SKB

TECHNICAL REPORT 88-10

GWHRT – A flow model for coupled groundwater and heat flow

Version 1.0

Roger Thunvik[1], and Carol Braester[2]

[1] Royal Institute of Technology, Stockholm, Sweden
[2] Israel Institute of Technology, Haifa, Israel

April 1988

SVENSK KÄRNBRÄNSLEHANTERING AB
SWEDISH NUCLEAR FUEL AND WASTE MANAGEMENT CO

BOX 5864 S-102 48 STOCKHOLM
TEL 08-665 28 00 TELEX 13108-SKB

GWHRT - A FLOW MODEL FOR COUPLED GROUNDWATER AND HEAT FLOW

Version 1.0

Roger Thunvik[1] and Carol Braester[2]

1 Royal Institute of Technology, Stockholm, Sweden

2 Israel Institute of Technology, Haifa, Israel

April 1988

GWHRT - A FLOW MODEL FOR COUPLED GROUNDWATER AND HEAT FLOW

ROGER THUNVIK
Royal Institute of Technology
Stockholm, Sweden

CAROL BRAESTER
Israel Institute of technology
Haifa, Israel

1988-04-01

ABSTRACT

A mathematical model describing the hydrothermal conditions around a hard rock repository for disposal of nuclear fuel waste is presented. The model was developed to study the effect of heat released from a radioactive waste repository on the flow times from the repository to the ground surface. The model consists of a set of coupled non-linear partial differential equations for heat and ground water flow. In addition there are two equations of state relating fluid density and viscosity to pressure and temperature.

The system of equations is solved numerically using the finite element method in one, two or three dimensions. The model has been successively developed and used as a research tool to include unsaturated flow, gas migration, discrete elements, stochastic analysis, etc. The model version described here is the basic one. The fractured rock is treated either as two overlapping continua in which the one represents the network of fractures and the other the solid blocks or as a single equivalent medium. The first approach assumes quasi-steady state heat transfer from the rock to the fluid, allowing a linear heat transfer function to be used. The second approach assumes instantaneous equilibrium between the fluid and the rock.

TABLE OF CONTENTS

NOMENCLATURE

Notation		Dimension	SI unit
a	equivalent block radius	L	m
c	compressibility	$M^{-1}Lt^{2}$	Pa^{-1}
C	specific heat capacity	$L^{2}t^{-2}K^{-1}$	$J/(kgK)$
D	thermal diffusivity	$L^{2}t^{-1}$	m^{2}/s
g	acceleration of gravity	Lt^{-2}	m/s^{2}
h	heat transfer coefficient	$ML^{-1}t^{-3}K^{-1}$	$W/(m^{3}K)$
k_{ij}	permeability tensor	L^{2}	m^{2}
n_i	normal vector	-	-
p	pressure	$ML^{-1}t^{-2}$	Pa
q	specific discharge	Lt^{-1}	m/s
q_r	specific flux normal to an exterior boundary	Lt^{-1}	m/s
Q^{T}	total heat flux	$ML^{2}t^{-3}$	W
r	radial coordinate	L	m
S	water saturation	-	-
S'	$\frac{dS}{dp}$	$M^{-1}Lt^{2}$	Pa^{-1}
t	time	t	s
T	temperature	K	K
v	fluid velocity	Lt^{-1}	m/s
w	heat transfer function	$ML^{-1}t^{-3}$	W/m^{3}
z	z-coordinate	L	m
α_1, α_2	coefficients in interpolation formula for radioactive decay	-	-
β	coefficient of thermal volume expansion of the fluid	K^{-1}	$1/K$

γ	coefficient to determine exponential decrease in permeability with depth	-	-
δ	coefficient for the approximation of the heat transfer at the surface of a sphere or a slab	-	-
Δt	time step	t	s
λ	thermal conductivity	$MLt^{-3}K^{-1}$	$W/(mK)$
θ	angle in cylindrical coordinate system	-	radians
μ	dynamic viscosity	$ML^{-1}t^{-1}$	Pas
ρ	density	ML^{-3}	kg/m^3
ϕ	porosity	-	-
ψ	basis function		
$<a,b>$	inner product		

$$<a,b> \equiv \int_R a \cdot b \, dR$$

{ }	column vector
[ ]	matrix

Superscripts

f	fluid
r	rock
o	reference value
*	equivalent medium
(n)	time level

Subscripts

i,j	indices used for Cartesian tensor notion, repeated indices indicate summation over these indices (i,j = 1,2,3)
	$p_{,t}$ partial time derivative
	$p_{,i}$ gradient of p
I,J,K,L	node indices, repeated indices indicate summation over these indices (I,J,K,L = 1,2,...N, where N the number of nodal points)

Notation used for stress calculation

a_i = acceleration component

e_{ij} = strain tensor

e = e_{kk}

E = Young's modulus (E > 0)

G = $\frac{E}{2(1+\nu)}$

K = $\frac{E}{3(1-2\nu)}$

s = σ_{kk}

T = temperature

x_i = Cartesian coordinate

X_i = body force vector

u_i = displacement vector

β = coefficient of thermal volume expansion

α = $\frac{E\beta}{1-2\nu}$

δ_{ij} = Kronecker delta

ν = Poissons's ratio $(-1 < \nu < 1/2)$

ρ^r = density of rock

λ = $\frac{E\nu}{(1+\nu)(1-2\nu)}$

σ_{ij} = stress tensor

Nomenclature used for gas migration calculation

symbol	parameter	dimension
B	cavern width	L
b_f	fracture width	L
c_f	compressibility of the fluid	$M^{-1}Lt^2$
c_r	compressibility of the rock matrix	$M^{-1}Lt^2$
g	acceleration of gravity	Lt^{-2}
k	permeability	L^2
l	fracture spacing	L
M	molecular weight of the gas	$Mmole^{-1}$
n_i	fracture frequency of fracture category i	-
p	pressure	$ML^{-1}t^{-2}$
p_c	capillary pressure	$ML^{-1}t^{-2}$
q_n	normal flux	Lt^{-1}
Q	mass rate of gas production	$ML^{-3}t^{-1}$
R	gas constant	$ML^2t^{-2}T^{-1}mole^{-1}$
t	time	t
S_{gr}	residual gas saturation	-
S_{wir}	irreducible water saturation	-
T	temperature	T
u	volumetric rate of flow per unit area	Lt^{-1}
v	velocity	Lt^{-1}
Z	correction factor for real gases	-
Greek		
ρ	density	ML^{-3}
μ	dynamic viscosity	$ML^{-1}t^{-1}$

θ	contact angle	*degrees*
o	porosity	-
Φ	potential	-
ψ	basis function	-
ξ	interface elevation above datum (z = 0)	

Subscripts

i	fracture category
w	water
g	gas
(ι)	iteration level
I,J,K	node number

1. MATHEMATICAL FORMULATION OF THE FLOW MODEL

1.1 FLUID FLOW

1.1.1 Basic equations for the saturated-unsaturated fluid flow

The present flow model considers the following equation of continuity for the fluid

$$(\phi S \rho^f)_{,t} + (\rho^f q_i)_{,i} = 0 \qquad (1.1\text{-}1)$$

where

$$q_i = -\frac{k_{ij}}{\mu}(p_{,j} - \rho^f g_j) \qquad (1.1\text{-}2)$$

which is analogous to Darcy's law.

Expanding the time derivative in the continuity equation for the fluid, one obtains

$$(\phi S \rho^f)_{,t} = \phi S\left(\frac{\partial \rho^f}{\partial p} p_{,t} + \frac{\partial \rho^f}{\partial T} T^f_{,t}\right)$$

$$+\phi \rho^f \frac{\partial S}{\partial p} p_{,t} + S \rho^f \frac{\partial \phi}{\partial p} p_{,t}$$

Fluid compressibility, rock compressibility and the coefficient of thermal volume expansion of the fluid are defined as

$$c^f = \frac{1}{\rho^f}\frac{\partial \rho^f}{\partial p} \qquad (1.1\text{-}3a)$$

$$c^r = \frac{1}{\phi}\frac{\partial \phi}{\partial p} \qquad (1.1\text{-}3b)$$

$$\beta = -\frac{1}{\rho^f}\frac{\partial \rho^f}{\partial T^f} \qquad (1.1\text{-}3c)$$

Making use of the previous definitions, one obtains

$$(\phi S \rho^f)_{,t} = \phi S \rho^f c^f p_{,t} - \phi S \rho^f \beta T^f_{,t}$$

$$+\phi \rho^f S' p_{,t} + S \rho^f \phi c^r p_{,t}$$

or

$$(\phi S \rho^f)_{,t} = \phi \rho^f \left(S(c^f + c^r) + S'\right) p_{,t} - \phi S \rho^f \beta T^f_{,t}$$

The equation for the fluid flow may now be written as

$$\phi\rho^f\left(S\left(c^f+c^r\right)+S'\right)p_{,t}-\phi S\rho^f\beta T^f_{,t} \qquad (1.1\text{-}4)$$
$$-\left(\rho^f\frac{k_{ij}}{\mu}\left(p_{,j}-\rho^f g_j\right)_{,i}\right)_{,i}=0 \quad ,$$

1.1.2 Finite element formulation of the flow equations

The flow equation is solved numerically, using the Galerkin finite element method. The flow domain is discretized by a mesh of line elements in one dimension, quadrilateral elements in two dimensions and hexahedral elements in three dimensions. The flow equation considered is:

$$\phi\rho^f\left(S\left(c^f+c^r\right)+S'\right)p_{,t}-\phi S\rho^f\beta T^f_{,t} \qquad (1.1\text{-}4)$$
$$-\left(\rho^f\frac{k_{ij}}{\mu}\left(p_{,j}-\rho^f g_j\right)_{,i}\right)_{,i}=0$$

The Galerkin formulation of the flow equation is

$$\left\langle\phi\rho^f\left(S\left(c^f+c^r\right)+S'\right)p_{,t}-\phi S\rho^f\beta T^f_{,t} \quad , \quad \psi_I\right\rangle \qquad (1.1\text{-}5)$$
$$-\left\langle\left(\rho^f\frac{k_{ij}}{\mu}\left(p_{,j}-\rho^f g_j\right)\right)_{,i} \quad , \quad \psi_I\right\rangle=0$$

where ψ_I represents the basis functions, chosen such that the essential boundary conditions will be satisfied.

Applying Green's theorem, we obtain

$$\left\langle\phi\rho^f\left(S\left(c^f+c^r\right)+S'\right)p_{,t}-\phi S\rho^f\beta T^f_{,t} \quad , \quad \psi_I\right\rangle \qquad (1.1\text{-}6)$$
$$+\left\langle\left(\rho^f\frac{k_{ij}}{\mu}\left(p_{,j}-\rho^f g_j\right)\right) \quad , \quad \psi_{I,i}\right\rangle$$
$$-\int_s\rho^f\frac{k_{ij}}{\mu}\left(p_{,j}-\rho^f g_j\right)\psi_I\,dS_i=0$$

Assume the trial functions to be of the form

$$p\approx p_J\psi_J \qquad (1.1\text{-}7a)$$

$$T\approx T_J\psi_J \qquad (1.1\text{-}7b)$$

Substitution of the trial functions into the previous equation yields

$$\left\langle\phi\rho^f\left(S\left(c^f+c^r\right)+S'\right)p_{J,t}\psi_J-\phi S\rho^f\beta T^f_{J,t}\psi_J \quad , \quad \psi_I\right\rangle \qquad (1.1\text{-}8)$$
$$+\left\langle\left(\rho^f\frac{k_{ij}}{\mu}\left(p_J\psi_{J,j}-\rho^f g_j\right)\right) \quad , \quad \psi_{I,i}\right\rangle$$

$$-\int_S \rho^f \frac{k_{ij}}{\mu}\left(p_{J,j}\psi_J - \rho^f g_j\right)\psi_I dS_i = 0$$

The previous equation may be written in matrix form as

$$A_{IJ}^{11} P_{J,t} + A_{IJ}^{12} T_{J,t}^f + A_{IJ}^{13} P_J - C_I^1 = 0 \qquad (1.1\text{-}9)$$

where

$$A_{IJ}^{11} = \int\int\int_R \phi\rho^f\left(S\left(c^f + c^r\right) + S'\right)\psi_J\psi_I dR$$

$$A_{IJ}^{12} = \int\int\int_R \phi S\rho^f\beta\psi_J\psi_I dR$$

$$A_{IJ}^{13} = \int\int\int_R \rho^f \frac{k_{ij}}{\mu}\psi_{J,j}\psi_{I,i} dR$$

$$C_I^1 = \int\int_S q_r\psi_I dS + \int\int\int_R \rho^f \frac{k_{ij}}{\mu}\rho^f g_j\psi_{I,i} dR$$

I,J = 1,2,...N

1.1.3 <u>Finite element formulation for axi-symmetric flow</u>

In cylindrical coordinates (r,θ,z) the inner product may be defined as

$$\langle\phi,\psi\rangle = \int_{z_1}^{z_2}\int_{r_1}^{r_2}\int_{\theta_1}^{\theta_2}\phi\psi r\, d\theta\, dr\, dz \qquad (1.1\text{-}10)$$

When the flow is assumed to be axi-symmetric, it is convenient to first perform the integration over θ, i.e. for $\theta_1 = 0$ and $\theta_2 = 2\pi$

$$\langle\phi,\psi\rangle = 2\pi\int_{z_1}^{z_2}\int_{r_1}^{r_2}\phi\psi r\, dr\, dz \qquad (1.1\text{-}11)$$

On acccount of (1.1-11) equation (1.1-8) may be written as

$$2\pi\int_{z_1}^{z_2}\int_{r_1}^{r_2}\phi\rho^f\left(S\left(c^f + c^r\right) + S'\right)p_{,t}\psi_I r\, dr\, dz +$$

$$-2\pi\int_{z_1}^{z_2}\int_{r_1}^{r_2}\phi S\rho^f\beta T_{,t}^f\psi_I r\, dr\, dz$$

$$+2\pi\int_{z_1}^{z_2}\int_{r_1}^{r_2}\rho^f\frac{k_{ij}}{\mu}\left(p_{,j} - \rho^f g_j\right)\psi_{I,i} r\, dr\, dz$$

$$-2\pi\int_{z_1}^{z_2}\rho^f\frac{k_{ij}}{\mu}\left(p_{,j} - \rho^f g_j\right)\psi_I r n_i dz = 0 \qquad (1.1\text{-}12)$$

Substitution of the trial functions into the previous equation, we obtain

$$2\pi \int\int \phi \rho^f \left(S\left(c^f + c^r\right) + S^s \right) p_{J,t} \psi_J \psi_I \, r \, dr \, dz -$$

$$-2\pi \int_{z_1}^{z_2} \int_{r_1}^{r_2} \phi S \rho^f \beta T^f_{J,t} \psi_J \psi_I \, r \, dr \, dz +$$

$$+2\pi \int_{z_1}^{z_2} \int_{r_1}^{r_2} \rho^f \frac{k_{ij}}{\mu} \left(p_J \psi_{J,j} - \rho^f g_j \right) \psi_{I,i} \, r \, dr \, dz -$$

$$2\pi \int_{z_1}^{z_2} q_r \psi_I \, r \, dz = 0 \qquad \text{(1.1-13)}$$

$$I, J = 1, 2, \ldots N$$

where q_r is the normal flux being prescribed along the exterior boundary, here assumed to be vertical, of the flow domain. The integration limits are the same as in the preceding equations and will therefore be omitted from now on.

Equation (1.1-13) may be written in a more compact form as

$$A^{11}_{IJ} P_{J,t} + A^{12}_{IJ} T^f_{J,t} + A^{13}_{IJ} P_J - C^1_I = 0 \qquad \text{(1.1-14)}$$

where

$$A^{11}_{IJ} = 2\pi \int_{z_1}^{z_2} \int_{r_1}^{r_2} \phi \rho^f \left(S\left(c^f + c^r\right) + S^s \right) \psi_J \psi_I \, r \, dr \, dz$$

$$A^{12}_{IJ} = 2\pi \int_{z_1}^{z_2} \int_{r_1}^{r_2} \phi S \rho^f \beta \psi_J \psi_I \, r \, dr \, dz$$

$$A^{13}_{IJ} = 2\pi \int_{z_1}^{z_2} \int_{r_1}^{r_2} \rho^f \frac{k_{ij}}{\mu} \psi_{J,j} \psi_{I,i} \, r \, dr \, dz$$

$$C^1_I = 2\pi \int_{z_1}^{z_2} q_r \psi_I \, r \, dz + \int_{z_1}^{z_2} \int_{r_1}^{r_2} \rho^f \frac{k_{ij}}{\mu} \rho^f g_j \psi_{I,i} \, r \, dr \, dz$$

$$I, J = 1, 2, \ldots N$$

1.1.4 Boundary conditions for the fluid flow equation

The previous set of governing equations must be supplemented with the appropriate boundary and initial conditions for the problem. These are of the following types:

(i) Prescribed pressure at a fixed boundary.

(ii) Prescribed groundwater flux normal to a fixed boundary (e.g. zero groundwater flux due to impervious barriers).

(iii) Prescribed pressure at a moving boundary, whose position is to be determined as part of the solution.

The first type of boundary condition is relevant to the top boundary of the flow domain when the aquifer is to be considered unconfined. The top boundary may either be a fixed or a moving boundary. If it is a moving boundary then its position is usually unknown and therefore a part of the

solution. If the top of the aquifer is confined by an impervious layer then the second type of boundary condition should be considered with zero normal flux prescribed at the boundary.

The lateral extent of the flow domain should in general be considered infinite. This is also true of the bottom of the flow domain. However, the calculations must be limited to a finite part of the flow domain. In certain cases a finite part may be localized using symmetry reasoning by considering an infinite series of identical repositories. In general, imaginary boundaries must be imposed upon the flow domain and they must be chosen in such a way that their effects on the solution values become negligible.

The third type of boundary condition is treated by solving the flow problem as a saturated-unsaturated flow problem.This requires that characteristic curves for capillary pressure versus saturation and relative permeability versus saturation be known.

1.2 HEAT FLOW

1.2.1 Basic equations of the heat flow in saturated fluid flow

The equation of the conservation of thermal energy for the fluid is written as

$$\left(\phi\rho^f C^f T^f\right)_{,t} - \left(\phi\lambda^f T^f_{,i}\right)_{,i} + \left(\rho^f C^f q_i T^f\right)_{,i} + w^f = 0 \qquad (1.2\text{-}1)$$

and the equation of the conservation of thermal energy for the rock is written as

$$\left((1-\phi)\rho^r C^r T^r\right)_{,t} - \left((1-\phi)\lambda^r T^r_{,i}\right)_{,i} + w^r = 0 \qquad (1.2\text{-}2)$$

where w^f and w^r represent heat sources and also the exchange of heat between the fluid and rock media.

Expanding the time derivative and the convective terms and substituting the equation of the conservation of mass for the fluid (Eq. 1.1-1 with S=1) into the equation of the thermal energy balance of the fluid, balance of the fluid, one obtains

$$\phi\rho^f C^f T^f_{,t} - \left(\phi\lambda^f T^f_{,i}\right)_{,i} + \rho^f C^f q_i T^f_{,i} + w^f = 0 \qquad (1.2\text{-}3)$$

Similarly and by making use of the definition of rock compressibility, the equation of the thermal balance for the rock may be written as

$$(1-\phi)\rho^r C^r T^r_{,t} - \phi\rho^r C^r T^r c^r p_{,t} - \left((1-\phi)\lambda^r T^r_{,i}\right)_{,i} + w^r = 0 \qquad (1.2\text{-}4)$$

The heat flow equations are supplemented by the equations state, relating the fluid density and viscosity to pressure and temperature.

$$\rho^f = \rho^f\left(p, T^f\right) \qquad (1.2\text{-}5)$$

$$\mu = \mu\left(p, T^f\right) \qquad (1.2\text{-}6)$$

In the present computer model it is assumed that viscosity is a function of temperature only.

Currently, quasi-steady heat exchange is considered. This means that the heat transfer function in equations (1.2-3) and (1.2-4) may be replaced by a linear heat transfer function of the following form

$$w = h\left(T^f - T^r\right) \tag{1.2-7}$$

where h is the heat transfer coefficient, being a function of the characteristic block size, thermal diffusivity of the rock, etc.(see Thunvik and Braester, 1980). Substituting the previous relationship into the equations for the heat flow, we obtain

$$\phi\rho^f C^f T^f_{,t} - \left(\phi\lambda^f T^f_{,i}\right)_{,i} + \rho^f C^f q_i T^f_{,i} + h\left(T^f - T^r\right) = 0 \tag{1.2-8}$$

$$\begin{aligned}&(1-\phi)\rho^r C^r T^r_{,t} - \phi\rho^r C^r T^r c^r p_{,t} \\ &-\left(\left(1-\phi\lambda^r\right)T^r_{,i}\right)_{,i} - h\left(T^f - T^r\right) = 0\end{aligned} \tag{1.2-9}$$

If it may be assumed that the fluid and rock media will attain thermal equilibrium instantaneously, then the two heat balance equations may be added to each other, and the heat flow may be described by the following equation

$$(\rho C)^* T_{,t} - \left(\lambda^* T_{,i}\right)_{,i} + \rho^f C^f q_i T^f_{,i} = 0 \tag{1.2-10}$$

where

$$(\rho C)^* = \phi\rho^f C^f + (1-\phi)\rho^r C^r \tag{1.2-11}$$

$$\lambda^* = \phi\lambda^f + (1-\phi)\lambda^r \tag{1.2-12}$$

1.2.2 Numerical formulation of the equations for the heat flow

Applying Galerkin's method to the heat flow equations, we obtain the fluid and rock temperature equations

$$\begin{aligned}&\left\langle \phi\rho^f C^f T^f_{,t} \;,\; \psi_I \right\rangle - \left\langle \left(\phi\lambda^f T^f_{,i}\right)_{,i} \;,\; \psi_I \right\rangle + \\ &\left\langle \rho^f C^f q_i T^f_{,i} \;,\; \psi_I \right\rangle + \left\langle w \;,\; \psi_I \right\rangle = 0\end{aligned} \tag{1.2-13}$$

$$\begin{aligned}&\left\langle (1-\phi)\rho^r C^r T^r_{,t} \;,\; \psi_I \right\rangle - \left\langle \rho^r C^r T^r \phi c^r p_{,t} \;,\; \psi_I \right\rangle - \\ &\left\langle \left((1-\phi)\lambda^r T^r_{,i}\right)_{,i} \;,\; \psi_I \right\rangle - \left\langle w \;,\; \psi_I \right\rangle = 0\end{aligned} \tag{1.2-14}$$

Applying Green's theorem to all second derivative terms, we obtain

$$\langle \phi \rho^f C^f T^f_{,t} \ , \ \psi_I \rangle + \langle \phi \lambda^f T^f_{,t} \ , \ \psi_{I,i} \rangle + \langle \rho^f C^f q_i T^f_{,i} \ , \ \psi_I \rangle + \langle w \ , \ \psi_I \rangle - \int_S \phi \lambda^f T^f_{,i} \psi_I dS_i = 0 \tag{1.2-15}$$

$$\langle (1-\phi) \rho^r T^r_{,t} \ , \ \psi_I \rangle - \langle \rho^r C^r T^r c^r p_{,t} \ , \ \psi_I \rangle + \langle (1-\phi) \lambda^r T^r_{,i} \ , \ \psi_{I,i} \rangle - \langle w \ , \ \psi_I \rangle - \int_S (1-\phi) \lambda^r T^r_{,i} \psi_I dS_i = 0 \tag{1.2-16}$$

Substitution of the trial functions yields

$$\langle \phi \rho^f C^f T^f_{J,t} \psi_J \ , \ \psi_I \rangle + \langle \phi \lambda^f T^f_J \psi_{J,i} \ , \ \psi_{I,i} \rangle + \langle \rho^f C^f q_i T^f_J \psi_{J,i} \ , \ \psi_I \rangle + \langle w \ , \ \psi_I \rangle - \int_S \phi \lambda^f T^f_{,i} \psi_I dS_i = 0 \tag{1.2-17}$$

$$\langle (1-\phi) \rho^r T^r_{J,t} \psi_J \ , \ \psi_I \rangle - \langle \rho^r C^r T^r_J \phi_J c^r p_{,t} \ , \ \psi_I \rangle + \langle (1-\phi) \lambda^r \psi_J T^r_{J,i} \ , \ \psi_{I,i} \rangle - \langle w \ , \ \psi_I \rangle - \int_S (1-\phi) \lambda^r T^r_{,i} \psi_I dS_i = 0 \tag{1.2-18}$$

The equations for the heat flow are written in matrix form as

$$A^{12}_{IJ} T^f_{J,t} + A^{22}_{IJ} T^f_J + C^2_I = 0 \tag{1.2-19}$$

$$A^{31}_{IJ} T^r_{J,t} + A^{33}_{IJ} T^r_J + C^3_I = 0 \tag{1.2-20}$$

where

$$A^{12}_{IJ} = \langle \phi \rho^f C^f \psi_J , \psi_I \rangle$$

$$A^{22}_{IJ} = \langle \phi \lambda^f \psi_{J,i} \ , \ \psi_{I,i} \rangle + \langle \rho^f C^f q_i \psi_{J,i} \ , \ \psi_I$$

$$A^{23}_{IJ} = 0$$

$$C^2_I = - \int_S \phi \lambda^f T^f_{,i} \psi_I dS_i + \langle w \ , \ \psi_I \rangle$$

$$A^{31}_{IJ} = \langle (1-\phi) \psi_J \ , \ \psi_I \rangle$$

$$A^{32}_{IJ} = 0$$

$$A_{IJ}^{33} = \langle (1-\phi)\lambda^r \psi_{J,i} \ , \ \psi_{I,i} \rangle - \langle \rho^r C^r c^r p_{,i} \psi_J \ , \ \psi_I \rangle$$

$$C_I^3 = -\int_S (1-\phi)\lambda^r T_{,i}^r \psi_I dS_i + \langle w \ , \ \psi_I \rangle$$

If quasi-linear heat exchange between the fluid and rock media is assumed, then the previous matrix system may be written as

$$A_{IJ}^{12} T_{J,t}^f + A_{IJ}^{22} T_J^f + A_{IJ}^{23} T_J^r + C_I^2 = 0 \qquad (1.2\text{-}21)$$

$$A_{IJ}^{31} T_{J,t}^f + A_{IJ}^{32} T_J^f + A_{IJ}^{33} T_J^r + C_I^3 = 0 \qquad (1.2\text{-}22)$$

where

$$A_{IJ}^{12} = \langle \phi \rho^f C^f \psi_J \ , \ \psi_I \rangle$$

$$A_{IJ}^{22} = \langle \phi \lambda^f \psi_{J,i} \ , \ \psi_{I,i} \rangle + \langle \rho^f C^f q_i \psi_{,i} \ , \ \psi_I \rangle + \langle h \psi_J \ , \ \psi_I \rangle$$

$$A_{IJ}^{23} = + \langle h \psi_J \ , \ \psi_I \rangle$$

$$C_I^2 = -\int_S \phi \lambda^f T_{,i}^f \psi_I dS_i$$

$$A_{IJ}^{31} = \langle (1-\phi) \psi_J \ , \ \psi_I \rangle$$

$$A_{IJ}^{32} = \langle h \psi_J \ , \ \psi_I \rangle$$

$$A_{IJ}^{33} = \langle (1-\phi)\lambda^r \psi_{J,i} \ , \ \psi_{I,i} \rangle$$

$$- \langle \rho^r C^r c^r p_{,i} \psi_J \ , \ \psi_I \rangle + \langle h \psi_J \ , \ \psi_I \rangle$$

$$C_I^3 = -\int_S (1-\phi)\lambda^r T_{,i}^r \psi_I dS_i$$

1.2.3 Boundary conditions for the heat flow equations

The previous set of governing equations must be supplemented with the appropriate boundary and initial conditions for the problem. These are of the following types:

(i) Prescribed temperature at a boundary.

