3899          -113386.7255
173      47201.56354          -6165.854157          -427.1390291
174            4          3
175     2.1440086172E-02       -41709871.89          481382148.5          -2485406358.
176      7659060113.          -1.5792199493E+10     2.3076949210E+10     -2.4632037870E+10
177     1.9459644275E+10      -1.1374149500E+10     4848161653.          -1457258484.
178      288780262.3          -32528460.21          1273394.775          35276.97191
179      3197.157391           507.6631098
180            4          4
181     1.5533968119E-09       226.5986382          -792.8889551          1084.912510
182     -721.6681939           227.7329731          -23.10384026          -1.760029944
```

****
****
****
********** No. of pages 4 2299-MNI Terminal: 56 23 MAR 84 15:41:09
****
****
****

```
**********
**********   Pdn =   7          2299-MNI          Terminal:103
**********
```

LISTING OF 2299MNI *BEZOUT LAST UPDATED ON 17-JAN-84 AT 19:40:50

```
 1          REAL PXY(2,4,0:80),VXY(4,4,0:100)
 2          REAL IC(2,2,0:40),PXY1(2,4,0:80)
 3          INTEGER DPXY(2,4),DVXY(4,4)
 4         INTEGER DIC(2,2),DPXY1(2,4)
 5          CALL READP (2,4,DPXY1,PXY1)
 6          CALL READP (2,2,DIC,IC)
 7         CALL POLPOL (IC,PXY1,PXY,DIC,DPXY1,DPXY,2,2,4)
 8          CALL RREDUCE (PXY,2,4,DPXY,VXY,DVXY)
 9          WRITE (40,105)
10   105    FORMAT (' EXTRACT X & Y FROM UNIMODULAR  VXY ')
11          CALL WRITP (4,4,DVXY,VXY)
12          STOP
13          END
```

```
**********
**********   No. of pages     1   2299-MNI          Terminal:103
**********
```

```
**********************************************************************
**********************************************************************
**************
**************   Pdn =   7            2299-MNI          Terminal:103
**************
**********************************************************************
**********************************************************************

LISTING OF 2299MNI *BEZTEST  LAST UPDATED ON  9-FEB-84 AT 13:24:13

   1          REAL A(2,2,0:50),B(2,2,0:50),X(2,2,0:50),Y(2,2,0:50)
   2          REAL AX(2,2,0:120),BY(2,2,0:120),AXBY(2,2,0:120)
   3          INTEGER DA(2,2),DB(2,2),DX(2,2),DY(2,2),DAX(2,2)
   4          INTEGER DBY(2,2),DAXB(2,2)
   5  **********************
   6          CALL READP (2,2,DA,A)
   7          CALL READP (2,2,DB,B)
   8          CALL READP (2,2,DX,X)
   9          CALL READP (2,2,DY,Y)
  10          CALL POLPOL (A,X,AX,DA,DX,DAX,2,2,2)
  11          WRITE (40,103)
  12    103   FORMAT (' ----- AX ------- ')
  13          CALL WRITP (2,2,DAX,AX)
  14          CALL POLPOL (B,Y,BY,DB,DY,DBY,2,2,2)
  15          WRITE (40,104)
  16    104   FORMAT (' ----- BY ------- ')
  17          CALL WRITP (2,2,DBY,BY)
  18          CALL POLYADD (AX,BY,DAX,DBY,2,2,AXBY,DAXB)
  19           WRITE (40,105)
  20    105    FORMAT (' -------  AX + BY = G  --------')
  21          CALL WRITP (2,2,DAXB,AXBY)
  22          STOP
  23          END

**********************************************************************
**********************************************************************
**************
**************   No. of pages     1   2299-MNI          Terminal:103
**************
**********************************************************************
**********************************************************************
```

```
**********************************************************************************************
**********************************************************************************************
**************
**************   Pdn =   7            2299-MNI          Terminal: 72                  26 MAR
**************
**********************************************************************************************
**********************************************************************************************

LISTING OF 2299MNI *ABL      LAST UPDATED ON 22-DEC-83 AT 19:50:54

  1  12 10 10 8
  2  5 3 3 0
  3     7.5755130418E-02   -2.884327023        34.46731430       -127.4427131
  4      151.0625789        4.794461592       -137.5682800        110.6236461
  5     -34.78404251        2.355751527        0.8756741594      -8.1393905892E-02
  6    -1.3602867683E-02
  7     0.3248118447        4.453754070       -43.04162022        147.6454078
  8     -242.8463943        205.3538822       -84.62088489        11.10560491
  9      2.021624925       -0.3533256278      -4.4053833833E-02
 10     -2.563611077        23.25947071       -90.58333916        176.4686526
 11     -186.0516105        107.3443700       -30.49467355        1.964018857
 12     0.7792087153       -7.4126312177E-02  -1.2453192001E-02
 13    -0.6523873876        1.062334130        0.7865093805       -2.683868564
 14      2.113167588       -0.6913630998       6.2637109198E-02   1.3190203004E-02
 15    -1.7305899812E-03
 16    -1.3784434946E-03   -7.7797823480E-03   8.3117034640E-03   4.9767877851E-03
 17    -7.2579504312E-03    1.8579566640E-03
 18     8.6382416831E-05    6.4562763213E-03  -1.1092778602E-02   6.0171219775E-03
 19    -3.0225540985E-03    8.1466174595E-03  -6.7393866548E-03   1.7009274452E-03
 20     7.9237259110E-05
 21   EXTRACT X & Y FROM UNIMODULAR  VXY
 22            0          8          9          0         12         26         27          4
 23            0         12         13          0         16         31         32          8
 24            1          1
 25      0.000000000
 26            1          2
 27    -0.2039320399       -0.7482070589       8.034386852       -16.78767038
 28      15.89449561       -7.614016408        1.500870285        0.1427457736
 29    -9.4431453242E-02
 30            1          3
 31     -935266.6116        157619.2885        13631900.54       -38632291.11
 32      49303130.95       -34456923.38        13013412.98       -1862838.372
 33     -348281.9455        132177.0039
 34            1          4
 35      0.000000000
 36            2          1
 37      10.92242213       -84.18400886        245.9516808       -189.3241616
 38     -361.3882420        876.5444438       -776.8446034        345.5498255
 39     -69.75126886        0.2250478285        1.870991846      -8.2452308546E-02
 40    -2.3763579433E-02
 41            2          2
 42      3.718517404       -18.90493922        25.69838026        38.53271907
 43     -93.97350589        21.71326705        34.53054351        7.413227757
 44     -34.27019202        13.39595110        3.286808666       -3.385880148
 45    -0.6521018347       -1.455315841        5.396518565       -1.888634680
 46     -3.129873305        2.461778328        4.9487238499E-02  -0.4161973987
 47    -0.2454702749        0.4514676020      -0.2652760627        8.4993808208E-02
 48    -1.3957983722E-02    3.4037620380E-04   1.8199486294E-04
 49            2          3
 50      7694764.864       -38917348.83        65779856.18        36557239.04
```

```
 51     -210945183. 5          161570787. 9          28259304. 43          -34343751. 39
 52     -66574237. 44          69198651. 76          -13036646. 35         -9863816. 555
 53      3137827. 964         -1646808. 348          11206830. 36          -10521703. 47
 54     -2610773. 443          8328364. 657         -3286380. 651          -756008. 2490
 55      175103. 4537          1076957. 428         -1058736. 549          506882. 4032
 56     -140994. 2312          20046. 34256         -190. 2707349          -254. 7407128
 57            2                     4
 58     4. 4371748712E-04   -3. 2585102593E-04   -8. 7620503222E-04    1. 0965975798E-03
 59    -2. 5177739455E-04
 60            3                     1
 61      1. 000000000
 62            3                     2
 63     0. 3333028252          1. 032719409          1. 421502059          -14. 81557375
 64      14. 38357538          3. 997684746         -13. 69282176          8. 407533795
 65     -1. 838652525         -0. 1362207278        0. 1086198426         -5. 7090322076E-03
 66    -1. 8527503416E-03
 67            3                     3
 68      674539. 6760          2645803. 711          309218. 9204          -27362758. 01
 69      44796039. 82         -13301644. 47         -27994679. 25          32668646. 82
 70     -14658677. 19          2344909. 143          362597. 8566         -160748. 3752
 71      5077. 850139          2593. 320136
 72            3                     4
 73      0. 000000000
 74            4                     1
 75      1. 521407893          69. 98742324         -1200. 691408          8176. 524040
 76     -30919. 26928          71524. 56028         -104475. 9659          97009. 90956
 77     -56013. 67500          18296. 03028         -2067. 831378         -601. 3537414
 78      203. 8368397         -4. 900304059         -4. 127676264         0. 1686314849
 79     4. 0486598602E-02
 80            4                     2
 81     0. 5179603615          25. 18592522         -343. 8745700          1823. 362447
 82     -5019. 266348          7356. 328481         -4827. 397689          391. 5824135
 83     -357. 6015119          2499. 185238         -2048. 886214          244. 2245843
 84      457. 1928663         -315. 3735183          239. 0994081         -313. 1669384
 85      158. 2431986          180. 8061973         -275. 8470788          111. 2511007
 86      1. 424930449          16. 91087792         -47. 75836999          38. 75947632
 87     -16. 63930449          3. 744691225        -6. 7139268475E-02    -0. 2132006668
 88     5. 1185089319E-02   -1. 2111003941E-04    -1. 7006677678E-03    2. 0871203378E-04
 89            4                     3
 90      1071820. 498          52145736. 70         -708310275. 9          3850128453.
 91    -1. 1378401342E+10     1. 9416913276E+10    -1. 7998288608E+10     7093430742.
 92     -1116616507.           4656417521.          -6822883214.          3235083923.
 93      393702784. 7         -1126847014.           827846885. 4         -867195549. 5
 94      704496503. 5          81625893. 39         -703279459. 0          560026890. 4
 95     -149282222. 3          25343855. 66         -101530393. 2          129509604. 4
 96     -80929092. 63          29223452. 11         -5325627. 499         -248485. 9594
 97      379209. 9502         -71628. 14965         -2534. 363869          2708. 615167
 98     -292. 1371043
 99            4                     4
100     2. 2121226528E-04   3. 1948782080E-03    -2. 6885262822E-02    8. 0304576856E-02
101    -0. 1050449466       6. 0949947057E-02    -1. 1844043469E-02    -1. 3930150597E-03
102     4. 7360041572E-04
```

```
**********************************************************************************************
**********************************************************************************************
**************
**************   Pdn =   6            2299-MNI          Terminal: 76                  17 J
**************
**********************************************************************************************
**********************************************************************************************

LISTING OF 2299MNI *AL       LAST UPDATED ON 22-DEC-83 AT 19:09:01

  1  12 10
  2  5 3
  3          3          1
  4     7.5755130418E-02   -2.884327023        34.46731430        -127.4427131
  5      151.0625789        4.794461592       -137.5682800         110.6236461
  6     -34.78404251        2.355751527        0.8756741594       -8.1393905892E-02
  7    -1.3602867683E-02
  8          3          2
  9     0.3248118447       4.453754070       -43.04162022         147.6454078
 10     -242.8463943       205.3538822       -84.62088489         11.10560491
 11      2.021624925      -0.3533256278      -4.4053833833E-02
 12          4          1
 13    -1.3784434946E-03  -7.7797823480E-03   8.3117034640E-03    4.9767877851E-03
 14    -7.2579504312E-03   1.8579566640E-03
 15          4          2
 16     8.6382416831E-05   6.4562763213E-03  -1.1092778602E-02    6.0171219775E-03

**********************************************************************************************
**********************************************************************************************
**************
**************   No. of pages    1   2299-MNI          Terminal: 76                  17 J
**************
**********************************************************************************************
**********************************************************************************************
```

```
**********************************************************************************************
**********************************************************************************************
**************
**************   Pdn =   6            2299-MNI          Terminal: 76                  17 .
**************
**********************************************************************************************
**********************************************************************************************

LISTING OF 2299MNI *BL       LAST UPDATED ON 22-DEC-83 AT 19:33:44

  1  10 8
  2  3 0
  3          3          3
  4      2.563611077          -23.25947071          90.58333916          -176.4686526
  5      186.0516105          -107.3443700          30.49467355          -1.964018857
  6     -0.7792087153         7.4126312177E-02     1.2453192001E-02
  7          3          4
  8      0.6523873876          -1.062334130         -0.7865093805          2.683868564
  9      -2.113167588          0.6913630998         -6.2637109198E-02     -1.3190203004E-02
 10      1.7305899812E-03
 11          4          3
 12      3.0225540985E-03     -8.1466174596E-03     6.7393866548E-03     -1.7009274452E-03
 13          4          4
 14     -7.9237259110E-05

**********************************************************************************************
**********************************************************************************************
**************
**************   No. of pages     1   2299-MNI          Terminal: 76                  17 .
**************
**********************************************************************************************
**********************************************************************************************
```

```
**********************************************************************************************
**********************************************************************************************
**************
**************   Pdn =   6          2299-MNI          Terminal: 48                22 DE
**************
**********************************************************************************************
**********************************************************************************************

LISTING OF 2299MNI *ABXY     LAST UPDATED ON 22-DEC-83 AT 20:09:57

  1  12 10
  2  5 3
  3     7.5755130418E-02   -2.884327023        34.46731430       -127.4427131
  4      151.0625789        4.794461592       -137.5682800        110.6236461
  5     -34.78404251        2.355751527        0.8756741594      -8.1393905892E-02
  6    -1.3602867683E-02
  7     0.3248118447        4.453754070       -43.04162022        147.6454078
  8     -242.8463943        205.3538822       -84.62088489        11.10560491
  9      2.021624925       -0.3533256278      -4.4053833833E-02
 10    -1.3784434946E-03   -7.7797823480E-03   8.3117034640E-03    4.9767877851E-03
 11    -7.2579504312E-03    1.8579566640E-03
 12     8.6382416831E-05    6.4562763213E-03  -1.1092778602E-02    6.0171219775E-03
 13  10 8
 14  3 0
 15      2.563611077       -23.25947071        90.58333916       -176.4686526
 16      186.0516105       -107.3443700        30.49467355       -1.964018857
 17    -0.7792087153        7.4126312177E-02    1.2453192001E-02
 18     0.6523873876       -1.062334130       -0.7865093805        2.683868564
 19     -2.113167588        0.6913630998      -6.2637109198E-02   -1.3190203004E-02
 20     1.7305899812E-03
 21     3.0225540985E-03   -8.1466174596E-03   6.7393866548E-03   -1.7009274452E-03
 22    -7.9237259110E-05
 23  0 8
 24  12 26
 25      0.000000000
 26    -0.2039320399       -0.7482070589        8.034386852       -16.78767038
 27      15.89449561       -7.614016408         1.500870285        0.1427457736
 28    -9.4431453242E-02
 29      10.92242213       -84.18400886        245.9516808       -189.3241616
 30     -361.3882420        876.5444438       -776.8446034        345.5498255
 31     -69.75126886        0.2250478285        1.870991846      -8.2452308546E-02
 32    -2.3763579433E-02
 33      3.718517404       -18.90493922        25.69838026        38.53271907
 34     -93.97350589        21.71326705        34.53054351        7.413227757
 35     -34.27019202        13.39595110        3.286808666       -3.385880148
 36    -0.6521018347       -1.455315841        5.396518565       -1.888634680
 37     -3.129873305        2.461778328        4.9487238499E-02  -0.4161973987
 38    -0.2454702749        0.4514676020      -0.2652760627        8.4993808208E-02
 39    -1.3957983722E-02    3.4037620380E-04    1.8199486294E-04
 40  0 12
 41  16 31
 42      1.000000000
 43     0.3333028252        1.032719409         1.421502059       -14.81557375
 44      14.38357538        3.997684746        -13.69282176        8.407533795
 45     -1.838652525       -0.1362207278        0.1086198426      -5.7090322076E-03
 46    -1.8527503416E-03
 47      1.521407893        69.98742324       -1200.691408        8176.524040
 48     -30919.26928        71524.56028       -104475.9659        97009.90956
 49     -56013.67500        18296.03028        -2067.831378       -601.3537414
 50      203.8368397       -4.900304059        -4.127676264        0.1686314849
```

```
 51      4.0486598602E-02
 52      0.5179603615          25.18592522          -343.8745700          1823.362447
 53     -5019.266348           7356.328481          -4827.397689           391.5824135
 54     -357.6015119           2499.185238          -2048.886214           244.2245843
 55      457.1928663          -315.3735183           239.0994081          -313.1669384
 56      158.2431986           180.8061973          -275.8470788           111.2511007
 57      1.424930449           16.91087792          -47.75836999           38.75947632
 58     -16.63930449           3.744691225         -6.7139268475E-02      -0.2132006668
 59      5.1185089319E-02     -1.2111003941E-04     -1.7006677678E-03      2.0871203378E-04
 60    ------ AX --------
 61          22          36          15          29
 62           1           1
 63      3.547732081           21.30181880          -765.1655981           6269.975177
 64     -26628.59842           65824.37733          -86686.77945           9296.004354
 65      191528.6479          -398381.1080           451683.2966          -332396.7555
 66      162648.4646          -50058.64848           7456.380380           541.4924252
 67     -395.8662251           32.75281882           7.870951750          -1.101580155
 68     -0.1013328947          1.2028621922E-02      1.0468767796E-03
 69           1           2
 70      1.192369599           10.95234001          -240.1640954           1465.446795
 71     -4268.168386           5315.887798           3351.790948          -21801.84190
 72      32259.70252          -20990.98373           2985.780008           305.2273059
 73      7132.558712          -9398.578777           4498.542515          -44.95913161
 74     -1190.962491           1446.193241          -1721.647267           1106.176467
 75      258.5610267          -999.1048694           699.8891864          -194.3316130
 76      69.84687652          -160.9982354           185.8011610          -119.0875960
 77      47.99621639          -12.04265279           1.465755499           9.9224117343E-02
 78     -5.8182353560E-02      3.8966863186E-03      8.6256441077E-04      -7.9298325928E-05
 79     -8.0175714503E-06
 80           2           1
 81      9.4350522123E-04      6.3246157224E-02     -0.6434293329           2.571133853
 82     -4.488379683           1.322547950           8.461720607          -16.88350457
 83      16.11657990          -8.957790472           2.854565424          -0.4101257806
 84     -1.9934745722E-02      1.2019187125E-02     -2.3252147241E-04      -1.4298835607E-04
 85           2           2
 86      6.0232331412E-04      2.4992629563E-02     -0.1680333964           0.3547287437
 87      1.5071114773E-02     -1.085305281           1.463477919          -0.4420149128
 88     -0.3804379360         -3.7538931702E-04      0.4864156073          -0.3325956359
 89      2.3179317243E-02      5.2824634632E-02     -2.2069389461E-02      4.6897990339E-02
 90     -8.1083111978E-02      3.3427044278E-02      3.9253042338E-02      -4.5857240261E-02
 91      1.1555579775E-02      3.3687312880E-03      3.1052129938-02       -8.1904083325E-03
 92      6.2067220491E-03     -2.6291031213E-03      6.6846421845E-04      -8.6587599360E-05
 93      2.9256415179E-08      1.0950852896E-06
 94    ------ BY --------
 95          24          39          16          31
 96           1           1
 97      3.556158398           20.78319797          -768.2792218           6382.365462
 98     -27542.51314           69600.71760          -95262.72456           16923.58616
 99      205598.3459          -458273.5720           555683.0477          -444206.0630
100      242484.8232          -87259.50457           17295.26260           157.2650049
101     -1065.295314           236.6317543          -2.659016784          -6.523814111
102      0.7069703250          6.3392822322E-02     -1.1903562204E-02      -2.4219449622E-04
103      7.0065701912E-05
104           1           2
105      1.192369622           10.77577660          -241.6873266           1500.114444
106     -4484.554844           5960.604461           2574.287423          -22679.79566
107      36940.73082          -28413.91110           8760.538975          -1696.851916
108      7932.547912          -11425.58820           6827.496335          -1087.433111
109     -1295.342412           1836.330802          -2095.856455           1489.358203
110      24.03061749          -1075.072823           950.8870878          -373.2118107
```

```
111      119.9892289          -176.3098310          223.6693352          -163.4798635
112      75.09231250          -21.25249774          2.565719319          0.5955192032
113     -0.3287545612         5.1635264237E-02      3.5601041603E-03     -2.6533411188E-03
114      3.4099858008E-04     9.1494428279E-06     -5.6961126952E-06     3.6119495461E-07
115            2     1
116      2.9020019071E-03     -1.3692229049E-02     0.1018788829         -0.6495862814
117      2.449958151          -5.667410116          8.278389181          -7.686799340
118      4.438370080          -1.449727292          0.1638492907         4.7649622224E-02
119     -1.6151472483E-02     3.8828666244E-04      3.2706575365E-04     -1.3361896663E-05
120     -3.2080471039E-06
121            2     2
122      9.6638406097E-04     -1.5895040154E-03     2.5377331923E-02     -0.1944465778
123      0.5697083259         -0.7902553257         0.4306915506         0.1084110145
124     -0.1447960166         -0.1035094066         0.1370941801         -1.8044456445E-02
125     -3.5222067655E-02     2.4781196987E-02     -1.8948357504E-02     2.4817641237E-02
126     -1.2538757330E-02     -1.4326587504E-02     2.1857366457E-02     -8.8152322924E-03
127     -1.1290758320E-04     -1.3399716155E-03     3.7842423376E-03     -3.0711946681E-03
128      1.3184528813E-03     -2.9671906888E-04     5.3199316126E-06     1.6893436478E-05
129     -4.0557661849E-06     9.5964275734E-09      1.3475625258E-07     -1.6538251657E-08
130           24    39           16    31
131            1     1
132      7.103890478          42.08501678          -1533.444820          12652.34064
133     -54171.11157          135425.0949          -181949.5040          26219.59051
134      397126.9938          -856654.6800          1007366.344          -776602.8185
135      405133.2878          -137318.1530          24751.64298          698.7574302
136     -1461.161539          269.3845731          5.211934966          -7.625394266
137      0.6056374303         7.5421444244E-02     -1.0856685425E-02     -2.4219449622E-04
138      7.0065701912E-05
139            1     2
140      2.384739221          21.72811662          -481.8514220          2965.561239
141     -8752.723229          11276.49226          5926.078371          -44481.63756
142      69200.43334          -49404.89484          11746.31898          -1391.624610
143      15065.10662          -20824.16698          11326.03885          -1132.392243
144     -2486.304904          3282.524043          -3817.503723          2595.534670
145      282.5916442          -2074.177692          1650.776274          -567.5434237
146      189.8361054          -337.3080663          409.4704962          -282.5674595
147      123.0885289          -33.29515053          4.031474817          0.6947433205
148     -0.3869369148         5.5531950555E-02      4.4226685710E-03     -2.7326394447E-03
149      3.3298100863E-04     9.1494403022E-06     -5.6961143855E-06     3.6119308788E-07
150            2     1
151      3.8455071283E-03     4.9553928175E-02     -0.5415504500         1.921547571
152     -2.038421532          -4.344862165          16.74010979          -24.57030391
153      20.55494998          -10.40751776          3.018414714          -0.3624761584
154     -3.6086218205E-02     1.2407473787E-02      9.4544281242E-05     -1.5635025273E-04
155     -3.2080471039E-06
156            2     2
157      1.5687073751E-03     2.3403125547E-02     -0.1426560645         0.1602821659
158      0.5847794406         -1.875560607          1.894169469          -0.3336038983
159     -0.5252339527         -0.1038847960         0.6235097873         -0.3506400924
160     -1.2042750413E-02     7.7605831619E-02     -4.1017746966E-02     7.1715631575E-02
161     -9.3621869308E-02     1.9100456774E-02      6.1110408796E-02     -5.4672472554E-02
162      1.1442672192E-02     2.0287596725E-03      6.8947636368E-03     -1.1261603001E-02
163      7.5251749303E-03     -2.9258221901E-03     6.7378415006E-04     -6.9694162883E-05
164     -4.0265097698E-06     1.1046817171E-06      1.3475625258E-07     -1.6538251657E-08
```

```
**********************************************************************************************
**********************************************************************************************
**************
**************   Pdn =   6            2299-MNI          Terminal: 62                16 JAN
**************
**********************************************************************************************
**********************************************************************************************

LISTING OF 2299MNI *ABLC-14  LAST UPDATED ON 16-JAN-84 AT 15:58:24

  1  8 7 6 5
  2  5 3 3 0
  3     0.0    0.0          34.46731430          -127.4427131
  4      151.0625789          4.794461592          -137.5682800          110.6236461
  5      -34.78404
  6     0.0          4.453754070          -43.04162022          147.6454078
  7     -242.8463943          205.3538822          -84.62088489          11.10560491
  8      0.0          -23.25947071          90.58333916          -176.4686526
  9      186.0516105          -107.3443700          30.49467355          0.0
 10     0.6523873876          -1.062334130          -0.7865093805          2.683868564
 11     -2.113167588          0.6913630998          0.0   0.0
 12    -1.3784434946E-03    -7.7797823480E-03    8.3117034640E-03    4.9767877851E-03
 13    -7.2579504312E-03    1.8579566640E-03
 14     8.6382416831E-05    6.4562763213E-03    -1.1092778602E-02    6.0171219775E-03
 15     3.0225540985E-03    -8.1466174596E-03    6.7393866548E-03    -1.7009274452E-03
 16     0.0
 17   EXTRACT X & Y FROM UNIMODULAR  VXY
 18           0          0          1          0          0          16          17          5
 19           4          24          25          13          5          26          27          15
 20           1          1
 21      0.000000000
 22           1          2
 23      1.000000000
 24           1          3
 25     0.1887844871          -0.2066691969
 26           1          4
 27      0.000000000
 28           2          1
 29      0.000000000
 30           2          2
 31    -0.9250081920          3.692977904          -1.939979428          -32.09055336
 32      135.1533656          -227.5161933          13.49620595          533.3661622
 33     -830.4023886          470.5763115          52.02824612          -224.2332885
 34      243.1355786          -307.1141973          284.5686364          -144.7091099
 35      33.33947666
 36           2          3
 37     0.1335702899          0.6509847786          -1.046223215          -5.771322941
 38      32.26674167          -70.88356540          49.56846327          97.90200733
 39     -266.9974454          260.4561024          -87.43150260          -53.08440220
 40      92.24233916          -108.2270310          117.1931886          -86.13040663
 41      36.20089152          -6.890242865
 42           2          4
 43     7.2828959292E-02    -0.1699506693          0.1137996376          0.2904734964
 44    -0.5335927939          0.2840288763
 45           3          1
 46      1.813480746          -23.21808079          -16.87668112          26.99425015
 47     -19.48195619
 48           3          2
 49    -0.5697981555          2.908452450          -4.867995782          -1.117095641
 50      62.28545195          -231.7659213          291.3205890          170.3128412
```

51	-938.7638790	1161.462225	-559.6504311	-220.5748879
52	759.6025684	-1212.532035	1437.178232	-1014.321136
53	173.9574280	404.5278307	-559.9059741	547.5374792
54	-464.6931342	292.9826781	-119.1164615	26.59542151
55	-2.380318555			
56	3 3			
57	8.2278303584E-02	0.3095097413	-1.270748653	0.7115642582
58	11.96134095	-56.52861407	102.5179380	-27.60243862
59	-212.7152367	413.4688995	-345.8546865	74.08794995
60	188.9786662	-385.8936912	521.9099775	-488.5085665
61	242.4693984	40.41693711	-189.3050040	219.0819002
62	-200.8859861	151.3483415	-83.03783487	29.63850643
63	-5.945821621	0.4919385240		
64	3 4			
65	9.7467848845E-03	-0.1244377423	0.2699239291	-9.6209878702E-03
66	-0.5503262381	0.7260728495	-0.3938940885	-7.7380467701E-05
67	0.1944687810	-0.4559895910	0.6685562771	-0.5213412146
68	0.1766519854	-2.0278638782E-02		
69	4 1			
70	92.40051275	215.1188439	-617.8989036	1985.688577
71	-1589.021855	859.3109678		
72	4 2			
73	7.1826855307E-11	-14.00003232	20.94122492	-52.71800724
74	261.7375100	-575.3810653	296.9944957	617.1848937
75	-1265.184968	-157.6070654	6957.734183	-25821.69362
76	55467.43865	-78653.65933	77290.21848	-48475.42940
77	3491.498015	37589.44764	-57512.30336	56779.84943
78	-45809.49930	31138.47010	-16347.84608	5888.233753
79	-1226.499446	107.3733766	-0.5617759370	
80	4 3			
81	-1.0371728566E-11	2.021591151	-2.703216128	-0.3295341925
82	38.63726059	-128.1416937	137.2640872	95.01463772
83	-401.2472680	255.3099261	1331.898235	-6303.828902
84	15804.82795	-26311.62289	30846.48084	-25124.91646
85	10677.51872	6374.719503	-18628.01165	22605.17630
86	-20382.76871	15345.87254	-9521.582346	4490.203410
87	-1448.460610	273.7500833	-22.29682410	0.1161017817
88	4 4			
89	-1.2286471997E-12	-0.1496920662	-6.8487592391E-02	0.7742881973
90	-1.369747319	1.185023429	1.238756352	-10.56316231
91	31.30925888	-48.89941393	55.01604535	-46.75140422
92	27.33112895	-8.212160357	0.9029635028	-4.7859355946E-03

```
**********************************************************************************
**********************************************************************************
**************
**************   Pdn =   6              2299-MNI          Terminal: 76                17
**************
**********************************************************************************
**********************************************************************************

LISTING OF 2299MNI *ABCXY-1  LAST UPDATED ON 17-JAN-84 AT 19:14:20

  1  8 7
  2  5 3
  3      0.0    0.0          34.46731430         -127.4427131
  4       151.0625789          4.794461592         -137.5682800          110.6236461
  5       -34.78404
  6      0.0          4.453754070         -43.04162022          147.6454078
  7      -242.8463943          205.3538822         -84.62088489          11.10560491
  8     -1.3784434946E-03   -7.7797823480E-03    8.3117034640E-03    4.9767877851E-03
  9     -7.2579504312E-03    1.8579566640E-03
 10      8.6382416831E-05    6.4562763213E-03   -1.1092778602E-02    6.0171219775E-03
 11  6 5
 12  3 0
 13      0.0         -23.25947071          90.58333916         -176.4686526
 14       186.0516105         -107.3443700          30.49467355          0.0
 15      0.6523873876         -1.062334130         -0.7865093805          2.683868564
 16      -2.113167588          0.6913630998         0.0    0.0
 17      3.0225540985E-03   -8.1466174596E-03    6.7393866548E-03   -1.7009274452E-03
 18  0.0
 19  0 0
 20  0 16
 21       0.000000000
 22       1.000000000
 23       0.000000000
 24     -0.9250081920          3.692977904         -1.939979428         -32.09055336
 25       135.1533656         -227.5161933          13.49620595          533.3661622
 26      -830.4023886          470.5763115          52.02824612         -224.2332885
 27       243.1355786         -307.1141973          284.5686364         -144.7091099
 28       33.33947666
 29  4 24
 30  5 26
 31      -1.813480746          23.21808079          16.87668112         -26.99425015
 32       19.48195619
 33      0.5697981555         -2.908452450          4.867995782          1.117095641
 34      -62.28545195          231.7659213         -291.3205890         -170.3128412
 35       938.7638790         -1161.462225          559.6504311          220.5748879
 36      -759.6025684          1212.532035         -1437.178232          1014.321136
 37      -173.9574280         -404.5278307          559.9059741         -547.5374792
 38       464.6931342         -292.9826781          119.1164615         -26.59542151
 39       2.380318555
 40      -92.40051275         -215.1188439          617.8989036         -1985.688577
 41       1589.021855         -859.3109678
 42     -7.1826855306E-11      14.00003232         -20.94122492          52.71800724
 43      -261.7375100          575.3810653         -296.9944957         -617.1848937
 44       1265.184968          157.6070654         -6957.734183          25821.69362
 45      -55467.43865          78653.65933         -77290.21848          48475.42940
 46      -3491.498015         -37589.44764          57512.30336         -56779.84943
 47       45809.49930         -31138.47010          16347.84608         -5888.233753
 48       1226.499446         -107.3733766         0.5617759370
 49    ------ AX -------
 50            0          23          0          19
```