(ii) Prescribed temperature at a boundary.

(iii) Prescribed heat flux normal to a boundary (e.g. zero heat flux).

The first type is considered for the top and the bottom boundaries of the flow domain. The prescribed temperature at the top boundary corresponds to the temperature at the ground surface or at the water table if this is below the ground surface. As for the fluid flow equation, a boundary condition corresponding to an imaginary bottom of the flow domain must be imposed. The temperature at the bottom boundary will, together with the prescribed temperature at the top boundary, specify the natural geothermal gradient. In a similar way as for the fluid flow equation the considered flow domain is confined laterally by the introduction of imaginary vertical boundaries.

1.3 NUMERICAL PROCEDURE FOR FLUID AND HEAT FLOW

1.3.1 Numerical procedure for the flow equations

The flow equations are written in matrix form as

$$A_{IJ}^{11} p_{J,t} + A_{IJ}^{12} T_{J,t}^{f} + A_{IJ}^{13} p_J + C_I^1 = 0 \qquad (1.3\text{-}1)$$

$$A_{IJ}^{12} T_{J,t}^{f} + A_{IJ}^{22} T_J^{f} + C_I^2 = 0 \qquad (1.3\text{-}2)$$

$$A_{IJ}^{31} T_{J,t}^{r} + A_{IJ}^{33} T_J^{r} + C_I^3 = 0 \qquad (1.3\text{-}3)$$

If a linear heat transfer function is considered, then the previous matrix system may be written as

$$A_{IJ}^{11} p_{J,t} + A_{IJ}^{12} T_{J,t}^{f} + A_{IJ}^{13} p_J + C_I^1 = 0 \qquad (1.3\text{-}4)$$

$$A_{IJ}^{12} T_{J,t}^{f} + A_{IJ}^{22} T_J^{f} + A_{IJ}^{23} T_J^{r} + C_I^2 = 0 \qquad (1.3\text{-}5)$$

$$A_{IJ}^{31} T_{J,t}^{r} + A_{IJ}^{32} T_J^{f} + A_{IJ}^{33} T_J^{r} + C_I^3 = 0 \qquad (1.3\text{-}6)$$

The system of matrix equations is non-linear. The non-linearities are treated by applying an iterative procedure in solving the matrix equations.

In the numerical scheme the fluid flow equation is solved separately. Thereafter the two heat flow equations are solved simultaneously. When the linear heat transfer function is applied, the fluid and rock temperatures are solved for implicitly in the two heat flow equations.

The present numerical scheme is based on direct iteration using weighted averages when solving for fluid flow coupled with heat flow, and Newton-Raphson iteration when solving for unsaturated flow. A brief statement of the application of the Newton-Raphson technique in the model is presented in the sequel.

Using the finite difference approach to approximate the time derivatives in the previous equations, we obtain

$$\frac{1}{\Delta t} A_{IJ}^{11}\left(p_J^{(n+1)} - p_J^{(n)}\right) + \frac{1}{\Delta t} A_{IJ}^{12}\left(T_J^{f(n+1)} - T_J^{f(n)}\right) \qquad (1.3\text{-}7)$$

$$+ A_{IJ}^{13} p_J^{(n+1)} + C_I^1 = 0$$

$$\frac{1}{\Delta t} A_{IJ}^{21}\left(T^{f(n+1)} - T_J^{f(n)}\right) + A_{IJ}^{22} T_J^{f(n+1)} \tag{1.3-8}$$
$$+ A_{IJ}^{23} T_J^{r(n+1)} + C_I^2 = 0$$

$$\frac{1}{\Delta t} A_{IJ}^{31}\left(T_J^{r(n+1)} - T_J^{r(n)}\right) + A_{IJ}^{32} T_J^{f(n+1)} \tag{1.3-9}$$
$$+ A_{IJ}^{33} T_J^{r(n+1)} + A_{IJ}^{34}\left(p_J^{(n+1)} - p_J^{(n)}\right) + C_I^3 = 0$$

where the figures enclosed by parentheses denote the time level. Rearranging, we obtain

$$\left(\frac{1}{\Delta t} A_{IJ}^{11} + A_{IJ}^{13}\right) p_J^{(n+1)} = \frac{1}{\Delta t} A_{IJ}^{11} p_J^{(n)} \tag{1.3-10}$$
$$- \frac{1}{\Delta t} A_{IJ}^{12}\left(T_J^{f(n+1)} - T_J^{f(n)}\right) - C_I^1$$

$$\left(\frac{1}{\Delta t} A_{IJ}^{21} + A_{IJ}^{22}\right) + T_J^{f(n+1)} + A_{IJ}^{23} T_J^{r(n+1)} \tag{1.3-11}$$
$$= \frac{1}{\Delta t} A_{IJ}^{21} T_J^{f(n)} - C_I^2$$

$$\left(\frac{1}{\Delta t} A_{IJ}^{31} + A_{IJ}^{32}\right) + T_J^{f(n+1)} + A_{IJ}^{33} T_J^{r(n+1)} \tag{1.3-12}$$
$$= \frac{1}{\Delta t} A_{IJ}^{31} T_J^{r(n)} - \frac{1}{\Delta t} A_{IJ}^{34}\left(p_J^{(n+1)} - p_J^{(n)}\right) - C_I^3$$

or

$$\left[A_{11}\right]\{p\} = \{C_1\} \tag{1.3-13}$$

$$\left[A_{22}\right]\{T^f\} + \left[B_{23}\right]\{T^r\} = \{C_2\} \tag{1.3-14}$$

$$\left[A_{32}\right]\{T^f\} + \left[B_{33}\right]\{T^r\} = \{C_3\} \tag{1.3-15}$$

Newton-Raphson iteration

The finite element formulation of the present flow equations leads to matrix problems of the following form

$$A_{IJ} p_J - B_I = R_I = 0 \tag{1.3-16}$$

Expanding the residual vector R_I in Taylor series, one obtains

$$R_I(p_J) = R_I\left(p_J^{(0)}\right) + \frac{\partial R_I\left(p_J^{(0)}\right)}{\partial p_J} \Delta p_J + \text{higher order terms} \tag{1.3-17}$$

where

$$\Delta p_J = p_J - p_J^{(0)} \tag{1.3-18}$$

Neglecting the higher order terms in (1.3-17) one may write

$$\frac{\partial R_I\left(p_J^{(0)}\right)}{\partial p_J} \Delta p_J = -R_I\left(p_J^{(0)}\right) \tag{1.3-19}$$

or

$$J_{IJ} \Delta p_J = -R_I\left(p_J^{(0)}\right) \tag{1.3-20}$$

On account of (1.3-16) one may write (1.3-19) as

$$A_{IJ} \Delta p_J = -R_I\left(p_J^{(0)}\right) \tag{1.3-21}$$

or

$$A_{IJ} \Delta p_J^{(n+1)} = -R_I\left(p_J^{(n)}\right) \tag{1.3-22}$$

which is solved successively until the stipulated criterion for convergence has been reached.

1.4 COUPLED FLOW AND ROCK STRESS

A preliminary set of equations for solving coupled fluid and rock stress is presented.

1.4.1 Basic equations for the rock displacements

For an isotropic rock mass the stress tensor may be defined as

$$\sigma_{ij} = \lambda e_{kk}\delta_{ij} + 2Ge_{ij} - \alpha(T - T_0)\delta_{ij} - p\delta_{ij} \qquad (1.4\text{-}1)$$

where λ and G are the Lame constants and $\alpha = E\beta/(1-2\nu)$, E is Young's modulus, ν is Poisson's ratio, β is the coefficient of thermal expansion, σ_{ij} is the total stress for an element consisting of a solid matrix and water. The first two terms terms in the right hand side of eq.(1.4-1), called the 'effective stress', determine the deformation of the solid matrix.

The equation of equilibrium is

$$\rho a_i = \frac{\partial \sigma_{ij}}{\partial x_j} + X_i \qquad (1.4\text{-}2)$$

where a_i are the acceleration components and X_i are the body force components per unit volume.

Substituting the definition of the stress tensor (1.4-1) into the equations of equilibrium, we obtain

$$(\lambda e_{kk}\delta_{ij})_{,j} + 2Ge_{ij,j} - (\alpha(T - T_0)\delta_{ij})_{,j} - (p\delta_{ij})_{,j} = 0 \qquad (1.4\text{-}3)$$

Expanding the first term on the left hand side of equation (1.4-3), we obtain

$$(\lambda e_{kk}\delta_{ij})_{,j} = \lambda\delta_{ij}e_{kk,j} = \lambda e_{kk,i} = \lambda e_{jj,i} \qquad (1.4\text{-}4)$$

$$(\lambda e_{kk}\delta_{ij})_{,j} = \lambda\frac{1}{2}(u_{j,ji} + u_{j,ji}) = \lambda u_{j,ji} \qquad (1.4\text{-}5)$$

Expanding the second term on the left hand side of equation (1.4-3), we obtain

$$2Ge_{ij,j} = 2G\left(\frac{1}{2}(u_{i,j} + u_{j,i})\right)_{,j} \qquad (1.4\text{-}6)$$

$$= 2G\left(\frac{1}{2}(u_{i,jj} + u_{j,ij})\right) = Gu_{i,jj} + Gu_{j,ij}$$

The third term of equation (1.4-3) becomes

$$\left(\alpha(T-T_0)\delta_{ij}\right)_{,j} = \alpha\delta_{ij}(T-T_0)_{,j} \qquad (1.4\text{-}7)$$
$$= \delta_{ij}\alpha(T-T_0)_{,j} = \alpha(T-T_0)_{,i}$$

and the fourth term becomes

$$\left(p\delta_{ij}\right)_{,j} = \delta_{ij}p_{,j} = p_{,i} \qquad (1.4\text{-}8)$$

Assuming quasi-steady conditions then equation (1.4-3) becomes

$$Gu_{i,jj} + (\lambda+G)u_{j,ji} - \alpha(T-T_0)_{,i} - p_{,i} = 0 \qquad (1.4\text{-}9)$$

1.4.2 Coupled equations for unsaturated flow and rock stress

1.4.2.1 Definitions

Rock density is assumed to be a function of temperature

$$\rho^r = \rho^r(T)$$

Fluid compressibility is defined as

$$c^f = \frac{1}{\rho^f}\frac{d\rho^f}{dp}$$

Thermal volume expansion coefficient for the fluid is defined as

$$\beta^f = \frac{1}{\rho^f}\frac{\partial\rho^f}{\partial T^f}$$

Thermal volume expansion coefficient for the rock is defined as

$$\beta^r = \frac{1}{\rho^r}\frac{\partial\rho^r}{\partial T^r}$$

Furthermore

u_i^r = rock displacement

$$v_i^r = u_{i,t}^r$$

$$v_{i,i}^r = \left(u_{i,t}^r\right)_{,i} = \left(u_{i,i}^r\right)_{,t} = e_{,t}$$

Darcy's law is:

$$q_i = S\phi\left(v_i^f - v_i^r\right) = -\frac{k_{ij}}{\mu}\left(p_{,i} - \rho^f g_i\right) \qquad (1.4\text{-}10)$$

1.4.2.2 Mass conservation equation for solid:

$$\left[(1-\phi)\rho^r\right]_{,t} + \left[(1-\phi)\rho^r v_i^r\right]_{,i} = 0 \qquad (1.4\text{-}11)$$

Expansion of the derivatives yields

$$(1-\phi)\rho_{,t}^r - \rho^r\phi_{,t} - \rho^r v_i^r\phi_{,i} + (1-\phi)\left(\rho^r v_i^r\right)_{,i} = 0 \qquad (1.4\text{-}12)$$

Rearranging and dividing the previous equation by ρ^r, we obtain

$$\phi_{,t} + v_i^r \phi_{,i} = \frac{1}{\rho^r}(1-\phi)\rho^r_{,t} + \frac{(1-\phi)}{\rho^r}\left(\rho^r v_i^r\right)_{,i} \tag{1.4-13}$$

Making use of the definition of the coefficient for thermal expansion of the rock, we obtain

$$\phi_{,t} + v_i^r \phi_{,i} = (1-\phi)\beta^r T_{,t} + (1-\phi)v_{i,i} + \frac{1-\phi}{\rho^r} v_i^r \rho^r_{,i} \tag{1.4-14}$$

Assuming (i) that v_i is small compared with $v_{i,i}$ and (ii) that the space derivative $\rho^r_{,i}$ is small compared with the time derivative $\rho^r_{,t}$, we obtain the following approximate equation.

$$\frac{d\phi}{dt} \approx (1-\phi)e_{,t} + (1-\phi)\beta^r \frac{dT}{dt} \tag{1.4-15}$$

or

$$\Delta\phi \approx (1-\phi)e + (1-\phi)\beta^r \Delta T \tag{1.4-16}$$

relating changes in porosity to changes in temperature

1.4.2.3 Mass conservation equation for unsaturated fluid flow

$$\left(S\phi\rho^f\right)_{,t} + \left(S\phi\rho^f v_i^f\right)_{,i} = 0 \tag{1.4-17}$$

$$\begin{aligned}&\phi\rho^f S_{,t} + S\phi\rho^f_{,t} + S\rho^f\phi_{,t} + v_i^f S\phi\rho^f_{,i} \\ &+\rho^f S v_i^f \phi_{,i} + \rho^f S\phi v^f_{i,i} + \rho^f\phi v_i^f S_{,i} = 0\end{aligned} \tag{1.4-18}$$

from (1.4-10)

$$q_{i,i} = \left(v_i^f - v_i^r\right)S\phi_{,i} + \left(v_i^f - v_i^r\right)\phi S_{,i} + S\phi\left(v^f_{i,i} - v^r_{i,i}\right) \tag{1.4-19}$$

$$\begin{aligned}&\rho^f S v_i^f \phi_{,i} + \rho^f S\phi v^f_{i,i} = \rho^f q_{i,i} \rho^f S v_i^r \phi_{,i} \\ &+\rho^f S\phi v^r_{i,i} - \rho^f\phi\left(v_i^f - v_i^r\right)S_{,i}\end{aligned} \tag{1.4-20}$$

Substitution of (1.4-20) and (1.4-13) into (1.4-18) yields

$$\begin{aligned}&\phi\rho^f S_{,t} + S\phi\rho^f_{,t} + v_i^f S\phi\rho^f_{,i} \\ &+\rho^f S\phi v^r_{i,i} + \rho^f\phi v_i^r S_{,i} + \rho^f q_{i,i} + \\ &\frac{S\rho^f}{\rho^r}(1-\phi)\rho^r_{,t} + \frac{S\rho^f(1-\phi)}{\rho^r}\rho^f v^r_{i,i} + \frac{S\rho^f(1-\phi)}{\rho^r} v_i^f \rho^r_{,i} = 0\end{aligned} \tag{1.4-21}$$

Making use of the definitions presented above, we obtain

$$\begin{aligned}&\phi\rho^f S^{'} p_{,t} + S\phi\rho^f c^f p_{,t} + S\phi\beta^f\rho^f T^f_{,t} + S\phi v_i^f\rho^f_{,i} + \rho^f\phi v_i^f S_{,i} + \rho^f q_{i,i} \\ &S\rho^f v^r_{i,i} + \frac{S\rho^f}{\rho^r}(1-\phi)\rho^r\beta^r T^r_{,t} + S(1-\phi)\frac{\rho^f}{\rho^r} v_i^r\rho^r_{,i} = 0\end{aligned} \tag{1.4-22}$$

or

$$S\phi\rho^f\beta^f T^f_{,t} + S\rho^f(1-\phi)\beta^r T^r_{,t} + \phi\rho^f\left(S' + Sc^f\right)p_{,t} + S\rho^f v^r_{i,i}$$

$$+\left(\rho^f q_i\right)_{,i} + \phi v^r_i\left(\rho^f S\right)_{,i} + S(1-\phi)\frac{\rho^f}{\rho^r}v^r_i\rho^r_{,i} = 0 \qquad (1.4\text{-}23)$$

or

$$S\phi\rho^f\beta^f T^f_{,t} + S(1-\phi)\rho^f\beta^r T^r_{,t} + \phi\rho^f\left(S' + Sc^f\right)p_{,t} + S\rho^f e_{,t}$$

$$\left(\rho^f q_i\right)_{,i} + \phi u^r_{i,i}\left(\rho^f S\right)_{,i} + S(1-\phi)\frac{\rho^f}{\rho^r}v^r_i\rho^r_{,i} = 0 \qquad (1.4\text{-}24)$$

where

$$q_i = -\frac{k_{ij}}{\mu}\left(p_{,j} - \rho^f g_j\right) \qquad (1.4\text{-}25)$$

2 PROGRAM STRUCTURE

2.1 INTRODUCTION

2.2 MAIN PROGRAM AND ITS SUBROUTINE CALLS

2.3 MASTER SUBROUTINE AND ITS SUBROUTINE CALLS

2.4 COMMON BLOCKS

2.5 DESCRIPTION OF VARIABLES IN COMMON BLOCKS

2.1 INTRODUCTION

The program consists of a main program and a set of subroutines. The main program (HFMAIN) is used to specify the dimensions of most of the arrays whose sizes are problem dependent.

There are basically four parameters values that need to be specified in the main program. These are: MXNPX for the maximum permitted number of nodal points, MXNEX for the maximum permitted number of elements, MXNDE for the maximum permitted number of nodes per element and MXFROX for the maximum permitted size of the front-width. The values of these parameters are specified using PARAMETER statements. A more detailed description of the variables in the main program as well as the input data will be given in Chapter 3.

The program structure and its subroutine interrelations are shown in the next paragraph.

The principal subroutine, containing the overall structure of the model, is subroutine GFSBX1. The main features of this subroutine and its interrelations with other subroutines are illustrated in the sequel.

2.2 PROGRAM ORGANIZATION

2.2.1 Main program

```
 Main       Subroutines
HFMAIN-|---JBAUX1-|---JOBNAM  Get current job identification
       |          |---XDATE1  Get current date from system
       |          |---XCLOCK  Get current time from system
       |---JBAUX3
       |---ZERO Disable underflow messages
       |   Transfer of external subroutine names
       |---HFEXRF-|---HFSBX2
       |          |---HFSBX3
       |          |---HFINS2
       |          |---HFINS4
       |          |---SETSTP  Time step control
       |          |---SETST1  Time step control
       |          |---SETST2  Time step control
       |          |---SETST3  Time step control
       |          |---SETST4  Time step control
       |          |---HFL2D2
       |          |---HFL2DB
       |          |---GFL2D2
       |          |---HFELT5  Selection of integration routine for
       |                      heat flow
       |   Master subroutine
       |---GFSBX1-|---XTIME0  Initialize time taking
       |          |   Input and initialization
       |          |---HFINS1---|---LBPHX
       |          |             |---HFINPT---|---TEXINU
       |          |
       |          |   Groundwater flow equations
       |          |---GFPHS2--|---GFNPG1---|---FNDNF1
       |          |           |             |---FNVSF1
       |          |           |---LBPHX  Test printout
       |          |           |---PORSLV---|--LBPHX
       |          |           |            |--PRSLVF
       |          |           |            |  Element integration
       |          |           |            |--HFELFX
       |          |           |   Fracture flow
       |          |           |---FRCSLV
       |          |           |   Two-phase flow
       |          |           |---TPHSLV
       |          |           |
       |          |           |---PHRES1  Newton-Raphson
       |          |           |---HFWRKP-->-|Update work arrays
       |          |           |---HFWAV1    | weighted average
       |          |           |---HFWAV2    | calculation
       |          |           |---PHSRC1
       |          |           |---PHSRC2
       |          |           |---HFSAVD  Store initial data on
       |          |                       disk file
       |          |---DLSCHK  Gas migration displacement check
       |          |---HFWRK1  Update nodal arrays
       |          |---DRAINF
       |          |---DRAINX
       |          |---DRAING
       |          |
```

```
|           |   Heat flow equations
|           |---HFTMS1---|---HFELT1 Heat flow integration
|           |
|           |   Solution of equations for rock stress
|           |---HFSTS1---|
|           |            |   Element integration
|           |            |---HFLGN4
|           |---SOLVUT---|---UTSLV3
|           |            |---UTSLV4
|           |---HFITER  Iteration control
|           |---PHMXC1  Pressure change control
|           |---HFWRK2  Update work arrays
|           |---HFOUT3  Printout solution data
|           |---HFLXNU  Calculation of nodal velocities
|           |---SOLVUT  Printout solution data
|           |---GFPRT1  Gas migration printout
|           |---SFRUT1  Two-phase flow printout
|           |---GFNPG3  Gas density abd viscosity
|           |---GMBLA1  Gas migration mass balance
|           |---HFSAVD---|---HFNDU2
|           |---GELIN2  Generation of mesh for gas migration
|           |---GFPRT1  Printout gas migration results
|           |---UNSPRT  Printout unsatureted data
|           |   Printout solution data
|           |---SOLVU2---|
|           |            |---UTSLV3
|           |            |---NDPRU2
|           |            |---NDPRU3
|           |            |---NDPRT2
|           |            |---HFWQUA Evaluate heat transfer
|           |                       function
|           |---HFWRK1  Update work arrays
|           |---UNSPRT  Printout unsatureted data
|           |---SOLVU2  Printout solution data
|           |---HFWRK1  Update work arrays
|           |---HFNDUX---|---HFNDU1
|           |---HFWRK2  Update work arrays
|           |---HFHED2  Printout intecation data
|           |---HFSAVD  Write solution data to disk file
|           |---HFBLS1---| Mass and Energy balance calculation
|           |            |
|           |   Ground water table localization
|           |---GFWTBL---|---ASORT
|           |---XTIME   Time taking
|           |---HFNFRZ  Set element connectivity positive
|           |---HFAX3D  Transfer pressure and temperature
|           |---HFIELN  Redefinition of element incidences
|           |---HFNDUT  Nodal data output
|           |---HFELUT  Element incidences output
|           |---HFNDUF  Nodal data output
|           |---HFELUF  Element incidences output
|           |---UNSUT1  Unsaturated data output
|           |---HFINP5  Rock stress parameter output
|---MFLOPS
|   Megaflopcount
Stop
```

2.2.2 Master subroutine

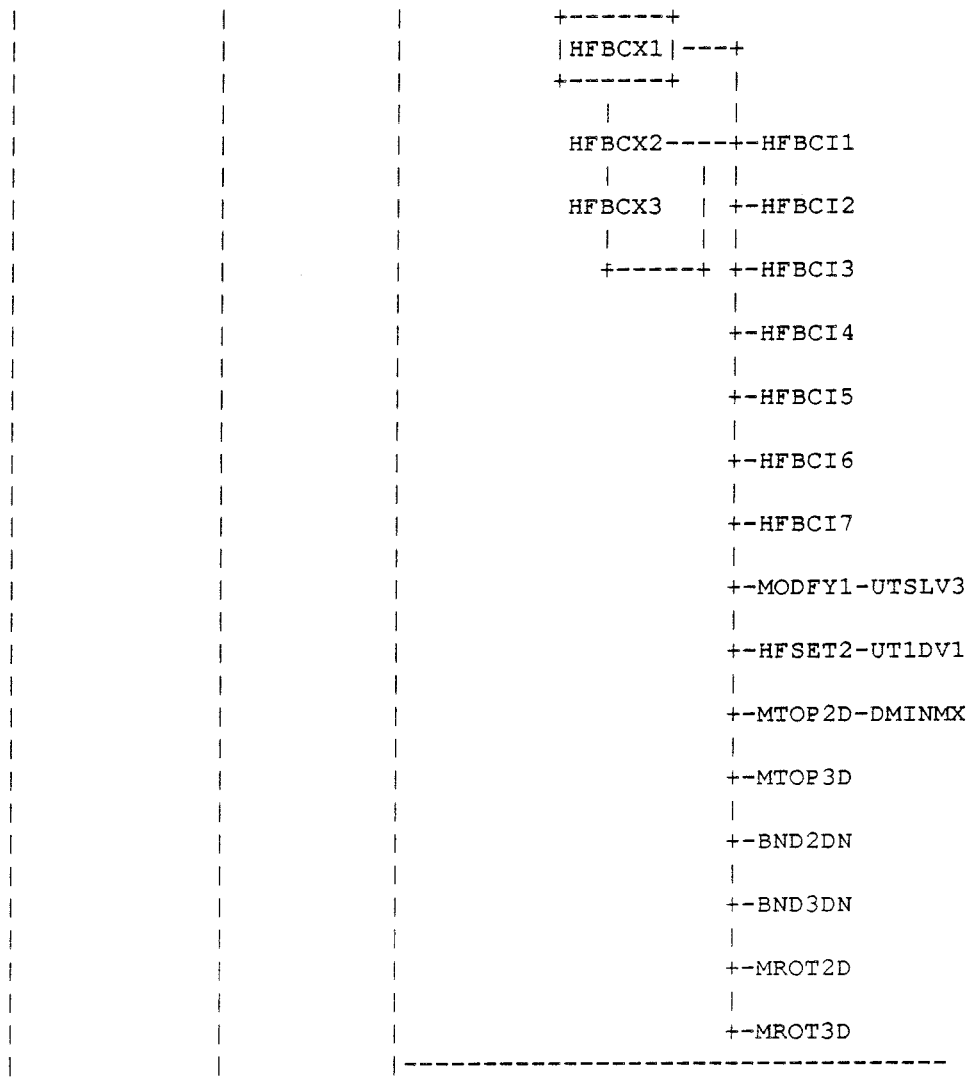
```
GFSBX1-|---XTIME0  Initialize time taking
       |   Input and initialization
       |---HFINS1---|---LBPHX
       |            |---HFINPT---|---TEXINU
       |            |            |---EXFUN1---|---FDENF---|-HFELG2
       |            |            |            |           |-HFELF2
       |            |            |            |           |-HFELT2
       |            |            |            |           |-HFLSG2
       |            |            |            |           |-HFLXN2
       |            |            |            |           |-GFNPF1
       |            |            |            |           |-GFNPF3
       |            |            |            |           |-VELXY4
       |            |            |            |
       |            |            |            |---FVISF---|-HFELG3
       |            |            |            |           |-HFELF3
       |            |            |            |           |-HFELT3
       |            |            |            |           |-HFLGS3
       |            |            |            |           |-HFLXN3
       |            |            |            |           |-GFNPF2
       |            |            |            |           |-GFNPF4
       |            |            |            |           |-VELXY5
       |            |            |            |
       |            |            |            |---FVISG---|-HFELG5
       |            |            |            |           |-HFLGS5
       |            |            |            |           |-EIGAS3
       |            |            |            |           |-GFNPGA
       |            |            |            |           |-GFNPGB
       |            |            |            |
       |            |            |            |---FHC2----|-HFPRM2
       |            |            |            |           |-HFLHC2
       |            |            |            |           |-FPERM3
       |            |            |            |
       |            |            |            |---FPOR----|-HFELG4
       |            |            |            |           |-HFELF4
       |            |            |            |           |-HFELT4
       |            |            |            |           |-HFLGS4
       |            |            |            |           |-HFELP3
       |            |            |            |
       |            |            |            |---FDCAY---|-WDCAY2
       |            |            |   Selection of functions
       |            |            |
       |            |            |---FNDNF1 Density = constant
       |            |            |---FNDNF2 Density = f(P,T)
       |            |            |---FNDNF3 Density = f(P,T)
       |            |            |---FNDNF4 Linear function of T
       |            |            |---FNDNF5 Density = f(exp(T))
       |            |            |---FNDNF6 Density = f(P,T)
       |            |            |---FNVSF1 Viscosity = constant
       |            |            |---FNVSF2 Viscosity = f(T)
       |            |            |---FNVSF5 Viscosity = f(exp(T))
       |            |            |---EIFLU1 Ei-function for fluid
```

```
|            |            |---EIGAS1 Ei-function for gas
|            |            |---GAUSSP Set Gauss points
|            |
|            |
|            |---HFSET1  Initialize work arrays
|            |---HFTEST  Ad Hoc input parameter settings
|            |---HFNDIN  Input nodal points
|            |---HFELIN  Input element incidences
|            |---HFAX3A  Transfer inital pressure and
|            |           temperature values
|            |---HFNDIF  Input nodal coordinates
|            |---HFELIF  Input nodal incidences
|            |---HFAX3A  Transfer of input nodal data to
|            |           work arrays (I.C. and B.C.)
|            |---ELFRIN  Input of block specification
|            |---ELREFM  Setup fracture plane reference arrays
|            |---NODSWP  Reorderring of nodal arrays
|            |---HFPERM---|---FUNHC2 Permeability function
|            |  (HFPRM2)
|            |
|            |            **** User controlled input ****
|            |---PRGSEL-|---------------------------------
|            |          | AMINMX  Find coordinate extremes
|            |          |         in  x- resp. y-direction
|            |          |
|            |          | DATAIO  Call  subroutine  to read
|            |          |         data from disc file
|            |          |
|            |          | DATAUO  Write data to disc file
|            |          |
|            |          | DRAINI  Input drain data
|            |          |
|            |          | GELIN1  Input mesh parameter  data
|            |          |         and  intial gas migration
|            |          |         conditions
|            |          |
|            |          | GFELHC  Input   parameters    for
|            |          |         parallel   plate    model
|            |          |         conceptualization and set
|            |          |         element or nodal permea-
|            |          |         bility arrays
|            |          |
|            |          | GFINJ1  Set full gas  displacement
|            |          |         at "injection" nodes
|            |          |
|            |          | GFRIN1  Input  parameter  data to
|            |          |         set  boundary and initial
|            |          |         conditions
|            |          |
|            |          | GFSEDT  Input fracture
|            |          |         distribution data
|            |          |
|            |          | GWSRCI  Input   point   mass
|            |          |         source/sinks    to     be
|            |          |         applied     to     frontal
|            |          |         equations
|            |          |
|            |          | HDR4B   Set  the  inital  pressure
```

distribution hydrostatic (Hydrocoin Level 1, Case 4)

HFAX3A — Transfer nodal data from work arrays -HH- and -LBHH- being read by subr.HFNDIN* to nodal arrays. This is for cases where the initial or boundary conditions have been set in grid data

HFBCX1 — Miscellaneous operations for input of boundary conditions, initial values, mesh modification, etc.



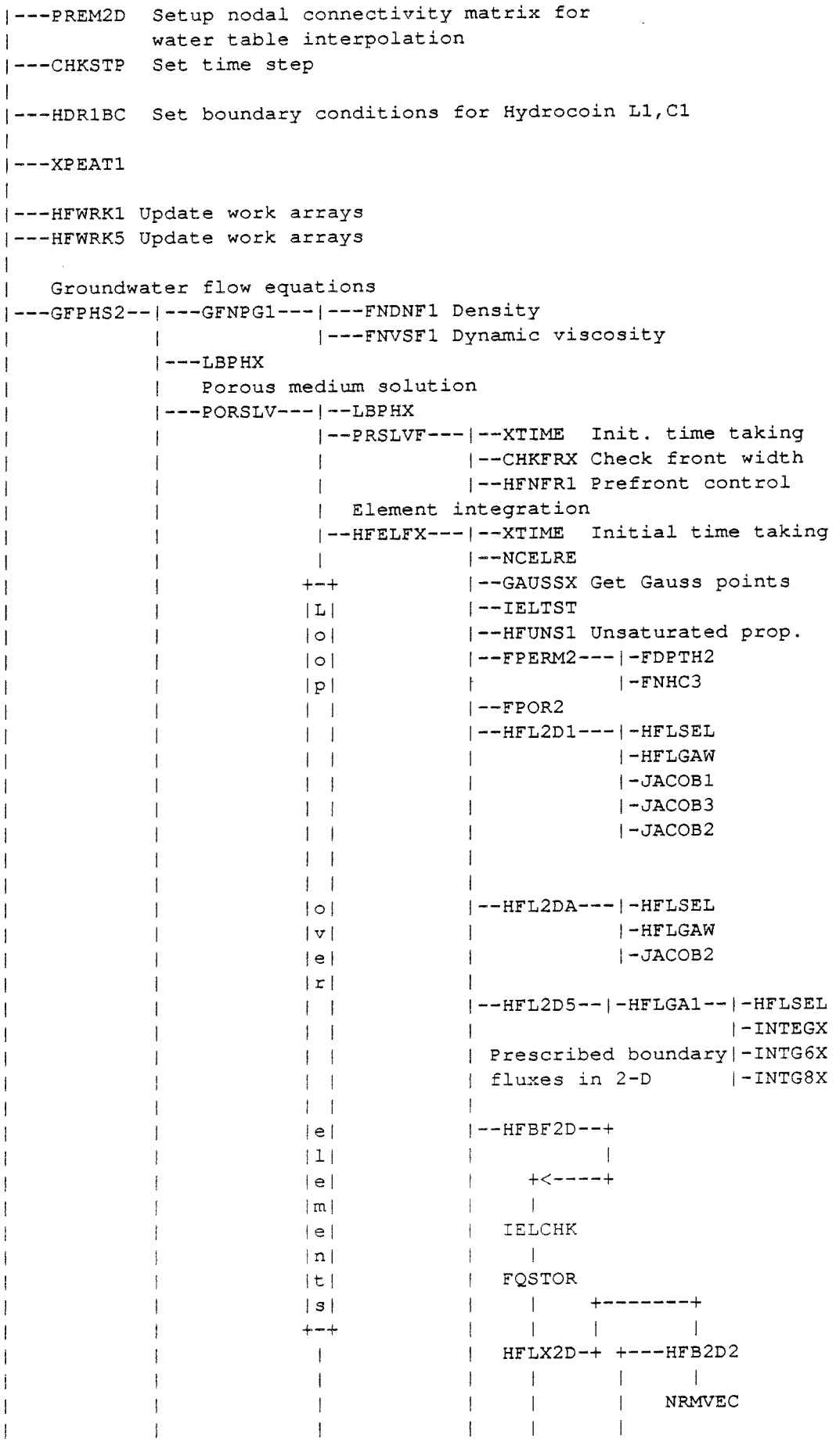
HFBELT — Find nodes on top boundary and set prescribed flux condition

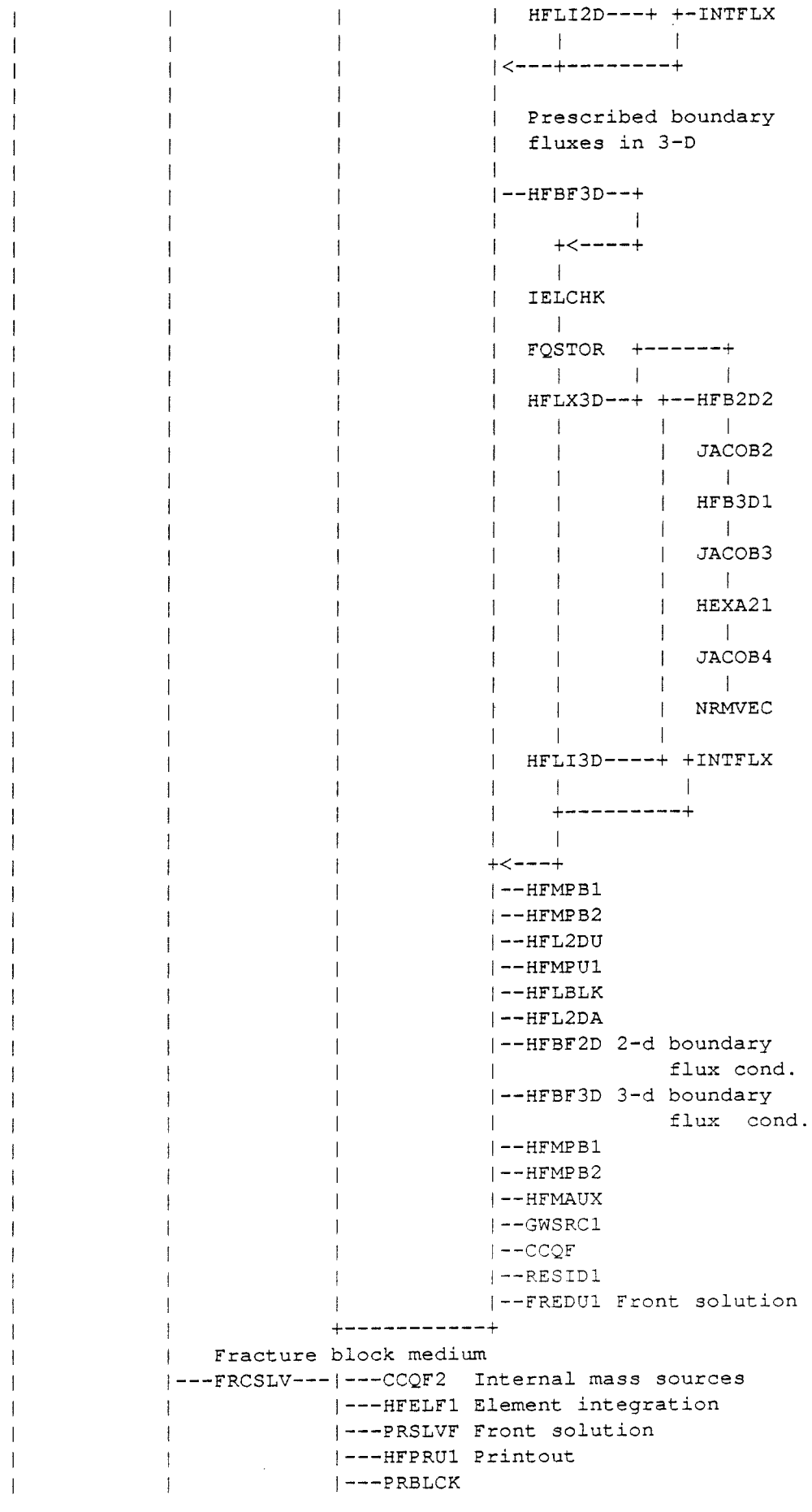
HFBFLW- — Input boundaries with

				prescribed flux conditions
				-CHKIEL
				-CHKSD2
				-CHKSD3
				-CHKND2
				-CHKND3
				-BLSORT
			HFELHC	Set element permeability (ref.value + region specification)
			HFELIF	Input element incidences (discrete system)
			HFELIN	Input element incidences
			HFELMP	Specify (unsaturated) material properties for elements or nodes
			HFELMP	Specify (unsaturated) material properties for
			HFELM1	Set element permeability according to values of -IELMAT-
			HFELPM	Specify (unsaturated) material properties for elements or nodes
			HFELP1	Set element porosity
			HFIELN	Redefine matrix of element incidences if 8-21 node hexahedral element is to be used
			HFINFR	Input fracture elements
			HFINP3	Input of material properties for unsaturated conditions
			HFINP4	Set the inital pressure distribution (hydrostatic)
			HFIN4P	Set the inital pressure distribution hydrostatic for unsaturated flow conditions
			HFINP5	Input data for stress solution and input of

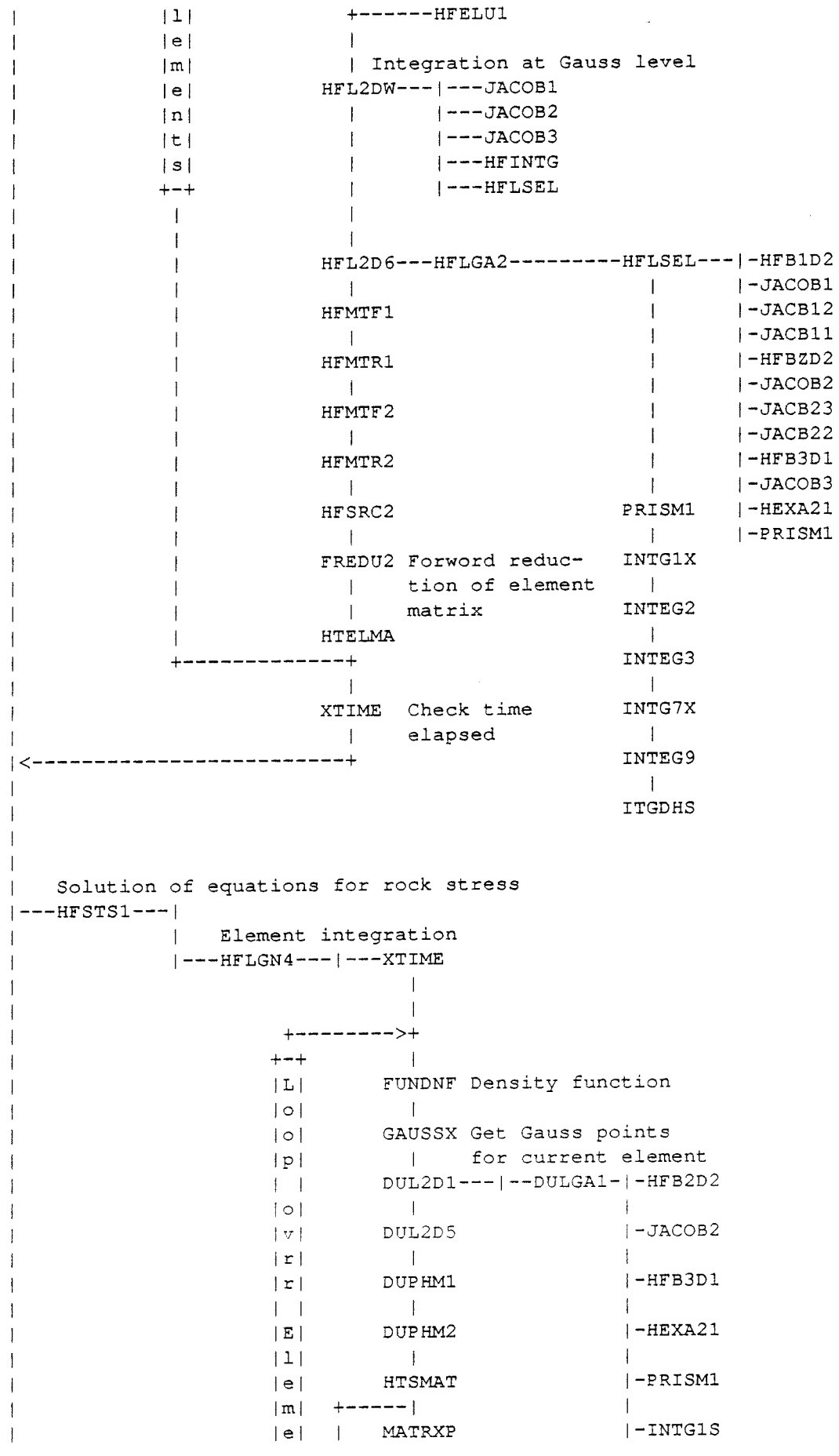
	initial and boundary conditions
HFMSRC	Input concentrated mass sources
HFNDIF	Input nodal points for discrete system of fractures and blocks
HFNDIN	Input nodal points
HFNFRC	Invoke subroutine to set nodal permeability for nodes associated with fractures
HFPERM	Set nodal permeabilities
HFPRMA	Set mid-side properties (permeability) as averages of adjaccent corner nodes
HFSRC1	Input concentrated and/or distributed heat sources
HYDSBS	Input subroutine for pressure, permeability, porosity, etc.
MVSDE1	Modify element mesh according to previous specification
NODSL2	Specify area within which nodes are to be selected for printout
NODSL4	Specify and set zones of specific material properties
PHXMIN	Set prescribed pressure on left hand boundary
PHXMAX	Set prescribed pressure on right hand boundary
PHYMIN	Set prescribed pressure on bottom boundary
PHYMAX	Set prescribed pressure on top boundary
PHYDRO	Sets hydrostatic boundary conditions for

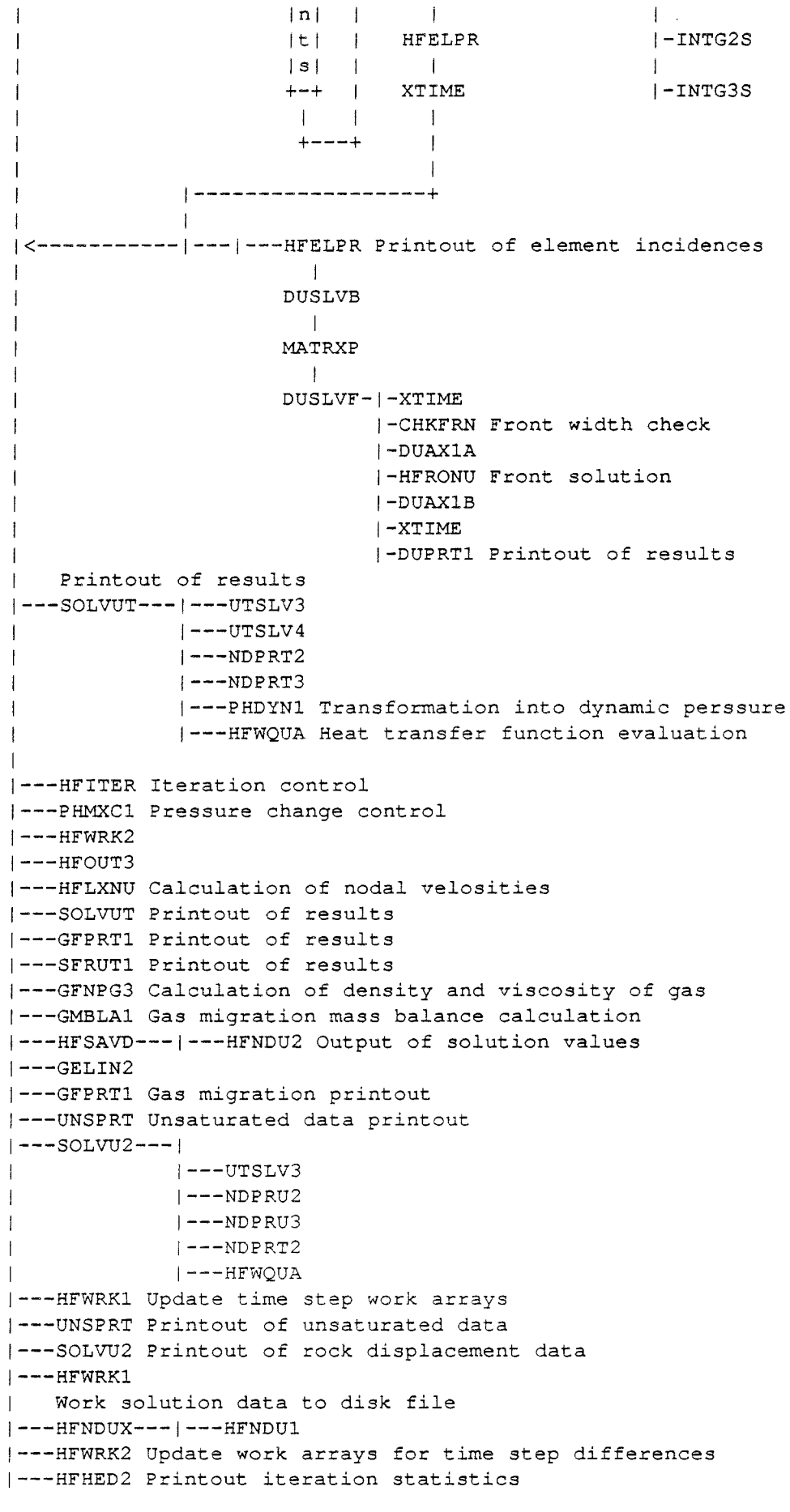
```
|              |           |          temperature     dependent
|              |           |          fluid density and a given
|              |           |          temperature  distribution
|              |           |
|              |           |
|              |           | TFXMIN   Set prescribed
|              |           |          temperature  on left hand
|              |           |          boundary
|              |           |
|              |           | TFXMAX   temperature on right hand
|              |           |          boundary
|              |           |
|              |           | TFYMIN   Set prescribed
|              |           |          temperature on bottom
|              |           |          boundary
|              |           |
|              |           | TFYMAX   Set prescribed
|              |           |          temperature on top
|              |           |          boundary
|              |           |
|              |           | SELNDS   Input node numbers for
|              |           |          selected  printout  of of
|              |           |          nodal data
|              |           |
|              |           | SFRIN1   Set boundary and initial
|              |           |          conditions  for  1-d  gas
|              |           |          flow test
|              |           |
|              |           | SHWTIN   Input the time dependent
|              |           |          energy   output   for
|              |           |          Hydrocoin Level 2, Case 2
|              |           |-----------------------------------
|              |
|              |---HFNDUV  Input solution values from
|              |           previous solution
|              |---UTLBHH
|              |---HFOUT1
|              |---HFOUT3
|              |---HFHEAD---|---JBNAM
|              |            |---XDATE1
|              |            |---XCLOCK
|              |
|              |---HFWSET
|              |---HFWUPD
|
|
|---AMINMX  Check minimum resp.maximum coordinates of flow
|           domain
|---PHYMIN  Boundary condition at bottom boundary
|---PHYMAX  Boundary condition at top boundary
|---PHXMIN  Boundary condition at lefthand lateral boundary
|---PHXMAX  Boundary condition at righthand lateral boundary
|
|---JBAUX1  Get job identification for current job
|---JBAUX3  Print out job identification
|
|---HFSAVD  Storage of solution data on disk file
|
```


---PREM2D Setup nodal connectivity matrix for water table interpolation
---CHKSTP Set time step
---HDR1BC Set boundary conditions for Hydrocoin L1,C1
---XPEAT1
---HFWRK1 Update work arrays
---HFWRK5 Update work arrays
Groundwater flow equations
---GFPHS2--
---GFNPG1---
---FNDNF1 Density
---FNVSF1 Dynamic viscosity
---LBPHX
Porous medium solution
---PORSLV---
--LBPHX
--PRSLVF---
--XTIME Init. time taking
--CHKFRX Check front width
--HFNFR1 Prefront control
Element integration
--HFELFX---
--XTIME Initial time taking
--NCELRE
--GAUSSX Get Gauss points
--IELTST
--HFUNS1 Unsaturated prop.
--FPERM2---
-FDPTH2
-FNHC3
--FPOR2
--HFL2D1---
-HFLSEL
-HFLGAW
-JACOB1
-JACOB3
-JACOB2
--HFL2DA---
-HFLSEL
-HFLGAW
-JACOB2
--HFL2D5--
-HFLGA1--
-HFLSEL
-INTEGX
-INTG6X
-INTG8X
Prescribed boundary fluxes in 2-D
--HFBF2D--
IELCHK
FQSTOR
HFLX2D--
HFB2D2
NRMVEC
Loop
over
element
s


HFLI2D
INTFLX
Prescribed boundary
fluxes in 3-D
HFBF3D
IELCHK
FQSTOR
HFLX3D
HFB2D2
JACOB2
HFB3D1
JACOB3
HEXA21
JACOB4
NRMVEC
HFLI3D
INTFLX
HFMPB1
HFMPB2
HFL2DU
HFMPU1
HFLBLK
HFL2DA
HFBF2D 2-d boundary
flux cond.
HFBF3D 3-d boundary
flux cond.
HFMPB1
HFMPB2
HFMAUX
GWSRC1
CCQF
RESID1
FREDU1 Front solution
Fracture block medium
FRCSLV
CCQF2 Internal mass sources
HFELF1 Element integration
PRSLVF Front solution
HFPRU1 Printout
PRBLCK

```
|            |             |---PRSLVF Front solution
|            |             |---HFPRU1 Printout
|            |   Simultaneous two-phase flow
|            |---TPHSLV---|---GFPRED
|            |             |---GFMPS3 Time derivatives
|            |             |---HFELG1 Element integration
|            |             |---PRSLVF Front solution
|            |             |---HFPRU1 Printout
|            |             |---SFRUT1 Printout
|            |             |---HFWRK2 Update work arrays
|            |             |---GFPRT1 Printout results
|            |             |---ADJUST
|            |             |---HFLGS1 Simultaneous element
|            |             |          integration
|            |             |---PRSLV2 Front solution
|            |             |---HFAX1B
|            |             |---GFNPG3 Gas density and visc.
|            |             |---GFELG1 Gas migration integration
|            |             |---HFNFRI
|            |             |---HFNFRZ
|            |             |---HFAX1A Work array transfer
|            |             |---HFPRU1 Printout gas migration
|            |             |---GMBLA1 Gas-water mass balance
|            |
|            |---PHRES1
|            |---HFWRKP  Update work arrays
|            |---HFWAV1  Weighted averages (2 time levels)
|            |---HFWAV2  Weighted averages (3 time levels)
|            |---PHSRC1  Internal mass source transfer
|            |---PHSRC2  Internal mass source transfer
|            |---HFSAVD Store solution data on disk file
|
|---DLSCHK Gas-water interface displacement control
|---HFWRK1 Update work arrays
|
|---DRAINF
|---DRAINX
|---DRAING
|
|   Heat flow equations - Integration of elements
|---HFTMS1---|---HFELT1----|
|                          |
|                         XTIME Check time elapesd
|                          |
|             +----------->|
|             |            |
|            +-+          HFUNS1 Evaluate unsaturated
|            |L|            |    conditions (saturation-
|            |o|            |    relative permeability)
|            |o|            |
|            |p|          FPERM2 Function for permeability
|            | |            |
|            |o|          FPOR2  Function for porosity
|            |v|            |
|            |e|          WDECAY Function for radioactive decay
|            |r|            |
|            | |          GAUSSX Selection of Gauss pionts
|            |E|            |
```


|l|
|e|
|m|
|e|
|n|
|t|
|s|
HFELU1
Integration at Gauss level
HFL2DW
JACOB1
JACOB2
JACOB3
HFINTG
HFLSEL
HFL2D6
HFLGA2
HFLSEL
HFB1D2
JACOB1
HFMTF1
JACB12
HFMTR1
JACB11
HFMTF2
HFBZD2
HFMTR2
JACOB2
HFSRC2
JACB23
FREDU2
Forword reduction of element matrix
JACB22
HFB3D1
HTELMA
JACOB3
PRISM1
HEXA21
XTIME
Check time elapsed
PRISM1
INTG1X
INTEG2
INTEG3
INTG7X
INTEG9
ITGDHS
Solution of equations for rock stress
HFSTS1
Element integration
HFLGN4
XTIME
|L|
|o|
|o|
|p|
| |
|o|
|v|
|r|
|r|
| |
|E|
|l|
|e|
|m|
|e|
FUNDNF Density function
GAUSSX Get Gauss points for current element
DUL2D1
DULGA1
HFB2D2
DUL2D5
JACOB2
DUPHM1
HFB3D1
DUPHM2
HEXA21
HTSMAT
PRISM1
MATRXP
INTG1S


|n|
|t|
|s|
HFELPR
XTIME
-INTG2S
-INTG3S
HFELPR Printout of element incidences
DUSLVB
MATRXP
DUSLVF-|-XTIME
|-CHKFRN Front width check
|-DUAX1A
|-HFRONU Front solution
|-DUAX1B
|-XTIME
|-DUPRT1 Printout of results
Printout of results
SOLVUT---|---UTSLV3
|---UTSLV4
|---NDPRT2
|---NDPRT3
|---PHDYN1 Transformation into dynamic perssure
|---HFWQUA Heat transfer function evaluation
|---HFITER Iteration control
|---PHMXC1 Pressure change control
|---HFWRK2
|---HFOUT3
|---HFLXNU Calculation of nodal velosities
|---SOLVUT Printout of results
|---GFPRT1 Printout of results
|---SFRUT1 Printout of results
|---GFNPG3 Calculation of density and viscosity of gas
|---GMBLA1 Gas migration mass balance calculation
|---HFSAVD---|---HFNDU2 Output of solution values
|---GELIN2
|---GFPRT1 Gas migration printout
|---UNSPRT Unsaturated data printout
|---SOLVU2---|
|---UTSLV3
|---NDPRU2
|---NDPRU3
|---NDPRT2
|---HFWQUA
|---HFWRK1 Update time step work arrays
|---UNSPRT Printout of unsaturated data
|---SOLVU2 Printout of rock displacement data
|---HFWRK1
Work solution data to disk file
|---HFNDUX---|---HFNDU1
|---HFWRK2 Update work arrays for time step differences
|---HFHED2 Printout iteration statistics

```
|---HFSAVD Write solution data to a file
|---HFBLS1---| Mass and Energy balance calculation
|             |
|             +----------------+
|             |                |
|          FFMBL1-|          FFMBL3-| Perform mass balance
|             |   |             |   |      for the fluid
|             |   |-FUNDNF      |   |-FUNDNF
|             |   |-FUNVSC      |   |-FUNVSC
|             |   |-FPERM2      |   |-FPERM2
|             |   |-HFUNS1      |   |-HFUNS1
|             |   |-HFBS2D      |   |-HFB3D1
|             |   |-JACOB2      |   |-JACOB3
|             |   |-NRMVEC      |   |-JACOB4
|             |                 |   |-NRMVEC
|             |                 |
|          HFWQUA             HFWQUA-| Evaluate the linear
|             |                 |      heat transfer
|             |                 |      function
|          TFEBL1-|           TFEBL3-| Perform energy
|             |   |             |   | balance for the fluid
|             |   |-FUNDNF      |   |-FUNDNF
|             |   |-FUNVSC      |   |-FUNVSC
|             |   |-FPERM2      |   |-FPERM2
|             |   |-HFUNS1      |   |-HFUNS1
|             |   |-HFB2D2      |   |-HFB3D1
|             |   |-JACOB2      |   |-JACOB3
|             |   |-FPOR2       |   |-JACOB4
|             |   |-WDECAY      |   |-NRMVEC
|             |   |-NRMVEC      |
|             |                 |
|          TREBL1-|           TREBL1-| Perform energy
|             |   |             |   | balance for the rock
|             |   |-HFB2D2      |   |-HFB3D1
|             |   |-JACOB2      |   |-JACOB3
|             |   |-NRMVEC      |   |-JACOB4
|             |                 |   |-NRMVEC
|          HFBLS1             HFBLS3 | Printout summary of
|                                |      mass- and
|             |                  |      energy balances
|             +<---------------+
|             |
|   Groundwater table interpolation
|---GFWTBL---|---ASORT
|---XTIME  Check time elapsed
|---HFNFRZ Reset connectivity matrix to positive values
|---HFAX3D Transfer nodal arrays to output work arrays
|---HFIELN Reset hexahedral element definition to original
|---HFNDUT Output of nodal points to disk file
|---HFELUT Output of element incidences to disk file
|---HFNDUF Output nodal points to disk file (discrete)
|---HFELUF Output element incidences to disk file (diskrete)
|---UNSUT1 Output unsaturated data to disk file
|---HFINP5 Output elasticity data to disk file
|
Return
GFSBX1
```

CONTENTS

2.3. LIST OF SUBROUTINES

2.3.1 Main programs

HFMAIN - Main program to specify the sizes of most of the problem dependent arrays.