```
 51          1          1
 52    0.000000000
 53          1          2
 54    0.000000000      -4.119759000       90.72878097      -431.6078685
 55     861.5251368       614.7542788      -10398.35275       36990.08883
 56    -71038.17374       61037.04290       54794.73902      -248628.7147
 57     373760.6436      -311372.1861       115252.7636       66479.68621
 58    -162505.2114       192455.2876      -178038.7747       127190.2232
 59    -65304.08796       22252.11074      -4428.298218       370.2550557
 60          2          1
 61    0.000000000
 62          2          2
 63  -1.4583479378E-03  -1.3432882478E-02   4.2247920227E-02  -5.6851588409E-02
 64   -0.1590276951       1.199092343      -3.160060712       3.470229740
 65    1.853124047       -11.15596219       15.46346413      -9.900092381
 66    0.8276678186       4.343651740      -6.004520585       6.694479656
 67    -5.936002522       3.532759188      -1.240559799       0.2006076977
 68   ----- BY -------
 69         10         31          7          27
 70          1          1
 71    -60.28092913       2.2351741791E-08  -1.1548399925E-07   4.9360096455E-08
 72   -7.4505805969E-08   3.9348378778E-08  -8.7730586529E-07   1.1790543795E-06
 73   -1.5050172806E-06   8.4936618805E-07  -2.6449561119E-07
 74          1          2
 75   -4.6858934493E-11  -4.119758997        90.72878097      -431.6078685
 76     861.5251368       614.7542787      -10398.35275       36990.08883
 77    -71038.17375       61037.04293       54794.73897      -248628.7230
 78     373760.7028      -311372.3508       115252.7334       66481.57937
 79    -162512.4539       192469.2000      -178050.0162       127177.9558
 80    -65255.08590       22182.46268      -4374.060781       353.6770282
 81    -15.57036948       31.56716728      -35.80224037       31.67368633
 82    -21.17581260       9.829638343      -2.834079905       0.3883911532
 83          2          1
 84   -5.4813436614E-03   8.4951639157E-02  -0.1503598886      -5.9519223503E-02
 85    0.3530433048      -0.3693427438       0.1772116965      -3.3137393970E-02
 86          2          2
 87    1.7222457502E-03  -1.3432882475E-02   4.2247920197E-02  -5.6851588372E-02
 88   -0.1596073265       1.207189217       -3.190306384       3.526400812
 89    1.867399069       -11.81062152       17.76994469      -13.31687071
 90    1.654384532       10.38774990       -19.71642090       24.23773286
 91   -20.53721083       9.474895594       2.090226558      -8.646681936
 92    10.32663386      -9.313659413       6.809921982      -3.815715202
 93    1.524971536      -0.4012368320       6.1278769466E-02  -4.0487491585E-03
 94         10         31          7          27
 95          1          1
 96    -60.28092913       2.2351741791E-08  -1.1548399925E-07   4.9360096455E-08
 97   -7.4505805969E-08   3.9348378778E-08  -8.7730586529E-07   1.1790543795E-06
 98   -1.5050172806E-06   8.4936618805E-07  -2.6449561119E-07
 99          1          2
100   -4.6858934493E-11  -8.239517997        181.4575619      -863.2157371
101     1723.050274       1229.508557      -20796.70551       73980.17766
102    -142076.3475       122074.0858       109589.4780      -497257.4377
103     747521.3464      -622744.5370       230505.4970       132961.2656
104    -325017.6653       384924.4876      -356088.7910       254368.1790
105    -130559.1739       44434.57341      -8802.358999       723.9320839
106    -15.57036948       31.56716728      -35.80224037       31.67368633
107    -21.17581260       9.829638343      -2.834079905       0.3883911532
108          2          1
109   -5.4813436614E-03   8.4951639157E-02  -0.1503598886      -5.9519223503E-02
110    0.3530433048      -0.3693427438       0.1772116965      -3.3137393970E-02
```

```
111          2          2
112    2.6389781240E-04  -2.6865764953E-02   8.4495840424E-02  -0.1137031768
113   -0.3186350216       2.406281560       -6.350367096       6.996630552
114    3.720523116      -22.96658371        33.23340882      -23.21696309
115    2.482052351       14.73140164       -25.72094148       30.93221251
116   -26.47321335       13.00765478         0.8496667591     -8.446074239
117    10.32663386       -9.313659413        6.809921982      -3.815715202
118    1.524971536       -0.4012368320       6.1278769466E-02  -4.0487491585E-03

**********************************************************************************
**********************************************************************************
**************
**************   No. of pages     3   2299-MNI          Terminal: 76              17
**************
**********************************************************************************
**********************************************************************************
```

```
**********************************************************************************************
**********************************************************************************************
**************
**************   Pdn =   6            2299-MNI          Terminal: 76                17 JAN 8
**************
**********************************************************************************************
**********************************************************************************************

LISTING OF 2299MNI *ABLC-16  LAST UPDATED ON 17-JAN-84 AT 21:17:30

  1  10 8 6 5
  2  5 3 3 0
  3     0.0  -2.8843          34.46731430         -127.4427131
  4      151.0625789          4.794461592         -137.5682800          110.6236461
  5       -34.78404  2.35575  .87567
  6     .32481          4.453754070        -43.04162022          147.6454078
  7     -242.8463943          205.3538822         -84.62088489          11.10560491
  8     2.021624
  9      0.0         -23.25947071          90.58333916         -176.4686526
 10      186.0516105         -107.3443700          30.49467355         0.0
 11     0.6523873876         -1.062334130        -0.7865093805          2.683868564
 12     -2.113167588         0.6913630998        0.0  0.0
 13    -1.3784434946E-03  -7.7797823480E-03   8.3117034640E-03   4.9767877851E-03
 14    -7.2579504312E-03   1.8579566640E-03
 15     8.6382416831E-05   6.4562763213E-03  -1.1092778602E-02   6.0171219775E-03
 16     3.0225540985E-03  -8.1466174596E-03   6.7393866548E-03  -1.7009274452E-03
 17      -7.923725E-05
 18  0 0
 19  0 0
 20  1.0
 21  0.0
 22  0.0
 23  1.0
 24   EXTRACT X & Y FROM UNIMODULAR  VXY
 25           0          0          0          0          5         20          6
 26           4         14         29         15          5         15         30         16
 27           1          1
 28     0.000000000
 29           1          2
 30     0.000000000
 31           1          3
 32     1.000000000
 33           1          4
 34     0.000000000
 35           2          1
 36     0.000000000
 37           2          2
 38     1.462832575         -4.178716165          2.751662061          6.861759566
 39    -13.16300285          6.854392123
 40           2          3
 41     5.244321530          13.61967147         -76.06919015         -184.6993399
 42     1944.532144         -5684.276661          6191.048474          8871.474801
 43    -41435.87466          57069.05209         -7730.069191         -88760.91710
 44     135105.2773         -73564.08208         -29530.43477          76055.89338
 45    -55663.92991          20986.26574         -3659.876761         -28.85523092
 46     82.12737494
 47           2          4
 48    5.2615661989E-03   0.1097812370        -0.2905895073          0.1540781071
 49    0.5281199703        -0.9324771268         0.4855703481
 50           3          1
```

```
 51      1.813480746          -23.21808079          -16.87668112          26.99425015
 52     -19.48195619
 53            3          2
 54     8.5402989905E-02      -2.775049825           6.545886404         -0.4432049035
 55     -13.25513990           17.62137041          -9.419257521         -9.2802511964E-02
 56      4.793155664          -10.85437988           16.46300548         -12.51510056
 57      4.449783212          -0.3704406524         -0.1051043512
 58            3          3
 59     0.3061736158          -8.192153595          -33.18408481          134.6605516
 60      467.2483881          -3574.703942           8854.680530         -6231.049277
 61     -22048.16086           68654.90979          -72967.43779         -20812.46672
 62      159350.5874          -206780.4897           109169.4882          45612.67320
 63     -155344.7532           196718.0682          -174480.5760          86179.74864
 64      30095.60520          -100276.8788           94831.89507         -51666.25289
 65      16842.90940          -2685.103540          -93.84452872          99.93812334
 66     -6.414438660          -1.259330412
 67            3          4
 68     7.7054643552E-03      6.3920035224E-03      -0.2068342414         0.4413369470
 69     2.2446464128E-02      -0.9767706489          1.241749601         -0.6379491147
 70     -3.5036685228E-02      0.3918370007         -0.8223493288          1.190769763
 71     -0.8894617590          0.3145815777         -2.6242297387E-02     -7.4456721268E-03
 72            4          1
 73      92.40051275           215.1188439          -617.8989036          1985.688577
 74     -1589.021855           859.3109678
 75            4          2
 76     -0.7283136638         -6.047161434           2.147008368           12.03924579
 77     -23.48880200           14.12764287          41.17932871          -265.0049772
 78      756.0361453          -1185.893345           1326.369848          -1146.942952
 79      654.7425245          -200.8478496           14.95682934           3.687319655
 80            4          3
 81     -2.611037718          -31.49782239          -150.9656024          282.4364345
 82      1416.706158          -7462.285499           14243.30982          -693.0920374
 83     -50198.29018           85361.98982          45924.66796          -506754.9612
 84      1316323.671          -2051984.180           1529756.777          1545742.575
 85     -6723900.244           10999009.16          -10909512.30          5655599.076
 86      1942264.246          -7481245.424           8431260.600         -5874899.371
 87      2720505.602          -799515.7449           118279.4795          4035.748537
 88     -4005.256049           248.5287622          44.18041431
 89            4          4
 90     -2.6196234794E-03      0.1798783113         -0.8098983667         0.6905670649
 91      1.1534656005E-02     -0.2150789589         -0.8212210330         5.840665737
 92     -23.19620192           58.13655150          -88.33391100          96.90876289
 93     -82.37337232           46.49560710          -14.20560553          1.059553159
 94      0.2612125271
```

****

****

********** No. of pages 2 2299-MNI Terminal: 76 .

****

****

****

```
**********************************************************************************************
**********************************************************************************************
**************
**************   Pdn =   6            2299-MNI          Terminal: 80                24 JA
**************
**********************************************************************************************
**********************************************************************************************

LISTING OF 2299MNI *ABCXY-6  LAST UPDATED ON 24-JAN-84 AT 12:31:38

  1  10 8
  2  5 3
  3     0.0  -2.8843          34.46731430         -127.4427131
  4      151.0625789          4.794461592         -137.5682800          110.6236461
  5       -34.78404  2.35575  .87567
  6     .32481          4.453754070        -43.04162022          147.6454078
  7     -242.8463943          205.3538822         -84.62088489          11.10560491
  8        2.021624
  9    -1.3784434946E-03  -7.7797823480E-03   8.3117034640E-03   4.9767877851E-03
 10    -7.2579504312E-03   1.8579566640E-03
 11     8.63824E-05   6.4562763213E-03  -1.1092778602E-02   6.0171219775E-03
 12     6 5
 13     3 0
 14       0.0        -23.25947071          90.58333916         -176.4686526
 15       186.0516105         -107.3443700          30.49467355         0.0
 16     0.6523873876         -1.062334130        -0.7865093805          2.683868564
 17     -2.113167588         0.6913630998
 18     3.0225540985E-03  -8.1466174596E-03   6.7393866548E-03  -1.7009274452E-03
 19     0.0
 20   0 0
 21   0 5
 22      0.000000000
 23      0.000000000
 24      0.000000000
 25      1.462832537         -4.178716049          2.751661978          6.861759383
 26     -13.16300246          6.854391916
 27   4 12
 28   5 13
 29      1.813480746         -23.21808079         -16.87668112          26.99425015
 30     -19.48195619
 31     8.5402996054E-02    -2.775049758          6.545886201         -0.4432048552
 32     -13.25513954          17.62136990         -9.419257236        -9.2802515486E-02
 33      4.793155530         -10.85437958          16.46300500         -12.51510019
 34      4.449783079        -0.3704406410        -0.1051043481
 35      92.40051275          215.1188439         -617.8989036          1985.688577
 36     -1589.021855          859.3109678
 37    -0.7283136463         -6.047160985          2.147007495          12.03924659
 38     -23.48880229          14.12764277          41.17932743         -265.0049702
 39      756.0361245         -1185.893311          1326.369810         -1146.942919
 40      654.7425051         -200.8478436          14.95682887          3.687319543
 41   ------ AX ------
 42            0      13          0          8
 43            1      1
 44      0.000000000
 45            1      2
 46     0.4751426363          5.157807606         -80.67988879          410.3232095
 47     -1064.362718          1369.715078         -39.94094564         -2969.917765
 48      5341.398160         -4926.175091          2603.207721         -712.3359166
 49      49.51152683          13.85700320
 50            2      1
```

```
 51       0.000000000
 52            2           2
 53     1.2636298534E-04   9.0834835493E-03  -4.2968127780E-02   7.3513839096E-02
 54    -1.2503058403E-02  -0.1439507742       0.2315561634      -0.1552376434
 55     4.1243712240E-02
 56    ------ BY --------
 57           10          18          7          15
 58            1           1
 59      60.28092913       -2.4680048227E-08   1.0197982192E-07  -7.7299773693E-08
 60    -3.4458935261E-08   -2.0535662770E-07   5.1874667406E-07  -1.5702098608E-06
 61     1.3783574104E-06   -9.0152025223E-07   2.5704503059E-07
 62            1           2
 63    -0.4751426371         -5.157807599        80.67988875      -410.3232094
 64     1064.362718         -1369.715078         39.94094524       2969.917764
 65    -5341.398157          4926.175087        -2603.207718       712.1962409
 66    -48.92213630         -15.20204074        -16.67229341       43.98385284
 67    -42.23504894          17.78690512        -3.164105379
 68            2           1
 69     5.4813436613E-03   -8.4951639159E-02   0.1503598886       5.9519223500E-02
 70    -0.3530433048         0.3693427438      -0.1772116965       3.3137393970E-02
 71            2           2
 72     2.5813517575E-04   -9.0834835583E-03   4.2968127786E-02   -7.3513839071E-02
 73     1.2381660470E-02    0.1471250886      -0.2606024255        0.2177578406
 74    -7.8209089715E-02   -5.6459912902E-02   0.1706475802       -0.2532500420
 75     0.2448185121       -0.1485971568       5.1276086092E-02   -7.5687581642E-03
 76           10          18          7          15
 77            1           1
 78      60.28092913       -2.4680048227E-08   1.0197982192E-07  -7.7299773693E-08
 79    -3.4458935261E-08   -2.0535662770E-07   5.1874667406E-07  -1.5702098608E-06
 80     1.3783574104E-06   -9.0152025223E-07   2.5704503059E-07
 81            1           2
 82    -7.2213879321E-10    6.6065695137E-09  -3.6787241697E-08   6.1467289925E-08
 83    -2.0861625671E-07    4.2468309403E-07  -4.0209852159E-07  -1.0132789612E-06
 84     2.8610229492E-06   -4.3809413910E-06   2.5779008865E-06  -0.1396757253
 85     0.5893905358        -1.345037540       -16.67229341        43.98385284
 86    -42.23504894          17.78690512       -3.164105379
 87            2           1
 88     5.4813436613E-03   -8.4951639159E-02   0.1503598886       5.9519223500E-02
 89    -0.3530433048         0.3693427438      -0.1772116965       3.3137393970E-02
 90            2           2
 91     3.8449816109E-04   -9.0381035988E-12   6.5938365879E-12   2.4556356948E-11
 92    -1.2139793256E-04    3.1743144673E-03  -2.9046262040E-02   6.2520197239E-02
 93    -3.6965377475E-02   -5.6459912902E-02   0.1706475802      -0.2532500420
 94     0.2448185121       -0.1485971568       5.1276086092E-02   -7.5687581642E-03
```

```
**********************************************************************************************
**********************************************************************************************
*************
*************   Pdn =   6              2299-MNI          Terminal: 63                  19 JAN
*************
**********************************************************************************************
**********************************************************************************************


ISTING OF 2299MNI *ABLC1-4* LAST UPDATED ON 19-JAN-84 AT 14:50:36


  1  10 8 6 5
  2  5 3 3 0
  3     0.0  -2.8843          34.46731430         -127.4427131
  4      151.0625789          4.794461592         -137.5682800          110.6236461
  5       -34.78404   2.35575   .87567
  6     .32481          4.453754070         -43.04162022          147.6454078
  7     -242.8463943          205.3538822         -84.62088489          11.10560491
  8        2.021624
  9      0.0         -23.25947071          90.58333916         -176.4686526
 10      186.0516105         -107.3443700          30.49467355         0.0
 11     0.6523873876         -1.062334130         -0.7865093805          2.683868564
 12     -2.113167588          0.6913630998
 13    -1.3784434946E-03   -7.7797823480E-03    8.3117034640E-03    4.9767877851E-03
 14    -7.2579504312E-03    1.8579566640E-03
 15     0.0    6.4562763213E-03   -1.1092778602E-02    6.0171219775E-03
 16     3.0225540985E-03   -8.1466174596E-03    6.7393866548E-03   -1.7009274452E-03
 17     -7.9237E-05
 18  0 0
 19  0 0
 20  1.0
 21  0.0
 22  0.0
 23  1.E02
 24   EXTRACT X & Y FROM UNIMODULAR  VXY
 25            0          0          1          0          0         22         23          6
 26            4         22         23          7          5         25         26         10
 27            1          1
 28      0.000000000
 29            1          2
 30      1.000000000
 31            1          3
 32     -73.63614915          138.0624846
 33            1          4
 34      0.000000000
 35            2          1
 36      0.000000000
 37            2          2
 38      6.495802313         -48.54934600          114.2459289          145.1054937
 39     -1933.546765          6784.309020         -12372.50887          9356.227222
 40      8984.574493         -31560.64081          36100.50926         -16127.22100
 41     -10157.54144          22162.07718         -17670.16052          7885.973877
 42     -1608.615342         -275.8499337          281.9059055         -76.09516406
 43      6.705747984          0.7701166697        -0.1481247055
 44            2          3
 45      367.2378438          272.1431265         -8910.249977          1304.351698
 46      163174.5672         -766106.0475          1847587.630         -2397135.857
 47      630154.5090          3564436.731         -7015642.969          6171672.453
 48     -1478601.964         -3034305.429          4360914.014         -3020279.013
 49      1207209.386         -201776.9039         -58842.99252          44523.98458
 50     -10999.67287          869.1038015          117.2315537         -20.45046487
```

```
 51          2          4
 52  -7.4585496161E-05   2.7141505907E-04  -2.7842920903E-04  -2.6573920991E-04
 53   7.6756995884E-04  -6.7417252004E-04   2.8872349306E-04
 54          3          1
 55   1.813480746       -23.21808079       -16.87668112        26.99425015
 56  -19.48195619
 57          3          2
 58  -0.6787900264      -14.31632626        91.82474771       -158.7580934
 59  -461.2314648        3733.280101       -11089.38503        16604.81990
 60  -6476.430443       -22177.35217        46346.45469       -39461.74280
 61   6254.985454        22684.10143       -27922.84093        16728.16390
 62  -5202.951833        160.0456780        545.6056393       -209.7553374
 63   27.64603992        1.466560876       -0.5442447619
 64          3          3
 65  -38.37514962       -1135.915642       -1956.585456        16991.33188
 66   14946.69271       -338300.5091        1331510.110       -2753743.044
 67   2769402.089        738902.7345       -6474634.792        9304517.464
 68  -5908779.298       -806791.0439        5187953.883       -5086894.366
 69   2692657.207       -730117.6046       -18079.99429        90773.24546
 70  -30995.09095        3708.889064        242.5531268       -75.13978404
 71          3          4
 72  -5.0051559683E-06   1.5253986106E-04  -4.3222977010E-04   2.0431077644E-04
 73   6.3707932449E-04  -1.0462127088E-03   6.5117558068E-04  -1.6347301152E-04
 74          4          1
 75   92.40051275        215.1188439       -617.8989036        1985.688577
 76  -1589.021855        859.3109678
 77          4          2
 78  -3.234123759       -45.22023306        156.5870239       -38.22590322
 79  -1103.342511        5194.294422       -8169.115833       -3756.584935
 80   32076.50853       -47108.45523        21920.34249        24540.53680
 81  -50913.12328        42600.12905       -15253.06566       -9654.582121
 82   19081.82669       -13769.06457        4360.496604        616.2924935
 83  -1021.104703        269.4320509        27.15824371       -23.29330843
 84   1.450002595        0.4320240689
 85          4          3
 86  -182.8400217       -4634.046454       -11628.84690        27155.67106
 87   70943.47802       -532345.8346        1315644.821       -849703.4736
 88  -2878478.688        7897840.970       -8118039.981        1219306.319
 89   7137173.822       -10166101.75        7004656.677       -1394949.892
 90  -2738047.831        3648385.295       -2222081.443        556639.5891
 91   160277.0911       -160816.1910        35198.62989        5464.764136
 92  -3322.704643        168.3783721        59.64631634
 93          4          4
 94   3.7134554525E-05   2.5607355548E-04  -3.9910962784E-05  -1.0334315327E-03
 95   1.1966225952E-03  -1.5316995106E-04  -3.6374266020E-04   3.4734554741E-04
 96  -5.8517686355E-04   2.6878890303E-04   1.2966729197E-04
```

```
**********
**********   Pdn =   6          2299-MNI          Terminal: 63          19
**********
```

LISTING OF 2299MNI *ABLC1-41 LAST UPDATED ON 19-JAN-84 AT 14:53:41

```
  1  10 8 6 5
  2  5 3 3 0
  3     0.0  -2.8843          34.46731430         -127.4427131
  4      151.0625789          4.794461592         -137.5682800          110.6236461
  5       -34.78404  2.35575  .87567
  6     .32481          4.453754070         -43.04162022          147.6454078
  7     -242.8463943          205.3538822         -84.62088489          11.10560491
  8        2.021624
  9      0.0         -23.25947071          90.58333916         -176.4686526
 10      186.0516105         -107.3443700          30.49467355         0.0
 11     0.6523873876         -1.062334130         -0.7865093805         2.683868564
 12     -2.113167588         0.6913630998
 13    -1.3784434946E-03  -7.7797823480E-03   8.3117034640E-03   4.9767877851E-03
 14    -7.2579504312E-03   1.8579566640E-03
 15     0.0   6.4562763213E-03  -1.1092778602E-02   6.0171219775E-03
 16     3.0225540985E-03  -8.1466174596E-03   6.7393866548E-03  -1.7009274452E-03
 17     -7.9237E-05
 18  0 0
 19  0 0
 20  1.0
 21  0.0
 22  0.0
 23  1.E05
 24   EXTRACT X & Y FROM UNIMODULAR  VXY
 25           0          0          0          0          0          6         23          7
 26           4          7         24          8          5         10         27         11
 27           1          1
 28      0.000000000
 29           1          2
 30      0.000000000
 31           1          3
 32      1.000000000
 33           1          4
 34      0.000000000
 35           2          1
 36      0.000000000
 37           2          2
 38    -7.4583994187E-05   2.7140959216E-04  -2.7842359498E-04  -2.6573386225E-04
 39     7.6755449403E-04  -6.7415893261E-04   2.8871767249E-04
 40           2          3
 41      5.689398217          11.77014642         -115.4381706         -75.71885817
 42      2414.210969         -9459.999840          17818.04496         -9220.085982
 43     -36869.34039          99563.17256         -104370.5825          2967.829937
 44      139993.5906         -196881.9337          121092.9702          7164.026084
 45     -83904.52958          83262.56538         -46871.13489          16654.97934
 46     -3487.839231          278.0654099          38.39796955         -7.691020692
 47           2          4
 48      1810.220290         -9518.687836          16387.93395         -7098.548723
 49     -15609.15474          27746.79028         -20217.06374          7001.279401
 50           3          1
```

```
 51      1.813480746          -23.21808079          -16.87668112          26.99425015
 52     -19.48195619
 53              3       2
 54   -5.0050550593E-06    1.5253678967E-04   -4.3222106214E-04    2.0430766227E-04
 55    6.3706610865E-04   -1.0461918525E-03    6.5116432384E-04   -1.6345452918E-04
 56              3       3
 57      0.3817944003          -9.369811083          -36.83335857          207.6568470
 58       387.8033432          -4834.574136           15785.38685         -23919.90230
 59      -172.0056318           74077.25898          -143849.9139           104880.2392
 60       62557.83249          -224379.2860           231066.9756          -90922.52435
 61      -58011.00938           113885.7461          -87780.58101           41660.18093
 62      -12687.66166           2241.527579          -122.0518267          -28.91760262
 63      4.354191953
 64              3       4
 65      -23.62703369          -4190.661821           16070.84540          -20500.32766
 66      -383.1984306           29262.21158          -34287.04588           17904.51035
 67      -3963.702067
 68              4       1
 69       92.40051275           215.1188439          -617.8989036           1985.688577
 70      -1589.021855           859.3109678
 71              4       2
 72    3.7133806698E-05    2.5606840364E-04   -3.9910147393E-05   -1.0334107154E-03
 73    1.1965984680E-03   -1.5316683098E-04   -3.6373547361E-04    3.4734216767E-04
 74   -5.8513543803E-04    2.6878787629E-04    1.2959208170E-04
 75              4       3
 76      -2.832632067          -31.28019973          -141.7734887           344.1994037
 77       1304.902329          -7268.741302           13196.20104           4605.033937
 78      -64722.20561           110959.1344          -34128.17819          -158474.3399
 79       282441.5237          -191059.4538          -37999.37045           212481.2893
 80      -230041.2032           125335.4959           1904.039410          -70135.06466
 81       66327.68731          -31956.26056           6833.365844           976.0260918
 82      -958.5516672           190.0259494           2.014165025          -3.452145389
 83              4       4
 84      -901.2707223          -9928.938026           12870.69060           15120.04054
 85      -45561.04497           33470.25597          -2044.491883          -11788.34259
 86       18145.67146          -17272.55019           4841.898369           3142.552273
```

```
**********
**********  Pdn =  6          2299-MNI          Terminal: 63          19 JAN 84
**********
```

.ISTING OF 2299MNI *ABLC1-42 LAST UPDATED ON 19-JAN-84 AT 14:56:17

```
 1  10 8 6 5
 2  5 3 3 0
 3     0.0  -2.8843          34.46731430         -127.4427131
 4      151.0625789          4.794461592         -137.5682800          110.6236461
 5       -34.78404  2.35575  .87567
 6     .32481          4.453754070         -43.04162022          147.6454078
 7     -242.8463943          205.3538822         -84.62088489          11.10560491
 8        2.021624
 9      0.0         -23.25947071          90.58333916         -176.4686526
10      186.0516105         -107.3443700          30.49467355         0.0
11     0.6523873876         -1.062334130         -0.7865093805         2.683868564
12     -2.113167588         0.6913630998
13    -1.3784434946E-03  -7.7797823480E-03    8.3117034640E-03    4.9767877851E-03
14    -7.2579504312E-03    1.8579566640E-03
15     0.0    6.4562763213E-03  -1.1092778602E-02    6.0171219775E-03
16     3.0225540985E-03  -8.1466174596E-03    6.7393866548E-03  -1.7009274452E-03
17     -7.9237E-05
18  0 0
19  0 0
20  1.0
21  0.0
22  0.0
23  1.E04
24   EXTRACT X & Y FROM UNIMODULAR  VXY
25            0          0          0          0          0          6         23          7
26            4          7         24          8          5         10         27         11
27            1          1
28      0.000000000
29            1          2
30      0.000000000
31            1          3
32      1.000000000
33            1          4
34      0.000000000
35            2          1
36      0.000000000
37            2          2
38    -7.4583939661E-05   2.7140939434E-04  -2.7842339713E-04  -2.6573366546E-04
39     7.6755394117E-04  -6.7415845101E-04   2.8871746779E-04
40            2          3
41      5.689398070          11.77014607         -115.4381674         -75.71885377
42      2414.210905         -9459.999630          17818.04465         -9220.086011
43     -36869.33938          99563.17075         -104370.5814          2967.831004
44      139993.5878         -196881.9312          121092.9695          7164.025206
45     -83904.52828          83262.56448         -46871.13450          16654.97923
46     -3487.839215          278.0654094          38.39796931         -7.691020663
47            2          4
48      1810.221566         -9518.694547          16387.94551         -7098.553730
49     -15609.16575          27746.80985         -20217.07800          7001.284312
50            3          1
```

```
 51      1.813480746         -23.21808079         -16.87668112          26.99425015
 52     -19.48195619
 53            3             2
 54     -5.0050515425E-06     1.5253667766E-04    -4.3222074893E-04     2.0430756138E-04
 55      6.3706562081E-04    -1.0461911109E-03     6.5116394823E-04    -1.6345584070E-04
 56            3             3
 57      0.3817944012         -9.369810739        -36.83335728          207.6568389
 58       387.8033127         -4834.573994         15785.38666          -23919.90270
 59      -172.0026245          74077.23520        -143849.9197          104880.2743
 60       62557.76405         -224379.2395         231067.0376          -90922.69146
 61      -58010.86388          113885.7274        -87780.66230          41660.26962
 62      -12687.70864          2241.541225        -122.0535319          -28.91770586
 63       4.354229960
 64            3             4
 65      -23.62705072         -4190.664776         16070.85674          -20500.34324
 66      -383.1970004          29262.23083        -34287.07176          17904.54356
 67      -3963.739462
 68            4             1
 69       92.40051275          215.1188439        -617.8989036          1985.688577
 70      -1589.021855          859.3109678
 71            4             2
 72      3.7133779543E-05     2.5606821049E-04    -3.9910127867E-05    -1.0334099638E-03
 73      1.1965976215E-03    -1.5316672339E-04    -3.6373530698E-04     3.4734212493E-04
 74     -5.8513802782E-04     2.6878597550E-04     1.2970946019E-04
 75            4             3
 76      -2.832631993         -31.28019835        -141.7734828          344.1993988
 77       1304.902287         -7268.741132         13196.20082          4605.033603
 78      -64722.20390          110959.1338        -34128.18518          -158474.4071
 79       282441.6099         -191059.0961        -38001.55234          212486.1879
 80      -230045.1143          125331.3067         1916.906040          -70146.82692
 81       66329.54137         -31949.93388         6826.253248          979.8413957
 82      -959.6690752          190.1667814         2.022526726          -3.455274619
 83            4             4
 84      -901.2713574         -9928.945018         12870.69969          15120.05118
 85      -45561.07708          33470.27948        -2044.490614          -11788.35981
 86       18145.78209         -17272.59592         4840.340902          3145.403087
```

```
**********************************************************************************************
**********************************************************************************************
**************
**************   Pdn =   6            2299-MNI          Terminal: 63                  19 JAI
**************
**********************************************************************************************
**********************************************************************************************

LISTING OF 2299MNI *ABLC1-43 LAST UPDATED ON 19-JAN-84 AT 15:00:49

  1  10 8 6 5
  2  5 3 3 0
  3     0.0  -2.8843          34.46731430        -127.4427131
  4      151.0625789          4.794461592        -137.5682800          110.6236461
  5       -34.78404  2.35575  .87567
  6     .32481          4.453754070        -43.04162022          147.6454078
  7     -242.8463943          205.3538822        -84.62088489          11.10560491
  8        2.021624
  9      0.0         -23.25947071          90.58333916        -176.4686526
 10      186.0516105         -107.3443700          30.49467355        0.0
 11     0.6523873876         -1.062334130        -0.7865093805         2.683868564
 12     -2.113167588         0.6913630998
 13    -1.3784434946E-03  -7.7797823480E-03    8.3117034640E-03    4.9767877851E-03
 14    -7.2579504312E-03    1.8579566640E-03
 15     0.0    6.4562763213E-03  -1.1092778602E-02    6.0171219775E-03
 16     3.0225540985E-03  -8.1466174596E-03    6.7393866548E-03  -1.7009274452E-03
 17     -7.9237E-05
 18  0 0
 19  0 0
 20  1.0
 21  0.0
 22  0.0
 23  5.E03
 24   EXTRACT X & Y FROM UNIMODULAR  VXY
 25           0          0          1          0          0         22         23          6
 26           4         22         23          7          5         25         26         10.
 27           1          1
 28      0.000000000
 29           1          2
 30      1.000000000
 31           1          3
 32     -73.63768588          138.0653658
 33           1          4
 34      0.000000000
 35           2          1
 36      0.000000000
 37           2          2
 38      6.495802301         -48.54934593          114.2459289          145.1054926
 39     -1933.546761         6784.309015         -12372.50888          9356.227282
 40      8984.574396         -31560.64077          36100.50936         -16127.22122
 41     -10157.54126          22162.07714         -17670.16061          7885.973992
 42     -1608.615419         -275.8499002          281.9058963         -76.09516264
 43      6.705747941          0.7701166487        -0.1481247031
 44           2          3
 45      367.2455069          272.1488039         -8910.435904          1304.378960
 46      163177.9722         -766122.0346          1847626.187         -2397185.889
 47      630167.6748          3564511.101         -7015789.380          6171801.279
 48     -1478632.864         -3034368.724          4361005.023         -3020342.063
 49      1207234.601         -201781.1280         -58844.21521          44524.91236
 50     -10999.90222          869.1219343          117.2339971         -20.45089132
```

```
 2          4
-7.4583939661E-05   2.7140939434E-04  -2.7842339713E-04  -2.6573366546E-04
 7.6755394117E-04  -6.7415845101E-04   2.8871746779E-04
 3          1
 1.813480746        -23.21808079        -16.87668112        26.99425015
-19.48195619
 3          2
-0.6787900257       -14.31632623         91.82474757       -158.7582129
-461.2311030         3733.280020        -11089.38885         16604.83731
-6476.464708        -22177.32251         46346.47331        -39461.83026
 6255.095403         22684.05220        -27922.87901         16728.23700
-5203.001137          160.0603282         545.6065150        -209.7573543
 27.64655050          1.466554909        -0.5442544178
 3          3
-38.37595043        -1135.939345        -1956.626278         16991.67970
 14946.96942        -338307.5373         1331538.159        -2753802.321
 2769464.810         738911.2399        -6474767.187         9304720.650
-5908922.778        -806789.0753         5188058.157        -5087011.166
 2692727.124        -730140.7272        -18078.41340         90775.40922
-30996.05384         3709.037398         242.5580757        -75.14268526
 3          4
-5.0050515425E-06   1.5253667766E-04  -4.3222074893E-04   2.0430756138E-04
 6.3706562081E-04  -1.0461911109E-03   6.5116394823E-04  -1.6345584070E-04
 4          1
 92.40051275         215.1188439        -617.8989036         1985.688577
-1589.021855         859.3109678
 4          2
-3.234123748        -45.22023292         156.5870236        -38.22590417
-1103.342506         5194.294381        -8169.115804        -3756.585245
 32076.50542        -47108.44418         21920.33410         24540.47999
-50912.81376         42599.65427        -15253.10788        -9653.617306
 19080.74500        -13768.86786         4361.089655         615.6859916
-1020.868098         269.4247748         27.13215700        -23.28549289
 1.449820950         0.4318759017
 4          3
-182.8438367        -4634.143148        -11629.08952         27156.23774
 70944.95801        -532356.9440         1315672.265        -849721.2492
-2878539.025         7898004.413        -8118207.254         1219334.788
 7137292.133        -10166236.21         7004740.415        -1395055.880
-2737892.111         3648297.604        -2222144.327         556777.7467
 160179.2760        -160786.3443         35200.28082         5460.700991
-3321.681553         168.3677178         59.62710434
 4          4
 3.7133779543E-05   2.5606821049E-04  -3.9910127867E-05  -1.0334099638E-03
 1.1965976215E-03  -1.5316672339E-04  -3.6373530698E-04   3.4734212493E-04
-5.8513802782E-04   2.6878597550E-04   1.2970946019E-04
```

```
**********************************************************************************************
**********************************************************************************************
**************
**************   Pdn =   6            2299-MNI          Terminal: 80                  24 JA
**************
**********************************************************************************************
**********************************************************************************************

LISTING OF 2299MNI *ABXYC3   LAST UPDATED ON 24-JAN-84 AT 10:50:34

  1  10 8
  2  5 3
  3     0.0  -2.8843          34.46731430         -127.4427131
  4      151.0625789          4.794461592         -137.5682800          110.6236461
  5       -34.78404  2.35575  .87567
  6     .32481          4.453754070        -43.04162022          147.6454078
  7     -242.8463943          205.3538822         -84.62088489          . 11.10560491
  8        2.021624
  9    -1.3784434946E-03  -7.7797823480E-03   8.3117034640E-03   4.9767877851E-03
 10    -7.2579504312E-03   1.8579566640E-03
 11     0.0   6.4562763213E-03  -1.1092778602E-02   6.0171219775E-03
 12    6 5
 13    3 0
 14      0.0         -23.25947071          90.58333916         -176.4686526
 15      186.0516105         -107.3443700          30.49467355         0.0
 16     0.6523873876         -1.062334130        -0.7865093805          2.683868564
 17     -2.113167588          0.6913630998
 18     3.0225540985E-03  -8.1466174596E-03   6.7393866548E-03  -1.7009274452E-03
 19     -7.9237E-05
 20    0 0
 21    0 6
 22       0.000000000
 23       0.000000000
 24       0.000000000
 25    -7.4583939661E-05   2.7140939434E-04  -2.7842339713E-04  -2.6573366546E-04
 26     .7.6755394117E-04  -6.7415845101E-04   2.8871746779E-04
 27    4 7
 28    5 10
 29       1.813480746         -23.21808079          -16.87668112          26.99425015
 30     -19.48195619
 31    -5.0050515425E-06   1.5253667766E-04  -4.3222074893E-04   2.0430756138E-04
 32     6.3706562081E-04  -1.0461911109E-03   6.5116394823E-04  -1.6345584070E-04
 33      92.40051275          215.1188439          -617.8989036          1985.688577
 34     -1589.021855          859.3109678
 35     3.7133779543E-05   2.5606821049E-04  -3.9910127867E-05  -1.0334099638E-03
 36     1.1965976215E-03  -1.5316672339E-04  -3.6373530698E-04   3.4734212493E-04
 37    -5.8513802782E-04   2.6878597550E-04   1.2970946019E-04
 38    ------ AX --------
 39             0          14           0          9
 40             1           1
 41       0.000000000
 42             1           2
 43    -2.4225609441E-05  -2.4402203945E-04   4.3285695964E-03  -2.4020218552E-02
 44     6.9234382447E-02  -0.1076976998         5.4480571585E-02   0.1271904974
 45    -0.3265065948          0.3839081843        -0.2770201454          0.1243240748
 46    -3.0366759542E-02   1.8434872235E-03   5.8367816210E-04
 47             2           1
 48       0.000000000
 49             2           2
 50       0.000000000        -4.8153452358E-07   2.5796371760E-06  -5.2570433706E-06
```

```
 51     3.0059425615E-06    6.2279575118E-06   -1.4465811057E-05    1.3960795879E-05
 52    -7.2591725806E-06    1.7372482207E-06
 53    ------ BY --------
 54           10          15          7          10
 55            1           1
 56      60.28092913       -2.4680048227E-08    1.0197982192E-07   -7.7299773693E-08
 57    -3.4458935261E-08   -2.0535662770E-07    5.1874667406E-07   -1.5702098608E-06
 58     1.3783574104E-06   -9.0152025223E-07    2.5704503059E-07
 59            1           2
 60     2.4225609428E-05    2.4402203926E-04   -4.3285695973E-03    2.4020218557E-02
 61    -6.9234579230E-02    0.1076985443       -5.4482325742E-02   -0.1271743898
 62     0.3267030234       -0.3858185474        0.2835790244       -0.1351183589
 63     3.9499099037E-02   -5.6089420116E-03   -8.8269121928E-05    8.9676334469E-05
 64            2           1
 65    -1.8401957674E-03   -0.1019970110        0.1993203440       -9.7820782275E-02
 66    -0.2271339801        0.3012535206       -0.1772116965        3.3137393970E-02
 67            2           2
 68    -1.8070408343E-08    4.8153452371E-07   -2.5796371764E-06    5.2570689412E-06
 69    -3.0060216179E-06   -6.2278788810E-06    1.4465837530E-05   -1.3960648869E-05
 70     7.5459175838E-06   -2.2304725367E-06    2.6774873703E-07
 71           10          15          7          10
 72            1           1
 73      60.28092913       -2.4680048227E-08    1.0197982192E-07   -7.7299773693E-08
 74    -3.4458935261E-08   -2.0535662770E-07    5.1874667406E-07   -1.5702098608E-06
 75     1.3783574104E-06   -9.0152025223E-07    2.5704503059E-07
 76            1           2
 77    -1.3544720900E-14   -1.8829382498E-13   -8.8107299234E-13    4.7748471843E-12
 78    -1.9678282115E-07    8.4447856352E-07   -1.7541574380E-06    1.6107574993E-05
 79     1.9642859297E-04   -1.9103630384E-03    6.5588789785E-03   -1.0794284114E-02
 80     9.1323394942E-03   -3.7654547881E-03    4.9540904017E-04    8.9676334469E-05
 81            2           1
 82    -1.8401957674E-03   -0.1019970110        0.1993203440       -9.7820782275E-02
 83    -0.2271339801        0.3012535206       -0.1772116965        3.3137393970E-02
 84            2           2
 85    -1.8070408343E-08    1.3183898417E-16   -4.3021142204E-16    2.5570545681E-11
 86    -7.9056469704E-11    7.8630796318E-11    2.6472601888E-11    1.4700979323E-10
 87     2.8674500316E-07   -4.9322431597E-07    2.6774873703E-07
```

```
**********************************************************************************************
**********************************************************************************************
**************
**************   Pdn =   6            2299-MNI          Terminal: 64                19 JAN 8
**************
**********************************************************************************************
**********************************************************************************************

LISTING OF 2299MNI *ABLC-51  LAST UPDATED ON 19-JAN-84 AT 18:40:30

  1  8 7 6 5
  2  5 3 3 0
  3     0.0  -2.8043          34.46731430         -127.4427131
  4      151.0625789          4.794461592         -137.5682800          110.6236461
  5       -34.78404
  6     .32481          4.453754070         -43.04162022          147.6454078
  7     -242.8463943          205.3538822         -84.62088489          11.10560491
  8      0.0          -23.25947071          90.58333916          -176.4686526
  9      186.0516105          -107.3443700          30.49467355          0.0
 10     0.6523873876          -1.062334130         -0.7865093805          2.683868564
 11     -2.113167588          0.6913630998
 12    -1.3784434946E-03   -7.7797823480E-03    8.3117034640E-03    4.9767877851E-03
 13    -7.2579504312E-03    1.8579566640E-03
 14     0.0    6.4562763213E-03   -1.1092778602E-02    6.0171219775E-03
 15     3.0225540985E-03   -8.1466174596E-03    6.7393866548E-03   -1.7009274452E-03
 16     -7.9237E-05
 17  0 0
 18  0 0
 19  1.0
 20  0.0
 21  .1
 22  1.E02
 23   EXTRACT X & Y FROM UNIMODULAR  VXY
 24           0          0          0          0          0          7          22          8
 25           4          7          22          8          5          9          24          10
 26           1          1
 27      0.000000000
 28           1          2
 29      0.000000000
 30           1          3
 31      1.000000000
 32           1          4
 33      0.000000000
 34           2          1
 35      0.000000000
 36           2          2
 37     3.3266653250E-02   -0.1208148094          0.1088854640          0.1192497894
 38    -0.3336236017          0.2903917116         -0.1266540365          1.9964028928E-02
 39           2          3
 40      8.389781308          17.41765345         -173.7394825         -133.6968722
 41      3572.822374         -13732.77456          25269.32857         -11593.64531
 42     -55302.77859          143122.1748         -146103.9593         -45.39412074
 43      200815.0948         -278768.8463          172592.7105          5874.715801
 44     -114799.9674          117927.2903         -69758.21243          27132.10083
 45     -6893.613288          1038.710036         -69.84240671
 46           2          4
 47     -5.984836114          31.47008319         -54.18075331          23.46877395
 48      51.60600260         -91.73468756          66.84038089         -25.05080737
 49      3.569148499
 50           3          1
```

```
 51      1.813480746          -23.21808079          -16.87668112          26.99425015
 52     -19.48195619
 53              3           2
 54      1.3741148975E-03     -7.1154563317E-02      0.1848290029         -7.4057899794E-02
 55     -0.2796048546          0.4497625374         -0.2834834797          7.0625981700E-02
 56              3           3
 57      0.3465489418         -15.88024340          -59.08757742          304.7195839
 58      610.8019258          -7122.001711           22815.98222          -33345.70659
 59     -2937.427258           109089.3169          -204980.2503          143953.0739
 60      96148.08415          -322987.9947           327722.8727          -125715.7837
 61     -86018.78395           163644.8524          -124603.3610          58299.03213
 62     -17456.70598           3098.851235          -247.0788114
 63              3           4
 64      7.8114208928E-02      13.85490172          -53.13241516          67.77638942
 65      1.267280128          -96.74546092           113.3573592          -59.19861621
 66      12.62644017
 67              4           1
 68      92.40051275           215.1188439          -617.8989036          1985.688577
 69     -1589.021855           859.3109678
 70              4           2
 71     -1.6562769065E-02     -0.1449348125         -1.8274957503E-02      0.4024605537
 72     -0.5053570927          2.7725664317E-02      0.1757789052         -0.2127029106
 73      0.3827589760         -0.3207069201
 74              4           3
 75     -4.177096183          -55.97299738          -242.1937144          445.7851776
 76      1945.593414          -10624.53317           18694.43740          8522.723648
 77     -94744.09620           156255.6459          -41355.92896          -233301.2487
 78      410128.1995          -288322.8060          -34454.57802          314070.2770
 79     -401847.1065           289031.3174          -69218.58922          -111265.5137
 80      161349.0358          -111217.1579           45882.65278          -10907.26227
 81      1121.965072
 82              4           4
 83      2.979724402           32.82642848          -42.55226525          -49.98892402
 84      150.6310522          -110.6573669           6.761446201          38.96815209
 85     -78.55905263           107.1075936          -57.33565237
```

****

****

****

********** No. of pages 2 2299-MNI Terminal: 64

****

****

****

```
**********************************************************************************************
**********************************************************************************************
**************
**************   Pdn =   6            2299-MNI          Terminal: 64                  19
**************
**********************************************************************************************
**********************************************************************************************

LISTING OF 2299MNI *ABLC-S1  LAST UPDATED ON 19-JAN-84 AT 19:43:48

  1  8 7 6 5
  2  5 3 3 0
  3     0.0  -2.8843          34.46731430         -127.4427131
  4      151.0625789          4.794461592         -137.5682800          110.6236461
  5       -34.78404
  6     .32481          4.453754070         -43.04162022          147.6454078
  7     -242.8463943          205.3538822         -84.62088489          11.10560491
  8      0.0          -23.25947071          90.58333916          -176.4686526
  9      186.0516105          -107.3443700          30.49467355          0.0
 10     0.6523873876          -1.062334130         -0.7865093805          2.683868564
 11     -2.113167588          0.6913630998
 12    -1.3784434946E-03   -7.7797823480E-03    8.3117034640E-03    4.9767877851E-03
 13    -7.2579504312E-03    1.8579566640E-03
 14     0.0    6.4562763213E-03   -1.1092778602E-02    6.0171219775E-03
 15     3.0225540985E-03   -8.1466174596E-03    6.7393866548E-03   -1.7009274452E-03
 16     -7.9237E-05
 17  0 0
 18  0 0
 19  1.0
 20  0.0
 21  1.0
 22  -1.0E05
 23   EXTRACT X & Y FROM UNIMODULAR  VXY
 24            0          0          0          0          0          7          22          8
 25            4          7          22          8          5          9          24          10
 26            1          1
 27       0.000000000
 28            1          2
 29       0.000000000
 30            1          3
 31       1.000000000
 32            1          4
 33       0.000000000
 34            2          1
 35       0.000000000
 36            2          2
 37      3.3268714205E-02   -0.1208222740          0.1088921203          0.1192573612
 38     -0.3336444742          0.2904098314         -0.1266619282          1.9965271738E-02
 39            2          3
 40       8.389778435          17.41765258          -173.7394165          -133.6969253
 41       3572.821182          -13732.76790          25269.30975          -11593.61369
 42      -55302.80486          143122.1620          -146103.8891          -45.49532799
 43       200815.1698          -278768.8568          172592.6668          5874.774322
 44      -114800.0098          117927.3104          -69758.21857          27132.10183
 45      -6893.613276          1038.710000          -69.84240221
 46            2          4
 47      -5.984463312          31.46812488          -54.17737833          23.46731205
 48       51.60278801          -91.72897322          66.83621708          -25.04924669
 49       3.568926095
 50            3          1
```

```
 51      1.813480746          -23.21808079          -16.87668112          26.99425015
 52     -19.48195619
 53            3          2
 54      1.3742006470E-03    -7.1158969542E-02      0.1848404121         -7.4059059363E-02
 55     -0.2796260950         0.4497870093         -0.2834913385          7.0625343096E-02
 56            3          3
 57      0.3465489794         -15.88023654          -59.08756602          304.7202806
 58       610.8056961         -7122.008014           22813.92922         -33545.41315
 59      -2938.098513          109090.0158          -204979.0393          143949.3681
 60       96151.81669         -322987.4415           327717.1429         -125709.2313
 61      -86021.39723          163643.1400          -124600.2554          58296.90335
 62      -17455.87132          3098.666598          -247.0611812
 63            3          4
 64      7.8109343066E-02      13.85403868          -53.12910548          67.77153716
 65      1.268725140          -96.73999898           113.3485533          -59.19256696
 66      12.62475327
 67            4          1
 68       92.40051275          215.1188439          -617.8989036          1985.688577
 69      -1589.021855          859.3109678
 70            4          2
 71     -1.6563795172E-02     -0.1449437796         -1.8276178467E-02      0.4024856215
 72     -0.5053883980          2.7727024834E-02      0.1758209965         -0.2127032519
 73      0.3827752376         -0.3206851347
 74            4          3
 75     -4.177094753          -55.97297366          -242.1936510          445.7849358
 76      1945.593093          -10624.52891           18694.43194          8522.787512
 77     -94744.09214           156255.1656          -41353.99038         -233304.5625
 78      410128.5604          -288313.9273          -34470.62766          314079.1282
 79     -401836.5585           289007.3934          -69200.68424         -111266.9002
 80      161339.6158          -111207.7485           45877.99030         -10905.98967
 81      1121.818949
 82            4          4
 83       2.979538791           32.82438368          -42.54961462          -49.98581023
 84       150.6216694          -110.6504407           6.756048063           38.97126483
 85      -78.55161450           107.0945168          -57.32461630
```

****
****
****
********** No. of pages 2 2299-MNI Terminal: 64
****
****
****

```
**********************************************************************************************
**********************************************************************************************
**************
**************   Pdn =   6            2299-MNI          Terminal: 64                19 JAN 8
**************
**********************************************************************************************
**********************************************************************************************

LISTING OF 2299MNI *ABLC-53  LAST UPDATED ON 19-JAN-84 AT 19:06:09

  1  8 7 6 5
  2  5 3 3 0
  3     0.0  -2.8843          34.46731430         -127.4427131
  4      151.0625789          4.794461592         -137.5682800          110.6236461
  5       -34.78404
  6     .32481          4.453754070         -43.04162022          147.6454078
  7     -242.8463943          205.3538822         -84.62088489          11.10560491
  8      0.0          -23.25947071          90.58333916         -176.4686526
  9      186.0516105          -107.3443700          30.49467355          0.0
 10     0.6523373876          -1.062334130         -0.7865093805          2.683868564
 11     -2.113167588          0.6913630998
 12    -1.3784434946E-03   -7.7797823480E-03    8.3117034640E-03    4.9767877851E-03
 13    -7.2579504312E-03    1.8579566640E-03
 14     0.0    6.4562763213E-03   -1.1092778602E-02    6.0171219775E-03
 15     3.0225540985E-03   -8.1466174596E-03    6.7393866548E-03   -1.7009274452E-03
 16     -7.9237E-05
 17  0 0
 18  0 0
 19  1.0
 20  0.0
 21  1.0
 22  -1.0E03
 23   EXTRACT X & Y FROM UNIMODULAR  VXY
 24           0          0          0          0          0          7          22          8
 25           4          7          22          8          5          9          24          10
 26           1          1
 27      0.000000000
 28           1          2
 29      0.000000000
 30           1          3
 31      1.000000000
 32           1          4
 33      0.000000000
 34           2          1
 35      0.000000000
 36           2          2
 37     3.3269011576E-02   -0.1208233539          0.1088930937          0.1192584271
 38    -0.3336474560          0.2904124266         -0.1266630597          1.9965449948E-02
 39           2          3
 40      8.389778408          17.41765253         -173.7394159         -133.6969249
 41      3572.821171         -13732.76786          25269.30967         -11593.61366
 42     -55302.80464          143122.1615         -146103.8885         -45.49542511
 43      200815.1690         -278768.8555          172592.6657          5874.774628
 44     -114800.0094          117927.3097         -69758.21804          27132.10156
 45     -6893.613185          1038.709982         -69.84240052
 46           2          4
 47     -5.984409803          31.46784350         -54.17689390          23.46710221
 48      51.60232661         -91.72815302          66.83561943         -25.04902266
 49      3.568894152
 50           3          1
```

```
 51      1.813480746          -23.21808079          -16.87668112          26.99425015
 52     -19.48195619
 53              3           2
 54      1.3742129360E-03    -7.1159605559E-02      0.1848420643         -7.4059211969E-02
 55     -0.2796291637         0.4497904666         -0.2834923455          7.0625188105E-02
 56              3           3
 57      0.3465489797         -15.88023648          -59.08756579          304.7204081
 58      610.8062837          -7122.009371           22815.92276          -33545.37201
 59     -2938.176351           109070.0895          -204978.8566           143948.8082
 60      96152.37802          -322987.3445           327716.2531          -125708.2212
 61     -86021.79364           163642.8670          -124599.7689           58296.57080
 62     -17455.74102           3098.637774          -247.0584278
 63              3           4
 64      7.8108644592E-02      13.85391481          -53.12863043          67.77083315
 65      1.268951127          -96.73922185           113.3472686          -59.19167007
 66      12.62449988
 67              4           1
 68      92.40051275           215.1188439          -617.8989036          1985.688577
 69     -1589.021855           859.3109678
 70              4           2
 71     -1.6563943227E-02     -0.1449450749         -1.8276341201E-02      0.4024892187
 72     -0.5053929132          2.7727250272E-02      0.1758273830         -0.2127029973
 73      0.3827774480         -0.3206814637
 74              4           3
 75     -4.177094740          -55.97297343          -242.1936500          445.7849351
 76      1945.593086          -10624.52889           18694.43311          8522.794840
 77     -94744.09243           156255.1008          -41353.70727          -233305.0585
 78      410128.6043          -288312.5518          -34473.08839          314080.4432
 79     -401834.8615           289003.8848          -69197.92062          -111267.0782
 80      161338.1199          -111206.2685           45877.25910          -10905.79033
 81      1121.796067
 82              4           4
 83      2.979512150           32.82409019          -42.54923424          -49.98536300
 84      150.6203221          -110.6494450           6.755213730          38.97176288
 85     -78.55049175           107.0925545          -57.32293533
```

```
*****************************************************************************************
*****************************************************************************************
**************
**************   No. of pages     2   2299-MNI          Terminal: 64
**************
*****************************************************************************************
*****************************************************************************************
```

```
**********************************************************************************
**********************************************************************************
**************
**************   Pdn =   6            2299-MNI          Terminal: 64                19
**************
**********************************************************************************
**********************************************************************************

LISTING OF 2299MNI *ABLC-54  LAST UPDATED ON 19-JAN-84 AT 19:02:23

  1  8 7 6 5
  2  5 3 3 0
  3     0.0  -2.8843          34.46731430         -127.4427131
  4      151.0625789          4.794461592         -137.5682800          110.6236461
  5       -34.78404
  6     .32481          4.453754070         -43.04162022          147.6454078
  7     -242.8463943          205.3538822         -84.62088489          11.10560491
  8      0.0         -23.25947071          90.58333916         -176.4686526
  9      186.0516105         -107.3443700          30.49467355         0.0
 10     0.6523873876         -1.062334130         -0.7865093805          2.683868564
 11     -2.113167588          0.6913630798
 12    -1.3734434946E-03   -7.7797823480E-03    8.3117034640E-03    4.9767877851E-03
 13    -7.2579504312E-03    1.8579566640E-03
 14     0.0    6.4562763213E-03   -1.1092778602E-02    6.0171219775E-03
 15     3.0225540985E-03   -8.1466174596E-03    6.7393866548E-03   -1.7009274452E-03
 16     -7.9237E-05
 17  0 0
 18  0 0
 19  1.0
 20  0.0
 21  1.0
 22  1.E02
 23   EXTRACT X & Y FROM UNIMODULAR  VXY
 24           0          0          0          0          0          7         22          8
 25           4          7         22          8          5          9         24         10
 26           1          1
 27     0.000000000
 28           1          2
 29     0.000000000
 30           1          3
 31     1.000000000
 32           1          4
 33     0.000000000
 34           2          1
 35     0.000000000
 36           2          2
 37     3.3266523309E-02   -0.1208143452         0.1088850805         0.1192492360
 38    -0.3336221994         0.2903905131        -0.1266535195         1.9963948940E-02
 39           2.          3
 40      8.389781456          17.41765181         -173.7394861         -133.6968306
 41      3572.822400         -13732.77561          25269.33355         -11593.65838
 42     -55302.75776          143122.1589         -146103.9699         -45.34636800
 43      200815.0283         -278768.7982          172592.7047          5874.686466
 44     -114799.9290          117927.2626         -69758.19928          27132.09667
 45     -6893.612476          1038.709959         -69.84240545
 46           2          4
 47     -5.984859603          31.47020867         -54.18096590          23.46886603
 48      51.60620512         -91.73504747          66.84064324         -25.05090600
 49      3.569162734
 50           3          1
```

```
 51      1.813480746          -23.21808079          -16.87668112          26.99425015
 52     -19.48195619
 53            3            2
 54      1.3741094967E-03    -7.1154285342E-02      0.1848282985         -7.4057755046E-02
 55     -0.2796035596         0.4497607452         -0.2834825379          7.0625788228E-02
 56            3            3
 57      0.3465489395         -15.88024372          -59.08757389          304.7195781
 58       610.8017643         -7122.001653           22815.98526         -33545.72117
 59      -2937.393165          109089.4792          -204980.2637          143953.1821
 60       96147.93285         -322987.9294           327722.9566         -125715.9359
 61      -86018.68912          163644.8483          -124603.4015          58299.06602
 62      -17456.72017          3098.854446          -247.0791200
 63            3            4
 64      7.8114515471E-02      13.85495608          -53.13262363          67.77666936
 65      1.267251561          -96.74582796           113.3578427         -59.19889962
 66      12.62650653
 67            4            1
 68       92.40051275          215.1188439          -617.8989036          1985.688577
 69      -1589.021855          859.3109678
 70            4            2
 71     -1.6562704371E-02     -0.1449342437         -1.8274843722E-02      0.4024589152
 72     -0.5053551102         2.7725654261E-02       0.1757772193         -0.2127023978
 73      0.3827564773        -0.3207068261
 74            4            3
 75     -4.177096257          -55.97299778          -242.1937034          445.7852267
 76      1945.593276         -10624.53358           18694.44057          8522.712812
 77     -94744.08511          156255.6620          -41356.02245         -233301.0780
 78      410128.0842         -288322.9393          -34454.15574          314069.8070
 79     -401846.9570          289031.6230          -69219.07182         -111265.2344
 80      161349.0491         -111217.2920           45882.74714         -10907.29383
 81      1121.969218
 82            4            4
 83      2.979736096          32.82655731          -42.55243256          -49.98911867
 84      150.6316392         -110.6577937           6.761585814          38.96816819
 85     -78.55930619          107.1079984          -57.33609393
```

****
****
****
********** No. of pages 2 2299-MNI Terminal: 64
****
****
****

```
**********************************************************************************
**********************************************************************************
**************
**************   Pdn =   6            2299-MNI          Terminal: 64              19 JAN
**************
**********************************************************************************
**********************************************************************************

LISTING OF 2299MNI *ARXYC4-50 LAST UPDATED ON 19-JAN-84 AT 18:56:47


  1  8 7
  2  5 3
  3     0.0    0.0          34.46731430         -127.4427131
  4      151.0625789          4.794461592         -137.5682800          110.6236461
  5      -34.78404
  6     0.0          4.453754070         -43.04162022          147.6454078
  7     -242.8463943          205.3538822         -84.62088489          11.10560491
  8    -1.3784434946E-03   -7.7797823480E-03    8.3117034640E-03    4.9767877851E-03
  9    -7.2579504312E-03    1.8579566640E-03
 10     8.6382416831E-05    6.4562763213E-03   -1.1092778602E-02    6.0171219775E-03
 11  6 5
 12  3 0
 13      0.0         -23.25947071          90.58333916         -176.4686526
 14      186.0516105         -107.3443700          30.49467355         0.0
 15     0.6523873876         -1.062334130         -0.7865093805         2.683868564
 16     -2.113167588         0.6913630998         0.0   0.0
 17     3.0225540985E-03   -8.1466174596E-03    6.7393866548E-03   -1.7009274452E-03
 18  0.0
 19  0 0
 20  0 7
 21      0.000000000
 22      0.000000000
 23      0.000000000
 24     3.3266653250E-02   -0.1208148094         0.1088854640         0.1192497894
 25    -0.3336236017         0.2903917116         -0.1266540365         1.9964028928E-02
 26  4 7
 27  5 9
 28      1.813480746          -23.21808079          -16.87668112          26.99425015
 29     -19.48195619
 30     1.3741148975E-03   -7.1154563317E-02    0.1848290029         -7.4057899794E-02
 31    -0.2796048546         0.4497625374         -0.2834834797         7.0625981700E-02
 32      92.40051275          215.1188439          -617.8989036          1985.688577
 33     -1589.021855          859.3109678
 34    -1.6562769065E-02   -0.1449348125         -1.8274957503E-02    0.4024605537
 35    -0.5053570927         2.7725664317E-02    0.1757789052         -0.2127029106
 36     0.3827589760         -0.3207069201
 37   ----- AX -------
 38            0          14           0          10
 39            1           1
 40       0.000000000
 41            1           2
 42       0.000000000          0.1481614923          -1.969930104          10.59668281
 43      -30.07193615          45.62873433          -20.80756868          -58.32743466
 44       143.3673026          -167.4724367          122.8940041          -59.13538604
 45       18.04224311          -3.093943483         0.2217126177
 46            2           1
 47       0.000000000
 48            2           2
 49     2.8736539076E-06    2.0434243044E-04   -1.1396276230E-03    2.2536371708E-03
 50    -1.1937094114E-03   -2.7965158157E-03    6.2822617283E-03   -6.0446937864E-03
```