DUMMYX - Subroutine associated with HFMAIN with dummy entries for model subroutines not to be included in current setup.

HFMAIG - Main program to specify the sizes of most of the problem dependent arrays (Gas migration).

DUMMYG - Subroutine associated with HFMAIG with dummy entries to replace model subroutines not to be included in current setup.

HFMFRN - Main program to specify the sizes of most of the problem dependent arrays (Discrete modelling with stochastically generated fracture networks).

DUMMYF - Subroutine associated with HFMFRN with dummy entries for replacement of subroutines that should not be included in current setup.

2.3.2 Master subroutines

GFSBX1 - Principal subroutine for the model containing the overall structure of the model.

HFITER - Subroutine for iteration control.

GFITER - Subroutine for iteration control for two-phase flow.

PHMXC1 - Subroutine to check if changes in nodal pressures exceed given tolerance values.

HFINS1 - Main control subroutine for input and initialization

PRGSEL - Subroutine to administer user controlled input. This subroutine contains most common blocks and work arrays used by the model programs.

MVSDE1 - Subroutine to modify the element grid. There are two options:
(i) to apply a uni-directional slope by moving any of the upper corner nodes in vertical direction.
(ii) to move any of the upper corners in horizontal direction.

HFPRMA - Subroutine to set midside nodal permeability to the average value of the two adjacent corner permeability values (currently only applicable to 8-nodes quadrilateral elements).

GFPHS2 - Subroutine for solution of pressure (fluid flow equation).

PORSLV - Porous medium flow solution.

FRCSLV - Fractured medium solution.

PRBLCK - Application of fluid pressure in fracture system to block system.

CCQF - Application of internal mass sources.

TPHSLV - Two-phase flow solution subroutine.

HFTMS1 - Subroutine for solution of fluid and rock temperatures.

HFSTS1 - Subroutine for solution of displacements.

HFEXRF - Subroutine to input and pass external references from *HFMAIN* OR *HFMFRN* to various subroutines.

2.3.3 Physical properties

FUNDNF - Generic function subroutine to relate the fluid density to pressure and temperature.

FNDNF1 - Function subroutine to relate the fluid density to pressure and temperature (Dummy subroutine).

```
      FUNCTION FNDNF1(P,T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/HFREF1/ PREF,TREF,HCREF,PRMREF,POREF,
     1               DNFREF,VSCREF
      FNDNF1=DNFREF
      RETURN
      ENTRY     FUNDNF(P,T)
      FUNDNF = DNFREF
      RETURN
      END
```

FNDNF2 - Function subroutine to relate the fluid density to pressure and temperature.

```
      FUNCTION FNDNF2(PP,TT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA A0,A1,A2,A3,A4,A5
     1  /1.000245779, -2.9782D-05, 6.536D-06, -2.830D-08,
     2    0.9115D-10,  -4.697D-10/
      V=TT*(TT*(TT*(TT*A4 + A3) + A2) + A1) + A0 + A5*PP
      FNDNF2=1000./V
      RETURN
      END
```

FNDNF3 - Function subroutine to relate the fluid density to pressure and temperature. Fourth degree polynomial for temperature and linear relation for pressure.

```
      FUNCTION FNDNF3(PP,TT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/CXA / A0,A1,A2,A3,A4,A5
      FNDNF3=A0+(A1+(A2+(A3+A4*TT)*TT)*TT)*TT + A5*PP
      RETURN
      END
      BLOCK DATA BLDNF3
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/CXA / A0,A1,A2,A3,A4,A5
      DATA A0,A1,A2,A3,A4,A5
     1  /9.9980D+02, 2.9782D-02,-6.5360D-03,
     2  2.8300D-05,-9.1150D-08, 4.697D-07/
      END
```

FNDNF4 - Function subroutine to relate the fluid density to pressure and temperature. Linear function for temperature.

```
      FUNCTION FNDNF4(PP,TT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/COEFF1/ POR,CF,CR,CTF,DF,DR
      COMMON/CXA    / A0,A1,A2,A3,A4,A5
      COMMON/HFREF1/ PREF,TREF,HCREF,PRMREF,
     1               POREF,DNFREF,VSCREF
      FNDNF4=DNFREF*(1.0 - CTF*(TT-TREF))
      RETURN
      END
```

FNDNF5 - Function subroutine to relate the fluid density to pressure and temperature. Exponential function of temperature (Used for Hydrocoin Level 2, Case 1). T is temperature and T0 is ambient rock temperature

```
      FUNCTION FNDNF5( P, T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/HFREF1/ PREF,  T0,HCREF,PRMREF,
     1               POREF,DNFREF,VSCREF
      DATA  DNFG /315.5/, TG/647.3/
      DATA  A1,A2,A3,A4,A5 /2.0233201, -0.49864401,
     1                -1.0282498, 0.9465529, -0.30178144/
      TT=T
      IF(TT.GT.350.) THEN
      WRITE(6,910) TT
  910 FORMAT(6X,'FNDNF5---T=',F7.2,' > 350 Degrees')
      TT = 350.
      ENDIF
      YG= (TG/ (TT + T0 + 273.3) - 1.0)**0.333333333333
      FNDNF5=DNFG*EXP(YG*(A1+YG*(A2+YG*(A3
     1                           +YG*(A4+YG*A5)))))
      RETURN
      END
```

FNDNF6 - Function subroutine to relate the fluid density to pressure and temperature.

```
      FUNCTION FNDNF6( P, T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/CXA / A0,A1,A2,A3,A4,A5
      TT=T
      IF(TT.GT.200.) THEN
      WRITE(6,910) TT
  910   FORMAT(6X,'FNDNF6---T=',F7.2,' > 200. Degrees')
      ENDIF
      FNDNF6=A0+(A1+(A2+(A3+A4*TT)*TT)*TT)*TT
      RETURN
      END
```

FUNVSF - Generic function subroutine to relate the dynamic viscosity of the fluid to the temperature of the fluid.

FNVSF1 - Function subroutine to relate the dynamic viscosity of the fluid to the temperature of the fluid.

```
      FUNCTION FNVSF1(T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/HFREF1/ PREF,TREF,HCREF,PRMREF,
     1               POREF,DNFREF,VSCREF
      FNVSF1=VSCREF
      RETURN
      ENTRY    FUNVSC(T)
      FUNVSC = VSCREF
      RETURN
      END
```

FNVSF2 - Function subroutine to relate the dynamic viscosity of the fluid to the temperature of the fluid.

```
      FUNCTION FNVSF2( T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      TT=T
      IF(TT.GT.200.) THEN
      WRITE(6,910) TT
  910 FORMAT(6X,'FNVSF2---T=',F7.2,' > 200.  Degrees')
      ENDIF
      FNVSF2=0.5858327123*(42.9 + TT)**(-1.538389244)
      RETURN
      END
```

FNVSF5 - Function subroutine to relate the dynamic viscosity of the fluid to the temperature of the fluid. Exponential function of temperature (Used for Hydrocoin Level 2, Case 1).
T is temperature and T0 is ambient rock temperature

```
      FUNCTION FNVSF5( T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/HFREF1/ PREF,  T0,HCREF,PRMREF,
     1               POREF,DNFREF,VSCREF
      DATA XMYR /39.06E-6/, TMY/647.27/
      DATA B1,B2,B3,B4,B5 /1.5537, -0.20276, 1.9107,
     1                     -0.63486, 0.0050468/
      TT=T
      IF(TT.GT.350.) THEN
      WRITE(6,910) TT
  910 FORMAT(6X,'FNVSF5---T=',F7.2,' > 350 degrees')
      ENDIF
      XMYG= (TMY/(TT + T0 + 273.3) - 1.0)**0.333333333333
      FNVSF5=XMYR*EXP(XMYG*B1+XMYG**3*(B2+XMYG**3*(B3
     1                          +XMYG*B4+XYG**24*B5)))
      RETURN
      END
```

FHCSET - Function subroutine to set the permeability.

```
      FUNCTION FHCSET(I,K)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/HFREF1/ PREF,TREF,HCREF,PRMREF,
                     POREF,DNFREF,VSCREF
      COMMON/ANISOT/ FHCX,FHCY,FHCZ
      GOTO (10,20,30),K
      WRITE(6,910) K
  910 FORMAT(/6X,'FUNHC1---Error:  K=',I5,'---Stop')
      STOP
   10 FHCSET=PRMREF*FHCX
      RETURN
   20 FHCSET=PRMREF*FHCY
      RETURN
   30 FHCSET=PRMREF*FHCZ
      RETURN
      END
```

FNHC1 - Function subroutine to set the permeability. Currently, upon each call this function will return the reference value given for the permeability. Used by the following subroutines: HFSUB1

```
      FUNCTION FNHC1(IEL,DEPTH)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/COEFF8/ CPOR,POR1,CPERM,PERM1,FRPERM
      FNHC1=1.0D0
      RETURN
      ENTRY    FUNHC2(IEL,DEPTH)
      FUNHC2=1.0
      RETURN
      END
```

FNHC3 - Function subroutine to set the permeability. Currently, upon each call this function will return the reference value given for the permeability.

```
      FUNCTION FNHC3(IEL,DEPTH)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/COEFF8/ CPOR,POR1,CPERM,PERM1,FRPERM
      COMMON/IMDPTH/ IMDPTH
      COMMON/XZERO / XF1,XF2,YF1,YF2,ZF1,ZF2
      LOGICAL L1,L2
      DATA L1/.TRUE./, L2/.TRUE./
      IF(DEPTH.LT.YF1 .OR.  DEPTH.GT.YF2) THEN
      FNHC3=1.0
      IF(L2) WRITE(6,910) IEL,IMDPTH,DEPTH,FNHC3,FRPERM
      L2=.FALSE.
      ELSE
      FNHC3=FRPERM
      IF(L1) WRITE(6,910) IEL,IMDPTH,DEPTH,FNHC3,FRPERM
  910 FORMAT(/6X,'FNHC3---IEL=',I4,' IMDPTH=',I2,'
     1          DEPTH=',F8.3/ 6X,9X,9X,'FNHC3=',1P,E12.4,'
     2          FRPERM=',1P,E12.4)
      L1=.FALSE.
      ENDIF
```

```
      RETURN
      END
      ------------------------------------------------------
```

FNHC2 - Function subroutine to relate the permeability to the depth below the ground surface.

```
      HFNFLX,VELXY1,HFLXNU,VELXY3
      FFMBL1,TFEBL1
      FFMBL3,TFEBL3
      FPERM2
      ------------------------------------------------------
      FUNCTION FNHC2(IEL,DEPTH)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/COEFF8/ CPOR,POR1,CPERM,PERM1,FRPERM
      IF(LFRAC1(IEL).NE.1) THEN
      FNHC2=EXP(CPERM*DEPTH)
      ELSE
      FNHC2=FRPERM
      ENDIF
      RETURN
      END
      ------------------------------------------------------
```

FPOR2 - Generic function subroutine to relate porosity to the the depth below the ground surface of an element.

FNPOR1 - Function subroutine to relate porosity to the the depth below the ground surface of an element.

```
      ------------------------------------------------------
      FUNCTION FNPOR1(IEL,XYZ,NPD)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION XYZ(3,8)
      COMMON/COEFF8/ CPOR,POR1,CPERM,PERM1,FRPERM
      COMMON/FECOM4/ NNODE,NDOF,NEK,NDIM
      FNPOR1=POR1
      RETURN
      ENTRY FPOR2(IEL,XYZ,NPD)
      FPOR2=POR1
      RETURN
      END
      ------------------------------------------------------
```

FNPOR2 - Function subroutine to relate porosity to the the depth below the ground surface of an element.

```
      ------------------------------------------------------
      FUNCTION FNPOR2(IEL,XYZ,NPD)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION XYZ(3,8)
      COMMON/COEFF8/ CPOR,POR1,CPERM,PERM1,FRPERM
      COMMON/FECOM4/ NNODE,NDOF,NEK,NDIM
      IF(LFRAC1(IEL).NE.1) THEN
        SUM=0.
        DO 20 J=1,NPD
20      SUM=SUM + XYZ(NDIM,J)
        DEPTH=SUM/NPD
        FNPOR2=POR1*EXP(CPOR*DEPTH)
      ELSE
        FNPOR2=POR1
      ENDIF
```

```
      RETURN
      END
      ------------------------------------------------------
```

FNPOR3 - Function subroutine to relate porosity to the the depth below the ground surface of an element.

```
      ------------------------------------------------------
      FUNCTION FNPOR3(IEL,XYZ,NPD)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION XYZ(3,8)
      COMMON/COEFF8/ CPOR,POR1,CPERM,PERM1,FRPERM
      COMMON/FECOM4/ NNODE,NDOF,NEK,NDIM
      COMMON/XZERO / XF1,XF2,YF1,YF2,ZF1,ZF2
      LOGICAL L1,L2
      DATA L1/.TRUE./, L2/.TRUE./
      DEPTH=FDPTH2(XYZ)
      IF(DEPTH.LT.YF1 .OR.  DEPTH.GT.YF2) THEN
        FNPOR3=POR1
        IF(L2) WRITE(6,910) FNPOR3,IEL,DEPTH
        L2=.FALSE.
      ELSE
        FNPOR3=POR1*CPOR
        IF(L1) WRITE(6,910) FNPOR3,IEL,DEPTH
  910   FORMAT(/6X,'FNPOR3---Now set porosity=',F7.5,
     1          ' in element=',I4,' DEPTH=',F8.4)
        L1=.FALSE.
      ENDIF
      RETURN
      END
      ------------------------------------------------------
```

FPERM2 - Subroutine to determine the permeabilities at the nodes of an element by multiplying the element permeability by a factor corresponding to the depth of respective node.

```
      ------------------------------------------------------
      SUBROUTINE FPERM2(IEL,XYZ,HCN)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION XYZ(3,8),HCN(3,8)
      COMMON/FECOM4/ NNODE,NDOF,NEK,NDIM
      COMMON/IMDPTH/ IMDPTH
      GOTO 5
      ENTRY FPERM3(FUNHC2)
      WRITE(6,920)
  920 FORMAT(/6X,'FPERM3---Now set function:
     1                                FUNHC2 - in FPERM2')
      RETURN
    5 CONTINUE
      IF(IMDPTH.NE.1) THEN
       DO 10 J=1,NNODE
       PERMC=FUNHC2(IEL,XYZ(NDIM,J))
       HCN(1,J)=HCN(1,J)*PERMC
       HCN(2,J)=HCN(2,J)*PERMC
       HCN(3,J)=HCN(3,J)*PERMC
   10 CONTINUE
        ELSE
      DEPTH=FDPTGH2(XYZ)
      PERMC=FUNHC2(IEL,DEPTH)
      DO 30 J=1,NNODE
```

```
      HCN(1,J)=HCN(1,J)*PERMC
      HCN(2,J)=HCN(2,J)*PERMC
      HCN(3,J)=HCN(3,J)*PERMC
   30 CONTINUE
      ENDIF
      RETURN
      END
      ------------------------------------------------------
```

FDPTH2 - Function to compute the depth of the centroid of an element. Used by the following subroutines: HFNFLX,VELXY1

```
      ------------------------------------------------------
      FUNCTION FDPTH2(XYZ)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION XYZ(3,8)
      COMMON/ELEMNO/ IEL
      COMMON/FECOM4/ NNODE,NDOF,NEK,NDIM
      IF(NNODE.GT.1) THEN
      SUM=0.0
      DO 10 J=1,NNODE
   10  SUM=SUM + XYZ(NDIM,J)
       FDPTH2=SUM/NNODE
      ELSE
      WRITE(6,910) IEL,NNODE,NDOF,NDIM
  910 FORMAT(/6X,'FDPTH2---IEL=',I4,' NNODE=',I4,' NDOF=',I1,
     1                    ' NDIM=',I2)
      WRITE(6,920) (XYZ(1,J),XYZ(2,J),J=1,3)
  920 FORMAT(/6X,'FDPTH2---XYZ: ',6F7.1)
      WRITE(6,930)
  930 FORMAT(6X,9X,'Execution terminated')
      STOP
      ENDIF
      RETURN
      END
      ------------------------------------------------------
```

WDCAY1 - Subroutine to apply a given function (DECAY) for the radioactive decay.

```
      ------------------------------------------------------
      SUBROUTINE WDCAY1(WHS,NS)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION WHS(NS)
      COMMON/CNTRL3/ NSTEP,ITER,TIME,DT,PHERMX,TFERMX,TRERMX
      COMMON/TIME1 / TIME1
      CHARACTER*6 DCAY
      LOGICAL L1,L2
      DATA L1/.TRUE./, L2/.TRUE./
      ENTRY WDECAY(WHS,NS)
      GOTO 10
      ENTRY WDCAY2(DCAY,DECAY)
      WRITE(6,910) DCAY
  910 FORMAT(/6X,'WDCAY2---Now specified
     1                          the decay function: ',A6)
      RETURN
      10 CONTINUE
      IF(L1) WRITE(6,901) DCAY,NS
  901 FORMAT(6X,'WDCAY1---Now entered
     1                             - DCAY=',A6,' NS=',I2)
```

```
      L1=.FALSE.
      IF(NS.GT.0) THEN
      WFAC=DECAY(TIME,DT,TIME1)
 60    DO 80 J=1,NS
       WHS(J)=WHS(J)*WFAC
 80    CONTINUE
        IF(L2) WRITE(6,920) TIME,TIME1,WFAC
920    FORMAT(/6X,'WDCAY1---TIME=',1P,E10.3,'
     1         TIME1=',1P,E10.3, ' WFAC=',1P,E10.3)
      L2=.FALSE.
      ENDIF
      RETURN
      END
      ------------------------------------------------------
```

WDCAY2 - Entry to *WDCAY1* .

DECAY1 - Function to interpolate the radioactive decay of a heat source.

```
      ------------------------------------------------------
      FUNCTION DECAY1(TIME,DT,TIME1)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      LOGICAL L1
      DATA L1/.TRUE./
      DECAY1=1.0
      IF(L1) WRITE(6,920) TIME,TIME1,DECAY1
920   FORMAT(/6X,'DECAY1---TIME=',1P,E10.3,'
     1            'TIME1=',1P,E10.3, 'DECAY1=',1P,E10.3)
      L1=.FALSE.
      RETURN
      ENTRY DECAY (TIME,DT,TIME1)
      DECAY=1.0
      RETURN
      END
      ------------------------------------------------------
```

DECAY2 - Function to interpolate the radioactive decay of a heat source. Exponential decay of two nuclides.

```
      ------------------------------------------------------
      FUNCTION DECAY2(TIME,DT,TIME1)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/HFREF3/ A1,AL1,A2,AL2
      LOGICAL L1
      DATA L1/.TRUE./
      C1=TIME-0.5*DT
      IF(TIME.GT.TIME1+0.001) GOTO 40
      IF(TIME1.LE.0.) GOTO 20
      DECAY2=C1/TIME1
      GOTO 60
 20   WRITE(6,910) TIME,TIME1
910   FORMAT(/6X,'DECAY2---Error:TIME=',1P,E10.3,'
     1                  TIME1=',1P,E10.3, 'Stop')
      STOP
 40   CONTINUE
      DECAY2=A1*EXP(-AL1*C1) + A2*EXP(-AL2*C1)
 60   CONTINUE
      IF(L1) WRITE(6,920) TIME,TIME1,DECAY2
920   FORMAT( 6X,'DECAY2---TIME=',1P,E10.3,'
     1            TIME1=',1P,E10.3,' DECAY2=',1P,E10.3)
      L1=.FALSE.
```

```
      RETURN
      END
      ------------------------------------------------------
```

DECAY3 - Function for time dependent energy output for Hydrocoin Level 2, Case 1.

```
      ------------------------------------------------------
      FUNCTION DECAY3(TIME,DT,TIME1)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      LOGICAL L1
      DATA L1/.TRUE./
      CALL SHWTOT(TIME,HWTOT)
      DECAY3 = HWTOT
      IF(L1) WRITE(6,920) TIME,TIME1,DECAY3
  920 FORMAT( 6X,'DECAY3---TIME=',1P,E10.3,'
     1                TIME1=',1P,E10.3,' DECAY3=',1P,E10.3)
      L1=.FALSE.
      RETURN
      END
      ------------------------------------------------------
```

CHKSTP - Dummy subroutine to control the time step scheme (SETSTP)

```
      ------------------------------------------------------
      SUBROUTINE CHKSTP
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/CNTRL2/ MXSTEP,MXITER,DTFAC,PHERTL,
     1               TFERTL,TRERTL
      COMMON/CNTRL3/ NSTEP,ITER,TIME,DT,PHERMX,
     1               THERMX,TRERMX
      WRITE(6,910) NSTEP,TIME,DT
  910 FORMAT(/6X,'CHKSTP---NSTEP=',I2,'
      TIME=',1P,E12.4,
     1          ' DT=',E12.4)
      RETURN
      ENTRY SETSTP(INUN)
      WRITE(6,920) INUN
  920 FORMAT(6X,'SETSTP(CHKSTP)---Dummy
     1                          function entry - INUN=',I2)
      RETURN
      END
      ------------------------------------------------------
```

CHKST1 - Subroutine to control the time step scheme (SETST1) Logarithmic time stepping.

```
      ------------------------------------------------------
      Description of parameters:
      DTIME(1) - Initial subinterval
      NCYCLE - Number of cycles
      NFAC - Number of subintervals within each cycle
      FAC - Array of length -NFAC- containing factors for
            subintervals
      Example of input:
           DTIME(1)=3.15576e7, NCYCLE=4, NFAC=7
           FAC(1)=0.5, FAC(2)=0.5, FAC(3)=1.0,
           FAC(4)=1.0, FAC(5)=2.0, FAC(6)=2.0,
           FAC(7)=2.0
       Output:
           DTIME(2)=1.58e7, DTIME(3)=1.58e7,
```

```
      DTIME(4)=3.16e7, DTIME(5)=3.16e7,
      DTIME(6)=6.31e7, DTIME(7)=6.31e7,
      DTIME(8)=6.31e7, DTIME(9)=1.58e8,
      DTIME(10)=1.58e8, DTIME(11)=3.16e8,
      etc.
      Note: the subintervals were here multiplied
            with 365.25*86400 to get the times
            in seconds

      SUBROUTINE CHKST1
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/CNTRL2/ MXSTEP,MXITER,DTFAC,
     1               PHERTL,TFERTL,TRERTL
      COMMON/CNTRL3/ NSTEP,ITER,TIME,DT,
     1               PHERMX,TFERMX,TRERMX
      COMMON/DTIMEP/ DTIMEP(101),MXNT,NT
      DIMENSION FAC(10)
      DTFAC=1.0
      IF(NSTEP.GT.NT) THEN
      WRITE(6,905) NSTEP,NT
  905 FORMAT(/6X,'CHKST1---Warning:NSTEP=',I3,' > NT=',I3)
      DT=DTIMEP(NT)
      ELSE
      DT=DTIMEP(NSTEP)
      ENDIF
      WRITE(6,910) NSTEP,TIME,DT,DTFAC
  910 FORMAT(/6X,'CHKST1---NSTEP=',I2,'
     1   TIME=',1P,E12.4, 'DT=',1P,E12.4,' DTFAC=',0P,F7.3)
      RETURN
      ENTRY SETST1(INUN)
      WRITE(6,915) INUN
  915 FORMAT(/6X,'SETST1 (Entry to  CHKST1*) - INUN=',I2)
      READ(INUN,820)
     1              DTIMEP(1),NCYCLE,NFAC,(FAC(I),I=1,NFAC)
  820 FORMAT(F10.0,2I5,10F5.0)
      WRITE(6,920) DTIMEP(1),NCYCLE,NFAC,(FAC(I),I=1,NFAC)
  920 FORMAT(/6X,'SETST1---DTIMEP(1)=',1P,E12.4/
     1        6X,9X,'NCYCLE   =',I12/
     2        6X,9X,'NFAC     =',I12/
     3        6X,'FAC:',1P,5E12.4/(6X,4X,5E12.4))
      MXNT=101
      NT=NCYCLE*NFAC + 1
      IF(NT.GT.MXNT) THEN
   80 WRITE(6,960) NT,MXNT
  960 FORMAT(/6X,'SETST1---Error: NT=',I2,'
     1                              > MXNT=',I2,' Stop')
      ENDIF
      POWER=365.25*86400.
      DTIMEP(1)=DTIMEP(1)*POWER
      DO 60 K=1,NCYCLE
        I=(K-1)*NFAC+1
        DO 50 J=1,NFAC
          NT=I+J
          IF(I+J.GT.MXNT) GOTO 70
   50   DTIMEP(I+J)=FAC(J)*POWER
   60 POWER=10.*POWER
   70 WRITE(6,940) NT
  940 FORMAT(/6X,'SETST1---NT=',I2,'
     1                    LOGARITHMIC TIME INCREMENTS')
```

```
      WRITE(6,950) (DTIMEP(IT),IT=1,NT)
  950 FORMAT(6X,1P,4E15.5)
      RETURN
      END
      ------------------------------------------------------
CHKST2  - Subroutine to control the time step scheme
      ------------------------------------------------------
      SUBROUTINE CHKST2
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/CNTRL2/ MXSTEP,MXITER,DTFAC,
     1               PHERTL,TFERTL,TRERTL
      COMMON/CNTRL3/ NSTEP,ITER,TIME,DT,
     1               PHERMX,TFERMX,TRERMX
      COMMON/DTIMEP/ DTIMEP(101),MXNT,NT
      DTFAC=1.0
      IF(NSTEP.GT.NT) THEN
      WRITE(6,905) NSTEP,NT
  905 FORMAT(/6X,'CHKST2---Warning:
     1                    NSTEP=',I3,' > NT=',I3)
      DT=DTIMEP(NT)
        ELSE
      DT=DTIMEP(NSTEP)
      ENDIF
      WRITE(6,910) NSTEP,TIME,DT,DTFAC
  910 FORMAT(/6X,'CHKST2---NSTEP=',I2,' TIME=',1P,E12.4,
     1           ' DT=',1P,E12.4,' DTFAC=',0P,F7.3)
      RETURN
      ------------------
      ENTRY SETST2(INUN)
      ------------------
      WRITE(6,915) INUN
  915 FORMAT(/6X,'SETST2 (Entry to  CHKST2*) - INUN=',I2)
      READ(INUN,820) DTIMEP(1),NCYCLE,NSUB
  820 FORMAT(F10.0,2I5)
      IF(DTIMEP(1).EQ.0.0) DTIMEP(1)=1.0E-1
      IF(NCYCLE.EQ.0) NCYCLE=5
      IF(NSUB.EQ.0) NSUB=1
      WRITE(6,920) DTIMEP(1),NCYCLE,NSUB
  920 FORMAT(/6X,'SETST2---DTIMEP(1)=',1P,E12.4/
     1        6X,9X,'NCYCLE   =',I12/
     2        6X,9X,'NSUB     =',I12)
      MXNT=101
      NT=4*NCYCLE + 1
      IF(NT.GT.MXNT) GOTO 80
      I=1
      POWER=DTIMEP(1)
   60 DTIMEP(I+1)=1.*POWER
      DTIMEP(I+2)=2.*POWER
      DTIMEP(I+3)=3.*POWER
      DTIMEP(I+4)=3.*POWER
      J=I+4
      IF(J.EQ.NT) GOTO 70
      I=I+4
      POWER=10.*POWER
      GOTO 60
   70 WRITE(6,940) NT
  940 FORMAT(/6X,'SETST2---NT=',I2,'
     1                    Logarithmic time increments')
```

```
      WRITE(6,950) (DTIMEP(IT),IT=1,NT)
  950 FORMAT(6X,1P,4E15.5)
      IF(NSUB.GT.1) THEN
      MXK=NSUB*NT
      DO 76 I=NT,1,-1
       DTIMEP(I)=DTIMEP(I)/REAL(NSUB)
       DO 75 J=1,NSUB
        K=NSUB*(I-1) + J
        IF(K.GT.MXNT) THEN
         MXK=MXNT
      GOTO 76
        ELSE
          DTIMEP(K)=DTIMEP(I)
        ENDIF
   75   CONTINUE
   76   CONTINUE
      NT=MXK
      WRITE(6,*) ' SETST2---NT=',NT
      WRITE(6,950) (DTIMEP(IT),IT=1,NT)
      ENDIF
      RETURN
   80 WRITE(6,960) NT,MXNT
  960 FORMAT(/6X,'SETST2---Error: NT=',I2,' >
     1                              MXNT=',I2,' Stop')
      STOP
      END
      ------------------------------------------------------
CHKST3    - Subroutine to control the time step scheme
            (SETST3)
      ------------------------------------------------------
      SUBROUTINE CHKST3
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/CNTRL2/ MXSTEP,MXITER,DTFAC,
     1               PHERTL,TFERTL,TRERTL
      COMMON/CNTRL3/ NSTEP,ITER,TIME,DT,
     1               PHERMX,TFERMX,TRERMX
      DATA DTFAC1/1.2/, DTFAC2/1.5/, ISTEP/7/
      WRITE(6,910) NSTEP,TIME,DT,DTFAC
  910 FORMAT(/6X,'CHKST3---NSTEP=',I2,' TIME=',1P,E12.4,
     1                ' DT=',1P,E12.4,' DTFAC=',0P,F7.3)
      IF(NSTEP-ISTEP) 10,20,30
   10 DTFAC=DTFAC1
      GOTO 40
   20 WRITE(6,920) DTFAC1,DTFAC2
  920 FORMAT(6X,'CHKST3---Now changing
     1  the time step multiplier','from',F6.3,' into',F6.3)
   30 DTFAC=DTFAC2
   40 WRITE(6,930) DTFAC
  930 FORMAT(6X,'CHKST3---Time step control---DTFAC=',F5.3)
      RETURN
      ------------------
      ENTRY SETST3(INUN)
      ------------------
      WRITE(6,915) INUN
  915 FORMAT(/6X,'SETST3 (Entry to  CHKST3*) - INUN=',I2)
      RETURN
      END
      ------------------------------------------------------
```

CHKST4 - Subroutine to control the time step scheme (SETST4) All time steps specified in input.

```
      ------------------------------------------------------
      SUBROUTINE CHKST4
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*20 DTIMES
      COMMON/CNTRL2/ MXSTEP,MXITER,DTFAC,PHERTL,TFERTL,TRERTL
      COMMON/CNTRL3/ NSTEP,ITER,TIME,DT,PHERMX,TFERMX,TRERMX
      COMMON/DTIMEP/ DTIMEP(101),MXNT,NT
      DTFAC=1.0
      IF(NSTEP.GT.NT) THEN
      WRITE(6,905) NSTEP,NT
  905 FORMAT(/6X,'CHKST4---Warning: NSTEP=',I3,' > NT=',I3)
      DT=DTIMEP(NT)
        ELSE
      DT=DTIMEP(NSTEP)
      ENDIF
      WRITE(6,910) NSTEP,TIME,DT,DTFAC
  910 FORMAT(/6X,'CHKST4---NSTEP=',I2,' TIME=',1P,E12.4,
     1          ' DT=',1P,E12.4,' DTFAC=',0P,F7.3)
      RETURN
      ------------------
      ENTRY SETST4(INUN)
      --------------------
      WRITE(6,915) INUN
  915 FORMAT(/6X,'SETST4 (Entry to  CHKST4*) - INUN=',I2)
      READ(INUN,816) DTIMES
  816 FORMAT(A20)
      WRITE(6,918) DTIMES
  918 FORMAT(/6X,'SETST4---',A20)
      MXNT = 101
      READ(INUN,820) NT,TFAC
  820 FORMAT(I5,F15.0)
      IF(NT.GT.MXNT) THEN
       WRITE(6,920) NT,MXNT
  920  FORMAT(6X,9X,' NT=',I3,' > MXNT=',I3,'Stop')
      STOP
      ENDIF
      IF(TFAC.EQ.0.0) TFAC=365.25*86400.
      READ(INUN,*,END=20) (DTIMEP(I),I=1,NT)
      WRITE(6,925) NT,TFAC,(DTIMEP(I),I=1,NT)
  925 FORMAT(6X,'SETST4---NT=',I3,
     1          ' Conversion factor (TFAC)=',1P,E13.6,
     2          ' TIMEP:'/(6X,1P,7E10.3))
      NT = NT-1
      DO 10 I=1,NT
   10  DTIMEP(I) = (DTIMEP(I+1) - DTIMEP(I))*TFAC
       WRITE(6,926) NT,(DTIMEP(I),I=1,NT)
  926 FORMAT(6X,'SETST4---NT=',I3,' DTIMEP:'/(6X,1P,7E10.3))
      RETURN
   20 WRITE(6,*) '     SETST4---End of file - INUN',INUN
      STOP
      END
      ------------------------------------------------------
```

2.3.4 Input/Output

2.3.4.1 Parameter data

HFINPT - Subroutine to input miscellaneous solution parameters, material properties etc.

TEXINU - Auxiliary subroutine to read and optionally print out header records for input of parameter data.

HFMSRC - Subroutine to input mass sources.

EXFUN1 - Subroutine for transfering of external function names for fluid density, dynamic viscosity, permeability, porosity and radioactive decay.

FDENF - Subroutine for transfering of selected function for fluid density.

FVISF - Subroutine for transfering of selected function for dynamic viscosity.

FVSG - Subroutine to transfering of function selected for dynamic viscosity of the gas to subroutines using gas viscosity.

FHC2 - Subroutine for transfering of selected function for permeability.

FPOR - Subroutine for transfering of selected function for porosity.

FDECAY - Subroutine for transfering of selected function for radioactive decay.

GAUSSP - Subroutine to set coordinates of sampling points for the integration formulae 3 by 3 (by 3) rule, or 4 by 4 (by 4) rule.

GAUSSX - Subroutine to check element type and set Gauss points accordingly.

HFHEAD - Subroutine to output headings, etc.

HFNDIN - Subroutine to input coordinates of nodal points and nodal data. This subroutine has the following additional entries:
HFNDUT - For the output of nodal data
HFELIN - For the input of element incidences
HFELUT - For the output of element incidences

HFNDPR - Subroutine to printout coordinates of nodal points.

HFELPR - Subroutine to printout the element incidences.

BLTYPX - Block data to initiallize the number of nodes per element , the number of degrees of freedom per nodes and the total number of degrees of freedom per element.

HFTEST - Subroutine for Ad Hoc setting of input parameters.

HFSET1 - Subroutine to initialize work arrays.

HFAX1A - Subroutine to transfer initial nodal pressure values previously being input to the arrays -HH- and -LBHH- in subroutine HFNDIN*.

HFAX2B - Entry (HFAX1A) to transfer pressure and fluid temperature data to the work arrays --HH- and -LBHH-.

HFWRK1 - Subroutine to store current values of pressure and temperatures (or alternatively gas pressure) into work arrays to be used during the next time step.

UTLBHH - Subroutine for the tabulation of nodal data.

HFOUT1 - Subroutine to printout work arrays. This subroutine has the following additional entries:
HFOUT2 - For the output of pressure arrays
HFOUT3 - For the output of temperature arrays

HFINFR - Subroutine to input list of fracture elements.

LFRAC1 - Function subroutine to check if a given element is included in the list of fracture elements as given to HFNFRC*.

SELEM1 - Subroutine to locate elements which constitutes either a vertical or horizontal fracture zone.

HFIELN - Auxiliary subroutine to redefine the matrix of the element incidences.

HFPERM - Subroutine to set the array for nodal permeabilities.

HFELHC - Subroutine to set the matrix for element permeabilities constant over each element.

HFELM1 - Subroutine to set element permeabilities according the value of IELMAT*.

HFELP1 - Sbroutine to set element porosity.

HFINP3 - Subroutine to input material properties for unsaturated conditions.

UNSPRT - Subroutine to printout unsaturated nodes.

BLINP3 - Block data to initialize printout parameter used in subroutine HFINP3*.

UNSIN1 - Subroutine to input pelationships for
Saturation- pressure
Saturation- relative permeability
Saturation- derivatives.

UNSUT1 - Alternative entry for storage of unsaturated data on dsic.

PSINP1 - Program to get hydraulic properties for unsaturated flow testing.

PSIFN1 - Function to get a capillary curve for testing.

HFUNS1 - Subroutine to find the hydraulic properties for unsaturated conditions.

HFINP4 - Subroutine with various options for setting the boundary conditions for some test examples.

MODFYX - Subroutine to modify an element mesh to account for the bore-hole radius.

PRMSE1 - Subroutine to set a linear variation in permeability for a given region.

HFELMP - Subroutine to specify material properties to be associated with elements unsaturated flow conditions saturetion-capillary pressure saturation-pelative permeability.

HFINP5 - Subroutine to input material properties of the rock.

HFINP6 - Subroutine to set boundary conditions for the solution of displacements for some test problems.

HFNDUV - Subroutine to read in solution values from permanent disc file. The file is read until "END OF FILE" is encountered.

HFNDU1 - Subroutine to read in solution values from a permanent disc file. The data for one time step is read and returned.

HFSAVD - Subroutine to write the solution values on a disc file for subsequent use.

HFNDUX - Subroutine for unformatted input of nodal values this subroutine reads data for one time step at each call upon the subroutine.

HFNDU2 - Alternative entry to output nodal data to a disc file.

HFBCX1 - Subroutine to input data of boundary conditions and/or initial values. The purpose of this subroutine is also to check which variable(s) to be taken into account.

HFBCX2 - Complementary subroutine to HFBCX1* to select subroutine to perform the action, as specified by the input to HFBCX1*

HFBCX3 - Complementary subroutine to HFBCX1* this subroutine is usded for displacements.

MROT2D - Subroutine to rotate an element mesh this subroutine rotates the mesh 90 degrees in the X-Y plane.

HFBN2U - Subroutine to prescribed displacements at either of the leftmost or rightmost boundaries.

PHYDRS - Program for integration of pressure under hydrostatic conditions.

HSITER - Subroutine to determine perssure under hydrostatic conditions with heat effects.

HFBCI1 - Subroutine to assign specified code and value to the set of nodes as specified by the input list.

HFBCI2 - Subroutine to assign specified code and value to nodes specified by the program itself.

HFBCI3 - Subroutine to assign a specified code and linear function values to the nodes as specified on the input list.

HFBCI4 - Subroutine to assign values according to a specified linear function to all nodal points.

HFBCI5 - Subroutine to assign nodal values to a given list of nodes.

MODFY1 - Subroutine to modify the nodal coordinates of the element mesh, e.g. to scale or tranform the original element mesh.

MTOP2D - Subroutine to modify the top boundary of the element mesh for a given unidirectional slope (2-D version).

MTOP3D - Subroutine to modify the top boundary of the element mesh for a given unidirectional slope (3-D version).

BND2DN - Subroutine to locate the top and bottom boundary nodes and to specify boundary conditions (2-D version).

BND3DN - Subroutine to locate the top and bottom boundary nodes and to specify boundary conditions (3-D version).

DMINMX - Auxiliary subroutine to find the maximum respectively minimum values of an array.

HFWAV1 - Subroutine for the calculation of weighted averages using the most recent iteration.

HFWAV2 - Alternative entry for the calculation of weighter averages using the most recent iteration values.

HFWAV3 - Subroutine for the calculation of weighted averages using the most recent iteration values.

HFWAV2 - Alternative entry for the calculation of weighted averages using the most recent iteration values.

SOLVUT - Subroutine for output of solution values. This subroutine calls in its turn the following subroutines:

UTSLV1 - For the output of one variable
UTSLV2 - For the output of two variables
UTSLV3 - For the output of three variables
UTSLV4 - For the output of four variables
UTSLV5 - For the output of five variables

UTSOLV - Subroutine for output of a single variable array. The present version of this subroutine is identical with *UTSLV1*

SOLVU2 - Subroutine to printout the displacement vectors at selected nodes.

UNSPRT - Subroutine to printout unsaturated nodes.

NDPRT2 - Subroutine for nodal printout. This version prints out coordinates, pressure, fluid velocities, temperature and heat soruces for a given set of nodes (2-D version)

NDPRT3 - Subroutine for nodal printout. This version prints out coordinates, pressure, fluid velocities, temperature and heat soruces for a given set of nodes (3-D version).

SELNDS - Subroutine to input nodes selected for printout by subroutines NDPRT2 or NDPRT3.

BLUTVL - Block data to initialize MXNPRT* and MXNHS*.

UTSLV1 - Subroutine to output solution values.

UTSLV2 - Subroutine to output values of two arrays.

UTSLV3 - Subroutine to output values of three arrays.

UTSLV4 - Subroutine to output values of four arrays.

UTSLV5 - Subroutine to output values of five arrays.

HFPRU1 - Subroutine to printout nodes with prescribed pressure.

2.3.4.2 Hydrostatic boundary

PHYDRO - Subroutine for the generation of hydrostatic boundary conditions.

PRITER - Complementary subroutine to *PHYDRO*.

2.3.4.3 Prescribed flux boundary

HFBFLW - Subroutine for the input of prescribed flux boundary conditions.

CHKIEL - Complementary subroutine to *HFBFLW* to find the element which contains four corner nodes defining one of the sides of the element.

CHKSD2 - Complementary subroutine to *HFBFLW* to indentify the element side which corresponds to four (global) nodes given as input to specify the element side under consideration (2-D quadri-lateral elements).

CHKSD3 - Complementary subroutine to *HFBFLW* to indentify the element side which corresponds to four (global) nodes given as input to specify the element side under consideration (3-D hexahedral elements).

CHKND2 - Complementary subroutine to *HFBFLW* to input the prescribed flux at a given element side (2-D quadri-lateral elements).

CHKND3 - Complementary subroutine to *HFBFLW* to input the prescribed flux at a given element side (3-D hexahedral elements).

FQSTOR - Complementary subroutine to *HFBFLX* to input the prescribed flux at an element side.

BLSORT - Complementary subroutine to sort the array of element numbers associated with elements containing prescribed flux boundaries.

INTFLX - Subroutine to compute the coefficients of an element at a Gauss point for an element subject to precribed flux.

IELCHK - Subroutine to check if a given element number is included among the list of elements subject to prescribed flux.

BLIELX - Block data to initialize local node number definition of element sides.

LSDEF1 - Subroutine to read or to print element side definitions.

LSDEF2 - Entry for printout of element side definitions.

2.3.5 Element matrix generation

2.3.5.1 Fluid flow

HFELF1 - Subroutine to compute and store the element matrix systems when solving for pressure. Matrix systems are reduced during assembly. Element matrices may optionally be stored on a disk file for subsequent use.

HFSEEP - Subroutine to check if current node is a seepage node and to set the boundary conditions accordingly.

NODSLC - Subroutine to locate nodes with in a region spcified as:
X1 < X < X2
Y1 < Y < Y2
Z1 < Z < Z2

NODLOC - Subroutine to locate element associated with given nodes

NODLC2 - Subroutine to find the last appearance in the element incidences for a given set of numbers.

NODSL2 - Subroutine to locate nodes.

NODSL3 - Subroutine to indentify elements and their related nodes for given region specifications.

NODSL4 - Subroutine to indentify elements and their related node for given region specifications.

PREM2D - Preprocessing subroutine for graphical display of an element mesh (in 2-D or 3-D).

IBASRT - Subroutine to sort two integer arrays into mutually ascending order.

ABSORT - Subroutine to sort two real arrays into mutually ascending order.

GFWTBL - Subroutine to locate the position of the water table assumed to be at zero pressure.

HFL2D1 - Subroutine to form the basic types of integraton matrices as required to solve the pressure equation. This subroutine administers the sampling points for the integration formulae.

HFL2DA - Subroutine to form the basic types of integraton matrices as required to solve the pressure equation. This subroutine administers the sampling points for the integration formulae.

HFL2D5 - Subroutine to form the basic types of integraton matrices as required to solve the pressure equation. This subroutine administers the sampling points for the integration formula (This version is for sxi-symmetric flow).

HFUNS1 - Subroutine to check unsaturated conditions.

HFUNS2 - Subroutine to find saturation corresponding to current pressure.

HFUNS3 - Subroutine to find the derivative of saturation versus pressure.

HFUNS4 - Subroutine to find relative permeability corresponding to current pressure.

LOCINT - Subroutine to perform linear interpolation in capillary-saturation and rel. permeability-saturation functions.

HFLGA1 - Subroutine to evaluate basis functions and Jacobians at the sampling points for the quadrature formulae.

RESID1 - Subroutine to compute the residual vector for Newton-Raphson (N-R) iteration.

PHRES1 - Subroutine to apply N-R corrections.

HFMAUX - Subroutine to transfer the effective matrix system to the work storage.

IELTST - Subroutine to check if current element should be supplied with printout messages.

HFBF2D - Subroutine to perform integration of prescribed boundary flux for 2-d element surfaces.

HFBF3D - Subroutine to perform integration of prescribed boundary flux for 3-d element surfaces.

HFMPH1 - Subroutine to setup matrix systems for each element (fluid flow equation).

HFMPH2 - Subroutine to apply finite differencing of the time derivatives to the element matrix systems (fluid flow equation).

HFLGAW - subroutine for calculation of integration matrices at Gauss point level.

HFLBLK - Subroutine to form integration matrices for a fractured medium.

HFLGAB - Computes basis function values at Gaussian quadrature points to form integration matrices block elements.

HFMPB1 - Compute the element contributions for the global matrix system.

HFMPB2 - Subroutine to rearrange the element matrix system into an effective matrix system.

HFL2DU - Forms basic integration matrices.

HFLGAU - Computes basic function values at Gaussian quadrature points to form integration matrices.

HFMPU1 - Compute the element contributions for the global matrix system.

HFMPU2 - Subroutine to rearrange the element matrix system into an effective matrix system.

AMOUT1 - Subroutine to printout the work matrix during the element assembly.

2.3.5.2 Heat flow

HFTMS1 - Subroutine to govern heat flow calculation.

HFSRCX - Subroutine to input distributed and concentrated heat sources.

HEATFN - Subroutine to apply a smooth transition for a distributed heat source.

HFSRC1 - Subroutine to input point or distributed heat sources.

HFSRC3 - Auxiliary subroutine to apply point heat sources associated with current element to the element matrix system.

HFSRC4 - Subroutine to evaluate heat sources associated with mass sources.

BLSRC4 - Block data to initialize some counter variables for heat sources associated with the mass sources.

SHWTIN - Subroutine to read time values (in days) and the corresponding total heat source values for Hydrocoin Level 2 Case 1.

SHWTOT - Subroutine for table look-up of the energy output corresponding to a given time value in days.

HFELT1 - Subroutine to compute and store the element matrix systems when solving for fluid and rock heat flow.

HFL2DW - Subroutine to form the basic types of integraton matrices as required to solve the pressure equation. this subroutine administers the sampling points for the integration formulae.

HFINTG - Heat flow integration at Gauss point level.

HFL2DT - Subroutine to update integration matrices.

HFL2D6 - Subroutine to form the basic types of integraton matrices as required to solve the pressure equation. this subroutine administers the sampling points for the integration formulae.

HFLGA2 - Subroutine to pick up basis function values and Jacobians associated with the sampling points.

HFLGZ1 - Subroutine to zero element matrices.

HFLGZ2 - Alternative entry to zero work array.

HFELU1 - Subroutine to prepare the input of element matrices to the band solution programs.

HFMTF1 - Subroutine to setup matrix systems for each element (fluid temperature equation).

HFMTF2 - Subroutine to apply finite differencing of the time derivatives to the element matrix systems (fluid temperature equation).

HFMTR1 - Subroutine to setup matrix systems for each element (rock temperature equation).

HFMTR2 - Subroutine to apply finite differencing of the time derivatives to the element matrix systems (rock temperature equation).

HEXA21 - Subroutine to compute values of the basis functions and their derivatives for a (8-20 nodes) curvilinear isoparametric element.

HFSRC2 - Auxiliary subroutine to check current element comprises any heat sources.

HFLMTM - Subroutine to form the system of simultaneous element matrices for the input to the frontal solutions programs.

HTELMA - Subroutine to form a simultaneous matrix system for an element.

HFWFUN - Subroutine to evaluate the heat transfer function for each nodal point.

HFWUPD - Alternative entry to update the temperature records.

HFTFUT - Alternative entry to output records of the fluid temperature.

HFWSET - Subroutine to zero fluid temperature records.

HFWFN1 - Subroutine to evaluate the heat transfer function for a node at a given time step.

TFPRE1 - Subroutine for simple prediction of the fluid temperature at current time level.

BLMAIN - Block data to initialize logical file numbers.

HFWCA1 - Subroutine to set array of heat transfer function values.

INTEG1 - Subroutine for the calculation of conductive element matrices.

INTEG2 - Subroutine for the calculation of element matrices associated with time derivatives, distributed heat sources, etc.Coefficients are assumed to be constent.

INTEG3 - Subroutine for calculation of element matrix associated with time derivatives, etc. The density of the fluid is varying over the element and is represented by functional coefficients.

INTEG4 - Subroutine for the calculation of convective matrices with fluid density and the velocity being represented by functional coefficients.

INTEG5 - Subroutine for the calculation of conductive element matrices with the fluid density represented by functional coefficients.

INTEG6 - Subroutine for the calculation of conductive element matrices with the fluid density and the conductivity being represented by functional coefficients. Off diagonal components of the conductivity tensor are assumed to be zero.

INTEG7 - Subroutine for the calculation of convective matrices with the fluid density, conductivity and pressure gradients being represented by functional coefficients. This subroutine computes velocities by Darcy's law.

INTEG8 - Subroutine for the calculation of the element matrices associated with the gravity term in Darcy's law with the fluid density being represented by functional coefficients.

INTEGU - Subroutine for the calculation of the element matrices associated with the derivatives for unsaturated flux. This version performs point integration at the nodal points.

INTG1X - Subroutine for calculation at a Gauss point the element matrix resulting from the flow divergence term. Fluid density and permeability are assumed to be constant over the element.
(INTEG1)

INTG4X - Subroutine for calculation of convective matrix. Fluid density and velocity represented by functional coefficients.

INTG5X - Subroutine for calculation at a Gauss point the element matrix resulting from the flow divergence term. Fluid density is varying over the element and is represented by functional coefficients - off diagonal components of conductivity tensor assumed to be zero.
(INTEG5)

INTG6X - Subroutine for calculation at a Gauss point the element matrix resulting from the flow divergence term. Fluid density and permeability are assumed to be varying over the element. Density and permeability represented by functional coefficients - off diagonal components of conductivity tensor assumed to be zero.

INTG7X - Subroutine for calculation of convective matrix.
Fluid density, conductivity and pressure gradients represented by functional coefficients.

INTG8X - Subroutine for calculation of element matrix associated with the gravity term in Darcy's law. Fluid density represented by functional coefficients.

INTEG9 - Subroutine for calculation of element matrix associated with the heat transfer function being represented by functional coefficients.

ITGDHS - Subroutine for calculation of element matrix associated with distributed heat sources being represented by functional coefficients.

INTEGX - Subroutine to evaluate the coefficients of the element matrix at a Gauss point. This subroutine is evaluating the matrix of the time derivative of pressure for unsaturated conditions The weighting is performed over the whole of the flow domain.

2.3.5.3 Basis functions

HFB2D2 - Subroutine for the calculation of basis function values and derivatives for a 2-D parabolic quadri-lateral element.

HFB3D1 - Subroutine for the calculation of basis function values and derivatives for a 3-D hexahedron element.

JACOB2 - Subroutine for the calculation of the Jacobian in 2-D.

JACOB3 - Subroutine for the calculation of the Jacobian in 3-D.

HEXA21 - Subroutine for calculation of 8-21 nodes hexahedral element.

PRISM1 - Subroutine for calculation of basis functions and their derivatives of a prismatic element.

JACOB4 - Subroutine for the calculation of the 2-D Jacobian for an element side belonging to a 3-D hexahedral element.

GFLGS1 - Subroutine to assemble the global matrix system for two phase flow of water and gas sharp interphase between gas and water.