```
 51     3.2811608231E-03  -9.8354933946E-04    1.2012599722E-04
 52    ----- BY -------
 53           10          14          7          10
 54            1           1
 55       60.28092913       -2.4680048227E-08    1.0197982192E-07  -7.7299773693E-08
 56    -3.4458935261E-08    -2.0535662770E-07    5.1874667406E-07  -1.5702098608E-06
 57     1.3783574104E-06    -9.0152025223E-07    2.5704503059E-07
 58            1           2
 59    -1.0805341642E-02    -0.1089196340         1.934563017       -10.63541633
 60       30.18033318       -45.72322059         20.84905327        58.32066016
 61     -143.3675473         167.4734079        -122.8952588        59.13631265
 62     -18.04266171         3.096049157        -0.2217249304
 63            2           1
 64     5.4813436613E-03    -8.4951639159E-02    0.1503598886       5.9519223500E-02
 65    -0.3530433048         0.3693427438        -0.1772116965       3.3137393970E-02
 66            2           2
 67     4.1533366152E-06    -2.2626290540E-04    1.1475853597E-03  -2.2114505748E-03
 68     1.1248634454E-03     2.8237798458E-03   -6.2792856047E-03   6.0296135290E-03
 69    -3.2508810793E-03     9.5816062943E-04   -1.2012967062E-04
 70           10          14          7          10
 71            1           1
 72       60.28092913       -2.4680048227E-08    1.0197982192E-07  -7.7299773693E-08
 73    -3.4458935261E-08    -2.0535662770E-07    5.1874667406E-07  -1.5702098608E-06
 74     1.3783574104E-06    -9.0152025223E-07    2.5704503059E-07
 75            1           2
 76    -1.0805341642E-02     3.9241858261E-02   -3.5367087738E-02  -3.8733524154E-02
 77     0.1083970325        -9.4486244721E-02    4.1484590853E-02  -6.7745004781E-03
 78    -2.4476647377E-04     9.7119808197E-04   -1.2547704391E-03   9.2661217786E-04
 79    -4.1860225611E-04     1.0567350546E-04   -1.2312723811E-05
 80            2           1
 81     5.4813436613E-03    -8.4951639159E-02    0.1503598886       5.9519223500E-02
 82    -0.3530433048         0.3693427438        -0.1772116965       3.3137393970E-02
 83            2           2
 84     7.0269905228E-06    -2.1920474956E-05    7.9577366847E-06   4.2186596033E-05
 85    -6.8845966006E-05     2.7264030052E-05    2.9761236249E-06  -1.5080257441E-05
 86     3.0279743854E-05    -2.5388710043E-05   -3.6733971420E-09

********************************************************************************
********************************************************************************
**************
**************   No. of pages     2   2299-MNI          Terminal: 64              1
**************
********************************************************************************
********************************************************************************
```

```
**********************************************************************************************
**********************************************************************************************
**************
**************   Pdn =   7          2299-MNI          Terminal: 72          26 I
**************
**********************************************************************************************
**********************************************************************************************

LISTING OF 2299MNI *GDASH    LAST UPDATED ON 25-JAN-84 AT 14:57:24

   1   0 0
   2   0 6
   3      0.000000000
   4      0.000000000
   5      0.000000000
   6    -7.4583939661E-05   2.7140939434E-04  -2.7842339713E-04  -2.6573366546E-04
   7     7.6755394117E-04  -6.7415845101E-04   2.8871746779E-04
   8   4 7
   9   5 10
  10      1.813480746        -23.21808079        -16.87668112         26.99425015
  11     -19.48195619
  12    -5.0050515425E-06    1.5253667766E-04  -4.3222074893E-04   2.0430756138E-04
  13     6.3706562081E-04   -1.0461911109E-03   6.5116394823E-04  -1.6345584070E-04
  14      92.40051275         215.1188439        -617.8989036         1985.688577
  15     -1589.021855         859.3109678
  16     3.7133779543E-05    2.5606821049E-04  -3.9910127867E-05  -1.0334099638E-03
  17     1.1965976215E-03   -1.5316672339E-04  -3.6373530698E-04   3.4734212493E-04
  18    -5.8513802782E-04    2.6878597550E-04   1.2970946019E-04
  19  0  0
  20  6  0
  21  .016588
  22  0.0
  23  -16.8928E02 -93.6308E03 182.9653E03 -89.7843E03 -208.4868E03 276.6054E03
  24  -162.6766E03
  25  -.553E08
  26    THE NEW     X
  27           0          0         12          6
  28           1          1
  29      0.000000000
  30           1          2
  31      0.000000000
  32           2          1
  33     0.1259931576         6.524867476        -38.58821654         82.87287205
  34     -36.17611379        -171.5654338         372.1826876        -285.0581551
  35     -74.88113752         370.1690565        -371.5327144         189.5306153
  36     -46.96757602
  37           2          2
  38      4124.491863        -15008.93951         15396.81386         14695.07170
  39     -42445.73295         37280.96234        -15966.07597
  40    THE NEW     Y
  41          13          7         16         10
  42           1          1
  43     3.8536952084E-02   -0.1741917030        -14.74769044         68.93011648
  44     -112.2621400        -14.88013045         328.1919978        -496.2722109
  45      222.3814488         272.7257221        -514.1010995         384.3846624
  46     -151.1419053         26.59044042
  47           1          2
  48      276.7793503        -8435.278275         23901.80742        -11298.20814
  49     -35229.72883         57854.36843        -36009.36634         9039.107991
  50           2          1
```

51	1.470010354	-0.3410450097	-27.36396592	81.93870465
52	30.34394226	-326.1356998	399.7851943	60.76686806
53	-613.1615731	681.5723860	-324.8227511	-58.54084207
54	276.8410551	-286.0409752	142.4932069	-7.846851503
55	-21.10069397			
56	2	2		
57	-2053.498009	-14160.57204	2207.030071	57147.57100
58	-66171.84847	8470.119803	20114.56248	-19208.01951
59	32358.13294	-14863.86445	-7172.933149	

****
****
****
********** No. of pages 2 2299-MNI Terminal: 72 26
****
****
****

```
**********************************************************************
**********************************************************************
**************
**************   Pdn =   7            2299-MNI          Terminal:103
**************
**********************************************************************
**********************************************************************

LISTING OF 2299MNI *FPANA    LAST UPDATED ON  7-SEP-83 AT  9:36:59

  1         REAL P(2,2,0:40),PDEN(2,2,0:40),PDEN1(0:20),PDS(0:20),PZ(2,2,0:30)
  2         REAL PDENZ(2,2,0:30),PP(2,2,0:40),PPDEN(2,2,0:30),K(2,2,0:1)
  3         REAL KPP(2,2,0:30),VG(2,2,0:40),VVD(2,2,0:40)
  4          REAL VGT(2,2,0:30)
  5         INTEGER DP(2,2),DPDEN(2,2),DPZ(2,2),DPDENZ(2,2),DPP(2,2)
  6         INTEGER DPPDEN(2,2),DK(2,2),DKPP(2,2),DVG(2,2)
  7          INTEGER DVGT(2,2)
  8
  9         CALL READP (2,2,DP,P)
 10         CALL READP (2,2,DPDEN,PDEN)
 11         CALL READS (NP,PDEN1)
 12         CALL SPECSC (PDEN1,5,PDS)
 13  *
 14  *
 15         CALL CONJTRA (P,PZ,DP,DPZ,2,2)
 16         CALL CONJTRA (PDEN,PDENZ,DPDEN,DPDENZ,2,2)
 17         CALL POLMUL (P,PZ,PP,DPZ,DP,DPP,2,2)
 18         CALL POLMUL (PDEN,PDENZ,PPDEN,DPDENZ,DPDEN,DPPDEN,2,2)
 19  *
 20  *
 21         WRITE (3,111)
 22   111   FORMAT (' INPUT MATRIX K ')
 23         FOR I=1,2
 24          FOR J=1,2
 25            DK(I,J) = 0
 26            READ (3,*) K(I,J,0)
 27          END FOR !J!
 28         END FOR !I!
 29  *
 30          WRITE (40,123)
 31  123     FORMAT(' THE   K  MATRIX ')
 32  *
 33          CALL WRITP (2,2,DK,K)
 34         CALL POLPOL (K,PPDEN,KPP,DK,DPPDEN,DKPP,2,2,2)
 35         CALL POLYADD (KPP,PP,DKPP,DPP,2,2,VG,DVG)
 36          CALL TRANSPOSE (VG,VGT,DVG,DVGT,2,2)
 37         CALL WRITP (2,2,DVGT,VGT)
 38          CALL WRITP (2,2,DPP,PP)
 39         CALL MULSPECT(VGT,2,DVGT,VVD)
 40         WRITE (40,112)
 41   112   FORMAT (' SPECTRAL FACTORIZATION OF DENOMINATOR ')
 42         CALL WRITS(5,PDS)
 43         WRITE (40,113)
 44   113   FORMAT (' SPECTRAL FACTORIZATION OF NUMERATOR ')
 45          CALL WRITP(2,2,DVG,VVD)
 46         STOP
 47         END
```

```
**********
**********
**********
**********   Pdn =   7          2299-MNI          Terminal: 103          27
**********
**********
**********
**********
```

LISTING OF 2299MNI *FPAND LAST UPDATED ON 26-AUG-83 AT 17:06:13

```
  1  6 6
  2  6 5
  3  4.107 21.024 -38.0575 -8.5742 35.1879 -15.7239 2.036
  4  -.244 -21.3202 53.9147 -43.74 9.92 2.2529 -.773
  5  -1.73 8.7925 -17.7802 17.8465 -8.9035 1.7932 -1.8573E-2
  6  -.402 1.8183 -3.0907 2.4656 -.91299 .12178
  7  5 0
  8  0 5
  9  1.0 -3.405 4.496 -2.8621 .87059 -9.9879E-2
 10  0.0
 11  0.0
 12  1.0 -3.405 4.496 -2.8621 .87059 -9.9879E-2
 13  5
 14  1.0 -3.405 4.4964 -2.8621 .87059 -9.9879E-2
 15          6          6          6          5
 16          1          1
 17     2.036000000       -15.72390000        35.18790000       -8.574200000
 18    -38.05750000        21.02400000        4.107000000
 19          1          2
 20   -1.8573000000E-02     1.793200000       -8.903500000        17.84650000
 21    -17.78020000        8.792500000       -1.730000000
 22          2          1
 23   -0.7730000000        2.252900000        9.920000000       -43.74000000
 24     53.91470000       -21.32020000       -0.2440000000
 25          2          2
 26     0.1217800000       -0.9129900000        2.465600000       -3.090700000
 27     1.818300000        -0.4020000000
 28          6          6          6          6
 29          1          1
 30     4267.763124       -1939.228816       -1091.118393        1144.158037
 31    -232.0476407        -25.03873240        8.393983290
 32          1          2
 33    -1690.009775        2625.411528       -1076.777358       -64.65221048
 34     120.1750804       -7.209571100       -3.174711000
 35          2          1
 36    -1690.009775       -1461.139100        2868.749517       -1713.876516
 37     443.3183605       -40.32593329       -0.4893176540
 38          2          2
 39     5398.620237       -3932.126018        1356.861880       -48.57602273
 40    -91.54036713        15.88185144        0.1886120000
 41  THE   K  MATRIX
 42          0          0          0          0
 43          1          1
 44    0.1000000000
 45          1          2
 46    0.000000000
 47          2          1
 48    0.000000000
 49          2          2
 50    0.1000000000
```

```
**********************************************************************************
**********************************************************************************
**************
**************   Pdn =   7            2299-MNI          Terminal: 62
**************
**********************************************************************************
**********************************************************************************

LISTING OF 2299MNI *APA      LAST UPDATED ON 16-FEB-84 AT 12:43:39

    1          SUBROUTINE INVERT (FF,NROW,DEN1,NB,DD,VO,DVO,VV,DV,DF1)
    2 ***************************************************************
    3 *  THIS SUBR INVERT A MATRIX FF AND THE OUTPUT COMES TO
    4 * VO & DVO . DEN1 OS IS THE DENOMINATOR & NB IS ITS DEG.
    5 * THE SUBR. USES AN ALTERED VERSION OF RREDUCE SUBR.
    6 * WE CALL IT REDUC1 .
    7 * FOR FF TO BE INVERTABLE , BESIDE OTHER THINGS , FF
    8 * SHOULD BE SQUARE (NROW*NROW)
    9 * FF IS NOT RATIONAL
   10 ***************************************************************
   11 * INVERT PRODUCES A N INVERSION OF MATRIX FF
   12 * WITHOUT INTRODUCING REDUNDANT POLES AND ZEROS
   13 * OR COMMON FACTOR BETWEEN THE NUMERATOR AND
   14 * DENOMINATOR .
   15 ***************************************************************
   16          INTEGER DD(NROW,NROW),DVO(NROW,NROW),DV(NROW,NROW)
   17          REAL FF(NROW,NROW,0:25),VO(NROW,NROW,0:30),VV(NROW,NROW,0:30)
   18          INTEGER DF1(NROW,NROW)
   19          REAL FF1(8,8,0:30)
   20          REAL DEN1(0:30)
   21          EPSI = 1.E-9
   22 **************
   23 * GENERATE THE IDENTITY MATRIX
   24 **************
   25          FOR I=1,NROW
   26            FOR J=1,NROW
   27              DV(I,J)=0
   28              IF (I.EQ.J) THEN
   29                VV(I,J,0)-= 1.0
   30              ELSE
   31                VV(I,J,0) = 0.0
   32              END IF
   33            END FOR !J!
   34          END FOR !I!
   35 ***************
   36          FOR I1=1,NROW-1
   37            FOR J1=NROW,I1+1,-1
   38               CALL REDUC1 (FF,NROW,DD,VV,DV,I1,J1)
   39            END FOR !J1!
   40          END FOR !I1!
   41 ****************************
   42 * NOW FF IS LOWER TRIANGULAR. GENERATE THE
   43 * DET. OF FF BY MULTIPLYING ALL THE DIAG.
   44 * ELEMENTS TOGETHER.
   45 * THEN SAVE THE DIAG. ELEMENTS FOR LATER USE . WHEN
   46 * FF ENTERS SUBR. OFFDIAG ALL ITS ELEMENTS WILL BE
   47 * CHANGED .
   48 ****************************
   49          CALL DETDIAG (FF,DD,DEN1,NB,NROW)
   50 ******************
```

```
* NOW GENERATE FF1. THIS WILL BE NEEDED LATER
* THE INVERSE OF FF ,(= VV*FF1 ) . THE SUBR. WILL
* AVOID PRODUCING HIGH ORDER POLY.
* IN THE NUMERATOR & DENOMINATOR .(SEE MANUAL)
*********************************************
        CALL DIAINV (FF,FF1,DD,DF1,NROW)
************
        CALL OFFDIG (FF,VV,DD,DV,NROW)
************
******************
        CALL POLPOL (VV,FF1,VO,DV,DF1,DVO,NROW,NROW,NROW)
************
        RETURN
        END
        SUBROUTINE  REDUC1 (FF,NROW1,DD,VV,DV,II,JJ)
*************************************************************
* THIS SUBROUTINE IS MAINLY USED IN INVERSION & DETERMINANT
* EVALUATION .
* ELEMENTARY ROW OPERATION
* TO ZEROS THE COLUMNS , STARTING FROM THE LAST COLUMN AND THE
* HIGHER ORDER COEFF. THE SUB WILL CHECK EACH TIME FOR THE
* MINI. POLY. AND CONTINUE FOR REDUCE OR INTERCHANGE COLUMN.
* ***********************************************************
        INTEGER DD(NROW1,NROW1), DV(NROW1,NROW1)
        REAL  FF(NROW1,NROW1,0:25),VV(NROW1,NROW1,0:30)
        REAL  MTP
          I=II
          J=JJ
        EPSILON = 10E-13
         EPF = 1.0E-4
         NCOLUMN1 = NROW1
            DO
              TES=ABS(FF(I,I,DD(I,I)))
            IF ((DD(I,J).GE.DD(I,I)).AND.(TES.GT.EPSILON)) THEN
              MTP = FF(I,J,DD(I,J))/FF(I,I,DD(I,I))
* MTP = MULTIPLIER THE COEFF. OF HIGHER POLY/COEFF. OF LOWER POLY .*
*
              K2 = DD(I,J)-DD(I,I) ! DIFFERENCE IN POWER
              FOR I1 =1,NROW1
                FOR K=0,DD(I1,I)
                  FF(I1,J,K+K2)=FF(I1,J,K+K2)-MTP*FF(I1,I,K)
                END FOR ! K !
*
* VV-IS THE UNIT MATRIX ,INITIALLY, AND OF DIMENSION NCOLUMN1*NCOLUMN1*
                FOR K=0,DV(I1,I)
                  VV(I1,J,K+K2)=VV(I1,J,K+K2)-MTP*VV(I1,I,K)
                END FOR ! K !
                IF ((DV(I1,I)+K2).GE.(DV(I1,J)))  THEN
                  KD1=DV(I1,I)+K2
                   DV(I1,J) =KD1
                WHILE (ABS(VV(I1,J,KD1)).LE.EPF)
                  DV(I1,J) = DV(I1,J)-1
                   IF (DV(I1,J).LE.0)  DV(I1,J)=0
                  EXIT WHILE IF (DV(I1,J).LE.0)
                    KD1 = KD1-1
                 END WHILE
                 END IF
                IF ((I1.NE.I).AND.((DD(I1,I)+K2).GE.DD(I1,J))) THEN
                  KD=DD(I1,I)+ K2
                  DD(I1,J)=KD
```

```
111                    WHILE (ABS(FF(I1,J,KD)).LE.EPF)
112                       DD(I1,J)=DD(I1,J)-1
113                     IF (DD(I1,J).LE.0) DD(I1,J)=0
114                     EXIT WHILE IF(DD(I1,J).LE.0)
115                        KD =KD-1
116                    END WHILE
117                     END IF
118 * IF DD(I1,I)+K2 IS LESS THAN DD(I1,J) THE MULTIPLCATON &
119 * ADDTION DOES NOT CHANGE THE DEG OF FF(I1,J,K).
120 * WE CAN NOT JUST ADD TO THE DEG OF FF(I1,J,K) BECAUSE IN THE
121 * CASE OF DD(I1,I)+K2=DD(I1,J) CANCELLATON MAY HAPPEN .
122 * I.E. FF(I1,J,DD(I1,J))=0 AND HENCE DEG OF FF(I1,J,K) IS NO
123 * LONGER =DD(I1,J) . SO YOU HAVE TO SEARCH IN FF(I1,J,K)
124 * FOR THE HIGHEST AVAILABLE COEFF. IT IS NOT NECESSARY
125 * THAT THE NEW DEG FOR FF(I1,J,K) = DD(I1,J)-1 BECAUSE THAT
126 * COEFF. MAY ALREADY WAS ZERO.
127 ************************************************************************
128                END FOR ! I1 !
129                DD(I,J) = DD(I,J)-1
130                IF (DD(I,J).LT.0) THEN
131                  DD(I,J) = 0
132                END IF
133              ELSE
134                 IRR=I
135                CALL RCHANG(FF,NROW1,NCOLUMN1,DD,VV,DV,IRR)
136              END IF
137            EXIT DO  IF ((DD(I,J).EQ.0).AND.(ABS(FF(I,J,0)).LT.EPF))
138            UNTIL ( ((DD(I,J).EQ.0).AND.(ABS(FF(I,J,0)).LT.EPF)))
139          RETURN
140          END
141          SUBROUTINE RCHANG (FF,NROW1,NCOLUMN1,DD,VV,DV,IRR)
142 ************************************************************
143 * ELEMENTRY ROW OPERATION, NCOLUMN1>=NROW1
144 * SORTING FOR THE LOWER POLY AND PUT IT IN DIAGONALLY .
145 * MATRIX VV COTAINS THE IDENTITY MATRIX INITIALLY .
146 * THE SUBR. CHECH THAT THE DIAG. ELEMENTS ARE MNOT ZERO OTHERWISE
147 **INRTERCHANGE AND CHECH THAT IT IS NOT INTERCHANGED BY AN OTHER ZERO
148 * IRR  POINTS TO THE REQUIRED ROW TO BE ARRANGED .
149 ************************************************************
150          INTEGER DD(NROW1,NCOLUMN1),DV(NCOLUMN1,NCOLUMN1)
151           INTEGER  SMALL
152          REAL FF(NROW1,NCOLUMN1,0:25),VV(NCOLUMN1,NCOLUMN1,0:30)
153          EPSILON =10E-13
154          I = IRR
155            TEST7 = ABS(FF(I,I,0))
156            IF ((DD(I,I).EQ.0).AND.(TEST7.LE.EPSILON)) THEN
157 *
158 * IF THE DIAG. ELEMENT =0 INTERCHANGE WITH LAST COLUMN IF IT IS NOT ZERO
159 * ELSE SEARCH THE COLUMN (COLUMN1-1),TILL YOU FIND A NON ZERO ELEMENT IN
160 * THAT ROW.
161 *
162              FOR J2= NCOLUMN1,I+1,-1
163                IF (DD(I,J2).NE.0) THEN
164                  L=J2
165                  GO TO 10 ! EXIT FROM FOR J2
166                ELSE IF (ABS(FF(I,J2,0)).GT.EPSILON) THEN
167                  L=J2
168                  GO TO 10
169                  END IF
170              END FOR !J2!
```

```
171              END IF
172            SMALL = DD(I,I)
173           FOR J=I+1, NCOLUMN1
174             IF (DD(I,J).LT.SMALL) THEN
175               SMALL =DD(I,J)
176              EXIT FOR
177             END IF
178           END FOR !J!
179 * FIND THE COLUMN WITH THE MINI DEGREE , L
180           L=I-1
181             DO
182              L=L+1
183             UNTIL (SMALL.EQ.DD(I,L))
184    10     CONTINUE
185 *  INTERCHANGE ELEMENTS
186 *  INTERCHANGE DEGREE
187             FOR I1=1,NROW1 ! BECAUSE YOU INTERCHANGE ALL ELEM. IN COLUMN
188              NX=DD(I1,I)
189              DD(I1,I) = DD(I1,L)
190              DD(I1,L) = NX
191              IF (DD(I1,I).GT.DD(I1,L)) THEN
192                KK1 = DD(I1,I)
193              ELSE
194                KK1 = DD(I1,L)
195             END IF
196              FOR K= 0,KK1
197                 X = FF(I1,I,K)
198                 FF(I1,I,K) = FF(I1,L,K)
199                 FF(I1,L,K) = X
200              END FOR ! KK1
201             END FOR !I1!
202 ************************************************
203 * NOW THE IDENTITY MATRIX
204             FOR I1 = 1,NCOLUMN1
205              NX1 = DV(I1,I)
206              DV(I1,I) = DV(I1,L)
207              DV(I1,L) = NX1
208              IF (DV(I1,I).GT.DV(I1,L)) THEN
209                KK2 = DV(I1,I)
210              ELSE
211                KK2 = DV(I1,L)
212              END IF
213              FOR K=0,KK2
214                 Y = VV(I1,I,K)
215                 VV(I1,I,K) = VV(I1,L,K)
216                 VV(I1,L,K) = Y
217               END FOR ! KK2!
218             END FOR ! I1!
219 * SEE ALSO SUBR1 CCHANG
220           RETURN
221           END
222           SUBROUTINE POLPOL (FF,F1,GG,DD,DF,DG,NROW,NCOLUMN,NCOLF1)
223 ********************************************************************
224 * TO MULTIPLY TWO ELEMENTS FF(I,J)*F1(J,M) .
225 * THE RESULT IS KEPT IN GG(I,M) . NOTE THEORDER OF MULT.
226 * (FF*F1) <> (F1*FF) .
227 * MULTIPICATION IN THE SCALAR CASE CAN BE WHEN I=J=M=1
228 * J IS THE MOST INNER LOOP , THEN I, ALL ARE CONTAINED WITHIN M
229 ********************************************************************
230           INTEGER DD(NROW,NCOLUMN),DG(NROW,NCOLF1),DF(NCOLUMN,NCOLF1)
```

```
231           REAL FF(NROW,NCOLUMN,0:25),F1(NCOLUMN,NCOLF1,0:25)
232           REAL GG(NROW,NCOLF1,0:30)
233           EPSI = 1.E-9
234            FOR M=1,NCOLF1
235              FOR I=1,NROW
236                MAX=0
237                FOR K = 0,30
238                  GG(I,M,K)=0.0
239                END FOR !K!
240               FOR J=1,NCOLUMN
241               ND1 = DD(I,J)+DF(J,M)
242           FOR K=0,ND1
243            FOR L=0,K
244               GG(I,M,K)=GG(I,M,K)+FF(I,J,L)*F1(J,M,K-L)
245             END FOR !L!
246           END FOR !K!
247                      IF (MAX.LT.ND1) THEN
248                        MAX = DD(I,J)+DF(J,M)
249                      END IF
250                      DG(I,M) = MAX
251                  END FOR !J!
252                   WHILE (ABS(GG(I,M,DG(I,M))).LE.EPSI)
253                     DG(I,M) = DG(I,M)-1
254                     IF (DG(I,M).LE.0) DG(I,M)=0
255                     EXIT WHILE IF (DG(I,M).EQ.0)
256                   END WHILE
257                END FOR !I!
258             END FOR !M!
259           RETURN
260           END
261           SUBROUTINE READP (NROW1,NCOLUMN1,DD,FF)
262 ******************************************************
263 * THIS SUBR. READS IN POLY. MATRICES NOT IN A RATIONAL FORN
264 * IT READS THE DIMENSION OF THE MATRIX M*N, THE DEG. OF EACH
265 * ELEMENT ROW BY ROW AND FINALLY IT READS THE COFFE. OF EACH
266 ****
267 * ELEMENT FROM LOWER TO HIGHER DEGREE ROW BY ROW.
268 * THE MAX. SIZE OF MATRIX IN THIS SUBR NROW1*NCOLUMN1
269 * THE MAX POWER DEG. OF POLY IS 20
270 *******
271 *******
272           REAL FF(NROW1,NCOLUMN1,0:20)
273           INTEGER DD(NROW1,NCOLUMN1)
274 * READ THE DEGREE FIRST ,THIS WILL ECONOMISE THE STORAGE
275 ********
276          FOR I=1,NROW1
277            PRINT *, ' INPUT DEGREES OF NEW ROW '
278            READ (40,*) ( DD(I,J),J=1,NCOLUMN1)
279           END FOR !I!
280.* READ THE COFFE. FROM SMALL****
281           FOR I=1,NROW1
282             PRINT *, ' INPUT COEFF. FOR NEW ROW '
283             FOR J=1,NCOLUMN1
284               PRINT *, ' INPUT COEFF. FOR NEW COLUMN '
285                 READ (40,*) (  FF(I,J,K),K=0,DD(I,J))
286             END FOR !J!
287           END FOR !I!
288           RETURN
289          END
290           SUBROUTINE WRITS(ND,DEN1)
```

```
291 ******************************************
292 * THIS SUBR. WRITES THE OUTPUT IN A FORMAT
293 * READABLE BY SUBR. READS .
294 * IT PRINTS THE DEG. OF THE SCALAR POLY.
295 * THEN THE COEFF. OF THE POLY. FROM LOW TO
296 * HIGH AS USUAL
297 ******************************************
298         REAL DEN1(0:30)
299         WRITE (40,*) ND
300         WRITE (40,*) (DEN1(J),J=0,ND)
301         RETURN
302         END
303         SUBROUTINE WRITP(NROW,NCOLUMN,DD,FF)
304 ************************************************
305 * THEIIS SUBROUTINE WRITES THE OUTPUT IN A FORMAT
306 * THAT IS READABLE BY SUBR. READP .
307 * IT FIRST PRINTS THE DIFFERENT DEGREES THEN
308 * THE ELEMENT COMPONENTS .
309 ************************************************
310         REAL FF(NROW,NCOLUMN,0:20)
311         INTEGER DD(NROW,NCOLUMN)
312         WRITE (40,*)((DD(I,J),J=1,NCOLUMN),I=1,NROW)
313         FOR I=1,NROW
314           FOR J=1,NCOLUMN
315             WRITE (40,*) I,J
316           WRITE (40,*)(FF(I,J,K),K=0,DD(I,J))
317           END FOR !J!
318         END FOR !I!
319         RETURN
320         END
321         SUBROUTINE OFFDIG (FFF,VVV,DDD,DVV,NROW)
322 ********************************************************
323 * THIS SUBROUTINE IS PART OF THE INVERSION OF MATRICES
324 * THE SUBR. PERFORM THE NON ELEMENTRY OPERATION
325 * OF ZEROING THE LOWER PART OFF THE MATRIX & MAKE IT
326 * DIAG. THE SAME OP IS CARRIED COCURRENTLY ON THE
327 * UNIT MATRIX.
328 ********************************************************
329         REAL FFF(NROW,NROW,0:30),VV1(8,8,0:30)
330         REAL VVV(NROW,NROW,0:30),FF1(8,8,0:30)
331         INTEGER DDD(NROW,NROW),DVV(NROW,NROW)
332         EPSI = 1.E-9
333         FOR I=2,NROW
334           FOR J=1,I-1
335               FOR I1=1,NROW
336                ND1=DDD(I,I)+DVV(I1,J)
337                 FOR K=0,ND1
338                   VV1(I1,J,K) = 0.0
339                   FOR L=0,K
340                    VV1(I1,J,K)=VV1(I1,J,K)+FFF(I,I,L)*VVV(I1,J,K-L)
341                   END FOR !L!
342                 END FOR !K!
343                ND2=DDD(I,J)+DVV(I1,I)
344                 FOR K=0,ND2
345                   FOR L=0,K
346                    VV1(I1,J,K)=VV1(I1,J,K)-FFF(I,J,L)*VVV(I1,I,K-L)
347                   END FOR !L!
348                 END FOR !K!
349               IF (ND1.GT.ND2) THEN
350                 DVV(I1,J) = ND1
```

```
351                 ELSE
352                   DVV(I1,J) = ND2
353                 END IF
354                 FOR K=0,DVV(I1,J)
355                   VVV(I1,J,K) = VV1(I1,J,K)
356                 END FOR !K!
357                 WHILE (ABS(VVV(I1,J,DVV(I1,J))).LE.EPSI)
358                   DVV(I1,J)=DVV(I1,J)-1
359                   IF (DVV(I1,J).LE.0) DVV(I1,J)=0
360                   EXIT WHILE IF (DVV(I1,J).EQ.0)
361                 END WHILE
362               END FOR !I1!
363               FOR I1=1,NROW
364                 NDD1 = DDD(I,I)+DDD(I1,J)
365                   FOR K=0,NDD1
366                     FF1(I1,J,K)=0.0
367                     FOR L=0,K
368                     FF1(I1,J,K)=FF1(I1,J,K)+FFF(I,I,L)*FFF(I1,J,K-L)
369                     END FOR !L!
370                   END FOR !K!
371                    NDD2 = DDD(I,J)+DDD(I1,I)
372                   FOR K=0,NDD2
373                     FOR L=0,K
374                     FF1(I1,J,K)=FF1(I1,J,K)-FFF(I,J,L)*FFF(I1,I,K-L)
375                     END FOR !L!
376                   END FOR !K!
377               END FOR !I1!
378               FOR I1=1,NROW
379                   IF (NDD1.GT.NDD2) THEN
380                      DDD(I1,J) = NDD1
381                   ELSE
382                      DDD(I1,J) = NDD2
383                   END IF
384                   FOR K=0,DDD(I1,J)
385                     FFF(I1,J,K) = FF1(I1,J,K)
386                   END FOR !K!
387                  WHILE (ABS(FFF(I1,J,DDD(I1,J))).LE.EPSI)
388                    DDD(I1,J)=DDD(I1,J)-1
389                    IF (DDD(I1,J).LE.0) DDD(I1,J)=0
390                    EXIT WHILE IF (DDD(I1,J).EQ.0)
391                  END WHILE
392                END FOR !I1!
393 *             END IF
394             END FOR !J!
395           END FOR !I!
396           RETURN
397           END
398           SUBROUTINE DETDIAG (FF,DD,DEN1,ND1,NROW)
399 ************************************************************
400 * THIS SUBR. MULTIPLIES THE DIAG. ELEMENTS . IT IS
401 * USED TO EVALUATE THE DET. OF MATRIX FF
402 * ************************************************************
403           REAL FF(NROW,NROW,0:25),DEN1(0:30),G(0:30)
404           REAL GG(0:30)
405           INTEGER DD(NROW,NROW)
406           EPSI = 1.E-9
407           ND1 = 0
408           DEN1(0) = 1.0
409           FOR I=1,NROW
410             FOR K=0,DD(I,I)+ND1
```

```
411             G(K) = 0.0
412             FOR L=0,K
413               G(K) = G(K)+DEN1(L)*FF(I,I,K-L)
414             END FOR !L!
415             GG(K)=G(K)
416           END FOR !K!
417           FOR K=0,ND1+DD(I,I)
418             DEN1(K) = GG(K)
419           END FOR !K!
420           ND1 = ND1+DD(I,I)
421         END FOR !I!
422         WHILE (ABS(DEN1(ND1)).LE.EPSI)
423           ND1=ND1-1
424           IF (ND1.LE.0) ND1=0
425           EXIT WHILE IF(ND1.EQ.0)
426         END WHILE
427         RETURN
428         END
429
430         SUBROUTINE DIAINV (FF,FF1,DD,DF1,NROW)
431 ************************************************************
432 *THIS WILL GENERATE FF1 .  IT WILL BE NEEDED LATER
433 * THE U INVERSE = VV*FF1 .THIS FORM AVOIDS PRODUCING
434 * HIGH ORDER POLY IN THE NUMERATOR & DENOMINATOR .
435 * SEE MANUAL .
436 ************************************************************
437         REAL FF(NROW,NROW,0:25),FF1(NROW,NROW,0:30)
438         REAL G(8,0:30),GG(8,0:30)
439         INTEGER DD(NROW,NROW),DF1(NROW,NROW)
440         EPSI = 1.E-9
441         FF1(1,1,0)=1.0
442         DF1(1,1)=0
443         FOR K=1,25
444           FF1(1,1,K) = 0.0
445         END FOR !K!
446         FOR I=2,NROW
447           DF1(I,I) = 0
448           FF1(I,I,0) = 1.0
449           FOR K=1,25
450             FF1(I,I,K) = 0.0
451           END FOR !K!
452           FOR J=1,I-1
453             FOR K=0,DF1(I,I)+DD(J,J)
454               G(I,K) = 0.0
455               FOR L=0,K
456                 G(I,K)=G(I,K)+FF1(I,I,L)*FF(J,J,K-L)
457               END FOR !L!
458               GG(I,K) = G(I,K)
459             END FOR !K!
460             FOR K=0,DF1(I,I)+DD(J,J)
461               FF1(I,I,K) = GG(I,K)
462             END FOR !K!
463             DF1(I,I) = DF1(I,I)+DD(J,J)
464           END FOR !J!
465             WHILE (ABS(FF1(I,I,DF1(I,I))).LE.EPSI)
466               DF1(I,I) = DF1(I,I)-1
467             IF (DF1(I,I).LE.0) DF1(I,I)=0
468             EXIT WHILE IF (DF1(I,I).EQ.0)
469           END WHILE
470         END FOR !I!
```

```
471           FOR I=1,NROW
472             FOR J=1,NROW
473               IF (I.NE.J) THEN
474               DF1(I,J) = 0
475               FOR K=0,30
476                 FF1(I,J,K) = 0.0
477               END FOR !K!
478               END IF
479             END FOR !J!
480           END FOR !I!
481           RETURN
482           END
483           SUBROUTINE MULSPECT (VV,NROW,DD,AA,DAA)
484 ************************************************************
485 * N IS THE MAX. DEG. IN VV.
486 * NROW = L , THE RANK OF VV .
487 * AA IS THE OUTPUT MATRIX.
488 * THIS SUBROUTINE PRODUCES THE LEFT SPECTRAL FACTORS
489 * OF VV. D.D* WHICH COMES OUT FROM Z.Z*
490 * TO GET THE RIGHT SPECTRAL FACTOR E*.E OF VV' WHICH COMES FROM Z*.Z
491 * (1) GET THE LEFT SPECTRAL OF VV'
492 * VV' = (Z*.Z)' , HAVING OBTAINED A LEFT FACTOR ,SAY, D
493 * THEN  E = D'
494 *  VV IS UINPUT MATRIX
495 *DD POINTS TO THE DEG. OF VV , AS USUAL
496 ************************************************************
497           INTEGER Q,R,P,DD(NROW,NROW),DAA(NROW,NROW)
498           REAL  VV(NROW,NROW,0:20),AA(NROW,NROW,0:10)
499           DOUBLE PRECISION A(40,40),R1,B(40,40),S,EPSI
500           EPSI = 1.D-12
501           EPSI1 = 1.E-6
502 *
503 * TO FIND THE MAX. DEG. IN VV.
504 *
505            MAX = DD(1,1)
506            FOR I=1,NROW
507              FOR J=1,NROW
508                 IF (MAX.LT.DD(I,J)) THEN
509                   MAX = DD(I,J)
510                 END IF
511               END FOR !J!
512             END FOR !I!
513              N = MAX
514              L = NROW
515 ***************
516 * FILL THE LOWER PART OF THE TOEBLITZ MATRIX .  THE UPPER DIAGONAL
517 * PART IS THE TRANSPOSE OF THE LOWER DIAGONAL .
518 ***************
519           FOR LL=0,NROW
520             FOR KK=0,MAX
521                II = KK*NROW
522                JJ = LL*NROW
523               FOR J=1,NROW
524                 J1 = JJ+J
525                 FOR I=1,NROW
526                   I1=II+I+LL*NROW
527                   B(I1,J1) = VV(I,J,KK)
528                 END FOR !I!
529               END FOR !J!
530           END FOR !K!
```

```
      END FOR !LL!
********
* FILL THE UPPER PART
********
       Q=L*N+L
      FOR I=1,Q
        FOR J=I,Q
          B(I,J) = B(J,I)
        END FOR !J!
      END FOR !I!
******************
       FOR I=1,Q
         FOR J=1,Q
*          WRITE (40,*) I,J
*          WRITE (40,*) B(I,J)
         END FOR !J!
       END FOR !I!
********
        S= 0.0
        P= L*N
        Q=P + L
        I= 0
        J= P
       GO TO 26
12      J =0
13      J= J+1
       IF ( A(J,J).LT.EPSI1)   GOTO 13
        R= P
16     R=R +1
        R1 = 0.0
       IF (J.EQ.1)   GO TO 23
         K=0
20       K =K +1
         R1 = R1+ A(J,K)*A(R,K)
       IF (K.LT.(J-1))   GO TO 20
23       A(R,J) = ( B(R,J)-R1)/A(J,J)
       IF (R.LT.Q)   GO TO 16
       IF (J.LT.P)   GO TO 13
26       J=J+1
         R1=0.0
       IF (J.EQ.(P+1))   GOTO 33
         K=0
30       K= K+1
         R1 = R1+ A(J,K)**2
       IF (K.LT.(J-1))   GOTO 30
33       A(J,J) =SQRT(B(J,J)-R1)
       IF (J.EQ.Q)    GOTO 46
         R=J
36       R=R+1
         R1=0.0
       IF (J.EQ.(P+1))   GOTO 43
         K=P
40       K= K+1
         R1 = R1+A(J,K)*A(R,K)
        IF (K.LT.(J-1))   GOTO 40
43       A(R,J)= ( B(R,J)-R1)/A(J,J)
        IF (R.LT.Q)   GOTO 36
         GOTO 26
46      IF (N.EQ.0)   GOTO 58
         J1=P+1
```

```
591          DO 60  J=J1,Q
592               S=S+ A(J,J)-A(J-L,J-L)
593   60     CONTINUE
594         IF ((ABS(S).LT.EPSI).OR.(I.GE.30)) GO TO 57
595           I=I+1
596          DO 70  K= 1,P
597             DO 80  J=1,P
598                A(J,K)= A(J+L,K+L)
599   80        CONTINUE
600   70     CONTINUE
601           GOTO 12
602    57     FOR J=1,NROW
603            FOR KK=0,MAX
604               LL = KK*NROW
605              FOR I=1,NROW
606                AA(I,J,KK) = A(I+LL,J)
607              END FOR !I!
608            END FOR !KK!
609          END FOR !J!
610          FOR I=1,NROW
611            FOR J=1,NROW
612              DAA(I,J) = MAX
613              WHILE (ABS(AA(I,J,DAA(I,J))).LE.EPSI)
614                DAA(I,J)=DAA(I,J)-1
615                IF (DAA(I,J).LE.0) DAA(I,J)=0
616                EXIT WHILE IF (DAA(I,J).EQ.0)
617              END WHILE
618            END FOR !J!
619          END FOR !I!
620   58     RETURN
621          END
622           SUBROUTINE SPECSC (DEN1,ND1,AL)
623 ************************************************************************
624 * THIS SUBR. TO EVALUATE THE SPECTRAL FACTORIZATION OF
625 * SCALAR POLYNOMIAL . DEN CARRIES THE POLY COEFF. ND1 POINT TO THE
626 * DEG OF THE POLY . THE OUTPUT COEFF ARE IN AL ***
627 * THE MAX. POLY. DEG. IS 20
628 ************************************************************************
629          REAL LA(0:20),AL(0:20),W(0:20),R(0:20),ZE(0:20)
630          REAL PH(0:20),DEN1(0:20)
631          TEST = 0.0
632          EPS  = 10E-13
633          FOR J=0,ND1
634            AL(J) = DEN1(J)/SQRT(DEN1(0))
635          END FOR ! J !
636          DO
637            FOR J=0,ND1
638              W(J) = AL(J)
639            END FOR !J!
640            FOR K = 0,ND1
641              FOR J=0,ND1-K
642                R(J)=AL(ND1-K-J)
643              END FOR !J!
644              LA(K) = AL(ND1-K)/R(ND1-K)
645              FOR J=0,ND1-K-1
646                AL(J) = AL(J)-LA(K)*R(J)
647              END FOR !J!
648              ZE(ND1-K) = 2.0*DEN1(ND1-K)/AL(0)
649              IF (K.LT.(ND1-1)) THEN
650                FOR J=1,ND1-K-1
```

```
651                 DEN1(J) = DEN1(J)-ZE(ND1-K)*AL(ND1-K-J)/2.0
652               END FOR !J!
653             END IF
654           END FOR !K!
655           ZE(0) = DEN1(0)/AL(0)
656           K = K-1 ! BECAUSE K NOW = ND1+1
657            DO
658              K = K-1
659              FOR J=0,ND1-K
660                PH(J) = ZE(ND1-K-J)
661                ZE(J) = ZE(J)-LA(K)*PH(J)
662              END FOR !J!
663             EXIT DO IF(K.EQ.0)
664           UNTIL (K.EQ.0)
665           FOR J =0,ND1
666             AL(J) = (ZE(J)+W(J))/2.0
667             TEST  = TEST +AL(J)**2
668           END FOR !J!
669           TEST = TEST-DEN1(0)
670         UNTIL ((TEST.GE.EPS)) ! FINISHES DO AT THE START OF THE SUBR.
671       RETURN
672       END
673       SUBROUTINE READS(ND,DEN1)
674 ****************************************
675       REAL DEN1(0:20)
676 *     WRITE (3,105)
677 *105  FORMAT (' INPUT THE POLY. DEG.   ND  ')
678       READ (40,*). ND
679 *     WRITE (3,115)
680 *115  FORMAT (' INPUT THE POLY COEFF.FROM LOW TO HIGH ')
681       READ (40,*) ( DEN1(J),J=0,ND)
682       RETURN
683       END
684       SUBROUTINE CCHANG(FF,NROW1,NCOLUMN1,DD,VV,DV,JRR)
685 * COLUMN ELEMENTARY OPERATION.  MROW1>=NCOLUMN1
686 * DV POINTS TO THE VARIOUS DEG. OF THE RESULTING IDENTITY
687 * MATRIX .  JRR POINTS TO THE COLUMN NEEDS TO BE ARRANGED
688 ************************************************************
689       INTEGER DD(NROW1,NCOLUMN1),DV(NROW1,NROW1),SMALL
690       REAL FF(NROW1,NCOLUMN1,0:20),VV(NROW1,NROW1,0:20)
691       EPSILON = 10E-13
692       J = JRR
693       TEST6 = ABS(FF(J,J,0))
694       IF ((DD(J,J).EQ.0).AND.(TEST6.LE.EPSILON))  THEN
695         FOR I2 =NROW1,J+1,-1
696           IF (DD(I2,J).NE.0) THEN
697             L = I2
698             GO TO 10
699           ELSE IF  (ABS(FF(I2,J,0)).GT.EPSILON) THEN
700             L = I2
701             GO TO 10
702           END IF
703         END FOR ! I2 !
704         END IF
705       SMALL = DD(J,J)
706       FOR I = J+1,NROW1
707         IF (DD(I,J).LT.SMALL) THEN
708           SMALL = DD(I,J)
709         EXIT FOR
710         END IF
```

```
711             END FOR ! I !
712             L = J-1
713             DO
714               L = L+1
715             UNTIL ( SMALL.EQ.DD(L,J))
716    10     CONTINUE
717             FOR J1 = 1,NCOLUMN1
718               NX = DD(J,J1)
719               DD(J,J1) = DD(L,J1)
720               DD(L,J1) = NX
721               IF (DD(J,J1).GT.DD(L,J1)) THEN
722                 KK1 = DD(J,J1)
723               ELSE
724                 KK1 = DD(L,J1)
725               END IF
726               FOR K=0,KK1
727                 X = FF(J,J1,K)
728                 FF(J,J1,K) = FF(L,J1,K)
729                 FF(L,J1,K) = X
730               END FOR !KK1!
731             END FOR !J1!
732 *********************************
733             FOR J1=1,NROW1
734               NX1 = DV(J,J1)
735               DV(J,J1) = DV(L,J1)
736               DV(L,J1) = NX1
737               IF (DV(J,J1).GT.DV(L,J1)) THEN
738                 KK2 = DV(J,J1)
739               ELSE
740                 KK2 = DV(L,J1)
741               END IF
742               FOR K = 0,KK2
743                 Y = VV(J,J1,K)
744                 VV(J,J1,K) = VV(L,J1,K)
745                 VV(L,J1,K) = Y
746               END FOR ! KK2 !
747             END FOR ! J1!
748 * SEE ALSO SUBR. RCHANG FOR MORE DETAILS
749           RETURN
750           END
751           SUBROUTINE  CREDUCE (FF,NROW1,NCOLUMN1,DD,VV,DV)
752 ************************************************************
753 *
754 * ELEMENTARY ROW OPERATION
755 * TO ZEROS THE COLUMNS , STARTING FROM THE LAST COLUMN AND THE
756 * HIGHER ORDER COEFF. THE SUB WILL CHECK EACH TIME FOR THE
757 * MINI. POLY. AND CONTINUE FOR REDUCE OR INTERCHANGE COLUMN.
758 * ************************************************************
759           INTEGER DD(NROW1,NCOLUMN1), DV(NROW1,NROW1)
760           REAL  FF(NROW1,NCOLUMN1,0:20),VV(NROW1,NROW1,0:20)
761           REAL  MTP
762 **********
763 * GENERATE THE IDENTITY MATRIX
764 **********
765           FOR I=1,NROW1
766             FOR J=1,NROW1
767               DV(I,J) = 0
768               IF (I.EQ.J) THEN
769                 VV(I,J,0) = 1.0
770               ELSE
```

```
771               VV(I,J,0) = 0.0
772             END IF
773           END FOR !J!
774         END FOR !I!
775 **********
776         EPSILON = 10E-13
777         EPF = 1.0E-4
778         FOR J=1,NCOLUMN1
779           FOR I=NROW1,J+1,-1
780             DO
781               TES=ABS(FF(J,J,DD(J,J)))
782             IF ((DD(I,J).GE.DD(J,J)).AND.(TES.GT.EPSILON)) THEN
783               MTP = FF(I,J,DD(I,J))/FF(J,J,DD(J,J))
784 * MTP = MULTIPLIER THE COEFF. OF HIGHER POLY/COEFF. OF LOWER POLY .*
785 *
786               K2 = DD(I,J)-DD(J,J) ! DIFFERENCE IN POWER
787               FOR J1 =1,NCOLUMN1
788                 FOR K=0,DD(J,J1)
789                   FF(I,J1,K+K2)=FF(I,J1,K+K2)-MTP*FF(J,J1,K)
790                 END FOR ! K !
791                 IF ((J1.NE.J).AND.((DD(J,J1)+K2).GE.DD(I,J1))) THEN
792                   KD=DD(J,J1)+ K2
793                   DD(I,J1)=KD
794                  WHILE (ABS(FF(I,J1,KD)).LE.EPF)
795                     DD(I,J1)=DD(I,J1)-1
796                   IF (DD(I,J1).LE.0) DD(I,J1)=0
797                   EXIT WHILE IF(DD(I,J1).LE.0)
798                      KD =KD-1
799                  END WHILE
800                   END IF
801               END FOR ! J1 !
802 *
803 * VV IS THE UNIT MATRIX , INITIALLY, AND OF DIMENSION NCOLUMN1*NCOLUMN1*
804               FOR J1=1,NROW1
805                  FOR K=0,DV(J,J1)
806                    VV(I,J1,K+K2)=VV(I,J1,K+K2)-MTP*VV(J,J1,K)
807                  END FOR ! K !
808                  IF ((DV(J,J1)+K2).GE.(DV(I,J1)))  THEN
809                    KD1=DV(J,J1)+K2
810                     DV(I,J1) =KD1
811                  WHILE (ABS(VV(I,J1,KD1)).LE.EPF)
812                    DV(I,J1) = DV(I,J1)-1
813                     IF (DV(I,J1).LE.0)  DV(I,J1)=0
814                    EXIT WHILE IF (DV(I,J1).LE.0)
815                      KD1 = KD1-1
816                   END WHILE
817                   END IF
818               END FOR !J1!
819 ************************************************************************
820 * IF DD(I1,I)+K2 IS LESS THAN DD(I1,J) THE MULTIPLCATON &
821 * ADDTION DOES NOT CHANGE THE DEG OF FF(I1,J,K).
822 * WE CAN NOT JUST ADD TO THE DEG OF FF(I1,J,K) BECAUSE IN THE
823 * CASE OF DD(I1,I)+K2=DD(I1,J) CANCELLATON MAY HAPPEN .
824 * I.E. FF(I1,J,DD(I1,J))=0 AND HENCE DEG OF FF(I1,J,K) IS NO
825 * LONGER =DD(I1,J) . SO YOU HAVE TO SEARCH IN FF(I1,J,K)
826 * FOR THE HIGHEST AVAILABLE COEFF. IT IS NOT NECESSARY
827 * THAT THE NEW DEG FOR FF(I1,J,K) = DD(I1,J)-1 BECAUSE THAT
828 * COEFF. MAY ALREADY WAS ZERO.
829 ************************************************************************
830               DD(I,J) = DD(I,J)-1
```

```
831               IF (DD(I,J).LT.0) THEN
832                 DD(I,J) = 0
833               END IF
834             ELSE
835                IRR=J
836               CALL CCHANG(FF,NROW1,NCOLUMN1,DD,VV,DV,IRR)
837             END IF
838           EXIT DO  IF ((DD(I,J).EQ.0).AND.(ABS(FF(I,J,0)).LT.EPF))
839           UNTIL ( ((DD(I,J).EQ.0).AND.(ABS(FF(I,J,0)).LT.EPF)))
840           END FOR ! I !
841         END FOR ! J !
842         RETURN
843         END
844         SUBROUTINE TRANSPOSE (FF,FFT,DD,DDT,NROW1,NCOLUMN1)
845
846
847 ********************************************************************
848 * THIS SUBROUTINE TO TRANSPOSE MATRIX FF TO FFT
849 ********************************************************************
850         INTEGER DD(NROW1,NCOLUMN1),DDT(NCOLUMN1,NROW1)
851         REAL FF(NROW1,NCOLUMN1,0:20),FFT(NCOLUMN1,NROW1,0:20)
852         FOR I = 1,NROW1
853           FOR J=1,NCOLUMN1
854             FOR K=0,DD(I,J)
855               FFT(J,I,K) = FF(I,J,K)
856             END FOR !K!
857             DDT(J,I)=DD(I,J)
858           END FOR !J!
859         END FOR !I!
860         RETURN
861         END
862         SUBROUTINE TRANFO (FFT,FFTZ,DDT,NROW1,NCOLUMN1)
863 * ********************************************************************
864 * FFT IS THE TRANSPOSE OF FF
865 * THIS SUBROUTINE TO TRANSFORM FFT( Z^-1) TO FFT(Z). THE NEW
866 * MATRIX WILL BE FFTZ(Z)
867 * Z^N HAS BEEN EXTRACTED AS A COMMON FACTOR
868 ********************************************************************
869         INTEGER DDT(NCOLUMN1,NROW1)
870         REAL FFT(NCOLUMN1,NROW1,0:20)
871         REAL FFTZ(NCOLUMN1,NROW1,0:20)
872         FOR I=1,NCOLUMN1
873           FOR J= 1,NROW1
874             FOR K=0,DDT(I,J)
875               FFTZ(I,J,DDT(I,J)-K)=FFT(I,J,K)
876             END FOR !K!
877           END FOR !J!
878         END FOR !I!
879         RETURN
880         END
881         SUBROUTINE CONJTRAN (FF,FFZ,DD,DDZ,NROW1,NCOLUMN1)
882 ****************************************************
883 * THIS SUBROUTINE TO TRANSPOSE FF(Z^-1) TO FF(Z)
884 ****************************************************
885         INTEGER DD(NROW1,NCOLUMN1),DDZ(NCOLUMN1,NROW1)
886         REAL FF(NROW1,NCOLUMN1,0:20),FFZ(NCOLUMN1,NROW1,0:20)
887         FOR I = 1,NROW1
888           FOR J=1,NCOLUMN1
889             DDZ(J,I)=DD(I,J)
890             FOR K=0,DD(I,J)
```

```
891               FFZ(J,I,DDZ(J,I)-K) = FF(I,J,K)
892             END FOR !K!
893           END FOR !J!
894         END FOR !I!
895         RETURN
896         END
897         SUBROUTINE POLMUL(FF,FFTZ,ZZ,DDT,DD,DDZ,NROW1,NCOLUMN1)
898 ************************************************************************
899 * FFTZ IS THE TRANSPOSE -VE OF FF ' SEE YOULA FOR NOTATION
900 * THIS SUBR. CAM BE USED WITHIN MATMUL DEPENDING ON I, & M
901 * ZZ WILL ACCUMOLATE THE CALCULATION OF ELEMENT MULTIPLCATION
902 ************************************************************************
903 * THE OUTPUT IS IN MATRIX ZZ.  ZZ & G SHOULD INTIALLY BE ZEROED
904 *THE SUBROUTINE CAN BE USED IN SCALAR CASE , IN FACT DEPENDING
905 * ON II,MM & JJ
906 ************************************************************************
907         INTEGER DDT(NCOLUMN1,NROW1),DD(NROW1,NCOLUMN1)
908         INTEGER DDZ(NCOLUMN1,NCOLUMN1)
909         REAL G(10,10,0:25),FF(NROW1,NCOLUMN1,0:25)
910         REAL ZZ(NCOLUMN1,NCOLUMN1,0:25),FFTZ(NCOLUMN1,NROW1,0:25)
911           EPSI = 1.E-6
912         FOR M=1,NCOLUMN1
913         FOR I=1,NCOLUMN1
914            MAX=0
915               FOR K=0,20
916                ZZ(I,M,K)=0.0
917               END FOR !K!
918         FOR J=1,NCOLUMN1
919         FOR K=0,DD(J,M)+DDT(I,J)
920            G(I,M,K) = 0.0
921           FOR L=0,K  ! LOCAL COUNTER
922             G(I,M,K) = G(I,M,K)+FFTZ(I,J,L)*FF(J,M,K-L)
923           END FOR !L!
924         END FOR !K!
925 ************************************************************
926 * ZZ NOW CONTAINS THE INVERSE PART OF Z-DOMAIN (1/Z)
927 * ZZ = F(A)1/Z + F(A)'Z .  IN THE ABOVE WE HAVE CALCULATED F(A).
928 ************************************************************************
929            FOR K=0,DD(J,M)
930              ZZ(I,M,K)=ZZ(I,M,K)+G(I,M,DDT(I,J)+K)
931            END FOR !K!
932           IF (DD(J,M).GT.MAX) MAX=DD(J,M)
933           IF (DDT(I,J).GT.MAX) MAX=DDT(I,J)
934         END FOR !J!
935           WHILE (ABS(ZZ(I,M,MAX)).LE.EPSI)
936             MAX = MAX-1
937           END WHILE
938           DDZ(I,M) = MAX
939         END FOR !I!
940         END FOR !M!
941         RETURN
942         END
943         SUBROUTINE DET (FF,NROW,DD,VV,DV,DEN1,NB)
944 ************************************************************
945 ************************************************************
946         INTEGER DD(NROW,NROW),  DV(NROW,NROW)
947         REAL FF(NROW,NROW,0:25) ,VV(NROW,NROW,0:30)
948         REAL DEN1(0:30)
949         EPSI = 1.E-12
950 *************
```

```
951 * GENERATE THE IDENTITY MATRIX
952 *************
953          FOR I=1,NROW
954            FOR J=1,NROW
955              DV(I,J)=0
956              IF (I.EQ.J) THEN
957                VV(I,J,0) = 1.0
958              ELSE
959                VV(I,J,0) = 0.0
960              END IF
961            END FOR !J!
962          END FOR !I!
963 ***************
964          FOR I1=1,NROW-1
965            FOR J1=NROW,I1+1,-1
966                CALL REDUC1 (FF,NROW,DD,VV,DV,I1,J1)
967            END FOR !J1!
968          END FOR !I1!
969 ****************************
970 * NOW FF IS LOWER TRIANGULAR. GENERATE THE
971 * DET. OF FF BY MULTIPLYING ALL THE DIAG.
972 * ELEMENTS TOGETHER.
973 * THEN SAVE THE DIAG. ELEMENTS FOR LATER USE . WHEN
974 * FF ENTERS SUBR.OFFDIAG ALL ITS ELEMENTS WILL BE
975 * CHANGED .
976 ******************************
977          CALL DETDIAG (FF,DD,DEN1,NB,NROW)
978 ********************
979          RETURN
980          END
981         SUBROUTINE SUMRAT (IIR,GP,GG,M,N,GN1,D1,DD1,DN1)
982 ************************************************************
983 * SUMATION OF RATIONAL FUNCTION
984 ************************************************************
985 * DN1 DEG. OF N1
986 * DD1 DEG. OF D1
987 * M OS THE DEG. OF P
988 * N IS THE DEG. OF G
989 * G IS THE DENOMINATOR  & P IS THE NUMERATORA
990 * THE FINAL RESULT GOES TO N1,D1.
991 * N1 IS THE NUMERATOR AND D1 IS THE DENOMINATOR .
992 ************************************************************
993         INTEGER  DN1,DD1,M(10),N(10)
994         REAL GP(8,0:8),GG(8,0:8),GN1(0:30),D1(0:30)
995         REAL GG(0:30),GN2(0:30),D2(0:30),GD(0:30)
996         EPSI = 1.E-9
997         FOR K=0,M(1)
998           GN1(K) = GP(1,K)
999         END FOR !K!
1000        FOR K=0,N(1)
1001         D1(K) = GG(1,K)
1002        END FOR !K!
1003         DN1 = M(1)
1004         DD1 = N(1)
1005        J = M(1)+M(2)+N(2)+N(1)
1006        FOR I=2,IIR
1007          FOR K=0,J
1008            GN2(K) = 0.0
1009            D2(K) = 0.0
1010            FOR L=0,K
```

```
1011                GN2(K)=GN2(K)+GN1(L)*GG(I,K-L)+D1(L)*GP(I,K-L)
1012                D2(K)=D2(K)+D1(L)*GG(I,K-L)
1013              END FOR !L!
1014                GD(K) = D2(K)
1015                GG(K) = GN2(K)
1016            END FOR !K!
1017              DD1 = DD1+M(I)
1018              DN1 = DN1+N(I)
1019             IF (M(I).LT.N(I))
1020               J=N(I)
1021             ELSE
1022               J=M(I)
1023             END IF
1024             IF (DD1.LT.DN1)
1025               J = J+DN1
1026             ELSE
1027               J = J+DD1
1028             END IF
1029             FOR K=0,J
1030               D1(K) = GD(K)
1031               GN1(K) = GG(K)
1032             END FOR !K!
1033 *             DD1 = J
1034 *            WHILE (ABS(D1(DD1)).LT.EPSI)
1035 *              DD1 = DD1-1
1036 *             IF (DD1.EQ.0) DD1=0
1037 *             EXIT WHILE IF (DD1.EQ.0)
1038 *            END WHILE
1039          END FOR !I!
1040        RETURN
1041        END
1042          SUBROUTINE POMSMU (Z1,Z2,D1,D2,GG,DG,NROW1,NCOLUMN1)
1043 ********************************************************************
1044 * TO MULTIPLY A POLY. MATRIX BY A SCALAR POLY .
1045 * THE COEFF. OF TERM IN ELEMENT GG(K)=[Z(I)*Z2(K-I) ,FOR I=0,K
1046 * [ = TOTAL SUM
1047 * Z1 CARRIES THE SCALAR POLY & Z2 IS THE MULTDIMENSIONAL ONE
1048 * THE RESULT WILL GO TO GG
1049 * DG IS THE DEG. OF THE MATRIX GG
1050 ********************************************************************
1051          INTEGER D1 ,D2(NROW1,NCOLUMN1),DG(NROW1,NCOLUMN1)
1052          REAL Z1(0:D1),Z2(NROW1,NCOLUMN1,0:20),GG(NROW1,NCOLUMN1,0:20)
1053           FOR I=1,NROW1
1054             FOR J=1,NCOLUMN1
1055               FOR K=0,D1+D2(I,J)
1056                   GG(I,J,K)=0.0
1057                 FOR L=0,K
1058                 GG(I,J,K)=GG(I,J,K)+Z1(L)*Z2(I,J,K-L)
1059                 END FOR !L!
1060             END FOR !K!
1061               DG(I,J) = D1+D2(I,J)
1062           END FOR !J!
1063         END FOR !I!
1064         RETURN
1065        END
1066        SUBROUTINE  RREDUCE (FF,NROW1,NCOLUMN1,DD,VV,DV)
1067 ********************************************************************
1068 *
1069 * ELEMENTARY ROW OPERATION
1070 * TO ZEROS THE COLUMNS , STARTING FROM THE LAST COLUMN AND THE
```