HTSMAT - Subroutines to form simultaneous matrix systems for an element.

GFSMAT - Subroutine to form a simultaneous matrix system for an element.

GFSMA1 - Subroutine to form a simultaneous matrix system for an element.

GFSMA2 - Subroutine to form a simultaneous matrix system for an element.

GFSMA3 - Subroutine to form a simultaneous matrix system for an element.

GFSCHK - Subroutine to chek schemes for simultatneous matrix formulations.

GFSMTR - Subroutine to form a simultaneous matrix system for an element.

GFMINZ - Subroutine to initialize matrix systems.

GFMIZ1 - Subroutine to initialize matrix systems.

GFMIZ2 - Subroutine to initialize matrix systems.

GFMIZ3 - Subroutine to initialize matrix systems.

GFLGA2 - Computes basis function values at gaussian quadrature points to form integration matrices block elements.

GFS2M1 - Compute the element contributions for the global matrix system.

GFS2M2 - Subroutine to rearrange the element matrix system into an effective matrix system.

GFS1M2 - Subroutine to rearrange the element matrix system into an effective matrix system.

GFVEL3 - Subroutine to compute flow velocities in 2-d.

CHKPNT - Subroutine to check if a given point is located within a pre-specified sub-domain.

BLVEL3 - Block data to initialize coordinate limits.

HFLXNG - Subroutine to calculate nodal fluxes by averaging the the element contributions around the considered node.

EVDRVG - Subroutine to evaluate derivatives at nodal points.

BLDRVG - Block data to initialize local coordinates at nodal points.

HFLXNU - Subroutine to calculate nodal fluxes averaging the element contributions.

EVDRV1 - Subroutine to evaluate derivatives at nodal points.

BLDRV1 - Block data to initialize local coordinates at nodal points.

ELBLOC - Subroutine to locate boundary element sides.

ELBPRT - Subroutine to printout element sides located along the exterior boundary properties.

GFSETP - Subroutine to store and retrieve pressure arrays.

GFELHC - Subroutine to input model parameters for parallel plate fractured rock conceptualization.

EIFLU1 - Subroutine for testing with analytical solution data using the Ei-function.

EIFLU2 - Alternative entry for evaluation.

EIGAS1 - Subroutine for testing and comparing numerical and analytical solution data (Ei-function).

EIGAS3 - Alternative entry to specify gas viscosity function.

EIGAS2 - Alternative entry for evaluation.

2.3.6 Equation solution

2.3.6.1 Front solution

PRSLVF - Subroutine for the solution of the pressure equation.

TRSLVF - Subroutine for the solution of the temperature equations.

CHKFRX - Subroutine to check the frontwidth before enterring the front solution subroutines.

HFRNF1 - Front solution subroutine - symmetric storage mode.

HFRNF2 - Front solution subroutine - non-symmetric storage mode.

2.3.6.2 Band solution (currently not used)

PHSLVB - Subroutine for the solution of the pressure equation using a band-matrix solver.

MDCHK1 - Subroutine for the calculation of the maximum index difference between two nodes of an element.

TMSLVB - Subroutine for the solution of the temperature equations using a band-matrix solver.

HFLMU1 - Subroutine to organize the element matrices to facilitate the global matrix assembly.

FESLV1 - Linear equation solver - symmetric band storage mode.

BNDSLV - Linear equation solver - non-symmetric band storage mode.

BNDOUT - Subroutine to printout a band matrix system.

BNDTRI - Linear equation solver - non-symmetric band storage mode.

2.3.7 Heat transfer function

HFWFN1 - Subroutine for the transient heat transfer evaluation at a node.

HFWQUA - Subroutine for the quasi-steady state heat transfer evaluation at a node.

HFWFUN - Subroutine for the transient heat transfer function administration. This subroutine has the following additional entries:
HFWUPD - For updating the temperature records
HFTFUT - For printout of the temperature records

HFWSET - Subroutine to initialize the temperature records.

TFPRE1 - Subroutine for simple prediction of the fluid temperature at the next time level.

HFTFUT - Subroutine to printout records for the temperature history.

2.3.8 Mass and energy balance

HFBLS1 - Subroutine to output a table of material and energy balances (2-D version).

FFMBL1 - Subroutine to perform mass balance for the fluid (2-D version).

TFEBL1 - Subroutine to perform thermal energy balance for the fluid (2-D version).

TREBL1 - Subroutine to perform thermal energy balance for the rock (2-D version).

HFBLS3 - Subroutine to output a table of material and energy balances (3-D version).

FFMBL3 - Subroutine to perform mass balance for the fluid (3-D version).

TFEBL3 - Subroutine to perform thermal energy balance for the fluid (3-D version).

TREBL3 - Subroutine to perform thermal energy balance for the rock (3-D version).

NRMVEC - Subroutine to construct a normal vector to an element boundary.

2.3.9 Fluid velocities

VELXY1 - Subroutine for the calculation of flow velocities at interior points of elements.

VELXY3 - Subroutine for the calculation of flow velocities at interior points of elements. This version is for saturated-unsaturated flow.

VELSAV - Subroutine to store flow velocities into arrays for subsequent use. This subroutine has the following additional entry:
VELPLT - to call subroutine *VECPL1* to perform graphical display of velocity vectors.

VECPL1 - Subroutine to perform graphical display of velocity vectors.

VECPL2 - Subroutine to draw an arrow at a point representing the flow velocity at this point.

HFNFLX - Subroutine for the calculation of flow velocities at nodal points. Since the velocity field is discontinuous at the nodal points, a mean value is computed by taking the arithmetic mean mean value of all the elements being connected at a node.

HFLXNU - Subroutine for the calculation of flow velocities at nodal points. Since the velocity field is discontinuous at the nodal points, a mean value is computed by taking the arithmetic mean mean value of all the elements being connected at a node. This version is for saturated-unsaturated flow.

EVDRV1 - Subroutine to evaluate pressure gradient at the nodal points of an element.

2.3.10 Radioactive heat source

HFSCR1 - Subroutine to input concentrated or distributed heat sources.

HFSRC2 - Subroutine to check, during the element assembly, if an element contains any concentrated heat sources.

WDECAY - Subroutine to apply the function for the radioactive decay to the initial heat source values.

DECAY - Function for the radioactive decay.

HFSRC4 - Subroutine for calculation of implied heat sources due to mass sources.

2.3.11 Rock Stress

HFLGN4 - Subroutine to perform element integration for the displacement equations.

DUL2D1 - Subroutine to perform the evaluation of the various types of integrals associated with the terms in the displacement equations.

DUL2D5 - Such as for *DUL2D1* but for prismatic elements.

DULGA1 - Subroutine to perform sampling at Gauss points.

DUPHM1 - Subroutine to form matrix system for current element.

DUPHM2 - Subroutine to rearrange current element matrix system into an effective matrix system by the application of finite differences to the time derivatives.

HTSMAT - Subroutine to form a simultaneous matrix system of several sub-systems.

MATRXP - Subroutine to printout an element matrix system.

DUSLVF - Subroutine to prepare work arrays for the solution of the displacements using the frontal method.

HFRONU - Subroutine for front solution of the displacement equations.

INTG1S - Subroutine to evaulate the integrals associated with the stiffness matrix.

INTG2S - Subroutine to evaluate the integrals associated with the body force vector.

INTG3S - Subroutine to evaluate the integrals associated with the pressure and temperature gradients.

DUBLAT - Auxiliary subroutine to setup boundary conditions for some specific test cases used for the model verification.

BND2DU - Auxiliary subroutine to prescribe the displacements at the bottom resp. top boundaries (2-D flow domain).

BND3DU - Auxiliary subroutine to prescribe the displacements at the bottom resp. top boundaries (3-D flow domain).

DUPRT1 - Subroutine to printout a table of the nodal displacements.

2.3.12 Auxiliary

HFSET1 - Subroutine to initialize most of the work arrays

HFAX1A - Subroutine for various data transfer between work arrays. This subroutine has the following additional entries: HFAX1B,HFAX2A,HFAX2B,HFAX3A,HFAX3B.

HFWRK1 - Subroutine to update arrays of values from a previous time step. This subroutine has the following additional entries: HFWRK2,HFWRK3.

LMATUT - Subroutine for the printout of element matrices.

EKUT - Complementary subroutine to *LMATUT*.

EKUT1 - Complementary subroutine to *LMATUT*.

EKUT2 - Complementary subroutine to *LMATUT*.

FDPTH2 - Function to compute the mean depth of an element.

2.3.13 Subroutines used for Hydrocoin problems

HDR1BC - Subroutine to set boundary conditions for -Hydrocoin- Example 1, Level 1 called from: GFSBX1.

HDR4BC - Subroutine to set prescribed boundary pressure for -Hydrocoin- example no 4, Level 1

IOPR - control parameter
=0 zero radius
=1 prescribed pressure for nodes at r=RMIN
=2 prescribed pressure for nodes at r=RMAX

PHDYN1 - Subroutine to get dynamic pressure from total pressure.

PHDYN2 - Alternative entry to get total pressure from dynamic pressure.

HFRDF1 - Subroutine to redefine pressure as pressure head (m l-1 t-2) or hydraulic head (l)
IOP=1: Pressure to be transformed into pressure head
IOP=2: Pressure to be transformed into hydraulic head

2.3.14 Miscellaneous Ad Hoc subroutines

L1DGEN - Subroutine for the generation of a single row of 2-D parabolic elements to be used for 1-D analysis.

HFSET2 - Subroutine to set the initial temperature distribution.

HFSET3 - Alternative version of *HFSET2*.

HFSET4 - Alternative version of *HFSET2*.

UT1DV1 - Subroutine to output nodal data.

UT1DV3 - Subroutine for the output of nodal data (three-variables).

HFPLT1A - Subroutine for the printout and graphical of nodal data for 1-D analysis.

HFMEAN - Subroutine to integrate the mobility over an element in order to obtain the average mobility called from: HFELG1,HFLGS1.

GFINJ1 - Subroutine to adjust gas and water pressure to a given saturation at injection nodes called from: PRGSEL.

GFINJ2 - Subroutine to set some gas saturation at nodes adjacent to injection nodes called from: GFINJ1.

HFNSW5 - Subroutine to find pressure for given saturation called from: GFINJ1.

GFPRT1 - Subroutine to print out a parameter called from: GFPHS2.

GFPRED - Subroutine to extrapolate solution values for two phase flow of water and gas.

ADJUST - Subroutine to modify fluid and gas pressure conditions for "SFR-gas flow problem" called from: GFPHS2.

XPEAT1 - Subroutine to change pressure conditions at prescribed nodes
called from: GFSBX1
NDX1 - a node which will be freed if the pressure at node -NDX2- is less than zero (unsaturated)
NDX2 - a node which is located above node -NDX1-

LBPHX - Subroutine to printout prescribed node pressures.

HFTEST - Subroutine whose purpose is to printout messages on the use of the control parameter -IOPTST-.

PHMNMX - Subroutine to set prescribed pressure at vertical or horizontal boundaries.

PHXMIN - Sets prescribed values at nodes along:
X=XMIN, YMIN<= Y <= YMAX

PHXMAX - Sets prescribed values at nodes along:
X=XMAX, YMIN<= Y <= YMAX

PHYMIN - Sets prescribed values at nodes along:
XMIN <= X <= XMAX, Y = YMIN

PHYMAX - Sets prescribed values at nodes along:
XMIN <= X <= XMAX, Y = YMAX

PHXMIN - Subroutine to set prescribed pressure at the the vertical boundary x=XMIN, YMIN<= y <= YMAX

PHXMAX - Subroutine to set prescribed pressure at the the vertical boundary x=XMAX, YMIN<= y <= YMAX

PHYMIN - Subroutine to set prescribed pressure at the horizontal boundary XMIN<= x <= XMAX, y=YMIN

PHYMAX - Subroutine to set prescribed pressure at then horizontal boundary XMIN<= x <= XMAX, y=YMAX

AMINMX - Subroutine to find the maximum and minimum values of a one dimensional array.

PHCHK1 - Subroutine to check if solution values lie within the inverval of prescribed values this is just a rough feasibility check of the the results from a boundary value problem.

2.3.15 Gas migration subroutines

Gas migration--Evaluation of density and dynamic viscosity

GFNPG3 - Subroutine to evaluate "equations of state variables" at nodal points for the fluid and gas.

GFNPG1 - Subroutine to evaluate "equations of state variables" at nodal points for fluid phase.

GFNPG2 - Subroutine to evaluate "equations of state variables" at nodal points for the gas.

Gas migration input of boundary conditions and element mesh generation

GFRIN1 - Subroutine to set boundary and initial conditions for "SFR-gas flow problem".

GFSET1 - Subroutine to set hydrostatic initial conditions in water and gas flow domains.

HFNDP2 - Prints out the coordinates of the nodal points.

HFELP2 - Prints out the element incidences.

FEMLG1 - Subroutine to generate a 1-d element mesh.

SFRIN1 - Subroutine to set boundary and initial conditions for "SFR-gas flow problem".

SFRUT1 - Subroutine to set printout results.

SFRIN2 - Subroutine to set boundary and initial conditions for "SFR-gas flow problem"
called from: PRGSEL
this version performs the following:
1) sets all nodes initially hydrostati
2) reads in one card image from unit iun to localize and set cavern B.C.
3) reads in one card image from unit iun to specify the top boundary.

SFRUT2 - Subroutine to set printout results.

SFRIN3 - Subroutine to set boundary and initial conditions for 2-d gas migration problem.

SFRUT3 - Subroutine to set printout results.

GELINX - Subroutine to input parameters to setup gas migration problem
Water saturated phase (IPHASE=1),
Gas saturated phase (IPHASE=2)
Gas-water interphase (IPHASE=3)

GELIN2 - Subroutine to input parameters to setup gas migration problem.

GELINX - Subroutine to compute grid displacements for migration problem.

GELINY - Subroutine to compute grid displacements for migration problem after gas break-through.

GELSE1 - Subroutine to specify elements that belong to
water saturated phase (IPHASE=1),
gas saturated phase (IPHASE=2)
gas-water interphase (IPHASE=3)

GELIN3 - Subroutine to generate an element mesh for gas migration problem.

GFMINP - Subroutine to superimpose results from a previous grid mesh onto a new grid.

GFLINP - Subroutine for linear interpolation.

GELIN1 - Subroutine to input parameters to setup gas migration problem
water saturated phase (IPHASE=1),
gas saturated phase (IPHASE=2)
gas-water interphase (IPHASE=3)

GELIN2 - Subroutine to input parameters to setup gas migration problem.

GELINX - Subroutine to compute grid displacements for migration problem.

GELINY - Subroutine to compute grid displacements for migration problem after gas break-through.

GELSE1 - Subroutine to specify elements that belong to
water saturated phase (IPHASE=1),
gas saturated phase (IPHASE=2)
gas-water interphase (IPHASE=3)

GFMPS3 - Subroutine for evaluation of coefficient vectors for time derivatives of water and gas in connexion with two-phase flow.

GFMPS1 - Subroutine for evaluation of coefficient vectors for time derivatives of water in connexion with two-phase flow properties in connexion with two-phase flow.

GFMPS2 - Subroutine for evaluation of coefficient vectors for time derivatives of gas in connexion with two-phase flow.

GASMP2 - Subroutine for evaluation of the physical properties of the gas.

GFELG1 - Subroutine to assemble the global matrix system for two phase flow of water and gas sharp interphase between gas and liquid.

GFMAUX - Subroutine to transfer the effective matrix system to a work storage.

GFL2D1 - Subroutine to form integration matrices for a fractured medium block elements. (fluid and gas pressure)

GFLGAW - Computes basis function values at gaussian quadrature points to form integration matrices block elements.

GFINTG - Subroutine for calculation of integration matrices for gas-water displacement.

GFLGAB - Computes basis function values at gaussian quadrature points to form integration matrices block elements.

GFMPB1 - Compute the element contributions for the global matrix system block elements.

GFMPB2 - Subroutine to rearrange the element matrix system into an effective matrix system.

GFELGI - Subroutine to assemble the interface equations.

LMAT1E - Subroutine to form element matrix system for 1-d linear element.

GASMP2 - Subroutine for evaluation of the physical properties of the gas.

ZFCTOR - Subroutine for evaluation of the correction factor for real gas.

GASMPR - Block data to set the universal constant r of gases.

VISGA1 - Subroutine for evaluation of the gas viscosity.

VISGA2 - Subroutine for evaluation of the gas viscosity this version applies to hydrogen.

GWIELM - Subroutine to assemble interface equations to for a 3 by 3 element matrix system.

DLSCHK - Subroutine to check displacement length and to change the time step is to small or too high according to given criteria.

Gas migration-mass balance calculation

GMBLA1 - Subroutine to compute mass balance for 1-d gas gas migration.

GMBLA2 - Subroutine to compute mass balance for 1-d gas gas migration.

VELX2D - Subroutine to evaluate the fluid velocity at a point given as local coordinates physical properties such as mobility (hcn), density (dnfn) and porosity (porn) must be evaluated beforehand.

GFSED1 - Subroutine to store and retrieve gas-water interface parameter data.

GFSED2 - Entry to store gas-water parameter data for current fracture category in work arrays.

GFSED3 - Entry to retrieve store gas-water parameter data for current fracture category from work arrays.

GFSED4 - Subroutine to set nodal permeability.

GFSED1 - Subroutine to set element permeability.

GFSED5 - Subroutine to set line coordinates for current fracture category.

GFSED6 - Entry to update line coordinates for current fracture category.

GFSED7 - Entry to initialize line coordinates for all fracture categories.

GFSEDT - Subroutine to set fracture distribution.

PPROP1 - Subroutine to compute gas migration transport properties.

GFSETP - Subroutine to store and retrieve pressure arrays.

GFELHC - Subroutine to input model parameters for parallel plate fractured rock conceptualization.

EIFLU1 - Subroutine for testing with the Ei-function.

EIFLU2 - Alternative entry for evaluation.

EIGAS1 - Subroutine for testing with the Ei-function.

EIGAS3 - Alternative entry to specify gas viscosity function.

EIGAS2 - Alternative entry for evaluation.

Simultaneous solution for gas migration

GFLGS1 - Subroutine to assemble the global matrix system for two phase flow of water and gas sharp interphase between gas and water.

HTSMAT - Subroutines to form simultaneous matrix systems for an element.

GFSMAT - Subroutine to form a simultaneous matrix system for an element.

GFSMA1 - Subroutine to form a simultaneous matrix system for an element.

GFSMA2 - Subroutine to form a simultaneous matrix system for an element.

GFSMA3 - Subroutine to form a simultaneous matrix system for an element.

GFSCHK - Subroutine to chek schemes for simultatneous matrix formulations.

GFSMTR - Subroutine to form a simultaneous matrix system for an element.

GFMINZ - Subroutine to initialize matrix systems.

GFMIZ1 - Subroutine to initialize matrix systems.

GFMIZ2 - Subroutine to initialize matrix systems.

GFMIZ3 - Subroutine to initialize matrix systems.

GFLGA2 - computes basis function values at gaussian quadrature points to form integration matrices block elements.

GFS2M1 - compute the element contributions for the global matrix system block elements.

GFS2M2 - Subroutine to rearrange the element matrix system into an effective matrix system.

GFS1M2 - Subroutine to rearrange the element matrix system into an effective matrix system.

GFVEL3 - Subroutine to compute flow velocities in 2-d.

CHKPNT - Subroutine to check if a given point is located within a pre-specified sub-domain.

BLVEL3 - Block data to initialize coordinate limits.

HFLXNG - Subroutine to calculate nodal fluxes by averaging the element contributions.

EVDRVG - Subroutine to evaluate derivatives at nodal points.

BLDRVG - Block data to initialize local coordinates at nodal points.

2.3.16 Block fracture system Subroutines

ELREFMX - Main program to set up connectivity matrices for element sides.

ELFRIX - Subroutine to return index values associated with the book-keeping of the fracture elements and nodes.

ELFRIN - Subroutine to input a set of intervals to specify areas of fracture elements.

NODSWP - Subroutine to reorganize a nodal array according to the re-definition of the element mesh as given by a fractured medium.

ELMFRC - Subroutine to set up connectivity matrices between the fracture elements and the blocks.

FNODS1 - Subroutine to generate node numbers of fracture elements.

ELNODF - Subroutine to establish an element connectivity matrix which also includes the fracture elements.

CHKSDE - Subroutine to check the order of the nodes which define the opposite block side.

BNODS3 - Subroutine to generate node numbers on the block elements and to store these in a connectivity matrix for the blocks. an array relating the block node numbers to the original node numbering is also prepared.

ELFND6 - Subroutine to perform local numbering of adjacent fracture elements.

BNODS2 - Subroutine to perform local numbering of block nodes and adjacent fracture nodes.

BNODS1 - Subroutine to generate node numbers on the block elements and to store these in a connectivity matrix for the blocks. an array relating the block node numbers to the original node numbering is also prepared.

LFSUT1 - Subroutine to printout fracture element incidences.

ELBLOC - Subroutine to locate boundary element sides.

ELBPRT - Subroutine to printout element sides located along the exterior boundary.

FELTAB - Subroutine to set up cross reference tables between fracture nodes and block nodes.

IBLOCK - Subroutine to perform the integration along the block surfaces to account for the exchange of fluid between the blocks and the fractures.

IFRACT - Subroutine to perform the integration along either side of the fracture element to account for the exchange of fluid between current fracture and its adjoing blocks.

IBLCK1 - Subroutine to perform the integration along the block surfaces to account for the exchange of fluid between the blocks and the fractures.

IFRCT1 - Subroutine to perform the integration along either side of the fracture element to account for the exchange of fluid between current fracture and its adjoing blocks.

PHSRC1 - Subroutine to compute the nodal fluxes at the block nodes and to apply these to the fracture nodes.

PHSRC2 - Subroutine to compute the nodal fluxes at the block nodes and to apply these to the fracture nodes.

Evaluation of nodal fluxes for unsaturated flow

HFLXNU - Subroutine to calculate nodal fluxes averaging the element contributions.

EVDRV1 - Subroutine to evaluate derivatives at nodal points.

BLDRV1 - Block data to initialize local coordinates at nodal points.

2.3.17 Rock stress

HFLGN4S - Subroutine to assemble the global matrix system for the solution of the displacement equation.

HFBU2D - Subroutine to compute element matrices for boundaries with prescribed flux.

DUL2DX - Subroutine to compute element matrices for the integration along the boundaries with prescribed displacements.

DUX2D1 - Subroutine to perform integration along the element boundaries.

DUTEST - Subroutine to printout stress-strain tensors.

BLU2DC - Block data to initialize control parameter for stress integration
IOPDUX=0 integration matrices to be computed
IOPDUX=1 surface tractions to be computed.

HFBU3D - Subroutine to compute element matrices for boundaries with prescribed flux.

DUL3DX - Subroutine to compute element matrices for the integration along the boundaries with prescribed displacements.

DUX3D1 - Subroutine to perform integration along element boundaries.

DUTFLX - Subroutine to compute the coefficients of an element matrix at a gauss point for an element boundary subject to prescribed flux.

BLU3DC - Block data to initialize control parameter for stress integration
IOPDUX=0 integration matrices to be computed
IOPDUX=1 surface tractions to be computed.

HOOKE1 - Subroutines for elasticity computations.

HOOKE1 - Subroutine to setup elasticity matrices.

STRES2 - Subroutine to determine the stress tensor.

STRAI2 - Subroutine to determine the strain tensor.

STRESS - Subroutine to evaluate the stress tensor.

STRAIN - Subroutine to evaluate the strain tensor.

ESTRAI - Subroutine to determine the strain tensor.

ESTRAJ - Subroutine to setup the strain tensor in vector form.

ESTRAK - Subroutine to setup the strain tensor in matrix form.

PSTR2D - Subroutine to determine the principal stresses.

DUSLVF - Subroutine to prepare work arrays for the solution* of the displacements using the frontal method. (uses symmetric frontal solver)

DUAXLA - Subroutine to prepare work arrays for the front solution of the rock displacements.

DULOC1 - Subroutines to set boundary conditions for the the calculation of the displacements.

Printout of rock stress data

HFUTV2 - Subroutines to printout results.

SOLVU2 - Subroutine to output solution values.

NDPRU2 - Subroutine for nodal printout
this version prints out concentrated heat sources and selected nodal values
- two-d version -

NDPRUX - Alternative entry to specify functions for material properties.

FUNGX - Dummy functions for analytical results
FUNGY
SIGMXX

NDPRU3 - Subroutine for nodal printout
this version prints out concentrated heat sources and selected nodal values
- three-d version -

DUPRT1 - Subroutine to printout the nodal displacements.

DUPRT2 - Subroutine to printout the nodal displacements.

MATRXP - Subroutine to printout an element matrix system.

Basis function subroutines

HFLPAC - Subroutine for the evaluation of the basis functions and their derivatives for a given elelemt type.

HFBS1D - Subroutines for basis functions.

HFB1D2 - Calculates basis function values and derivatives.

HFB1DA - Alternative entry to compute only the basis functions.

JACOB1 - Computes the Jacobian in 1-d.

HFBS2D - Subroutines for basis functions.

HFB2DF - Subroutines for basis functions and their derivatives for a 2-d four nodes quadri-lateral element.

HFB2DG - Alternative entry for basis functions.

HFB2D2 - Calculates basis function values and derivatives.

HFB2DA - Alternative entry to compute only the basis functions.

JACOB2 - Computes the Jacobian in 2-d.

HFBS3D - Subroutines for basis functions.

HFB3D1 - Calculates basis function values and derivatives.

HFB3DA - Alternative entry to compute only the basis functions.

JACOB3 - Computes the Jacobian in 3-d.

JACB12 - Subroutines to compute the jacobian associated with 1-d fracture elements in a 3-d domain.

JACB11 - Subroutines to compute the jacobian associated with 1-d fracture elements in a 3-d domain.

JACB23X - Subroutines to compute the jacobian associated with 3-d fracture elements.

JACB22X - Subroutines to compute the jacobian associated with 3-d fracture elements.

COORDT - Subroutine to apply a linear coordinate transformation.

HEXA21 - Three-d hexahedral element (8 - 21 nodes).

HEX21X - Subroutine to find the basis functions for a three-d hexahedral element (8-21 nodes).

NRMVEC - Subroutine to construct a normal vector to an element surface.

VCROSS - Subroutine to perform vector product the output vector is normalized.

HFAREA - Subroutine to construct a normal vector to a surface.

VECMUL - Subroutine to perform the cross product of two vectors.

VECNRM - Subroutine to normalize a given vector.

Mass and energy balance calculation

GFBL2D - Subroutine for mass and energy balances in 2-d.

GFMBL1 - Subroutine to perform material balance for the fluid flow equation.

GFBLS2 - Subroutine to output a table of material and energy balances.

HFBL2D - Subroutine for mass and energy balances in 2-d.

HFBLS2 - Subroutine to output a table of material and energy balances.

HFBL3D - Subroutines for mass and energy balances in 3-d.

IPRCHK - Function to set printout control option.

Boundary flux subroutines

HFBF2D - Subroutine to compute element matrices for boundaries with prescribed flux.

HFLI2D - Subroutine to compute element integration for elements with prescribed flux.

HFLX2D - Subroutine to perform integration of fluxes fluid flow equation.

HFBF3D - Subroutine to compute element matrices for boundaries with prescribed flux.

HFLI3D - Subroutine to compute element matrices for boundaries with prescribed flux.

INTFLX - Subroutine to compute the coefficients of an element matrix at a Gauss point for an element boundary subject to prescribed flux.

HFLX3D - Subroutine to perform integration of fluxes fluid flow equation.

2.4 COMMON AREAS

The program contains the following common blocks:

```
 COMMON/AMBLCK/ AMBLCK(16,16)
 COMMON/AMBMNG/ AM(21,21,5),BM(21,5)
 COMMON/ANISOT/ FHCX,FHCY,FHCZ

 COMMON/BOTNDS/ JBTN(125),NBTNMX,NBTN

 COMMON/CJACOB/ AJ(3,3)
 COMMON/CNTRL1/ ICP1(40)
 COMMON/CNTRL2/ MXSTEP,MXITER,DTFAC,PHERTL,TFERTL,TRERTL
 COMMON/CNTRL3/ NSTEP,TIME,DT,ITER,PHERMX,TFERMX,TRERMX
 COMMON/CNTRL4/ PHMXDT,TFMXDT,TRMXDT
 COMMON/CNTRL5/ NPSTPS,NTSTPS
 COMMON/COEFF1/ POR,CF.,CR,CTF,DF,DR.
 COMMON/COEFF2/ SCF,SCR,HT
 COMMON/COEFF3/ HC(3,3),TCF(3,3),TCR(3,3)
 COMMON/COEFF4/ GRAVTY(3)
 COMMON/COEFF5/ TDIF,RCHAR,CAREA,WSHF,CDR,NSUM,EXTOL
 COMMON/COEFF6/ C11,C12,C21,C31,C32
 COMMON/COEFF8/ CPOR,POR1,CPERM,PERM1,FRPERM
 COMMON/COMFRC/ LFRACS(25),MXLFRC,NLFRC

 COMMON/DUFLUX/ DFLX(99,4,3),JLDLU(99),JUSIDE(99),MXNDLX,NDLX,NDEL
 COMMON/DTMAX1/ DTMAX

 COMMON/ELCTY1/ EM,PR,ALFA,BETA
 COMMON/EKMFKM/ EKM(21,21,5),FKM(21,5)
 COMMON/ELTYPE/ LTYPE
 COMMON/ELEMNO/ IEL
 COMMON/EQSOLV/ ICPSLV(10)
 COMMON/ERRMXN/ MXEDTN(3),MXE1IN(3),MXE2IN(3)

 COMMON/FECOM1/ MXNE,NE,MXNP,NP
 COMMON/FECOM2/ MXBAND,MBAND,MXNN,NN
 COMMON/FECOM4/ NNODE,NDOF,NEK,NDIM
 COMMON/FECOM6/ LNTYP(3,12)
 COMMON/FECOM7/ MXNEB,NEB,MXNBN,NBN,MXNFEL,NFEL,MXNFN,NFN
 COMMON/FRIC01/ QFRI(    15),PFRI(    15),XFRI(   125, 15),
1               CFRI(    15)
 COMMON/FRIC03/ MXFRI,NFRI
 COMMON/FRIC04/ RPERM(15),XLCAVI(15),FRWIDI(15),
1               QFACI(15)
 COMMON/FTCFR / FTCFR
 COMMON/FUNGA1/ TLOAD,VALFA,A,A2G,AK,IOP2D
 COMMON/FUNGA3/ IOPDUL,IBODY
 COMMON/FUNGA4/ IOPBC
 COMMON/FUNLHC/ FUNLHC

 COMMON/GAS3   / SLIR(3),SGR(3)
 COMMON/GAUBFN/ PBFN(21,27),PDFN(3,21,27),WTGT(27),NGT
 COMMON/GAUSS1/ XG1(4),WG1(4),NG1
 COMMON/GAUSS2/ XG2(4),YG2(4),WG2(4),NG2
 COMMON/GAUSSW/ XG(4),YG(4),ZG(4),WGX(4),WGY(4),WGZ(4),NGX,NGY,NGZ
 COMMON/GFELX1/ NELG,NELG1,NELG2,NELW,NELW1,NELW2,
1               NPG,NPG1,NPG2,NPW,NPW1,NPW2,
```

```
2                NELI,NELGWI,NPGWI1,NPGWI2
 COMMON/GFELX2/ XLGAS,XLWAT,XLGWI,XLTOT,DLSG
 COMMON/GFELX4/ QMG1,QMG2,QMW1,QMW2
 COMMON/GRVTY1/ GRVTY1(3)
 COMMON/GWMODE/ MODE

 COMMON/HEATMS/ NHMS1,NHMS2
 COMMON/HFMEB1/ FSUM(5),WFLXLB,TFLXT1,TFLXT2,TFLXT3,WSUM
 COMMON/HFMEB2/ THEAT,TNRG
 COMMON/HFMEB3/ THEATR,TENRGR
 COMMON/HFMEB4/ TDHS,TCHS
 COMMON/HFPLOT/ IPLOT
 COMMON/HFREF1/ PREF,TREF,HCREF,PRMREF,POREF,DNFREF,VSCREF
 COMMON/HFREF2/ GEOGRD,SLOPE,XREF,YREF,ZREF
 COMMON/HFREF3/ A1,AL1,A2,AL2
 COMMON/HFREF4/ CPR,CPRM
 COMMON/HFWIUN/ IUNIT1,IUNIT2
 COMMON/HGCUSH/ HG,HS
 COMMON/HSCOM1/ HS(75),IHS(75,3),NHS
 COMMON/HSCOM2/ WHS(21),NDS(21),NS

 COMMON/IAXSYM/ IAXSYM
 COMMON/ICPLOT/ ICPLOT(10)
 COMMON/ICRAY / ICRAY
 COMMON/IDISCR/ IDISCR
 COMMON/IDNFMS/ IDNFMS
 COMMON/IELREF/ IELREF(375)
 COMMON/IFMODE/ IFMODE,IFRCHK,IFRM
 COMMON/IFNPG1/ IFNPG1
 COMMON/IGBRK1/ IGBRK,NSTBRK
 COMMON/IMDPTH/ IMDPTH
 COMMON/IMSTRE/ IMSTRE
 COMMON/IMTYPE/ IMTYPE
 COMMON/INEWTN/ INEWTN
 COMMON/INPTEX/ INPTEX
 COMMON/IOPBC1/ IOPBC1(10)
 COMMON/IOPFLX/ IOPFLX
 COMMON/IOPHC / IOPHC
 COMMON/IOPINS/ IOPINS
 COMMON/IOPOR / IOPOR
 COMMON/IOPSLV/ IOPSLV
 COMMON/IOPTST/ IOPTST
 COMMON/IPRISM/ IPRISM
 COMMON/ITERX / ITERX
 COMMON/ITRPRT/ ITRPRT
 COMMON/IUFRON/ IU1,IU2,IU3,IU4
 COMMON/IUNITS/ IUNIT(10)
 COMMON/IUNSAT/ IUNSAT
 COMMON/INUN10/ INUN(10)
 COMMON/IWAVG / IWAVG
 COMMON/IWTBL / IWTBL

 COMMON/JBPRM1/ JBNAM,JBNR
 COMMON/JBPRM2/ IYR,MON,IDY,IHR,IMIN,ISEC
 COMMON/JNDS2D/ JNDS2D(3,4),NCHK2D
 COMMON/JNDS3D/ JNDS3D(4,6),NCHK3D,JSAP(8)
```

```
COMMON/LADHOC/ CGWI6X,CGWI8X,LADHOC,LGWI
COMMON/LBFLUX/ BFLX(375,3),JLFLX(375),JLSIDE(375),MXFLX,NFLX,NCEL
COMMON/LBSEEP/ LBSEEP(100),MXSEEP,NSEEP
COMMON/LHEX21/ IEL21
COMMON/LSFLAG/ LSFLAG

COMMON/MATMD1/ IMATM,IPORF,IPERF
----------------------------------------------------------------
COMMON/MATRX1/ AM1(21,21,3,3),VEC1(21,3)
COMMON/MATRX2/ AM2(21,21,3,3),VEC(21,3)
COMMON/MATRX3/ AA(42,42),CC(42)
----------------------------------------------------------------
COMMON/MFRONX/ MFRON
COMMON/MISCHK/ MISCHK
COMMON/MODPRT/ MODPRT,MPRT
COMMON/MOST10/ LFSIDE(375,6),LCON(375,6,2),LECONX(375,6)
COMMON/MTRX1 / AM1(8,8,3,3),VEC1(8,3)
COMMON/MTRX2 / AM2(8,8,3,3),VEC2(8,3)
COMMON/MTRX3 / AA(24,24),CC(24)
COMMON/MXLBHH/ MXLBHH
COMMON/MXNDE / MXNDE
COMMON/MXNSTP/ MXNSTP
COMMON/MXNPEL/ MX21

COMMON/NETOT / NETOT
COMMON/NEX   / NEX
COMMON/NODPRT/ NDSPRT(100),MXNPRT,NPRT
COMMON/NREADT/ NREADT
COMMON/NRUN  / NRUN

COMMON/PBASIS/ P(21)
COMMON/PHNTFN/ PHN(21),TFN(21),DNFN(21)
COMMON/PHSINK/ WPSINK(75),LPSINK(75),NPSINK(75),MXNMS,NMS
COMMON/PLOTTR/ IOPEN,ICLOS
COMMON/PLOTXY/ XMIN,XMAX,YMIN,YMAX,SX,SY,XORG,YORG,DX,DY
COMMON/PORNDS/ PORN(21)

COMMON/RUNMAX/ RUNMAX

COMMON/SPRIME/ SPRIME,ISPRIM

COMMON/TIME1 / TIME1
COMMON/TITLE / TEX(20)
COMMON/TITLE2/ TEX2(20)
COMMON/TITLE3/ TEX3(20)
COMMON/TOPNDS/ JTPN(125),NTPNMX,NTPN
COMMON/TWOPH5/ IPSINK
COMMON/TWOPH3/ IPHASE

COMMON/UNSAMG/ THXG(41,3),HCRG(41,3), NTHMXG(3)
COMMON/UNSAM1/ THX1(20,3),PHC(20,3),NTHX1
COMMON/UNSAM2/ THX2(20,3), HCRX(20,3),NTHX2
COMMON/UNSAM3/ THX3(20,3),DSWPC(20,3),NTHX3
COMMON/UNSAM4/ MXIMAX,NXIMAT
COMMON/UNSAM5/ IPRUNS
COMMON/UNSAM6/ WTBL,WLV1,WLV2
COMMON/UNSAM7/ XWTBL(125),YWTBL(125),MXWTBL,NWTBL
COMMON/UNSATN/ SWN(21),DSWN(21)
COMMON/UNSTEX/ UNSLBL(15,3)
```

```
      COMMON/VELARY/ XCE(612),YCE(612),ZCE(612),VXE(612),VYE(612),
     1               VZE(612),NL
      COMMON/VELMAX/ VELMAX

      COMMON/WEIGHT/ W11,W12,W21,W22,W23

      COMMON/XIETZE/ XI2(8),ET2(8),XI3(8),ET3(8),ZE3(8)
      COMMON/XIT4   / XI4(21),ET4(21),ZE4(21)
      COMMON/XYZHCN/ XYZ(3,21),HCN(3,21)
      COMMON/XZERO / XF1,XF2,YF1,YF2,ZF1,ZF2
```

2.5 DESCRIPTION OF VARIABLES APPEARING IN THE COMMON BLOCKS

A - Internal parameter computed by the program.

A1 - Coefficient in the function for the radioactive decay.

A2 - Coefficient in the function for the radioactive decay.

AA(42,42) - Element matrix.

AJ(3,3) - Matrix of Jacobian coefficients.

ALFA - Coefficient of thermal volume expansion of rock.

AL1 - Coefficient in the function for the radioactive decay.

AL2 - Coefficient in the function for the radioactive decay.

A2G - Internal parameter.

AK - Internal parameter.

AM(21,21,5) - See descriptions of the element matrix systems.

AM1(21,21,5) - See descriptions of the element matrix systems.

AM2(8,8,3,3) - See descriptions of the element matrix systems.

AMBLCK(16,16) - Matrix resulting from surface integration fluid flow transfer along block-fracture surfaces.

B1 - Not used.

B2(21,21) - See descriptions of the element matrix systems.

B3(21,21) - See descriptions of the element matrix systems.

BETA - Elasticity parameter related to the thermal volume expansion of the rock and Poisson's ration (see Nomenclature).

BFLX(375,3) - Matrix for prescribed boundary flux.

BM(21,5) - See descriptions of the element matrix systems.

C11 - Auxiliary variable, currently: C11=CF.+CR.

C12 - Auxiliary variable, currently: C12 = CTF.

C21 - Auxiliary variable, currently: C21 = SCF.

C31 - Auxiliary variable, currently: C12 = CTF.

C32 - Auxiliary variable, currently: C12 = CTF.

CAREA - Coefficient of the contact area used in the heat transfer function.

CC(42) - Array for right-hand side of the system of element of element equations.

CDR - Coefficient used in the heat transfer function whose value depends on the model approach considered (see Thunvik and Braester, 1980).

CF. - Fluid Compressibility.

CFRI(15) - Array for total number of fractures of the respective fracture category.

CPERM - Coefficient in permeability function (for depth variation of the permeability).

CPOR - Coefficient in porosity function.

CR - Rock compressibility.

CTF - Coefficient of thermal volume expansion of the fluid.

DF - Fluid density.

DLSG - Current gas-water displacement length.

DNFREF - Reference value of fluid density.

DNFN(21) - Array for fluid densities at nodal points of current element.

DNFREF - Reference value of the fluid density.

DP - Difference in pressure between current and previous iteration.

DR. - Rock density.

DSWP(20,3) - Matrix for derivatives of saturation versus pressure for various degree of saturation. This matrix is associated with the matrix THX3 which contains the corresponding saturation values.

DSWN(21) - Array for nodal values of the derivatives of saturation versus pressure for current element.

DT - Time increment.

DTFAC - Factor by which the time increment is multiplied during the execution.

DTMAX - Maximum permitted time increment.

DX - Plot scale in the x-direction (DX=(XMAX-XMIN)/SX

DY - Plot scale in the y-direction (DY=(YMAX-YMIN)/SY

EKM(21,21,5) - See descriptions of the element matrix systems.

EM - Young's modulus.

ET2 - Array for local y-coordinates of the nodal points of a 2-D parabolic element.

ET3 - Array for local y-coordinates of the nodal points of a 3-D hexahedral (8-nodes) element.

ET4 - Array for local y-coordinates of the nodal points of a 3-D hexahedral (8-21 nodes) element.

EXTOL - Tolerance value to be used as the criterion to stop summation of terms in infinite series at the heat transfer function evaluation

-30

Default: EXTOL = 10

FDLX(99,4,3) - Matrix for prescribed displacements at element sides along the boundary.

FHCX - Factor for the permeability component in the x-direction.

FHCY - Factor for the permeability component in the y-direction.

FHCZ - Factor for the permeability component in the z-direction.

FKM(21,5) - See descriptions of the element matrix systems.
Default: PERM1=PRMREF

FTCFR - Coefficient for linear fluid transfer function for blocks and fractures.

FRPERM - Coefficient to specify permeability of a fracture zone versus the surrounding rock. The permeability of the fracture zone is obtained by multiplying FRPERM by the rock permeability.
Default: FRPERM=1.0

FRWIDI(15) - Array for equivalent fracture with (or tube diameters) for each fracture category.

FSUM(5) - Array for contributions from the time derivatives to the total mass- and energy balance calculation.

GEOGRD - Geothermal gradient.

GRAVTY(3) - Vector for the acceleration of gravity.

GRVTY1(3) - Vector for the acceleration of gravity.

HC(3,3) - Matrix for the permeability tensor.

HCN(3,21) - Matrix for the permeability components at the nodal points of current element.

HCREF - Reference value of the hydraulic conductivity.

HCRF(41.3) - Matrix to hold the curves for saturation versus relative permeability of the fluid for 3 different materials.

HCRG(41,3) - Matrix to hold relative gas permeability curves for three different materials.

HCRX(20,3) - Matrix for the relative permeability at various degree of saturation. This matrix is input and is associated with THX2, which contains the corresponding saturation values.

HG - Thickness of gas cushion (in gas migration calculation).

HS - Depth to see bottom (in gas migration calculation).

HS(75) - Array for the effects of concentrated heat sources.

HT - Heat transfer coefficient.

IAXSYM - Indicates if the flow domain is axi-symmetric.

IBLNOD(375,8) - Connectivity matrix for block elements.

IBODY - Body force control parameter.

ICLOS - Indicates if the plotter has been closed.

ICP1(40) - Array for miscellaneous control parameters.

ICPLOT(10) - Array for plot control parameters.

ICPSLV(10) - Array for control parameters associated with the solution variables.

ICRAY - Control parameter for Cray runs.

IDISCR - Control parameter for discrete modelling.

IDNFMS - An integer to indicate if density is to be multiplied by volumetric sources to get
= 0 mass source input (no action)
= 1 volumetric input source to be multiplied with density (currently this is applied only to the gas phase).

IDTRED - See description of PHMXTL.

IDY - Day of execution.

IEL - Current element number.

IEL21 - Control parameter to specify type of hexahedral element used (see input description section 3.2.1).

IELREF(375) - Auxiliary array used during the establishment to keep block sides associated with fracture element.

IFRM - Read/write control parameter in front solver.

IFLNOD(450,4) - Connectivity matrix for the fracture elements.

IFNPG1 - An integer to indicate if fluid density and viscosity needs to be evaluated only once
= 0 evaluation at each time step
= 1 evaluation only at first time step.

IFMODE - Front solution control parameter
= 0 Direct elimination of element matrix systems.
= 1 Element matrices to be written to disk file.

IGBRK - Control parameter for gas break through.

IHR - Hour of execution.

IHS(75,3) - Matrix used for the concentrated heat sources.
IHS(IHS,1) = <lock node number of element containing the heat source>
IHS(IHS,2) = <element containing the heat source>
IHS(IHS,3) = <nodal number>
IHS is number of heat source.

IMATM - An integer variable that should be set equal to one if node oriented material properties are to be considered
=0 Element oriented properties
=1 Node oriented properties.

IMSTRE - Element matrix control parameter
= 0 Element matrix and the R.H.S to be written on separate records
= 1 Element matrix and the R.H.S to be written on the same record.

IMIN - Minute of execution.

IMDPTH - Indicates of the permeability of an element is to be related to the mean depth of the element.

IMTYPE - Control parameter to indicate type of format used for nodal coordinates function.

INEWTN - Indicates of Newton-Raphson iteration technique is to be used.

INPTEX - An integer used to indicate whether each input group should be proceeded with a text record
INPTEX=0 No text record to be input
INPTEX=1 each input group to be proceeded by a text record.

INUN(1) - Logical unit number for input to HFINPT, HFNDIN and HFELIN
Default: INUN(1)=5

INUN(2) - Logical unit number for input to subroutines HFSRC1, HFBCX1 and SELNDS
Default: INUN(2)=5

INUN(3) - Logical unit number for data from a previous execution to be read in. This is to continue the execution using solution values from the most recent time step in a previous execution as initial values. This is carried out by subroutines HFNDUV and HFNDU1. If no data are to be input in this way, then set:
INUN(3)=0

INUN(4) - Output unit used for storage of of input to subroutines HFINPT, HFNDIN and HFELIN
Default: INUN(4)=21

INUN(5) - Output unit for solution values of pressure and temperatures at each time step. This is carried out by subroutine HFSAVD. This output may later be used as input using INUN(3)
Default: INUN(5)=22

INUN(6) - Input unit for unsaturated data.
Default: INUN(6)=55

INUN(7) - Input unit for stress data.
Default: INUN(7)=56

INUN(8) - Input unit for element mesh.
Default: INUN(8)= 5

INUN(9) - Input unit for drain data.
Default: INUN(9)=57

INUN9 - Unit number for printout of gas displacement.

INUN(10) - Array for logical unit numbers of the input files.

IOPBC1 - Array for control option of B.C. settings
IOPBC1(3)=1 set B.C. on bottom boundary
IOPBC1(4)=1 set B.C. on top boundary.

IOPG - Gravity control parameter
IOPG=0 no gravity
IOPG=1 gravity.

IOPSLV - Solution parameter to specify the set of equations to be solved
= 0 Pressure, Fluid and Rock temperatures to be solved for
= 1 Pressure to be solved for
= 2 Fluid temperature to be solved for
= 3 Pressure and fluid temperature to be solved for
= 4 Rock temperature to be solved for
= 5 Fluid and rock temperatures to be solved for
= 6 Displacements to be solved for
= 7 Two-phase flow.

IOPBC - Control parameter for boundary conditions in stress calculations.

IOPDUL - Control parameter for stress calculation.

IOPHC - An integer variable which should be set equal one if node permeabilities are to be considered.

IOPOR - An integer variable which should be set equal one if node porosities are to be considered.

IOPEN - Indicates if the plotter has been opened.

IOPFLX - Indicates if prescribed boundary flux conditions are to be applied, or if the current purpose is to calculate the fluxes through an element boundary.

IOPHC - Indicates of permeability is to be associated with the nodal points rather that with the element volumes.

IOPINS - Control parameter to suppress input of parameter data and initialization.

IOPTST - Control parameter for special parameter settings.

IOP2D - Control parameter for solution of validation example of stress equations
= 0 Ideal stress
= 1 Plane stress
= 2 Plain strain.

IPHASE - Phase control parameter.
=1 gas
=2 water

IPSINK - Control parameter to specify to which phase point sources are to be applied.
= 0
= 1
= 2

IPRCHK - Front solution printout parameter.

IPERF - An integer variable which should be set equal a positive value if a function for permeability is to be called for each element.

IPORF - An integer variable which should be set equal a positive value if a function for porosity is to be called for each element.

IPLOT - Indicates of the solution values are to be graphically displayed.

IPRISM - Indicates if current element is a prismatic element.

IPRMX - Set control parameter for printout of nodes with maximum change in values between time steps.

IPRUNS - Indicates if the material properties associated with unsaturated conditions are to be printed out.

ISPRIM - Control parameter for usage of SPRIME.

ISEC - Second of execution.

ITER - Number of iterations in current time step.

ITERX - Iteration counter used for the solution of the fluid pressure.

ITRPRT - Control parameter to produce"trace" printout from fluid flow integration subroutines.

IUNSAT - Control parameter for unsaturated flow. A value than zero indicates that unsaturated flow should be considered.

IUNSAT - An integer that must be set equal to one when unsaturated flow is to be considered.

IUNIT(10) - Array for logical unit numbers of the intermediate data files.

IUNIT1 - Unit number of file for temperature history data used for the evaluation of the transient heat transfer function.

IUNIT2 - Unit number of file for temperature history data used for the evaluation of the transient heat transfer function.

IU1 - Unit number of file containing element matrix systems to be processed by the front solution subroutine.

IU2 - Unit number of file containing the right hand sides of the element matrix systems to be processed by the front solution subroutine.

IU3 - Unit number of work file for the front solution subroutine.

IU4 - Unit number of work file for the front solution subroutine.

IWTBL - An integer to indicate if the position of the water table is to be located in an unsaturated-saturated flow problem (only for NDIM=2).
IWTBL > 0 water table to be located and printed.

IWAVG - Control parameter for weighted average calculation for calculation of material properties:
IWAVG = 0 No weighting to be performed.
= 1 Two most recent values to be used.
= 2 Three most recent values to be used.
= 3 Most recent values and previous solution values to be used.
= 4 Most recent values plus iteration values solution values to be used.

IYR - Year of execution.

JBN(500) - Array for block nodes corresponding node numbers in the original number system.

JBNAM - Job identification name.
(CHARACTER*8)

JBNR - Job number (CHARACTER*4).

JBTN - Array for node numbers on the top boundary.

JFN(500) - Array for fracture nodes containing the node numbers in the original system.

JLDLU(99) - Array for element numbers associated with prescribed displacements.

JLFLX(375) - Array for element numbers subject to prescribed boundary flux conditions.

JLSIDE - Array for the element sides of the elements subject to prescribed flux conditions.

JNDS2D(3,4) - Matrix for element side definitions of 2-D.

JNS3D(4,6) - Matrix for element side definitions of 3-D (8-node hexahedral) element.

JSAP(8) - Auxiliary array specified in block data BLIELX to temporarily redefine the nodal numbering of an element type (currently not used).

JTPN - Array for node numbers at the top boundary.

JUSIDE(99) - Array for element side number of elements associated with the prescribed boundary displacements.

LADHOC - Logical control parameter which if set equal to will cause all time derivatives being held in the coefficient vectors to be set equal to zero.

LBSEEP(100) - Array for possible seepage nodes.

LCON(MXNE,NSIDES,2) - Work matrix where MXNE is the maximum permitted number of elements NSIDES is the maximum permitted number of sides

per element
LCON(IEL,ISIDE,1) = <adjacent element>
LCON(IEL,ISIDE,2) = <adjacent element side>

LECONX - Matrix to relate block element sides to fracture elements
LECONX(IEL,ISIDE) = <fracture element>
IEL is current block element.
ISIDE is current side of the block element.

LFRACS(25) - Array for element numbers for fracture elements.

LFSIDE(375,6) - Work matrix used to specify fracture elements associated with the sides of the respective element
LFSIDE(IEL,ISIDE)= <fracture element number>
IEL is current element
ISIDE is specifying a side of current element.

LGWI - Logical control parameter (currently not used).

LNFEL(450,2) - Work matrix for the fracture elements
LNFEL(IFEL,1) = <reference element number>
LNFEL(IFEL,2) = <side of the reference element being associated with current fracture element IFEL>.

LNTYP(3,12) - Matrix for specification of the various types of elements implemented.

LPSINK(75) - Array for element numbers associated with concentrated mass sources.

LSFLAG - Internal control parameter which is set true if current element is of the same type as the previous element.

LTYPE - Type code of current element.

MBAND - Current band-width.

MFRON - Current front width.

MISCHK - General printout control parameter.

MON - Month of execution.

MODE - Control parameters to indicate that fracture network generation is to be performed.

MODPRT - Printout control parameter to suppress the printout of intermediate time steps, e.g.
MODPRT=1 gives printout for every time step.
MODPRT=2 gives printout for time steps 1,3,5,7,...
MODPRT=3 gives printout for time steps 1,4,7,10,...
etc.
Default: MODPRT=1

MPRT - Same as MODPRT (currently used to replace MODPRT in subroutine GFSBX1).

NWTBL - Current number of points for the water table.

MXBAND - Maximum permitted band-width.

MXEDTN(3) - Array containing the nodes at which the maximum change from the previous time step has occurred for pressure, fluid and rock temperatures.

MXE1IN(3) - Array containing the nodes at which the maximum iteration errors have occurred for pressure, fluid and rock temperatures (most recent iteration).

MXE2IN(3) - Array containing the nodes at which the maximum iteration errors have occurred for pressure, fluid and rock temperatures (previous iteration).

MXFRI - Maximum permitted number of categories that may be treated with regard to the dimensions of work array QFRI, PFRI, XFRI, CFRI, RPERM, XLCAVI, FRWDI and QFACI.

MX21 - Maximum permitted number of nodes per element in the element connectivity matrix.

MXIMAX - Maximum permitted number of material properties for unsaturated flow.

MXITER - Maximum permitted number of iterations with a time step.

MXFLX - Maximum permitted number of element sides subject to prescribed boundary flux.

MXLBHH - Specifies the sizes of the work arrays HH and LB. The value of this parameter should not be less than NP*NDOF.

MXLFRC - Maximum permitted number of element numbers to be input to the array LFRACS.

MXNDE - Maximum number if nodes that current element type may take.

MXNMS - Maximum permitted number of concentrated mass sources.

MXNE - Maximum permitted number of elements.

MXNN - Maximum permitted number of unknowns in the global matrix system.

MXNMS - Maximum permitted number of point (fluid) sources).

MXNP - Maximum permitted number of nodal points.

MXNDLX - Maximum permitted number of element sides that could be specified using the arrays: DFLX,JLDLU and JUSIDE.

MXNPRT - Maximum permitted number of nodes to be selected for printout.

MXNSTP - Maximum permitted number of time step for which the temperature history will be updated.

MXSEEP - Maximum permitted number of possible seepage nodes.