```
1071 * HIGHER ORDER COEFF. THE SUB WILL CHECK EACH TIME FOR THE
1072 * MINI. POLY. AND CONTINUE FOR REDUCE OR INTERCHANGE COLUMN.
1073 * ***************************************************************
1074          INTEGER DD(NROW1,NCOLUMN1), DV(NCOLUMN1,NCOLUMN1)
1075          REAL  FF(NROW1,NCOLUMN1,0:20),VV(NCOLUMN1,NCOLUMN1,0:20)
1076          REAL  MTP
1077 *******
1078 * GENERATE THE IDENTITY MATRIX VV BESIDE THE INITIAL DEG. DV.
1079 *******
1080          FOR I=1,NCOLUMN1
1081            FOR J=1,NCOLUMN1
1082                DV(I,J)=0
1083                IF (I.EQ.J) THEN
1084                  VV(I,J,0) = 1.0
1085                ELSE
1086                  VV(I,J,0) = 0.0
1087                END IF
1088            END FOR !J!
1089          END FOR !I!
1090          EPSILON = 10E-13
1091          EPF = 1.0E-4
1092          FOR I=1,NROW1
1093            FOR J=NCOLUMN1,I+1,-1
1094              DO
1095                TES=ABS(FF(I,I,DD(I,I)))
1096              IF ((DD(I,J).GE.DD(I,I)).AND.(TES.GT.EPSILON)) THEN
1097                MTP = FF(I,J,DD(I,J))/FF(I,I,DD(I,I))
1098 * MTP = MULTIPLIER THE COEFF. OF HIGHER POLY/COEFF. OF LOWER POLY .*
1099 *
1100                K2 = DD(I,J)-DD(I,I) ! DIFFERENCE IN POWER
1101                FOR I1 =1,NROW1
1102                  FOR K=0,DD(I1,I)
1103                    FF(I1,J,K+K2)=FF(I1,J,K+K2)-MTP*FF(I1,I,K)
1104                  END FOR ! K !
1105                  IF ((I1.NE.I).AND.((DD(I1,I)+K2).GE.DD(I1,J))) THEN
1106                    KD=DD(I1,I)+ K2
1107                    DD(I1,J)=KD
1108                   WHILE (ABS(FF(I1,J,KD)).LE.EPF)
1109                      DD(I1,J)=DD(I1,J)-1
1110                    IF (DD(I1,J).LE.0) DD(I1,J)=0
1111                    EXIT WHILE IF(DD(I1,J).LE.0)
1112                       KD =KD-1
1113                   END WHILE
1114                    END IF
1115                 END FOR !I1!
1116 * VV IS THE UNIT MATRIX ,INITIALLY, AND OF DIMENSION NCOLUMN1*NCOLUMN1*
1117                FOR I1=1,NCOLUMN1
1118                  FOR K=0,DV(I1,I)
1119                    VV(I1,J,K+K2)=VV(I1,J,K+K2)-MTP*VV(I1,I,K)
1120                  END FOR ! K !
1121                  IF ((DV(I1,I)+K2).GE.(DV(I1,J)))  THEN
1122                    KD1=DV(I1,I)+K2
1123                     DV(I1,J) =KD1
1124                  WHILE (ABS(VV(I1,J,KD1)).LE.EPF)
1125                    DV(I1,J) = DV(I1,J)-1
1126                     IF (DV(I1,J).LE.0)  DV(I1,J)=0
1127                    EXIT WHILE IF (DV(I1,J).LE.0)
1128                      KD1 = KD1-1
1129                   END WHILE
1130                   END IF
```

```
1131                    END FOR !I1!
1132 ************************************************************************
1133 * IF DD(I1,I)+K2 IS LESS THAN DD(I1,J) THE MULTIPLCATON &
1134 * ADDTION DOES NOT CHANGE THE DEG OF FF(I1,J,K).
1135 * WE CAN NOT JUST ADD TO THE DEG OF FF(I1,J,K) BECAUSE IN THE
1136 * CASE OF DD(I1,I)+K2=DD(I1,J) CANCELLATON MAY HAPPEN .
1137 * I.E. FF(I1,J,DD(I1,J))=0 AND HENCE DEG OF FF(I1,J,K) IS NO
1138 * LONGER =DD(I1,J) . SO YOU HAVE TO SEARCH IN FF(I1,J,K)
1139 * FOR THE HIGHEST AVAILABLE COEFF. IT IS NOT NECESSARY
1140 * THAT THE NEW DEG FOR FF(I1,J,K) = DD(I1,J)-1 BECAUSE THAT
1141 * COEFF. MAY ALREADY WAS ZERO.
1142 ************************************************************************
1143                 DD(I,J) = DD(I,J)-1
1144                 IF (DD(I,J).LT.0) THEN
1145                   DD(I,J) = 0
1146                 END IF
1147               ELSE
1148                  IRR=I
1149                 CALL RCHANG(FF,NROW1,NCOLUMN1,DD,VV,DV,IRR)
1150               END IF
1151             EXIT DO  IF ((DD(I,J).EQ.0).AND.(ABS(FF(I,J,0)).LT.EPF))
1152             UNTIL ( ((DD(I,J).EQ.0).AND.(ABS(FF(I,J,0)).LT.EPF)))
1153             END FOR ! J !
1154           END FOR ! I !
1155           RETURN
1156           END
1157           SUBROUTINE POLYADD (Z1,Z2,D1,D2,NROW1,NCOLUMN1,VV,DV)
1158 ************************************************************************
1159 * ADDITION OF TWO POLYNOMIAL MATRICES .
1160 * FOR THE ADDTION TO BE DEFINED THE NO. OF ROWS & COLUMNS IN
1161 * Z1,Z2 MUST BE THE SAME .
1162 ************************************************************************
1163           INTEGER D1(NROW1,NCOLUMN1),D2(NROW1,NCOLUMN1)
1164           REAL Z1(NROW1,NCOLUMN1,0:20),Z2(NROW1,NCOLUMN1,0:20)
1165           REAL VV(NROW1,NCOLUMN1,0:20)
1166           INTEGER DV(NROW1,NCOLUMN1)
1167           FOR I = 1,NROW1
1168             FOR J = 1,NCOLUMN1
1169               IF (D1(I,J).GT.D2(I,J)) THEN
1170                 MAX = D1(I,J)
1171               ELSE
1172                 MAX = D2(I,J)
1173               END IF
1174                FOR K = 0,MAX
1175                  VV(I,J,K)=Z1(I,J,K)+Z2(I,J,K)
1176                END FOR !K!
1177                DV(I,J) = MAX
1178              END FOR !J!
1179            END FOR !I!
1180            RETURN
1181           END
```

```
**********
**********   Pdn =   7          2299-MNI          Terminal: 72
**********
```

LISTING OF 2299MNI *EXPANS LAST UPDATED ON 8-SEP-83 AT 15:54:23

```
      ******************************************************
      * IN THIS PROGRAM THRE DEGREE SHOULD NOT BE LESS
      * THAN 1.  IF A NUMERATOR IS A CONSTANT PUT THE
      * DEGREE 1 AND THE DATA WILL BE THE CONSTANT FOLLOWED
      * BY A ZERO ,SAY.
      ******************************************************
              REAL GG(8,0:12),DEN(0:30)
              INTEGER N1(8),DDEN
              WRITE (3,100)
  100         FORMAT (' INPUT NO. OF FUNCTION TO BE MULTIPLIED ...R ')
              READ (3,*) IR
  200         FORMAT (' INPUT DEG. OF FACTOR ')
  300         FORMAT ('  INPUT COEFF.    FROM LOW TO HIGH ')
              FOR I=1,IR
              WRITE (3,200)
              READ (3,*)  N1(I)
              WRITE (3,300)
              FOR K=0,N1(I)
                READ (3,*) GG(I,K)
              END FOR !K!
              END FOR !I!
              CALL EXPAN (IR,GG,N1,DEN,DDEN)
              CALL WRITS (DDEN,DEN)
              STOP
              END
              SUBROUTINE WRITS(ND,DEN1)
      ******************************************
      * THIS SUBR. WRITES THE OUTPUT IN A FORMAT
      * READABLE BY SUBR. READS .
      * IT PRINTS THE DEG. OF THE SCALAR POLY.
      * THEN THE COEFF. OF THE POLY. FROM LOW TO
      * HIGH AS USUAL
      ******************************************
              REAL DEN1(0:20)
              WRITE (40,*) ND
              WRITE (40,*) (DEN1(J),J=0,ND)
              RETURN
              END
             SUBROUTINE EXPAN (IIR,GG,N,D1,DD1)
      ***********************************************************
      * SUMATION OF RATIONAL FUNCTION
      ***********************************************************
      * DN1 DEG.  OF N1
      * DD1 DEG.  OF D1
      * M OS THE DEG.  OF P
      * N IS THE DEG.  OF Q
      * Q IS THE DENOMINATOR  & P IS THE NUMERATORA
      * THE FINAL RESULT GOES TO N1,D1.
      * N1 IS THE NUMERATOR AND D1 IS THE DENOMINATOR .
      ***********************************************************
```

```
      INTEGER  DN1,DD1,N(10)
      REAL GG(8,0:8),D1(0:30)
      REAL D2(0:30),GD(0:30)
      EPSI = 1.E-9
      FOR K=0,N(1)
       D1(K) = GG(1,K)
      END FOR !K!
       DD1 = N(1)
      J = N(1) + N(2)
      FOR I=2,IIR
        FOR K=0,J
          D2(K) = 0.0
          FOR L=0,K
            D2(K)=D2(K)+D1(L)*GG(I,K-L)
          END FOR !L!
            GD(K) = D2(K)
        END FOR !K!
           J = J+N(I+1)
           DD1=J
         FOR K=0,J
           D1(K) = GD(K)
         END FOR !K!
      END FOR !I!
        WHILE (ABS(D1(DD1)).LT.EPSI)
          DD1 = DD1-1
         IF (DD1.EQ.0) DD1=0
         EXIT WHILE IF (DD1.EQ.0)
        END WHILE
    RETURN
    END
```

****

********** No. of pages 2 2299-MNI Terminal: 72

****

```
**********************************************************************************
**********************************************************************************
**************
**************   Pdn =  7            2299-MNI          Terminal: 72
**************
**********************************************************************************
**********************************************************************************

LISTING OF 2299MNI *EXPANSX  LAST UPDATED ON  6-OCT-83 AT 14:36:01

  1  ******************************************************
  2  * IN THIS PROGRAM THRE DEGREE SHOULD NOT BE LESS
  3  * THAN 1. IF A NUMERATOR IS A CONSTANT PUT THE
  4  * DEGREE 1 AND THE DATA WILL BE THE CONSTANT FOLLOWED
  5  * BY A ZERO , SAY.
  6  ******************************************************
  7          COMPLEX GG(8,0:12),DEN(0:30)
  8          INTEGER N1(8),DDEN
  9          WRITE (3,100)
 10   100    FORMAT (' INPUT NO. OF FUNCTION TO BE MULTIPLIED ...R ')
 11          READ (3,*) IR
 12   200    FORMAT (' INPUT DEG. OF FACTOR ')
 13   300    FORMAT ('  INPUT COEFF.    FROM LOW TO HIGH ')
 14          FOR I=1,IR
 15          WRITE (3,200)
 16          READ (3,*)  N1(I)
 17          WRITE (3,300)
 18          FOR K=0,N1(I)
 19            READ (40,*) GG(I,K)
 20          END FOR !K!
 21          END FOR !I!
 22          CALL EXPAN (IR,GG,N1,DEN,DDEN)
 23          CALL WRITS (DDEN,DEN)
 24          STOP
 25          END
 26          SUBROUTINE WRITS(ND,DEN1)
 27  ******************************************
 28  * THIS SUBR. WRITES THE OUTPUT IN A FORMAT
 29  * READABLE BY SUBR. READS .
 30  * IT PRINTS THE DEG. OF THE SCALAR POLY.
 31  * THEN THE COEFF. OF THE POLY. FROM LOW TO
 32  * HIGH AS USUAL
 33  ******************************************
 34          COMPLEX DEN1(0:20)
 35          WRITE (40,*) ND
 36          WRITE (40,*) (DEN1(J),J=0,ND)
 37          RETURN
 38          END
 39         SUBROUTINE EXPAN (IIR,GG,N,D1,DD1)
 40  ********************************************************
 41  * SUMATION OF RATIONAL FUNCTION
 42  ********************************************************
 43  * DN1 DEG. OF N1
 44  * DD1 DEG. OF D1
 45  * M OS THE DEG. OF P
 46  * N IS THE DEG. OF G
 47  * G IS THE DENOMINATOR  & P IS THE NUMERATORA
 48  * THE FINAL RESULT GOES TO N1,D1.
 49  * N1 IS THE NUMERATOR AND D1 IS THE DENOMINATOR .
 50  ********************************************************
```

```
      INTEGER  DN1,DD1,N(10)
      COMPLEX GG(8,0:8),D1(0:30)
      COMPLEX D2(0:30),GD(0:30)
      EPSI = 1.E-9
      FOR K=0,N(1)
       D1(K) = GG(1,K)
      END FOR !K!
       DD1 = N(1)
      J = N(1) + N(2)
      FOR I=2,IIR
        FOR K=0,J
          D2(K) = 0.0
          FOR L=0,K
            D2(K)=D2(K)+D1(L)*GG(I,K-L)
          END FOR !L!
            GD(K) = D2(K)
        END FOR !K!
           J = J+N(I+1)
           DD1=J
         FOR K=0,J
           D1(K) = GD(K)
         END FOR !K!
      END FOR !I!
         WHILE (ABS(D1(DD1)).LT.EPSI)
           DD1 = DD1-1
          IF (DD1.EQ.0) DD1=0
          EXIT WHILE IF (DD1.EQ.0)
         END WHILE
    RETURN
    END
```

LISTING OF 2299MNI *PD LAST UPDATED ON 9-MAR-84 AT 10:24:30

```
1   ******************************************************
2   * IN THIS PROGRAM THRE DEGREE SHOULD NOT BE LESS
3   * THAN 1.  IF A NUMERATOR IS A CONSTANT PUT THE
4   * DEGREE 1 AND THE DATA WILL BE THE CONSTANT FOLLOWED
5   * BY A ZERO , SAY.
6   ******************************************************
7           REAL GG(15,0:30),DEN(0:30)
8           INTEGER N1(15),DDEN
9           WRITE (3,100)
10   100    FORMAT (' INPUT NO. OF FUNCTION TO BE MULTIPLIED ...R ')
11          READ (3,*) IR
12   200    FORMAT (' INPUT DEG. OF FACTOR ')
13   300    FORMAT ('  INPUT COEFF.    FROM LOW TO HIGH ')
14          FOR I=1,IR
15          WRITE (3,200)
16          READ (3,*)  N1(I)
17          WRITE (3,300)
18          FOR K=0,N1(I)
19            READ (3,*) GG(I,K)
20          END FOR !K!
21          END FOR !I!
22          CALL EXPAN (IR,GG,N1,DEN,DDEN)
23          CALL WRITS (DDEN,DEN)
24          STOP
25          END
26          SUBROUTINE WRITS(ND,DEN1)
27  ******************************************
28  * THIS SUBR. WRITES THE OUTPUT IN A FORMAT
29  * READABLE BY SUBR. READS .
30  * IT PRINTS THE DEG. OF THE SCALAR POLY.
31  * THEN THE COEFF. OF THE POLY. FROM LOW TO
32  * HIGH AS USUAL
33  ******************************************
34          REAL DEN1(0:20)
35          WRITE (40,*) ND
36          WRITE (40,*) (DEN1(J),J=0,ND)
37          RETURN
38          END
39         SUBROUTINE EXPAN (IIR,GG,N,D1,DD1)
40  ******************************************************
41  * SUMATION OF RATIONAL FUNCTION
42  ******************************************************
43  * DN1 DEG. OF N1
44  * DD1 DEG. OF D1
45  * M OS THE DEG. OF P
46  * N IS THE DEG. OF Q
47  * Q IS THE DENOMINATOR  & P IS THE NUMERATORA
48  * THE FINAL RESULT GOES TO N1,D1.
49  * N1 IS THE NUMERATOR AND D1 IS THE DENOMINATOR .
50  ******************************************************
```

```
      INTEGER  DN1,DD1,N(15)
      REAL GG(15,0:30),D1(0:30)
      REAL D2(0:30),GD(0:30)
      EPSI = 1.E-9
      FOR K=0,N(1)
       D1(K) = GG(1,K)
      END FOR !K!
       DD1 = N(1)
      J = N(1) + N(2)
      FOR I=2,IIR
        FOR K=0,J
          D2(K) = 0.0
          FOR L=0,K
            D2(K)=D2(K)+D1(L)*GG(I,K-L)
          END FOR !L!
            GD(K) = D2(K)
        END FOR !K!
           J = J+N(I+1)
           DD1=J
         FOR K=0,J
           D1(K) = GD(K)
         END FOR !K!
      END FOR !I!
         WHILE (ABS(D1(DD1)).LT.EPSI)
           DD1 = DD1-1
          IF (DD1.EQ.0) DD1=0
          EXIT WHILE IF (DD1.EQ.0)
         END WHILE
      RETURN
      END
```

****
****
****
********** No. of pages 2 2299-MNI Terminal: 72
****
****
****

```
**********************************************************************
**********************************************************************
**************
**************   Pdn =   7            2299-MNI          Terminal: 72
**************
**********************************************************************
**********************************************************************

LISTING OF 2299MNI *ONROOTS  LAST UPDATED ON 11-OCT-83 AT 16:11:58

      C     MARK 4 RELEASE NAG COPYRIGHT 1974
      C     MARK 4.5 REVISED
      C     MARK 5C REVISED
      * EXAMPLE :
      * S^2 + 3.S + 2.
      * THE  INPUT COEFF.  GOES FROM 1.0,3.0,2.0
      *
            INTEGER T, NIN, NOUT, I, IFAIL, N, NN, NOF
            DOUBLE PRECISION RE(10), XZ(10), YZ(10), PI, TOL, ZZ, X01AAF, X02AAF,
           * ONE,P1, P15, ZERO, A1P1
      C     *** IMPLEMENTATION DEPENDENT DECLARATION ***
      C     DOUBLE PRECISION DFLOAT
            DATA ONE /1.0D0/, P1 /0.1D0/, P15 /0.15D0/, ZERO /0.0D0/,A1P1 /
           *1.1D0/
            PI = P1*X01AAF(P1)
            TOL = X02AAF(P1)
              WRITE (3,101)
       101    FORMAT(' INPUT THE NO. OF COEFF.   N ')
            READ (3 ,*) N
         20 T = 0
            XZ(1) = P15
            YZ(1) = ZERO
            IFAIL = 1
            IF (N.LE.0) GO TO 160
            NN = N - 1
               WRITE(3,109)
       109     FORMAT(' INPUT THE COEFF. FROM HIGH TO LOW ')
               READ(3,*) ( RE(I),I=1,N)
              WRITE (3,103)
       103    FORMAT('      ROOTS OF THE POLY.       ')
              WRITE (3,104)
       104    FORMAT (' REAL          IMAGINARY          MAGNITUDE   ')
         40 NOF = N - 1
            CALL C02AEF(RE, N, XZ, YZ, TOL, IFAIL)
            IF (IFAIL.NE.0) GO TO 120
         60 I = NOF + 1
         80 I = I - 1
            IF (I.LT.N) GO TO 100
            ZZ = DSQRT(XZ(I)**2+YZ(I)**2)
            WRITE (3,*) XZ(I), YZ(I),ZZ
            GO TO 80
        100 IF (N.NE.1) GO TO 40
        120   WRITE(3,115) IFAIL
        115   FORMAT(' ERROR NUMBER  ',I4)
            IF (T.NE.20 .AND. IFAIL.EQ.2) GO TO 140
            N = N - 1
            GO TO 20
        140 T = T + 1
            XZ(1) = XZ(1)*A1P1*DCOS(DFLOAT(T)*PI) - YZ(1)*A1P1*
           *DSIN(DFLOAT(T)*PI)
```

```
      YZ(1) = XZ(1)*A1P1*DSIN(DFLOAT(T)*PI) + YZ(1)*A1P1*
     *DCOS(DFLOAT(T)*PI)
      GO TO 60
  160 STOP
      END
```

****
****
****
********** No. of pages 2 2299-MNI Terminal: 72
****
****
****

```
**********
**********
**********
**********   Pdn =   7            2299-MNI          Terminal: 72
**********
**********
**********

LISTING OF 2299MNI *PARTIAL  LAST UPDATED ON 11-APR-83 AT 22:24:28

  1  PROGRAM PARTIAL (INPUT,OUTPUT);
  2  (*PROGRAM TO EVALUATE PARTIAL FRACTION*)
  3  (* A POLLYNOMIAL AND TRANSFORM TO Z-DOMAIN*)
  4  TYPE
  5       ROOTS=ARRAY[1..10] OF REAL;
  6  VAR
  7       ZERORES,SIMPLERES,ZERO,POLE : ROOTS;
  8       I,J,L,P,Q,MU : INTEGER;
  9       PO : REAL;
 10  (* DEFINE A KRONEKER DELTAFUNCTION *)
 11    FUNCTION DELTA (II,JJ :INTEGER) :REAL;
 12     BEGIN
 13      IF II=JJ THEN
 14       DELTA := 1.0
 15      ELSE
 16       DELTA := 0.0;
 17     END; (*DELTA*)
 18   FUNCTION POWR(XX : REAL;NN :INTEGER) : REAL;
 19  VAR J : INTEGER;
 20      YY : REAL;
 21  BEGIN
 22    YY := 1.0;
 23   FOR J:=1 TO NN DO
 24     BEGIN
 25        YY := YY*XX;
 26    END;
 27      POWR := YY;
 28   END;
 29   BEGIN
 30      WRITELN('INPUT NO. OF ZEROS Q,POLES P,ZEROS AT ORIGIN MU');
 31      read(Q,P,MU);
 32      WRITELN('INPUT ZEROS OF NUMERATOAR');
 33   FOR I:=1 TO Q DO
 34     BEGIN
 35      READLN(ZERO[I]);
 36     END;
 37     WRITELN (' INPUT POLES OF DEN. ');
 38   FOR I:=1 TO P DO
 39    BEGIN
 40     READLN(POLE[I]);
 41  END;
 42  (*CALCULATE RESIDUES OF SIMPLE POLES*);
 43    SIMPLERES[1]:=1.0 ;
 44   FOR I:=1 TO P DO
 45     BEGIN
 46           PO := POLE[I] ;
 47       FOR J:=1 TO Q DO
 48        BEGIN
 49          IF I<>J THEN
 50        BEGIN
```

```
51          SIMPLERES[I]:= SIMPLERES[I]*(POLE[I]-ZERO[J]);
52          SIMPLERES[I]:= SIMPLERES[I]/(POWR(PO,MU) *(POLE[I]-POLE[J]))
53              END;
54         END;
55    END;
56  (*CALCULATE RESIDUE AT ORIGIN S=0.0*)
57      ZERORES[I]:=0.0;
58    FOR I:=1 TO MU DO
59     BEGIN
60      FOR J:=1 TO P DO
61       BEGIN
62          PO := POLE[J] ;
63            L := I-1 ;
64       ZERORES[I] :=ZERORES[J]+POWR(PO,L)*SIMPLERES[J];
65      END;
66       ZERORES[I]:= DELTA(I,MU)*DELTA(P,Q)-ZERORES[I];
67     END;
68    FOR I :=1 TO P DO
69    BEGIN
70     WRITELN('POLES AT',POLE[I] :8:5);
71     WRITELN('RESIDU AT',SIMPLERES[I] :8:5);
72  END;
73    FOR J:=1 TO MU DO
74    BEGIN
75     WRITELN('RESIDUE AAT ORIGIN',ZERORES[J] :8:5);
76    END;
77  END.
```

****
****
****
********** No. of pages 2 2299-MNI Terminal: 72
****
****
****

```
**********************************************************************
**********************************************************************
**************
**************   Pdn =   7            2299-MNI          Terminal: 72
**************
**********************************************************************
**********************************************************************

LISTING OF 2299MNI *SUMRATIO LAST UPDATED ON 11-AUG-83 AT 16:55:25

  1  ******************************************************
  2  * IN THIS PROGRAM THRE DEGREE SHOULD NOT BE LESS
  3  * THAN 1.  IF A NUMERATOR IS A CONSTANT PUT THE
  4  * DEGREE 1 AND THE DATA WILL BE THE CONSTANT FOLLOWED
  5  * BY A ZERO , SAY.
  6  ******************************************************
  7          REAL PP(8,0:12),QQ(8,0:12),NU(0:30),DEN(0:30)
  8          INTEGER M1(8),N1(8),DNU,DDEN
  9          WRITE (3,100)
 10   100    FORMAT (' INPUT NO. OF FUNCTION TO BE ADDED ...R ')
 11          READ (3,*) IR
 12   200    FORMAT (' INPUT DEG. OF NUMERATOR M & DEN  N ')
 13   300    FORMAT ('   INPUT NUMERATOR COEFF. THEN DEN COEFF. ')
 14          FOR I=1,IR
 15          WRITE (3,200)
 16          READ (3,*) M1(I),N1(I)
 17          WRITE (3,300)
 18          FOR K=0,M1(I)
 19            READ (3,*) PP(I,K)
 20          END FOR !K!
 21          WRITE (3,400)
 22   400    FORMAT (' NOW DEN. COEFF... ')
 23          FOR K=0,N1(I)
 24            READ (3,*) QQ(I,K)
 25          END FOR !K!
 26          END FOR !I!
 27          FOR K=0,2
 28          END FOR
 29          CALL SUMRAT (IR,PP,QQ,M1,N1,NU,DEN,DDEN,DNU)
 30          CALL WRITS (DNU,NU)
 31          WRITE (40,400)
 32          CALL WRITS (DDEN,DEN)
 33          STOP
 34          END
 35          SUBROUTINE WRITS(ND,DEN1)
 36  ******************************************
 37  * THIS SUBR. WRITES THE OUTPUT IN A FORMAT
 38  * READABLE BY SUBR. READS .
 39  * IT PRINTS THE DEG. OF THE SCALAR POLY.
 40  * THEN THE COEFF. OF THE POLY. FROM LOW TO
 41  * HIGH AS USUAL
 42  ******************************************
 43          REAL DEN1(0:20)
 44          WRITE (40,*) ND
 45          WRITE (40,*) (DEN1(J),J=0,ND)
 46          RETURN
 47          END
 48        SUBROUTINE SUMRAT (IIR,GP,GQ,M,N,GN1,D1,DD1,DN1)
 49  ******************************************************
 50  * SUMATION OF RATIONAL FUNCTION
```

```
51  ************************************************************
52  * DN1 DEG. OF N1
53  * DD1 DEG. OF D1
54  * M OS THE DEG. OF P
55  * N IS THE DEG. OF Q
56  * Q IS THE DENOMINATOR  & P IS THE NUMERATORA
57  * THE FINAL RESULT GOES TO N1,D1.
58  * N1 IS THE NUMERATOR AND D1 IS THE DENOMINATOR .
59  ************************************************************
60        INTEGER DN1,DD1,M(10),N(10)
61        REAL GP(8,0:8),GQ(8,0:8),GN1(0:30),D1(0:30)
62        REAL GQ(0:30),GN2(0:30),D2(0:30),GD(0:30)
63        EPSI = 1.E-9
64        FOR K=0,M(1)
65          GN1(K) = GP(1,K)
66        END FOR !K!
67        FOR K=0,N(1)
68         D1(K) = GQ(1,K)
69        END FOR !K!
70         DN1 = M(1)
71         DD1 = N(1)
72        J = M(1)+M(2)+N(2)+N(1)
73        FOR I=2,IIR
74          FOR K=0,J
75            GN2(K) = 0.0
76            D2(K) = 0.0
77            FOR L=0,K
78              GN2(K)=GN2(K)+GN1(L)*GQ(I,K-L)+D1(L)*GP(I,K-L)
79              D2(K)=D2(K)+D1(L)*GQ(I,K-L)
80            END FOR !L!
81              GD(K) = D2(K)
82              GQ(K) = GN2(K)
83          END FOR !K!
84            DD1 = DD1+M(I)
85            DN1 =DD1+ DN1+N(I)
86           IF (M(I).LT.N(I))
87             J=N(I)
88           ELSE
89             J=M(I)
90           END IF
91           IF (DD1.LT.DN1)
92             J = J+DN1
93             DN1=J
94           ELSE
95             J = J+DD1
96             DD1=J
97           END IF
98           FOR K=0,J
99             D1(K) = GD(K)
100            GN1(K) = GQ(K)
101          END FOR !K!
102       END FOR !I!
103          WHILE (ABS(D1(DD1)).LT.EPSI)
104            DD1 = DD1-1
105           IF (DD1.EQ.0) DD1=0
106           EXIT WHILE IF (DD1.EQ.0)
107          END WHILE
108           WHILE (ABS(GN1(DN1)).LT.EPSI)
109             DN1 = DN1-1
110            IF (DN1.EQ.0) DN1=0
```

```
111               EXIT WHILE IF (DN1.EQ.0)
112               END WHILE
113         RETURN
114         END
```

****
****
****
********** No. of pages   3   2299-MNI   Terminal: 72
****
****
****

```
**********************************************************************
**********************************************************************
**************
**************   Pdn =   7            2299-MNI          Terminal: 72
**************
**********************************************************************
**********************************************************************

LISTING OF 2299MNI *UNIMOD   LAST UPDATED ON 29-FEB-84 AT 11:52:57

     1           REAL X1(2,2,0:100),Y1(2,2,0:100),X(2,2,0:100),Y(2,2,0:100)
     2           REAL G1(2,2,0:35)
     3           INTEGER DX1(2,2),DY1(2,2),DG1(2,2),DX(2,2),DY(2,2)
     4            CALL READP(2,2,DX1,X1)
     5            CALL READP(2,2,DY1,Y1)
     6            CALL READP(2,2,DG1,G1)
     7            CALL POLPOL (X1,G1,X,DX1,DG1,DX,2,2,2)
     8            CALL POLPOL (Y1,G1,Y,DY1,DG1,DY,2,2,2)
     9           WRITE (40,102)
    10   102     FORMAT ('  THE NEW    X ')
    11            CALL WRITP (2,2,DX,X)
    12           WRITE (40,103)
    13   103     FORMAT ('  THE NEW    Y ')
    14            CALL WRITP (2,2,DY,Y)
    15           STOP
    16           END

**********************************************************************
**********************************************************************
**************
**************   No. of pages    1   2299-MNI          Terminal: 72
**************
**********************************************************************
**********************************************************************
```

```
**********************************************************************
**********************************************************************
**************
**************   Pdn =   7            2299-MNI          Terminal: 72
**************
**********************************************************************
**********************************************************************

LISTING OF 2299MNI *ZTRANS   LAST UPDATED ON 10-JUN-83 AT 19:45:25

  1  * AFTER GETTING THE ROOTS OF THE DEN. & NUMERATOR BY PROG.
  2  * ROOTS
  3  ****************************************
  4  *  VERY IMPORTANT   P > 1                *
  5  ****************************************
  6  ************************************************************
  7          REAL ZERRES(15),SIMRES(15),ZERO(15),POLE(15),PO
  8          REAL  KR
  9          INTEGER P,Q
 10          WRITE (3,101)
 11   101    FORMAT(' INPUT THE CONSTANT GAIN KR ')
 12          READ (3,*) KR
 13          WRITE (3,100)
 14   100    FORMAT(' INPUT NO. OF ZEROS Q, POLES  P, ZERO AT ORIGIN MU ')
 15          READ(3,*) Q,P,MU
 16          IF (Q.GT.0) THEN
 17          WRITE (3,200)
 18   200    FORMAT(' INPUT ZEROS OF NUMERATOR  ')
 19          READ (3,*) (ZERO(I),I=1,Q)
 20          END IF
 21          WRITE (3,300)
 22   300    FORMAT(' INPUT POLES OF DEN.   ')
 23          READ(3,*) (POLE(I),I=1,P)
 24  *************************
 25  * CALCULATE RESIDUES OF SIMPLE POLES
 26  *************************
 27          FOR I=1,P
 28             SIMRES(I) = KR
 29            PO = POLE(I)
 30            IF (Q.GT.0) THEN
 31            FOR J=1,Q
 32                SIMRES(I)=SIMRES(I)*(PO-ZERO(J))
 33            END FOR !J!
 34            FOR J=1,P
 35               IF (I.NE.J) THEN
 36               SIMRES(I) = SIMRES(I)/(PO-POLE(J))
 37               END IF
 38            END FOR !J!
 39              SIMRES(I) = SIMRES(I)/PO**MU
 40            ELSE IF (Q.EQ.0) THEN
 41               FOR J=1,P
 42             IF (I.NE.J) THEN
 43                 SIMRES(I)=SIMRES(I)/(PO-POLE(J))
 44             END IF
 45               END FOR !J!
 46               SIMRES(I) = SIMRES(I)/PO**MU
 47             END IF
 48         END FOR !I!
 49  *************************
 50  * CALCULATE RESIDUE AT THE ORIGIN   , S=0 .
```

```
      ************************
            IF (MU.GE.1) THEN
            FOR I=1,MU
                SUM =0.0
              FOR J=1,P
                PO = POLE(J)
                L  = I-1
                SUM = SUM+PO**L*SIMRES(J)
              END FOR !J!
              ZERRES(I)=KR*DELTA(I,MU)*DELTA(P,Q) -SUM
            END FOR !I!
            END IF
      ************************
      * THE ZERO AT THE ORIGIN OF THIS FORM
      * A1/S,A2/S^2, A3/S^3 , A4/S^4 ,....UP TO MU
      ************************
            WRITE (30,401)
      401   FORMAT('              POLES   AT            ')
           WRITE (30,400) ( POLE(I),I=1,P)
      400   FORMAT(6(4X,F12.3))
            WRITE (30,405)
      405   FORMAT(' THE RESIDUE OF THE ABOVE POLES    ')
            WRITE (30,400) (SIMRES(I),I=1,P)
             IF (Q.GT.0) THEN
            WRITE (30,406)
      406   FORMAT(' THE ZEROS OF THE ABOVE SYSTEM  ARE ')
            WRITE(30,400) (ZERO(I),I=1,Q)
             END IF
             IF (MU.GE.1) THEN
            WRITE (30,407)
      407   FORMAT(' THE ORDER OF THE POLES AT ORIGIN ')
            WRITE(30,408) (J,J=1,MU)
      408   FORMAT(4(8X,I2))
            WRITE (30,409)
      409   FORMAT(' THE RESIDUES AT THE ORIGIN ,S=0,  ARE ')
            WRITE(30,400) (ZERRES(J),J=1,MU)
             END IF
            STOP
            END
      **********************
      * DEFINE A KRONEKER DELTA FUNCTION
      **********************
             FUNCTION DELTA(II,JJ)
               IF (II.EQ.JJ) THEN
                  DELTA =1.0
               ELSE
                  DELTA =0.0
               END IF
               RETURN
             END
```

```
**************
**************   No. of pages     2   2299-MNI          Terminal: 72
**************
```

```
**********************************************************************
**********************************************************************
**************
**************   Pdn =   7            2299-MNI          Terminal: 72
**************
**********************************************************************
**********************************************************************

LISTING OF 2299MNI *ZTRANCX  LAST UPDATED ON 10-JUN-83 AT 19:46:02

  1  * AFTER GETTING THE ROOTS OF THE DEN. & NUMERATOR BY PROG.
  2  * ROOTS
  3  ****************************************
  4  *  VERY IMPORTANT   P > 1                *
  5  ****************************************
  6  ********************************************************
  7        COMPLEX   ZERRES(15),SUM,SIMRES(15),ZERO(15),POLE(15),PO
  8          REAL  KR
  9          INTEGER P,Q
 10          WRITE (3,101)
 11   101    FORMAT(' INPUT THE CONSTANT GAIN KR ')
 12          READ (3,*) KR
 13          WRITE (3,100)
 14   100    FORMAT(' INPUT NO. OF ZEROS Q, POLES  P, ZERO AT ORIGIN MU ')
 15          READ(3,*) Q,P,MU
 16          IF (Q.GT.0) THEN
 17          WRITE (3,200)
 18   200    FORMAT(' INPUT ZEROS OF NUMERATOR  ')
 19          READ (3,*) (ZERO(I),I=1,Q)
 20          END IF
 21          WRITE (3,300)
 22   300    FORMAT(' INPUT POLES OF DEN.   ')
 23          READ(3,*) (POLE(I),I=1,P)
 24  ************************
 25  * CALCULATE RESIDUES OF SIMPLE POLES
 26  ************************
 27        FOR I=1,P
 28           SIMRES(I) = KR
 29          PO = POLE(I)
 30          IF (Q.GT.0) THEN
 31          FOR J=1,Q
 32              SIMRES(I)=SIMRES(I)*(PO-ZERO(J))
 33          END FOR !J!
 34          FOR J=1,P
 35             IF (I.NE.J) THEN
 36             SIMRES(I) = SIMRES(I)/( (PO-POLE(J)))
 37             END IF
 38          END FOR !J!
 39            SIMRES(I) = SIMRES(I)/PO**MU
 40          ELSE IF (Q.EQ.0) THEN
 41             FOR J=1,P
 42           IF (I.NE.J) THEN
 43              SIMRES(I)=SIMRES(I)/( (PO-POLE(J)))
 44           END IF
 45             END FOR !J!
 46           SIMRES(I) = SIMRES(I)/PO**MU
 47           END IF
 48        END FOR !I!
 49  ************************
 50  * CALCULATE RESIDUE AT THE ORIGIN   , S=0 .
```

```
      ************************
            IF (MU.GE.1) THEN
           FOR I=1,MU
               SUM =0.0
             FOR J=1,P
               PO = POLE(J)
               L  = I-1
               SUM = SUM+PO**L*SIMRES(J)
             END FOR !J!
             ZERRES(I)=KR*DELTA(I,MU)*DELTA(P,Q) -SUM
           END FOR !I!
           END IF
      ************************
      * THE ZERO AT THE ORIGIN OF THIS FORM
      * A1/S,A2/S^2, A3/S^3 , A4/S^4 ,....UP TO MU
      ************************
           WRITE (30,401)
       401 FORMAT('              POLES   AT            ')
          WRITE (30,400) ( POLE(I),I=1,P)
       400 FORMAT(6(4X,F12.3))
           WRITE (30,405)
       405 FORMAT(' THE RESIDUE OF THE ABOVE POLES    ')
           WRITE (30,400) (SIMRES(I),I=1,P)
            IF (Q.GT.0) THEN
           WRITE (30,406)
       406 FORMAT(' THE ZEROS OF THE ABOVE SYSTEM  ARE ')
           WRITE(30,400) (ZERO(I),I=1,Q)
            END IF
            IF (MU.GE.1) THEN
           WRITE (30,407)
       407 FORMAT(' THE ORDER OF THE POLES AT ORIGIN ')
           WRITE(30,408) (J,J=1,MU)
       408 FORMAT(4(8X,I2))
           WRITE (30,409)
       409 FORMAT(' THE RESIDUES AT THE ORIGIN ,S=0,  ARE ')
           WRITE(30,400) ( ZERRES(J),J=1,MU)
            END IF
          STOP
           END
      **********************
      * DEFINE A KRONEKER DELTA FUNCTION
      **********************
            FUNCTION DELTA(II,JJ)
              IF (II.EQ.JJ) THEN
                 DELTA =1.0
              ELSE
                 DELTA =0.0
              END IF
              RETURN
            END
```

```
**************************************************************************
**************************************************************************
**************
**************  No. of pages     2   2299-MNI          Terminal: 72
**************
**************************************************************************
**************************************************************************
```

```
**********
**********   Pdn =   7            2299-MNI          Terminal: 72
**********
**********

LISTING OF 2299MNI *DISCR2   LAST UPDATED ON 27-MAY-83 AT 12:56:05

      1          DIMENSION A(0:15),B(0:15),Y(0:15),U(0:15)
      2          INTEGER A3(10)
      3          DIMENSION YR(0:250)
      4          WRITE(3,100)
      5   100    FORMAT (' INPUT BIASED VALUE ....BIAS ')
      6          READ (3,*) BIAS
      7          WRITE (3,200)
      8   200    FORMAT (' INPUT TYPE & MAGNITUDE OF INPUT ..U M&UO ')
      9          READ (3,*) M,UO
     10          WRITE (3,300)
     11   300    FORMAT (' INPUT MAX. NO. OF DELAYS ...N ')
     12          READ (3,*) N
     13          WRITE (3,400)
     14   400    FORMAT (' INPUT TIME DURATION OF EXP. ......LOFEX ')
     15          READ (3,*) LOFEX
     16          WRITE (3,500)
     17   500    FORMAT (' INPUT COEFF. OF F.B.    B MATRIX ')
     18          READ (3,*) (B(L),L=0,N-1)
     19          WRITE (3,600)
     20   600    FORMAT (' INPUT COEFF. OF F.F       A MATRIX ')
     21          READ (3,*) (A(L),L=0,N)
     22          WRITE(3,700)
     23   700    FORMAT(' INPUT SAMPLING INTERVAL IT  ')
     24          READ (3,*) IT
     25   C ***************************************************************
     26   C  THE DATA FEEDING FOR B STARTS FROM THE COEFF OF HIGHEST
     27   C  D AND DOWNWORDS.
     28   C  THE DATA FEEDING FOR A STARTS FROM THE CONSTANT TERM AO
     29   C  AND UPWORDS.
     30   C ***************************************************************
     31    1      FOR J=0,N
     32             Y(J)=0.0
     33             U(J)=0.0
     34            END FOR
     35            B(N) = 0.0
     36           FOR K = 0,LOFEX
     37            U(0)=UO*((K+1)*IT)**M
     38             Y(0)=0.0
     39            FOR I=0,N
     40               Y(0)= Y(0)+A(I)*U(I)-B(I)*Y(N-I)
     41            ENDFOR
     42             YR(K)= Y(0)+BIAS
     43             CALL DELAY(N,Y)
     44             CALL DELAY(N,U)
     45            ENDFOR
     46              WRITE (40,*) (YR(K),K=0,LOFEX)
     47           WRITE (3,777)
     48   777    FORMAT(' INPUT NO OF CHANGES NC.... ')
     49          READ(3,*) NC
     50          IF (NC.GT.0) THEN
```

```
      WRITE (3,888)
888   FORMAT(' INPUT POSI. OF POLES')
      READ (3,*) (A3(K),K=1,NC)
      WRITE (3,889)
889   FORMAT(' INPUT CHANGES .. ')
      READ (3,*) (A(A3(K)),K=1,NC)
        GOTO 1
        END IF
      STOP
      END
         SUBROUTINE DELAY (NN,XX)
           INTEGER  NN
           DIMENSION XX(0:NN)
        FOR J=NN-1,0,-1
           XX(J+1)=XX(J)
           END FOR
         RETURN
      END
```

```
*******************************************************************************
*******************************************************************************
***************
***************   No. of pages     2   2299-MNI          Terminal: 72
***************
*******************************************************************************
*******************************************************************************
```

```
**********************************************************************************
**********************************************************************************
**************
**************  Pdn =   7            2299-MNI          Terminal: 72
**************
**********************************************************************************
**********************************************************************************

LISTING OF 2299MNI *CJF      LAST UPDATED ON 16-FEB-84 AT 16:17:38

  1  C     G02CJF EXAMPLE PROGRAM TEXT
  2  C     MARK 7 RELEASE. NAG COPYRIGHT 1978.
  3  C     .. LOCAL SCALARS ..
  4        DOUBLE PRECISION COV, VT1, VT2
  5        INTEGER I, IFAIL, NOUT
  6  C     .. LOCAL ARRAYS ..
  7        DOUBLE PRECISION C(40,40), SIGSQ(1), THETA(40,1),
  8       * X(40,40), WK1(40,4), WK2(40),Y(40,1)
  9        INTEGER IPIV(40)
 10  C     .. SUBROUTINE REFERENCES ..
 11  C     G02CJF
 12  C     ..
 13        DATA NOUT /40/
 14         WRITE (3,115)
 15   115    FORMAT (' INPUT NO. OF CASES N  ')
 16         READ (3,*) N
 17         WRITE (3,116)
 18   116    FORMAT (' INPUT NO. OF INDEPENDENT VAR. M ')
 19         READ (3,*) M
 20         FOR J=1,N
 21           READ (40,*) (X(J,L),L=1,M)
 22         END FOR !J!
 23           READ (40,*) (Y(J,1),J=1,N)
 24        IFAIL = 0
 25        CALL G02CJF(X, 40, Y, 40, N, M, 1, THETA, 40, SIGSQ,
 26       * C, 40, IPIV, WK1, WK2, IFAIL)
 27         WRITE (40,121)
 28   121   FORMAT(' COEFF. VALUES  ')
 29         FOR J=1,M
 30           WRITE (40,*) THETA(J,1)
 31         END FOR !J!
 32         WRITE (40,122)
 33   122   FORMAT (' VARIANCE OF THETA  ')
 34         FOR J=1,M
 35           WRITE (40,*) SIGSQ(1)*C(J,J)
 36         END FOR
 37         WRITE (40,123)
 38   123   FORMAT (' VARIANCE OF RESIDUALS  ')
 39         WRITE (40,*) SIGSQ(1)
 40        STOP
 41        END
```

```
**********************************************************************
**********************************************************************
**************
**************   Pdn =   7            2299-MNI          Terminal: 72
**************
**********************************************************************
**********************************************************************

LISTING OF 2299MNI *CJF2     LAST UPDATED ON 21-FEB-84 AT 21:39:31

      C     G02CJF EXAMPLE PROGRAM TEXT
      C     MARK 7 RELEASE. NAG COPYRIGHT 1978.
      C     .. LOCAL SCALARS ..
            DOUBLE PRECISION COV, VT1, VT2
            INTEGER I, IFAIL, NOUT
      C     .. LOCAL ARRAYS ..
            DOUBLE PRECISION C(80,80), SIGSQ(1), THETA(80,1),
           * X(80,80), WK1(80,4), WK2(80),Y(80,1)
            INTEGER IPIV(80)
      C     .. SUBROUTINE REFERENCES ..
      C     G02CJF
      C     ..
            DATA NOUT /40/
             WRITE (3,115)
       115    FORMAT (' INPUT NO. OF CASES N  ')
             READ (3,*) N
             WRITE (3,116)
       116    FORMAT (' INPUT NO. OF INDEPENDENT VAR. M ')
             READ (3,*) M
             FOR J=1,N
               READ (40,*) (X(J,L),L=1,M)
             END FOR !J!
               READ (40,*) (Y(J,1),J=1,N)
            IFAIL = 0
            CALL G02CJF(X, 80, Y, 80, N, M, 1, THETA, 80,SIGSQ,
           * C, 80, IPIV, WK1, WK2, IFAIL)
             WRITE (40,121)
       121   FORMAT(' COEFF. VALUES  ')
             FOR J=1,M
               WRITE (40,*) THETA(J,1)
             END FOR !J!
             WRITE (40,122)
       122   FORMAT (' VARIANCE OF THETA  ')
             FOR J=1,M
               WRITE (40,*) SIGSQ(1)*C(J,J)
             END FOR
             WRITE (40,123)
       123   FORMAT (' VARIANCE OF RESIDUALS  ')
             WRITE (40,*) SIGSQ(1)
            STOP
            END
```

```
**********************************************************************
**********************************************************************
**************
**************   Pdn =   7            2299-MNI          Terminal: 72
**************
**********************************************************************
**********************************************************************

LISTING OF 2299MNI *SMINI    LAST UPDATED ON 18-OCT-83 AT 13:21:28

  1  ************************************************
  2  * THIS PROGRAM IS TO EVALUATE THE COMPLEX       *
  3  * MATRIX OF B1*A/K1*Z FOR A RANGE OF FREG.      *
  4  * THEN, THE DATA IS USED FOR MINIMIZATION       *
  5  * BY THE METOD OF LEAST-SQUARE BE ORTHOGONAL    *
  6  * DECOMPOSITION .                               *
  7  ************************************************
  8         REAL B(10),A(10),K1(10),XA(10)
  9         COMPLEX ZZ(10,10),XA1(10)
 10         COMPLEX B1,A1,K11,Z1
 11         REAL RE,IM
 12         WRITE (3,103)
 13  103    FORMAT (' INPUT THE CHOOSEN DEGREE OF K2 ')
 14         READ (3,*) M
 15          CALL READA (NA,A)
 16          CALL READA (NB,B)
 17          CALL READA (NK,K1)
 18          CALL READA (NXA,XA)
 19         NW = 0
 20         FOR W = 0,.2,.05
 21           NW = NW + 1
 22           W1 = W * 687.2727
 23           IM = -SIN(W1)
 24           RE = COS(W1)
 25           Z1 = CMPLX(RE,IM)
 26           FOR J = 1,NB
 27             B1 = B1 + B(J)*Z1**(J-1)
 28           END FOR
 29           FOR J = 1,NA
 30             A1 = A1 + A(J)*Z1**(J-1)
 31           END FOR
 32           FOR J = 1,NK
 33             K11 = K11 + K1(J)*Z1**(J-1)
 34           END FOR
 35           FOR J = 1,NXA
 36             XA1(NW) = XA1(NW) + XA(J)*Z1**(J-1)
 37           END FOR
 38           BAK = B1*A1/K11
 39           FOR J = 1,M
 40             ZZ(NW,J) = BAK*Z1**(J-1)
 41           END FOR
 42           WRITE (50,*)   (REAL(ZZ(NW,J)),J=1,M)
 43           END FOR !W!
 44         FOR J2=1,NW
 45           WRITE (50,*) (AIMAG( ZZ(J2,J)),J=1,M)
 46         END FOR
 47         FOR J1=1,NW
 48            WRITE (50,*) REAL( XA1(J1))
 49         END FOR
 50         FOR J3=1,NW
```

```
      WRITE (50,*) AIMAG(XA1(J3))
    END FOR
      STOP
      END
       SUBROUTINE READA (MB,BB)
          REAL BB(3)
          READ (40,*) MB
          READ (40,*) (BB(I),I=1,MB)
          RETURN
          END
```

****
****
****
********** No. of pages 2 2299-MNI Terminal: 72
****
****
****

```
**********
**********   Pdn =   7            2299-MNI          Terminal: 72
**********
```

LISTING OF 2299MNI *MMINI LAST UPDATED ON 28-FEB-84 AT 10:58:10

```
  1  ************************************************
  2  * DATA FILE IS MINI.D1
  3  * READ FIRST XA1  I.E ELEMENT XA(1,1) OR XA(1,2)
  4  * NEXT READ XA2 . I.E ELEMENT XA(2,1),OR XA(2,2)*
  5  * READ ALPA                                     *
  6  ************************************************
  7        COMPLEX Y1(20),DK,Y2(20),Z1,A(2,24)
  8        REAL XA1(18),XA2(18)
  9       REAL ALPA(2,4,0:40),IM, DK1(10),RE
 10        INTEGER DALPA(2,4)
 11       WRITE (3,103)
 12  103  FORMAT ('  INPUT THE CHOOSEN DEG. FOR K2 ')
 13        READ (3,*) M
 14        CALL READA (NXA1,XA1)
 15        CALL READA (NXA2,XA2)
 16        CALL READA (NDK1,DK1)
 17         FOR I=1,2
 18            READ (40,*)( DALPA(I,J),J=1,4)
 19         END FOR
 20        FOR I=1,2
 21          FOR J =1,4
 22           READ (40,*) LL,KK
 23           READ (40,*)(ALPA(I,J,K),K=0,DALPA(I,J))
 24          END FOR !J!
 25        END FOR !I!
 26       WRITE (3,*) LL,KK
 27        NW=0
 28        W = 0.0
 29       FOR NOW =1,20
 30         NW = NW + 1
 31         W1 = W*286.4789
 32         IM = -SIN(W1)
 33         RE = COS(W1)
 34         Z1 = CMPLX(RE,IM)
 35         Y1(NW) = (0.0,0.0)
 36         Y2(NW) = (0.0,0.0)
 37        DK = (0.0,0.0)
 38          FOR J = 1,NXA1+1
 39           Y1(NW) = Y1(NW)+ XA1(J)*Z1**(J-1)
 40          END FOR !J!
 41          FOR J = 1,NXA2+1
 42           Y2(NW) = Y2(NW)+ XA2(J)*Z1**(J-1)
 43          END FOR !J!
 44          FOR J=1,NDK1+1
 45            DK = DK + DK1(J)*Z1**(J-1)
 46          END FOR !J!
 47         FOR I = 1,2
 48          J = 1
 49          FOR L = 1,4
 50           FOR J = J,L*M   ! M IS THE DEGREE OF K2
```

```
51              FOR K =0,DALPA(I,L)
52            A(I,J) = A(I,J) + ALPA(I,L,K)*Z1**(K-1+J)
53              END FOR !K!
54               A(I,J) =   A(I,J)/DK
55             END FOR !J!
56            END FOR !L!
57            WRITE (50,*)(REAL(A(I,J)),J=1,4*M)
58           END FOR !I!
59            W = W + .01
60           END FOR !W!
61           FOR J = 1,NW
62             WRITE (50,*) REAL(Y1(J))
63             WRITE (50,*) REAL(Y2(J))   ! CHANNEL 50 TO DATA FILE CJFD1
64           END FOR !J!
65           STOP
66           END
67           SUBROUTINE READA (MB,BB)
68            REAL BB(3)
69           READ (40,*) MB
70           READ (40,*) (BB(I),I=1,MB+1)
71           RETURN
72           END
```

****
****
****
********** No. of pages 2 2299-MNI Terminal: 72
****
****
****

```
**********
**********
**********
**********   Pdn =   7            2299-MNI          Terminal: 72
**********
**********
**********

LISTING OF 2299MNI *CONTRL   LAST UPDATED ON 14-OCT-82 AT 15:14:10

  1  C        M=NO. OF COLUMNS
  2  C     C  N=NO. OF ROWS
  3  C     C  R=HIGHEST ORDER IN NUMER.
  4  C     C  Q=HIGHEST ORDER IN DENU.   Q>R
  5  C     C  Y=OUT PUT FROM PROCESS
  6  C     C  UO=INPUT SET VALUE
  7  C     C  AP=COEFF. OF NUMER. POLYNOMIAL IN PLANT MATRIX
  8  C     C  BP=COEFF. OF DENUM. POLYNOMIAL IN PLANT MATRIX
  9  C     C  AC=COEFF. OF NUMER. POLY. IN CONTROLLER MATRIX
 10  C     C  BC=COEFF. OF DENUM. POLY. IN CONTROLLER MATRIX
 11  C     C  ELEMENTS ARE READ ROW BY ROW
 12  C     C **********************************************
 13  C     C **********************************************
 14           REAL AP(4,4,6),BP(4,4,6),AC(4,4,6),BC(4,4,6)
 15           REAL UO(4),YO(4),W(4),E(4),Y(4)
 16           REAL YP(4,4,6),YC(4,4,6)
 17           REAL XG(1600),YG(1600,4)
 18           INTEGER RP,RC,QP,QC,T
 19           WRITE(3,150)
 20    150    FORMAT(21H INPUT NO. OF ROWS  N)
 21           READ (3,*) N
 22           WRITE(3,250)
 23    250    FORMAT(24H INPUT NO. OF COLUMNS  M)
 24           READ (3,*) M
 25           WRITE (3,350)
 26    350    FORMAT(40H IN PLANT PUT HIGHEST POWER IN NUMER. RP)
 27           READ (3,*) RP
 28           WRITE(3,450)
 29    450    FORMAT(37H INPUT HIGHEST POWER IN PLANT DEN. QP)
 30           READ (3,*) QP
 31           WRITE(3,550)
 32    550    FORMAT(33H INPUT TOTAL TIME OF OPERATION  T)
 33           READ (3,*) T
 34           WRITE(3,650)
 35    650    FORMAT(22H INPUT 'SET VALUE'  UO)
 36           READ (3,*) (UO(L),L=1,M)
 37           WRITE(3,750)
 38    750    FORMAT(27H INPUT INTEGRATION STEP  ST)
 39           READ (3,*) ST
 40            DO 100 I=1,N
 41              WRITE(3,850)
 42    850       FORMAT(14H READ NEXT ROW)
 43             DO 110 K=1,M
 44  C    IF Q NOT EQ. R THEN AA(I,K,R+1)=0.0
 45              WRITE(3,950)
 46    950       FORMAT(35H READ NEXT ELEMENT IN PLANT ROW  AP)
 47                READ(3,*) (AP(I,K,J),J=1,RP+1)
 48           WRITE (3,951)
 49  951      FORMAT(36H READ NEXT ELEMENT IN PLANT DEN.  BP)
 50                READ(3,*) (BP(I,K,J),J=1,QP)
```

```
 110  CONTINUE
 100  CONTINUE
   1   WRITE(3,1350)
1350   FORMAT(40H INPUT HIGHEST POWER IN CONTR NUMER.   RC)
       READ (3,*) RC
       WRITE(3,1300)
1300   FORMAT(39H INPUT HIGHEST POWER IN CONTR. DEN.   QC)
       READ (3,*) QC
      DO 200 I=1,N
           WRITE(3,850)
        DO 210 K=1,M
           WRITE(3,1050)
1050   FORMAT(40H READ NEXT ELEMENT IN CONTROLLER ROW  AC)
           READ(3,*) (AC(I,K,J),J=1,RC+1)
       WRITE (3,1051)
1051   FORMAT(22H READ NEXT ELEMENT  BC)
           READ(3,*) (BC(I,K,J),J=1,QC)
 210 CONTINUE
 200  CONTINUE
      WRITE(3,1150)
1150  FORMAT(38H INPUT OUTPUT INITIAL STEADY-STATE  YO)
      READ (3,*) (YO(L),L=1,M)
      DO 300 I=1,N
        DO 310 K=1,M
             Y(K)=0.0
          DO 320 J=1,QP
             YP(I,K,J)=0.0
             YC(I,K,J)=0.0
 320      CONTINUE
 310    CONTINUE
 300  CONTINUE
      TIME =0.0
      DO 400 L=1,T
        DO 410 K=1,M
          E(K) = UO(K)-Y(K)-YO(K)
 410    CONTINUE
        CALL WIEN(E,ST,N,M,RC,QC,AC,BC,YC,W)
        CALL WIEN(W,ST,N,M,RP,QP,AP,BP,YP,Y)
             TIME=TIME + ST
             XG(L)=TIME
          DO 420 K=1,M
             YG(L,K)= YO(K)+Y(K)
 420      CONTINUE
 400  CONTINUE
       DO 500 K=1,M
         WRITE(3,*) (YG(L,K),L=1,T)
500    CONTINUE
       WRITE(3,1250)
1250   FORMAT(23H RESPONSE O.K.  ? 1 OR 0)
          READ(3,*) RES
         IF (RES.EQ.0) GO TO 1
       STOP
       END
       SUBROUTINE WIEN(EE,H,NN,MM,RR,QQ,AA,BB,X,WW)
        REAL X1(4,4),X(4,4,6),XO1(4,4)
        REAL AA(4,4,6),BB(4,4,6)
        REAL EE(4),WW(4)
        INTEGER RR,QQ
       DO 10 I=1,NN
          WW(I)=0.0
```

```
111             DO 20 K=1,MM
112                XO1(I,K)=0.0
113                X1(I,K) =0.0
114               DO 30 J=1,RR
115                 X1(I,K)=X1(I,K)+BB(I,K,J)*X(I,K,J)
116     30        CONTINUE
117                X1(I,K)=X1(I,K)+EE(K)
118                X(I,K,GG)=X(I,K,GG)+X1(I,K)*H-.5*H*X1(I,K)*H
119                IF (GG.EQ.1) GOTO 70
120               DO 40 J=1,GG-1
121                X(I,K,J-1)=X(I,K,J-1)+H*X(I,K,GG-J+1)-.5*H*X(I,K,
122         1                    GG-J+1)*H
123     40        CONTINUE
124                DO 45 J=1,RR-1
125                  XO1(I,K)=XO1(I,K)+AA(I,K,J)*X(I,K,J)
126     45         CONTINUE
127     70    WW(I)=WW(I)+XO1(I,K)+AA(I,K,RR)*X(I,K,RR)+AA(I,K,RR+1)*EE(K)
128     20      CONTINUE
129     10    CONTINUE
130           RETURN
131           END
```

****
****
****
********** No. of pages 3 2299-MNI Terminal: 72
****
****
****

APPENDIX (I)

The On-Line UNIVERAL Compensators

```
?LIST

 10  REM  ..THE UNIVERSAL MULTIVARIABLE CONTROLLER FOR DISCRETE
 20  REM  ......SYSTEMS ....
 21  REM   .... A PROGRAM FOR HIERARCHICAL CONTROL OF THE DOUBLE EFFECT,
 22  REM   ....EVAPORATOR.....
 23  REM   ...... HONEYWELL 316  &  M6800 ....
 25  REM          D(0) GOES TO STEAM  VALVE
 30  REM          D(1) GOES TO INFLOW  VALVE
 35  DIM A1(1,1,11),B1(11),B2(11),E(1,11),Y(1,11)
 40  PRINT " INPUT SET VALUES  D1 , D3 ";: INPUT D1,D3
 42  PRINT " INPUT SET VALUE OF VARIABLES";: INPUT S7,S8
 43  PRINT " INPUT CONTROLLED CHANNELS   ";: INPUT M1,M2
 50  DIM A(13),B(86),C(7),D(29)
 70  FOR I=0,13:A(I)=0: NEXT I
100  FOR I=0,86:B(I)=0: NEXT I
116  PRINT "INPUT AMPLIT. RANGE  Y1 ";: INPUT Y1
118  PRINT "INPUT MIN. AMPLIT.  Y0 ";: INPUT Y0
121  PRINT "INPUT TIME MAX X1 ";: INPUT X1
122  PRINT "INPUT TIME MIN. X0  ";: INPUT X0
123  PRINT "INPUT CHANN. NO. TO BE PLOTTED J";: INPUT J
124  PRINT "INPUT CHANN.NO TO BE PLOTTED  K ";: INPUT K
126  PRINT "INPUT CHANN. NO. TO BE PLOTTED L ";: INPUT L
127  PRINT " INPUT CHANN. FOR HISTO. PLOTTING C1&C2 ";: INPUT C1,C2
128  PRINT "INPUT  SCALING FACTOR FOR C1 & C2 ";: INPUT F1
130  PRINT "INPUT SCAN INTERVAL  ";: INPUT S3
140  PRINT "INPUT MULTIPLES OF MICRO SCAN ";: INPUT S2
144  NEXT J1
160  PRINT " NO. OF SCANS ..";: INPUT S1
162  GOSUB 600: CALL (5,1,C(0))
163  CALL (5,2,C(0)): GOSUB 700
164  N=1
165  B(C1)=D1:B(C2)=D3
180  A(1)=S2
190  A(2)=S1
195  A(12)=1
200  S=S2*S3
210  REM  DATA FOR CONTROLLER NUMERATOR MATRIX
212  FOR I=0 TO 1
214  FOR J1=0 TO 1
216  FOR K=0 TO 11: READ A1(I,J1,K1): NEXT K1
218  NEXT J1
220  NEXT I
222  REM  DATA FOR CONTROLLER COMMON DENOMINATOR POLY.
224  FOR K1=0 TO 11: READ B1(K1): NEXT K1
226  FOR I1=0 TO 1
228  FOR J1=0 TO 1
230  FOR K1=0 TO 11
232  A1(I1,J1,K1)=A1(I1,J1,K1)/B1(0)
234  NEXT K1
236  NEXT J1
238  NEXT I1
240  FOR J1=0 TO 11
```

```
280  FOR I=0 TO 9
290  B(I)=B(I)*128/1024
300  NEXT I
310  B(10)=B(10)*.245E-01/716.8+.105E-01
320  B(11)=B(11)*.5E-01/1024+.222E-01
330  B(12)=B(12)*31.5/1024-2.52
360  B(14)=B(14)*52.636/1024+1.4
370  B(15)=B(15)*19.127/1024
400  B(37)=B(55)/(3270*S3)+.394E-02
405  GOSUB 900
410  REM   TO ENABLE PLOTTING B(75),B(76) IS REMOVED
415  REM    ..D(0) IN M6800 CORRESPONDS TO B(57) IN H316 & SO ON
420  REM     M6800 CONTROL OUTPUTS
425  B(75)=B(57)/.32E05:B(76)=B(58)/.32E05
502  IF N>1 THEN 512
504  C(5)=0:C(6)=B(J):C(0)=2
505  REM  F1 TO SCALE THE HISTO. OF STEAM ,INFEED
506  CALL (5,3,C(0))
507  L1=B(C1)*F1:L3=B(C2)*F1
508  Y1=B(J):V1=B(K):W1=B(L)
510  GOTO 513
512  GOSUB 540
513  N=N+1
520  CALL (2)
522  CALL (5,6,C(0))
523  C(7)=24: CALL (5,5,C(0))
524  END
540  X=X+S
541  L2=B(C1)*F1:L4=B(C2)*F1
542  Y2=B(J):V2=B(K):W2=B(L)
544  C(5)=X:C(6)=Y2:C(0)=3
546  GOSUB 1000
548  C(5)=X-S:C(6)=V1:C(0)=2
550  GOSUB 1000
552  C(5)=X:C(6)=V2:C(0)=3
554  CALL (5,3,C(0))
556  C(5)=X-S:C(6)=W1:C(0)=2
558  GOSUB 1000
560  C(5)=X:C(6)=W2:C(0)=3
562  GOSUB 1000
564  C(5)=X-S:C(6)=L1:C(0)=2: GOSUB 1000
566  C(5)=X:C(0)=3: GOSUB 1000
567  C(6)=L2: GOSUB 1000
568  C(5)=X-S:C(6)=L3:C(0)=2: GOSUB 1000
570  C(5)=X:C(0)=3: GOSUB 1000
572  C(6)=L4: GOSUB 1000
574  C(5)=X:C(6)=Y2:C(0)=2: GOSUB 1000
575  Y1=Y2:V1=V2:W1=W2
576  L1=L2:L3=L4
580  RETURN
600  C(7)=29: CALL (5,5,C(0)):C(1)=X0:C(3)=Y0
620  C(2)=X1:C(4)=Y1
630  RETURN
700  C(5)=X0:C(6)=Y0:C(0)=2: GOSUB 1000
710  C(5)=X1:C(6)=Y0:C(0)=3: GOSUB 1000
715  Y1=Y1+Y0
720  C(5)=X1:C(6)=Y1: GOSUB 1000
730  C(5)=X0:C(6)=Y1: GOSUB 1000
740  C(5)=X0:C(6)=Y0: GOSUB 1000
750  RETURN
800  FOR J2=10 TO 0 STEP -1
```

```
816  RETURN
900  E(0,0)=B(M1)-S7
902  E(1,0)=B(M2)-S8
906  Y(0,0)=D1
908  Y(1,0)=D2
912  FOR I=0 TO 1
914  FOR K1=0 TO 11
916  FOR J1=0 TO 1
918  Y(I,0)=Y(I,0)+A1(I,J1,K1)*E(I,K1)-B2(K1)*Y(I,11-K1)
920  NEXT J1
922  NEXT K1
924  IF Y(0,0)<=0 THEN Y(0,0)=0
926  IF Y(1,0)<=0 THEN Y(1,0)=0
928  D(0)=Y(0,0)*.32767E05/.4E-01
930  D(1)=Y(1,0)*.32767E05/.8E-01
932  IF ABS(D(0))>.32767E05 THEN 944
934  IF ABS(D(1))>.32767E05 THEN 944
936  GOSUB 800
938  GOSUB 810
940  NEXT I
942  CALL (4,0,3,D(0))
944  RETURN
950  REM  DATA FOR MATRIX A1
952  DATA 0
954  DATA 0
956  DATA 0
958  DATA 0
960  REM  DATA FOR MATRIX B1
962  DATA 0
1000  CALL (5,3,C(0)): RETURN
```

On-line SD BASIC program: Double-Effect Evaporatc

```
      PROGRAM ORIGIN :4000
      DATA ORIGIN    :5000
      REM ... PROGRAM FOR M6800 CONTROL OF TEMPERATURES
      REM           IN A DOUBLE-EFFECT EVAPORATOR
      REM
      REM             2 P+I CONTROL LOOPS
      REM     RPV1,RPV2 ARE INITIAL VALVE SETTINGS
      REM
      DIM A(11),B(86),D(29),E1(2),E2(2),TIMER/06/,PFLAG
      DIM KPS,KPF,KIS,KIF,AVS,AVF,RPV1,RPV2,TIS,TIF,L/0/
      DIM S,SV1,SV2,I,F,L1,S1,EN,LN,U,N,M
      REM
      ON ERROR GOTO 9999
      CALL SUB0
      PRINT "INPUT CONTROLLER PARAMETERS !"
      PRINT \ INPUT "KPS & TIS FOR STEAM " KPS,TIS
      INPUT "KPF & TIF FOR FEED " KPF,TIF
      INPUT "STEAM & FEED VALVE AVERAGE POS. AVS,AVF " AVS,AVF
      INPUT "SET POINTS SV1 & SV2 " SV1,SV2
      INPUT "A(0),PRINT FLAG " S,PFLAG
      FOR I=0 TO 11 \ A(I)=0 \ NEXT I
      INPUT "F,L1,S1,A(1) " F,L1,S1,A(1)
      REM
      FOR I=0 TO 86 \ B(I)=0 \ NEXT I
      FOR I=0 TO 29 \ D(I)=0 \ NEXT I
      REM
      A(0)=S \  A(2)=S1 \ A(3)=F \ A(4)=L1
      A(5)=33
      E1(1)=0.0 \ E2(1)=0.0
      KIS=KPS/TIS \ KIF=KPF/TIF
      REM
   10 CALL SUB1(A(0),B(0))
      IF B(59)=0 THEN 15
      SV1=B(57)*.04/32767 \ SV2=B(58)*.08/32767
   15 REM
      L=L+1 \ IF L<=6 THEN GOSUB 1000
      IF A(1)=1 THEN 18
      ELSE GOSUB 2000
   18 IF L=12 THEN L=0
      E1(1)=E1(2) \ IF A(1)=9 THEN  E2(1)=E2(2)
      REM
      IF PFLAG=0 THEN 20
      PRINT USING 901,SV1,SV2,AVS,AVF
  901 FORMAT "-##.###  -##.###  -#######.##  -#######.##"
      B(35)=B(54)/50.0 \ REM -- CONVERT TO SECS.
      PRINT USING 902,B(57),B(58),B(37),B(55),B(56)
  902 FORMAT "-#####.# -#####.# -##.### -#### -###"
      PRINT D(0),D(1)
      PRINT
   20 M=2
```

```
      CALL SUB4(M,D(0))
      CALL SUB2
      GOTO 10
      REM
 1000 B(36)=B(10)*.0245/716.8 + .105E-01
      E1(2)=SV1-B(36) \ IF ABS( E1(2) ) < 3.E-04 THEN RETURN
      RPV1=KPS*(E1(2)-E1(1)) + KIS*S*E1(2)
      AVS=AVS+RPV1 \ IF AVS>32.E3 THEN AVS=32.E3
      IF AVS<0.0 THEN AVS=0.0
      D(0)=INT(AVS)
      U=INT(AVS) \ N=1
   30 CALL SUB3(N,U) \ RETURN
      REM
 2000 B(37)=( B(55)/(3270*S) + .00394 )
      E2(2)=SV2-B(37) \ IF ABS( E2(2) ) < 3.0E-03 THEN RETURN
      RPV2=KPF*(E2(2)-E2(1)) + KIF*S*E2(2)
      AVF=AVF-RPV2 \ IF AVF>32.E3 THEN AVF=32.E3
      IF AVF<0.0 THEN AVF=0.0
      D(1)=INT(AVF)
      U=INT(AVF) \ N=2
   40 CALL SUB3(N,U) \ RETURN
      REM
 9999 EN=ERR \ LN=ELN
      CALL SUB99(EN,LN)
      END
```

REFERENCES

Airy, G B — "On The Regulator of the Clockwork for Effecting Uniform Movement of Equatoreals", Mem. Roy. Astran. Soc., Vol. 11, pp. 249-267, 1840.

Anderson, B D O — "An Algebraic Solution to the Spectral Factorization Problem", IEEE Trans. Autom. Contr., Vol. AC-12, pp. 410-414, Aug. 1967.

Anderson, B D O
Moore, J B
Kailath, — Optimal Filtering, Information and System Science Series, Ed. T. Prentice-Hall.

Astrom, K J
Borisson, U
Ljung, S
Wittenmark, B — "Theory and Applications of Self-tunning Regulators", 6th IFAC World Congress, Boston, 1975, 1977.

Athans, M — "Trends in Modern System Theory", AICHE, Symp : series, 1976, 72, (159), 52.

Baker, G. A., — Pade' Approximants, Addison-Wesley, Mass. 1981 (Two Vol).

Barkelew, C H — "Modern Process Control - State of the Art in Petrolium Refining", AICHE, Symp : Series, 1976, 72 (159), 52.

Barrett, J F — "Construction of Wiener Filters Using the Return Difference Matrix", Int. J. of Control, Vol. 26, No. 5, pp. 797-803, 1977.

Barnett, S — "Some Topics in Algebraic Systems Theory : A Survey", Int. J. of Control 19, 669-688, 1974.

Bliss, H — "Transmission of Heat To Boiling Liquids", Trans. A.I.CH.E., Vol 36, pp 491-513, 1940

Bitmead, R R
Yung, S Y
Anderson, B D O
Kailath, T — "Greatest Common Divisors via Generalized Sylvester and Bezout Matrices", IEEE Trans. Auto. Contr. Vol. AC-23, pp. 1043-1046, 1978.

Bolton, A G — "Design and Implementation of Digital Filters for Microprocessors Implementation Using Fixed Point Arithmatic", Proc. IEEE 128, Part G, No. 5, pp. 245-250, 1981.

Bonilla, C F
Perry, C W — "Heat Transmission to Boiling Binary Liquid Mixtures". Tran. A.I.CH.E.,Vol 37, 1941, pp685

Bose, N K
Mitra, S K — "Generalized Inverse of Polynomial Matrices", IEEE Trans. of Automatic Contr., Vol. AC-23, pp. 491-493, 1978.

Bowen, B A
Buhr, R J A — The Logical Design of Multiple-Microprocessor Systems, Prentice-Hall, 1980.

Bristol, E H — "Industrial Needs and Requirements for Multivariable Control", Chemical Process Control, AICHE Symposium Series No. 159, Vol. 72, pp. 88-98, 1976.

Bryant, G F — "Control of a Steel Plant", Modern Approaches to Control Systems Design, IEE Control Eng. Series 9. Ed. N. Munro, Pub. Peter Peregrinus Ltd., 1979.

Bruun, N G
Kummel, M — "Multiloop, Feedforward, Model and Optimal Control of an Evaporator", Automatica, Vol. 15, p. 269, 1979.

Buchhalt, F.
Kummel, M — "A Multivariable Self-tuning Regulator in Control of a Double Effect Evaporator", Automatica, Vol. 17, p. 737, 1981.

Buckley, P S — "Distillation Column Design Using Multivariable Control", Instrum. Technol., 1978, 25, (9), pp. 115-122.

Chantmakher, F R — Theory of Matrices, Chelsea Publishing Co., New York, 1959, 2 Vols.

Church, D F — "Cement and Projected Pulp and Paper Industry Problems in Process Control and Process Modelling", AICHE, Symp : series, 1976, 72, (159), 52.

Coleby, J M — Application of Digital Filtering Methods to State and Parameter Estimation in Process Plant, Ph.D. Thesis, University of Aston in Birmingham, 1974.

Coulson, J
Richardson, J — Chemical Engineering, Vol. 2, 3rd edition (SI Units), Pergamon Prss, 1978.

Davis, M C — "Factoring the Spectral Matrix", IEEE Trans. Autom. Contr., Vol. AC-8, pp. 296-305, Oct. 1963.

Douglas, J M — Process Dynamics and Control, Prentice-Hall, 1978.

Eckert, E R G
Sparrow E M — "Heat Transfer- A Review of 1973 Literature", International J. of Heat

Goldstein R J — and Mass Transfer, 17, 1287-1317, 1974.

Ekstrom, M P
Woods, J W — "Two Dimensional Spectral Factorization with Applications in Recursive Digital Filtering", IEEE, Trans. Acoustic, Speach, and Signal Processing, ASSP, P-24, 1976, pp. 115-128.

Eykhoff, P — "Identification and system Parameter Estimation", Proc. 3rd IFAC Symposium, The Hague 1973.

Fisher D G
Seborg, D E — Multivaraible Computer Control : A Case Study, North Holland, Amesterdam 1976.

Fisher, D G
Kuon, J — "Comparison and Experimental Evaluation of Multivariable Frequency-Domain Design Techniques" IFAC, ed. D.P. Atherton, Proc. of 4th IFAC International Symposium, Frederiction, Canada, 4-8 July 1977.

Forney, G D — "Convolutional Codes I : Algebraic Structure", IEEE Trans. Infor. Theory, IT-16, pp. 720-738, 1970 and IT-17, p. 360, 1971.

Foss, A S — "Critique of Chemical Process Control Theory", IEEE Trans. Autom. Control, Vol. AC-18, pp. 646-652, 1973.

Fraade, D J
Gast, S — "A Survey of Computer Networks and Distribued Control", Digital Computer Applications to Process Control, IFAC 1977.

Gustavvsonan, J — "Survey of the Application of Identification in Chemical and

Physical Processes", Automatica, 11, 3, 1975.

Hamming, R W — Digital Filters, Prentice-Hall Englewood Cliffs, 1977.

Heins, A — "On the Solution of linear difference-differential equation", J. Maths Physics, 19, 2, 153, (1940).

Hewitt, G F — Annular Two-Phase Flow, Pergamon Press, 1970.

Horowitz, I M — "Synthesis of Feedback Systems", Academic Press, New York, 1963.

Hassan, S — Comparative Study of Classical and Modern Control of a Double Effect Evaporator, Dept. of Chem. Eng, M.sc. Thesis, University of Aston in Birmingham, 1983

Hsu, C H — "A Proof of the Stability of Multivariable Feedback Systems", Proc. IEE, 56, 2061-2062, 1968.

Hubbard, M B — "The Characterization of flow regimes for horizontal Two-Phase Flows", in Proc. of 1966 Heat Transfer of Fluid Mech. Inst. M.A. Saad and J.A. Miller Editors, Stanford University Press 1966.

Hung, N T
Anderson, B D O — "Triangularization Technique for the Design of Multivariable Control Systems", IEEE Trans. Auto. Contr., Vol. AC-24, pp. 455-460, Jan. 1979.

Jonhson, C R
Foss, A S — "Towards Development of a Practical Benchmark Example for Adaptive

Franklin, G F
Monopolic, R V
Stein, G
Control", AICHE, Symp : Series, 1976, 72, (159), 52.

Johnson, D E
I.S.A., J.7, 46, 1960.

Juten, A J
Tremblay, P
MacGregor, J F
Wright, J D
"Multivariable Computer Control of a Butane Hydrogenalysis Reactor", J. AICHE 23, p. 722, 1977.

Kahne, S
Lefkowriz, I
Rose, C W
"Automatic Control by Distributed Intelligence Control", Scientific American, June 1979, pp. 54-66.

Kailath, T
Linear Systems, Englewood Cliffs, NJ Prentice-Hall, 1980.

Kalman, R E
"Irreducible Realization and the degree of a Rational Matrix", J. SIAM, Contr., Vol. 13, pp. 520-544, 1965.

Kalman, R. E.
"When is a Linear Control System Optimal ?", J. Basic Eng. Ser. D., Vol. 86, pp. 51-60, 1964.

Kamen, E W
"An Operator Theory of Linear Functional Differential Equations", J. of Differential Equation, 77, 1978, pp. 247-297.

Kamen, E W
"Lectures on Algebraic System Theory : Linear Systems over Rings" NASA Contractor Reprt 3016, 1978.

Kamen, E W
"Linear Systems with Commensurate time delays : Stabilization and Stabilization Independent of Delay", IEEE Trans. Autmat. Contr. , Vol. AC-27, pp. 367-375, April 1982.

Kern, D Q — Process Heat Transfer, McGraw-Hill New York, 1960.

Kerjsa, E A
Goody Roontz, J
Stevens, G H — "Frequency Response of a Forced-Single-Tube Boiler", NASA, TN-D-4039, 1967.

Kershenbaum, L S
Fortescue, T R — "On-line Control of Chemical Processes Plants" Proc 7th World Congress IFAC, Helsinki, p. 263, Pergamon, 1978.

Koppel, L B — "Dynamic and Control of A Class of Nonlinear, Tubular, Paramatrically Forced Heat Exchanger and Chemical Reactors"Ind. Eng. Chem. Funfamentals, 5,3,403-413, 1966.

Korol'kov, B P — Heat Transfer - Soviet Research, 2,3,74, May 1970.

Kreith, F — Principles of Heat Transfer, International Text Book Co., London, 1972.

Kung, S
Kailath, T
Morf, M — "A Generalized Resultant Matrix", Proc. 1976, IEEE Conference.

Kuo, B C — Digital Control Systems, 1980, 2nd Edition. Prentice-Hall.

Kurichara, H — "Optimal Control of Fluid Catalytic Cracker", Report E SL-R-309, M.I.T., June 1967.

Kwakcrmoak, H
Sivan, R — Linear Optimal Control Systems, Wiley, New York, 1972.

La Salle, J
Lefschet, S — Stability by Liapunov's Direct Method with Applications, New York : Academic Press, 1961.

Leigh, J R — Applied Control Theory, IEE Series in Control, Peter-Peregrinus Ltd. London, 1982.

MacDuffee, C C — The Theory of Matrices, Springer-Verlag, Berlin, 1933, Reprinted by Chelsea Publishing Co., New York, 1950.

MacFarlane, A J — "A Survey of Some Recent Resuls in Linear Multivariable Feedback Theory", Automatica, 1972, 8, p. 455.

MacFaralane, A J
Belletrultti, jj — "The Characteristic Locus Method", Automatica, 1973, Vol. 9, p. 575.

Mann, W — "Digital Control of Rotary Dryer in the Sugar Industry", Automatica, Vol. 19, No. 2, March 1983, pp. 131.

Marshal, J E — Control of Time-Delay Systems, IEEE Series in Control, Peter Pergrinus Ltd., 1979.

Masani, P — "Recent Trends in Multivariable Prediction Theory", Multivariable Analysis, P.R. Krishnaiah ed., Academic Press, New York, 1966, pp. 351-382.

Mason, S J — Electronic Circuits, Signals, and Systems, Wiley, 1965.

Matyka, P R
Cadzow, J A — "The Factorization of Discrete Process Spectral Matrices", IEEE Trans. Autom. Contr., Vol. AC-12, pp. 698-706, Dec. 1967.

Maxwell, J C — "On Governor's", Proc. Roy. Soc., London, Vol. 16, pp. 270-283, 1868.

Mayne, D Q — "The Design of Linear Multivariable Systems", Automatica, Vol. 9, pp 201-207, 1973.

McAdams, W H — Heat Transmission, McGraw-Hill, New York 1954, 3rd ed.

McClellan, M T — "The Exact Solution of Systems of Linear Equations with Polynomial Coefficients", J. ACM 20, 4, Oct. 1973, pp. 563-588.

McMiller, E L
Larson, R E — "Annular Heat Transfer Coefficient for Turbulant Flow", Trans. AICHE, Vol. 76, 1954, pp. 147-153.

Momen A — On-line Computer Analysis of Plant Dynamics , University of Aston in Birmingham, PhD. Thesis, 1978.

Morran, P D — "Extension of the Inverse Nyquist Method", Electronic Letters, 6, 800-801, 1970.

Nawari, M O — "APA: Algebraic Polynomial Algorithms", Internal Repot, June 1983, Chemical Engineering Dept., Aston University, Birmingham.

Nawari, M O — "Wiener-Hopf Controller: Off-line Software", Internal Reprt, Dept. of Chem. Eng. University of Aston in Birmingham, 1983.

Nawari, M O — "Private Notes" No-15, Dept. of Chemical Eng. University of Aston, 1982.

Newell, R B
Fisher, D G — "Model Development, Reduction, and Experimental Evaluation for an Evaporator", Ind. Eng. Chem., Proc., Des. Dev., 11(2), p. 213, 1972.

Newton, G C — "Compensation of Feedback Control System Subject to Saturation", J. of the Franklin Institute, Vol. 254, 1952.

Newton, G C — "Design of Control Systems for Minimum Bandwidth", Trans. AIEE, Vol. 74, pt. II, 1955.

Newton, G C
Gould, L A
Kaiser. J F — Analytical Design of Linear Feedback Controls, New York, Wiley, 1967.

Olsson, G — "State of the art in Sewage Treatment Plant Control", AICHE Symp: series, 1976, 72, (159), 52.

Padmanabhan, L
Lapidus, L — "Control of Chemical Reactors", Chemical Reactor Theory : A Review, ed. by L. Lapidus and N.R. Amundson, Prentice-Hall, Englewood Cliff. N.J., 1977.

Patel, R V — "Computation of Matrix Fraction Descriptions of Linear-time-invariant Systems", IEEE Trans. Auto. Contr., Vol. AC-26, pp.148-161, 1981.

Payne, S G — The Application of On-line Estimation to a Double Effect Evaporator, PhD. Thesis, University of Aston, U.K., 1974.

Perry, R H
Chilton, C H — Chemical Engineer's Handbook, 4th ed., McGraw-Hill, New York , 1973.

Pontryagin, L S
Boltyanski, V C
Gamkrelidze, R V
Mishchenka, E F
The Mathematical Theory of Optimal Processes, (English Translation), New York, Wiley, 1962.

Popov, V M
"Hyperstability and Optimality of Automatic Systems with Several Control Functions", Revue Roumanie de Sciences Techniques - Electrotechnics on Energetics, Vol. 9, Bucharst (1964).

Richalet, J A
Rault, J L
Papon, J
"Model Algorithmic Control of Industial Processes", 5th IFAC/IFIP Conference on Digital Computer Application to Process Control, the Hague, Proc. 1977, North Holland, Amsterdam.

Riddle, A C
Anderson, B D O
"Spectral Factorization-Computational Aspects", IEEE Trans. Autom. Contr. Vol. AC-11, pp.764-765, Oct. 1966.

Rosenbrock, H H
State-Space and Multivariable Theory, London, Nelson, 1970.

Rosenbrock, H H
"The Theory of A System", Int. J. Contr., Vol. 18, pp. 297-299, 1973.

Rosenbrock, H H
Computer-Aided Control System Design, London : Academic, 1974.

Rosenbrock, H H
"Distinctive Problems of Process Control", CEP, 58, 43, (1962).

Rosenbrock, H H
"On the Design of Linear Multivariable Control Systems", Proc. Third IFAC Congress, London, 1966, Vol. 1, pp. 1-16.

Rosenbrock, H H — "Designing of Multivariable Control Systems Using the Inverse Nyguist Array", Proc. Inst. Elct. Eng., Vol. 116, pp. 1929-1936, 1969.

Rosenbrock, H H
McMarran, P — "Good, Bad, or Optimal ?", IEEE, Trans., Autom. Contr. AC-16 (6), p. 552, 1971.

Sain, M K — Independent Algebraic System Theory, Academic Press, New York, 1981.

Sain, M K — "A Free-Modular Algorithm for Minimal Design of Linear Multivariable Systems", in Proc. sixth IFAC Congress, Part 1B, pp. 9.1.1-9.17., 1975.

Sain, M K — "The Growing Algebraic Presence in Systems Engineering : An Introduction", Proc. IEEE, Vol. 64, No. 1, pp. 96-111, 1976.

Sasaki, T — "Efficient Gaussian Elimination Method for Symbolic Determinants and Linear Systems", ACM Trans. Maths., Software (USA), Vol. 8, No. 3, pp. 277-289, Sept. 1982.

Sasaki, T — "Efficient Gaussian Elimination Method for Symbolic Determinants and Linear Systems", ACM Trans. Maths. Software (USA), Vol. 8, No. 3, Sept. 1982, pp. 277-289.

Schoeffler, J D
Rose, C W — "Distributed Computer Intelligence For Data Aquisition and Control", IEEE Trans. Nuc. Science, Vol. NS-23, No. 1, Feb. 1976.

Schuldt, S B
Smith, F B — "An Application of Quadratic Performance Synthesis Techniques to a

Fluid Catalytic Cracker", Proc. Joint Automati Control Conference 1971, p. 270.

Schuman, R — "Digital Parameter - Adaptive Control of an Air-Conditioning Plant", Automatic, Vol. 18, No. 5, Sept, 1982.

Schuman, R et al — "Towards Applicability of Parameter-Adaptive Control Algorithms", 8th IFAC Congress Kyato, 1981.

Shafii, A — "The Use of Microprocessors in the Control of Chemical Plants with Special Reference to the use of Distributed Processors", Dept. pf Chemical Eng., University of Aston in Birmingham, 1983.

Shimnar, R — "Process Control Research : An Evaluation of Present Status and Research Needs", AICHE, Symp : series, 1976, 72, (159), 52.

Silberstein J — "On a histero-Differential Equation arising in a Probability Problem", Philosophical Magazine, Vol. 29, No. 192, 75, (1940).

Stout, T M — "Economic Justification of Computer Control Systems", Automatica, Vol. 9, pp. 9-19, 1973.

Strintzis, M G — "A Solution to the Matrix Factorization Problem", IEEE, Trans. Infor. Theory, Vol. IT-18, pp. 225-232, March 1972.

Sydenham, P H — Hadbook of Measurement Science, Vol. 1, Theoretical Fundamentals, Edited

by Peter H, Wiley Interscinece Publication, p. 425, 1982.

Taitil, Y
Dukler, A E — "A Model for Predicting Flow Regime Transition in Horizontal and Near Horizontal Gas Liquid Flow", AICHE J., 22, 47, 1976.

Trevathan, V L — "Process Controls in the Chemical Industry", AICHE, Symp : Series, 1976, 72, (159), 52.

Tuel, W J — IBM Journal, March 1968.

Webb, R N — Development of an Adaptive Kalman Filter for Estimation in Chemical Plant, PhD. Thesis, University of Aston in Birmingham, 1977.

Weston, J E — "Extension of Analytical Design Techniques to Multivariable Feedback Control Systems", IEEE Trans. Automat. Contr., Vol. AC-17, pp. 613-620, Oct. 1972.

Weitzman, C — Distributed Micro/Mini Computer Systems, Prentice-Hall, 1980.

Wiener, N — Extrapolation, Interpolation, and Smoothing of Stationary Time Series, Technology Press, Cambridge, 1949.

Wiener, N — "Cybernetics : Or control and Communication in the Animal and the Machine", The M.I.T. Press, Cambridge, Mass, 2nd edition, 1961.

Wilson, G T — "The Factorization of Matricial Spectral Densities", SIAM J., Appl. Math., Vol. 23, No. 4, pp. 420-426, Dec. 1972.

Wolvich, W A — Linear Multivariable Systems, Springer-Verlag, New York, Inc., 1974.

Wong, E
Thomas, J B — "On Multidimensional Prediction and Filtering Problems and Factorization of Spectral Matrices", J. Franklin Inst., Vol. 272, pp. 87-99, 1961.

Wonham, W M — Linear Multivariable Control : A Geometric Approach, Springer-Verlag, Berlin, 1974.

Youla, D C — "On the Factorization of Rational Matrices", IRE Trans. Information Theory, IT-7, pp. 172-189, 1961.

Youla, D C — "Modern Wiener-Hopf Design of Optimal Controllers - Part I : The Single-input case" IEEE Trans. Automat. Contr., Vol. AC-21, No. 1, pp. 3-13, Feb. 1976.

Youla, D C — "Modern Wiener-Hopf Design of Optimal Controllers - Part II : The Multivariable Case" IEEE Trans. Automat. Contr., Vol. AC-21, pp. 319-338, June 1976.

Youla, D C — "Course Notes for EL610", Dept. of Electrical Engineering and Computer Science, Polytechnic Institute of New York, Route 110, Farmingdale, New York 11735, July,1, 1981.

Zadeh, L A — Linear Systems Theory : The State-Space approach", New York : McGraw-Hill, 1963.

BIBLIOGRAPHY OF RELEVANT INTEREST

1. Alos, A J — "Stabilization of a Class of Plants with Possible Loss of Outputs or Actuator Facilities", IEEE Trans. Autom. Contr. Vol. AC-28, No. 2, Feb. 1983, pp. 231-233.

2. Kucera, V — "Disturbance Rejection : A Polynomial Approach", IEEE Trans. on Autom. Contr. Vol. AC-28, No. 4, April 1983, pp. 508-511.

3 Zames, G, Francis, B A — "Feedback, Minimax Sensitivity, and Optimal Robustness", IEEE Trans. on Autom. Contr. Vol. AC-28, No. 5, May 1983.

4. Zames, G Bensoussan D — "Multivariable Feedback, Sensitivity, and Decentralized Control", IEEE Trans. on Autom. Contr. Vol. AC-28, No. 11, Nov. 1983.

5. Vidyasagar, M Viswanadham N — "Algebraic Design Techniques for Reliable Stabilization", IEEE Trans. on Autom. Contr. Vol. AC-27, No. 5, October 182, pp. 1085-1095.

6. Saeks, R Murray J — "Fractional Representation, Algebraic Geometry, and the Simultaneous Stabilization Problems", IEEE Trans. on Autom. Contr. Vol. AC-27, No. 4, August 1982, pp.895-903.

7. Vidyasagar, M — "Algebraic and Topological Aspects of the Regulator Problem for Lumped Linear Systems", Automatica, Vol. 19,1, Jan. 1983.

8. Khargonekar, P P Sontag E — "On the Relation Between Stable Matrix Fraction Factorization and Regulable Realizations of Linear Systems Over Rings", IEEE Trans. Autom. Contr. Vol. AC-27, No. 3, June 1982, pp. 627-638.

9 Ray, W H — Advanced Process Control, McGraw-Hill, 1980.

10 Friedly, J C — Dynamic Behaviour of Process, Prentice-Hall, 1972.

11 Azbel, D — Two Phase Flows in Chemical Engineering, Cambridge University Press, Cambridge, 1981.

12 Burghes, D N
Borrie, M S — Modelling With Differential Equation, Ellins Hrwood Ltd., Chicheston, 1981.

13 Doebelin, O E — Systems Modelling and Response : Theoretical and Experimental Approach, John Wilely, New York, 1980.

14 Munro, N — Modern Approaches to Control Systems Design, IEE Contr. Eng. Series 9, 1979.

15 Patel, R V
Munro, N — Multivariable Systems Theory and Design, Vol. 4, Pregamon Press, Oxford, 1982.

16 Dornhoff, L L
Hohn, F — Applied Modern Algebra, Macmillen Co., New York, 1978.

17 Sain, M — Introduction to Algebraic System Theory, Academic Press, 1981, New York.

18 Noble, B — Applied Linear Algebra , Prentice-Hall, Englewood, Cliffs, N.J., 1969.