MXSTEP - Maximum permitted number of time steps.

MXWTBL - Maximum permitted number of points for water table.

MXNEB - Maximum permitted number of block elements.

MXNBN - Maximum permitted number of block nodes.

MXNFEL - Maximum permitted number of fracture elements.

MXNFN - Maximum permitted number of fracture nodes.

NBN - Number of block nodes.

NBTN - Number of nodes along the bottom boundary selected for specification of boundary conditions,etc.

NBTNMX - Maximum permitted nodes along the bottom boundary that may be selected and stored in the array JBTN.

NCEL - Element counter for boundary flux elements.

NCHK3D - Number of element sides of 3-D elements.

NCHK2D - Number of element sides of 2-D elements.

NCEL - Auxiliary element counter used during element assembly of prescribed boundary fluxes.

NDSPRT(100) - Array for node numbers to be printout.

NDLX - Current number of element sides subject to prescribed boundary displacements.

NDEL - Auxiliary element counter used during element assembly of prescribed boundary displacements.

NDIM - Number of spatial dimensions.

NDOF - Number of degrees of freedom per node.

NDS(21) - Array for global node numbers associated with current element.

NE - Total number of elements.

NEK - Number of unknowns in a element matrix system.

NEX - Element counter.

NFLX - Number of element sides subject to prescribed flux.

NERI - Number of fracture categories.

NEB - Number of block elements.

NETOT - Total number of block and fracture element.

NEL - Current number of element sides subject to prescribed boundary fluxes.

NELG1 - Internal element counter in gas phase.

NELG2 - Internal element counter in the gas phase.

NELW1 - Internal element counter in the water phase.

NELW2 - Internal element counter in the water phase.

NELG - Number of element in gas phase.

NELGWI - Number of interface elements.

NELW - Number of element in water phase.

NELI - Number of interface elements.

NFEL - Number of fracture elements.

NEX - Element counter.

NFN - Number of fracture nodes.

NHS - Number concentrated heat sources specified using the matrices WHS and IHS.

NNPSDE - Number of nodes per element side.

NG - Number of sampling points in the Gauss quadrature formulae.

NGX - Number of Gauss point in X-drection.

NGY - Number of Gauss point in Y-drection.

NGZ - Number of Gauss point in Z-drection.

NG1 - Number of Gauss quadrature points (in each direction).

NG2 - Number points.

NGT - Total number of Gauss points.

NHMS1 - Pointer to the first implied heat source appearing in array HS.

NHMS2 - Pointer to the last implied heat source appearing in array HS.

NHS - Current number of concentrated heat source nodes.

NL - Number of points at which velocity vectors have been calculated and stored in the arrays: XCE, YCE, ZCE, VCE, VYE and VZE.

NLFRC - Current number of fracture element numbers stored in LFRACS.

NMS - Current number of concentrated mass sources.

NN - Number of equations in global matrix system.

NNODE - Number of nodes of current element.

NODS(21) - Array for global node numbers associated with current element.

NP - Total number of nodal points.

NPRT - Number of nodes selected for printout.

NPSINK - Number of mass sources.

NPSTPS - Number of time steps to proceed before solving for pressure.

NPG - Number of nodes in gas phase.

NPG1 - Internal node counter in gas phase.

NPG2 - Internal node counter in gas phase.

NPW - Number of nodes in water phase.

NPW1 - Internal node counter in water phase.

NPW2 - Internal node counter in water phase.

NPGWI1 - Internal counter of interface element.

NPGWI2 - Internal counter of interface element.

NREADT - An integer to indicate how many time steps that should be read from a previous solution file by subroutine HFNDUV. If a value of zero is set for this parameter the file will be read until the end. The idea of this is to enable a restart of a previous solution not only from the last time step that has been solved for. This is of interest in cases where the previous solution has degenerated at some stage and not been aborted.

NRUN - Realization number where solving for stochastically generated fracture networks.

NS - Number of heat sources associated with current element.

NSEEP - Number of possible seepage nodes.

NSTBRK - Time step for gas break through.

NSIDES - Number of element side.

NSTEP - Current time step.

NSUM - Number of summation terms in the evaluation of the transient heat transfer function.

NTPN - Number of nodes on the top boundary.

NTPNMX - Maximum permitted number of nodes on top boundary that may be specified using the array JTPN.

NTHMXG(3) - Array specifying the number of points used to describe the relationship between saturation and the relative gas-permeability for 3 different materials.

NTHX1 - Number of data points defining the relationship between the saturation and the capillary pressure.

NTHX2 - Number of data points defining the relationship between the saturation and the relative permeability.

NTHX3 - Number of data points defining the relationship between derivatives of the saturation versus capillary pressure and saturation.

NTSTPS - Number of time steps to proceed before solving for temperature.

NXIMAT - Number of material properties input for unsaturated flow conditions.

P(21) - Basis function values at the nodes of current element.

PATM - Pressure of 1 atmosphere (Pascal).

PBFN(21,27) - Matrix to hold basis function values associated with the Gauss points.

PDFN(321,27) - Matrix to hold the derivatives of the basis function values associated with the Gauss points.

PERM1 - Reference value of permeability.
Default: PERM1=PRMREF

PEPS - Maximum tolerated pressure difference (DP).

PFRI(15) - Array for equivalent fracture permeabilities of each fracture category.

PHC(41,3) - Matrix to hold the curves for saturation versus capillary pressure for 3 different materials.

PHN(21) - Array for pressure at nodal points associated with current element.

PHERTL - A tolerance value of the iteration error in the solution for pressure.

PHERMX - Maximum iteration error encountered in the solution for pressure.

PHMXDT - Maximum change in pressure from the previous time step.

PHMXTL - Tolerance value for maximum permitted nodal change in pressure between current and previous time step. If the change in pressure exceeds this value then the time step (DT) will be reduced by a factor of 0.5. IDTRED is the maximum permitted number of time step reductions before the job will be aborted.

PHMNTL - See explanation to PHMXTL.

PNEW - Adjusted pressure after N.R. iteration.

POLD - Current pressure at cavern.

POR - Porosity.

POR1 - Porosity.

POREF - Reference value of porosity.

PREF - Reference value of pressure.

PR - Poisson's ratio.

PRMREF - Reference value of permeability.

QC - Prescribed flux at cavern boundary.

QEPS - Maximum tolerated error in total flux calculation.

QFRI(15) - Array for fluxes per unit area of respective fracture class.

QFACI(15) - Array for multiplication factors associated with each fracture category to relate the flux per unit area to the total flux for the respective fracture category.

QMG1 - Gas flux at bottom boundary.

QMG2 - Gas flux at top boundary.

QMW1 - Water flux at bottom boundary.

QMW2 - Water flux at top boundary.

QNEW - Computed flux after N.R. iteration.

QOLD - Current flux at cavern boundary.

RCHAR - Characteristic block radius.

RPERM(15) - Rock permeability associated with respective fracture category.

RUNMAX - Maximum permitted CPU-time (seconds). This to avoid an abnormal end of the execution by exceceeding the maximum permitted CPU-time as specified on the JOB-card. The time-taking is performed in subroutines *GFSBX1*, *HFELFX* and *HFELT1*. The following computer residential subroutines are used:
XTIME0 - to initialize the clock
XTIME - to measure the time elapsed after the initialization of the clock

SCF - Specific heat capacity of the fluid.

SCR - Specific heat capacity of the rock.

SGR(3) - Array to hold residual gas saturation values for three different materials.

SLIR(3) - Array to hold residual gas saturation values for three different materials.

SLOPE - Unidirectional slope of the upper boundary.

SPRIME - Imposed value of the derivative of pressure versus saturation (Usage is controlled by ISPRIM below).

SX - Plot range in the x-direction.

SY - Plot range in the y-direction.

SWN(21) - Array for saturation at the nodes of current element.

TCF(3,3 - Matrix to hold the tensor components of the thermal conductivity of the fluid.

TCR(3,3) - Matrix to hold the tensor components of the thermal conductivity of the rock.

TCHS - Total effect of the concentrated heat sources.

TDIF - Thermal diffusivity.

TDHS - Total effect of the distributed (radioactive) heat sources.

TENRGR - Total energy in the rock medium.

TEX(20) - Alphanumeric information to identify current problem.

TEX2(20) - Alphanumeric information to identify the element mesh.

TEX3(20) - Alphanumeric information to identify previous execution (used for restart).

TFN(21) - Array for fluid temperatures at the nodes of current element.

TFERTL - Tolerance value of the iteration error in the solution for the fluid temperature.

TFERMX - Maximum iteration error encountered in the solution for the fluid temperature.

TFLXT1 - Conductive energy transport though all element sides for the fluid.

TFLXT2 - Convective energy transport though all element sides for the fluid.

TFLXT3 - Conductive energy transport though all element sides for the rock.

TFMXDT - Maximum change in the fluid temperature from the previous time step.

THEAT - Total heat content in the fluid medium.

THEATR - Total heat content in the rock medium.

THXG(41,3) - Matrix to hold saturation values associated with relative gas permeability curves for three different materials.

THX1(20,3) - Array for data points defining the relationship between saturation and capillary pressure.

THX2(20,3) - Array for data points defining the relationship between saturation and relative permeability.

THX3(20,3) - Array for data points defining the relationship between saturation and the derivatives of the saturation versus capillary pressure.

TIME - Time.

TIME1 - Time until subtime steps are to be solved for (see option ICP1(13)).

TLOAD - Total load used in test examples for comparisons with analytical solutions for plane stress-strain using the Kolosoff-Muskhelishvili method: Fung,Y.C. (1965),"Foundations of Solid Mechanics", Prentice-Hall.

TNRG - Total energy (associated with the heat) in the system.

TREF - Reference value of temperature.

TRERTL - Tolerance value of the iteration error in the solution for the rock temperature.

TRERMX - Maximum iteration error encountered in the solution for the rock temperature.

TRMXDT - Maximum change in rock temperature from the previous time step.

UNSLBL(15,3) - Matrix for alphanumeric text description(max.60 characters) of 3 different materials.

VALFA - Angle.

VEC(8) - See description of element matrix systems.

VEC1(8) - See description of element matrix systems.

VEC2(8) - See description of element matrix systems.

VELMAX - The maximum discharge encountered at the points (XCE, YCE, ZCE) selected for calculation of discharges.

VSCREF - Reference value of the dynamic viscosity of the fluid

VXE(625) - Array for x-components of the specific discharge vectors.

VYE(625) - Array for x-components of the specific discharge vectors.

VZE(625) - Array for x-components of the specific discharge vectors.

W11,W12 - Weights to be applied to the two most recent iterative solutions.

W21,W22,W23 - Weights to be applied to the three most recent iterative solutions.

WFLXLB - Sum of the fluxes through all element sides.

WG(4) - Array for weights to be given to the sampling points in the quadrature formulae.

WG1(4) - Array for weights associated with Gauss quadrature points (see XG1).

WGX(4) - Weights associated with X-coordinates.

WG2(4) - Array for weights associated with barycentric coordinates (see XG2, XG2).

WGY(4) - Weights associated with Y-coordinates.

WGZ(4) - Weights associated with Z-coordinates.

WHS(8) - Array for the heat sources at the nodal points of current element.

WLV1 - Water level in the well. This applies to test cases used for model verification.

WLV2 - Water level at the lateral boundary. This applies to test cases used for model verification.

WPSINK(5) - Array for point sinks with fluid.

WSHF - Block shape factor
Default: WSHF=1.0

WTBL - Initial level of the water table. This applies to test cases used for model verification.

WSUM - Total exchange of heat between the fluid and the rock media.

WTGT(27) - Array to hold weights associated with the Gauss points.

XCE(625) - Array for x-coordinates of points at which specific discharge vectors are to be calculated.

XF1, XF2 - X-coordinates (used in conjunction with YF1, YF2, ZF1, ZF2) to specify a test region, whose material properties may be modified by various user supplied input subroutines.

XFRI(125,15) - Auxiliary matrix for the geometrical descriptions of each fracture category.

XI2(8) - Array for local x-coordinates of the nodal points of a 2-D parabolic element.

XI3(8) - Array for local x-coordinates of the nodal points of a 3-D hexahedral (8 nodes) element.

XI4(21) - Array for local x-coordinates of the nodal points of a 3-D hexahedral (8-21 nodes) element.

XLCAVI(15) - Array for representative lengths along caverns for each fracture category.

XLGAS - Gas saturation (expressed as length along segment).

XLWAT - Water saturation (expressed as length along segment)

XLGWI - Length of interface element. The element is split into two halves where one is for gas and the other one for water.

XLTOT - Total length of segment.

XMIN,XMAX - Extreme values in the X-direction of the flow domain.

XG(4) - X-coordinates of Guass points.

XG1(4) - Array for Gauss quadrature points.

XG2(4) - Array for barycentric coordinates.

XORG - X-origin on the graphical display.

XREF - Reference coordinate in the x-direction.

XWTBL(125) - Array for X-coordinates of water table.

XYZ(3,21) - Matrix to hold coordinates of the nodal points associated with current element.

YCE(625) - Array for y-coordinates of the points at which specific discharge vectors are to be calculated.

YF1, YF2 - Y-coordinates (used in conjunction with XF1, XF2, ZF1, ZF2) to specify a test region, whose material properties may be modified by various user supplied input subroutines.

YG(4) - Y-coordinates of Guass points.

YG2(4) - Array for barycentric coordinates.

YMIN,YMAX - Extreme values in the y-direction of the flow domain.

YORG - Y-origin on the graphical display.

YREF - Ditto in the y-direction.

YWTBL(125) - Array for Y-coordinates of water table.

ZCE(625) - Array for z-coordinates of the points at which specific discharge vectors are to be calculated.

ZE3(8) - Array for local z-coordinates of the nodal points of a 3-D hexahedral (8 nodes) element.

ZE4(8) - Array for local z-coordinates of the nodal points of a 3-D hexahedral (8-21 nodes) element.

ZF1, ZF2 - Z-coordinates (used in conjunction with XF1, XF2, YF1, YF2) to specify a test region, whose material properties may be modified by various user supplied input subroutines.

ZG(4) - Z-coordinates of Guass points.

ZMIN,ZMAX - Extreme values in the z-direction of the flow domain.

ZREF - Ditto in the z-direction.

Description of element matrix variables

P r e s s u r e

AM(I,J,1) - Element matrix associated with the time derivative in the fluid flow equation.

AM(I,J,2) - Element matrix associated with the divergence of the flux in the fluid flow equation.

BM(I,1) - Element vector associated with the time gravity in the fluid flow equation.

EKM(I,J,1) - Element matrix associated with the time derivative in the fluid flow equation.

EKM(I,J,2) - Element matrix associated with the divergence of the flux in the fluid flow equation.

FKM(I,1) - Element vector associated with the time gravity in the fluid flow equation.

T e m p e r a t u r e

AM(I,J,1) - Element matrix associated with the heat conduction in the fluid.

AM(I,J,2) - Element matrix associated with the heat conduction in the rock.

AM(I,J,3) - Element matrix associated with the time derivative of the rock temperature.

AM(I,J,4) - Element matrix associated with the time derivative of the fluid temperature.

AM(I,J,5) - Element matrix associated with the convection of the fluid.

BM(I,1) - Element vector associated with the distributed heat sources.

BM(I,2) - Element vector associated with the heat transfer function.

EKM(I,J,1) - Element matrix associated with the heat conduction in the fluid.

EKM(I,J,2) - Element matrix associated with the heat conduction in the rock.

EKM(I,J,3) - Element matrix associated with the time derivative of the rock temperature.

EKM(I,J,4) - Element matrix associated with the time derivative of the fluid temperature.

EKM(I,J,5) - Element matrix associated with the convection of the fluid.

FKM(I,1) - Element vector associated with the distributed heat sources.

FKM(I,2) - Element vector associated with the heat transfer

3 INPUT DATA AND USE OF THE PROGRAM

3.1 MAIN PROGRAM PREPARATION

3.1.1 Nodal data arrays

- 3.1.1.1 Nodal coordinates
- 3.1.1.2 Pressure data arrays
- 3.1.1.3 Fluid temperature data arrays
- 3.1.1.4 Rock temperature data arrays
- 3.1.1.5 Heat transfer function data arrays
- 3.1.1.6 Distributed heat source data arrays
- 3.1.1.7 Rock displacement data arrays
- 3.1.1.8 Gas migration data arrays

3.1.2 Element data arrays

- 3.1.2.1 Element incidences arrays
- 3.1.2.2 Permeability data array
- 3.1.2.3 Work storage

3.1.3 Matrix problem arrays

- 3.1.3.1 Front solution arrays
- 3.1.3.2 Band solution arrays

3.1 MAIN PROGRAM PREPARATION

The main task of the main program HFMAIN is to specify the dimensions of most of the problem dependent arrays. The problem dependent arrays may be divided into the following main categories:

(i) Nodal data

Refers to any information being specified at the nodal points in the element mesh.

MXNPX - Maximum permitted number of nodal points

(ii) Element data

Refers to information valid for the whole element

MXNEX - Maximum permitted number of elements

MXNDE - Maximum permitted number of nodes per element

(iii) Work arrays for the frontal solver

MXFRO - Maximum permitted frontwidth

MXNNX - Parameter which either equal to the frontwidth (out of core solution) or equal to the frontwidth plus the maximum permitted number of unknowns in front solution (in-core solution).
For in-core solution:
```
PARAMETER (MXNNX=MXFROX+MXNPX)
```
For out-of-core solution:
```
PARAMETER (MXNNX=MXFROX)
```

MXWORK - Parameter which is set equal to two times the maximum permitted number of unknowns (e.g. MXWORK = 2*MXNPX)

The previous variables are set in the beginning of the main program using Parameter statements. MXNPX, MXNEX and MXFROX and MXNDE are primary variables, while MXLBHX, MXNNX and MXWORK are evaluated by the parameter statements.

The following parameters are used to specify the size of the arrays dimensioned in HFMAIN:

MXNE - Maximum permitted number of elements

NE - Current number of elements

MXNNDE - Maximum permitted number of nodes per element

NNODE - Current number of nodes per element

MXNP - Maximum permitted number of nodal points

NP - Current number of nodal points

MXNDOF - Maximum permitted number of degrees of freedom per node

NDOF - Current number of degrees of freedom per node

MXNN - Maximum permitted number of solution values in the system of matrix equations (MXNN=MXNP*MXNDOF)

NN - Current number of solution values in the matrix system (NN=NP*NDOF)

MXBAND - Maximum permitted bandwidth in a banded matrix system

MBAND - Current bandwidth

MXFRON - Maximum permitted frontwidth due to the work arrays EQUAT , PVCOL, VECRV, GLOAD and NACVA. These arrays are used by the front solution subroutines HFRON1 and HFRON2

The following set of arrays are dimensioned in the main program:

3.1.1 Nodal data arrays

3.1.1.1 Nodal coordinates

NODES - Node numbers

XN - X-coordinates

YN - Y-coordinates

ZN - Z-coordinates

3.1.1.2 Pressure data arrays

PH - Values of pressure

LBPH - Codes (=0 active value, =1 prescribed value)

PH1I - Values of the pressure at the last iteration

PH2I - Values of the pressure at the last but one iteration

PHPTS - Values of the pressure at the previous time step

PHDT - Values of the pressure difference between current and previous time step

PHDX - Values of specific (Darcy) velocities in x-direction

PHDY - Values of specific (Darcy) velocities in y-direction

PHDZ - Values of specific (Darcy) velocities in z-direction

LBPHDX - Codes associated with PHDX

LBPHDY - Codes associated with PHDY

LBPHDZ - Codes associated with PHDZ

PERM - Values of the intrinsic permeability

VISCF - Values of the dynamic viscosity of the fluid

DENSF - Values of the fluid density

PORSTY - Nodal array for porosity at respective node

NJN - Work area used by subroutine HFLXNU in the calculation of the discharge vectors at the nodal points

SW - Values of saturation.

3.1.1.3 Fluid temperature data arrays

TF - Values of the fluid temperature

LBTF - Codes associated with TF (=0 active value, =1 prescribed value)

TF1I - Values of the fluid temperature at the last iteration

TF2I - Values of the fluid temperature at the last but one iteration

TFPTS - Values of the fluid temperature at the previous time step

TFDT - Values of the fluid temperature difference between current time step and the previous time step

3.1.1.4 Rock temperature data arrays

TR - Values of the rock temperature

LBTR - Codes associated with TF (=0 active value, =1 prescribed value)

TR1I - Values of the rock temperature at the last iteration

TR2I - Values of the rock temperature at the last but one iteration

TRPTS - Values of the rock temperature at the previous time step

TRDT - Values of the rock temperature difference between current time step and previous time step

3.1.1.5 Heat transfer function data arrays

TIRA - Initial temperatures of the rock

HFWA - Values of the heat transfer function

HFWA1 - Values of the heat transfer function at the previous time step

HFWA2 - Values of the heat transfer function at the last but one time step

3.1.1.6 Distributed heat source array

DHS - Array of heat source strength at nodal points

3.1.1.7 Rock displacements

DU(NP,3) - Matrix for rock displacement vectors.

LBDU(NP,3) - Associated matrix for codes indicating if the displacements at a node is free or prescribed.

3.1.1.8 Gas migration data arrays

PG - Values of the rock temperature

LBPG - Encoded integer array associated with PG (= 0 active value, = 1 prescribed value)

PG1I - Values of the gas pressure at the last iteration

PG2I - Values of the gas pressure at the last but one iteration

PGPTS - Values of the gas pressure at the previous time step

PGDT - Values of the gas pressure difference between current time step and previous time step

AF11 - Coefficients associated with time derivatives of the fluid pressure (Fluid flow equation).
$\phi \rho^l \left[S \left(c^l + c^r \right) - S' \right]$

AF12 - Coefficients associated with time derivatives of the gas pressure (Fluid flow equation).
$\phi \rho^l S'$

AG11 - Coefficients associated with time derivatives of the fluid pressure (Gas flow equation).

$$\phi \left(1 - S_{wir} \right) p^g \frac{M}{ZRT} S'$$

AG12 - Coefficients associated with time derivatives of the gas pressure (Gas flow equation).

$$\phi \left(1 - S_{wir} \right) \frac{M}{ZRT} \left[(1 - S)\left(1 + c^r p^g \right) - p^g S' \right]$$

ZRTM - $\dfrac{M}{(ZRT)}$

3.1.2.1 Element data arrays

IELNOD - Matrix of element incidences (MXNE,MXNNDE)

IELTYP - Array of element type codes
= 1 1-D three-nodes quadratic element
= 5 2-D eight node parabolic element
= 7 3-D hexahedral eight node element
=11 3-D hexahedral curvilinear 8-21 nodes element
=12 3-D prismatic curvilinear 6-15 nodes element

IELMAT - Element matrix for material number and phase code for respective element
IELMAT(IEL,1) is material number for current element
IELMAT(IEL,2) is phase code for current element

IELS - Array for element numbers. Used as an auxiliary array for sorting and renumbering of element incidences.

3.1.2.2 Permeability data array

HCEL - Matrix of the intrinsic permeability in x-, y- or z-direction
HCEL(IEL,1) is permeability in the x-direction
HCEL(IEL,2) is permeability in the y-direction
HCEL(IEL,3) is permeability in the z-direction

3.1.2.3 Work storage

NDFRO - Array used by subroutine CHKFRN to calculate the maximum frontwidth

3.1.3 Matrix problem arrays

3.1.3.1 Front solution arrays

WFRONT - HFRON1: Vector of size MXFRON*(MXFRON+1)/2
HFRON2: Matrix of size MXFRON*MXFRON
HFMAIN: Matrix of size MXFROX*MXNNX

CFRON - Vector of length MXNPX

WFRONX - Work matrix for frontal sowers

IWORK - Integer work matrix for front sowers

Note:
The specific nodal fluxes by subroutine HFLXNU are used only for printout together with the pressure values. The advective fluxes considered in the heat flow equation are evaluated by Gauss quadrature during the element integration.

3.1.3.2 Work arrays for solution of the system of linear equation

WFRONT - Global matrix of size: MXNN*MXBAND
CFRON - Right hand side vector to WFRONT of size MXNN

In solving the system of algebraic equations resulting from the element integration either a front or a band solution technique has been used. When using the frontal method, the parameters MXNN and MXBAND will have a different meaning than the explanation that was given previously for these parameters.

In the frontal method the forward elimination procedure according to Gauss's elimination method is performed at the same time as the element matrices are being assembled. Thus, only limited a portion of the matrix system needs to be held in core throughout the solution process. This will in most cases substantially reduce the size of the core required to solve the matrix problem. The part of the matrix held in core is called the 'front' and the number of variables involved in the front determines the 'frontwidth'. The frontwidth will depend upon the structure of the problem and the order in which the element matrices are being assembled, but not on the numbering of the nodes. The computer time is roughly proportional to the number of nodal variables multiplied by the square of the frontwidth.

There is no direct formula available to calculate the frontwidth, as for band-matrix problems. Therefore, the program checks current frontwidth before entering either of the two front solvers, which are employed by the program. The maximum size of the front- width encountered in the problem will then be checked against the size of the work arrays associated with the front solution subroutines.

First the size of the work array WFRON(MXNN,MXBAND) will be examined. For the symmetric front solver (HFRNF1) the permissible front width with regard to the dimensions of this array is equal to

```
MFRON=SQRT(2*MXNN*MXBAND + 0.25)
```

and for the non-symmetric front solver (HFRNF2) the permissible frontwidth is equal to

```
MFRON=SQRT(MXNN*MXBAND)
```

The second step is to compare the value of MFRON with the value of MXFRON representing the dimensions of the work arrays WFRONX, and IWORKX. The smallest value of MFRON and MXFRON represents the largest permissible frontwidth in the program. Finally, this value will be checked against the actual frontwidth encountered in the program.

3.2 INPUT DATA PREPARATION

3.2.1 Compulsory input

The input is grouped into three major categories:

a) Specification of array sizes and selection of subroutines. This is performed in the Main program.

b) Encoded control and parameter data and physical parameters.

c) User controlled input for boundary and initial conditions, permeability and porosity, etc.

Data set 1----------Subroutine selection

5-10 A6 CHKSTP Subroutine for time march scheme. Input and initialization of time step data are perfor med by an alternative entry *SETSTP*. Currently the following subroutines are implemented:

CHKSTP, CHKST1, CHKST2, CHKST3, CHKST4

CHKSTP is a dummy subroutine that will will have no effect and the time stepping will be performed according to the values of DT and DTFAC (see below).

CHKST1 is subroutine for logarithmic time time stepping. The following input record must be supplied:

```
READ(INUN,820) DTIMEP(1),NCYCLE,
                NFAC,
               (FAC(I),I=1,NFAC)
     820 FORMAT(F10.0,2I5,10F5.0)
```

where

DTIME(1) - Initial subinterval
NCYCLE - Number of cycles
NFAC - Number of subintervals within each cycle
FAC - Array of length -NFAC- containing factors for subintervals

Example of input:

```
DTIME(1)=3.15576e7,NCYCLE=4,NFAC=7
FAC(1)=0.5,FAC(2)=0.5,FAC(3)=1.0,
FAC(4)=1.0,FAC(5)=2.0,FAC(6)=2.0,
FAC(7)=2.0
```

Output:

```
DTIME(2)=1.58e7, DTIME(3)=1.58e7,
DTIME(4)=3.16e7, DTIME(5)=3.16e7,
DTIME(6)=6.31e7, DTIME(7)=6.31e7,
DTIME(8)=6.31e7, DTIME(9)=1.58e8,
DTIME(10)=1.58e8, DTIME(11)=3.16e8,
etc.
```

Note:
the subintervals were here multiplied with 365.25*86400 to get the times in seconds

			CHKST4 is a subroutine that will perform the time stepping according to the input being read by the alternative entry SETST4. Input description: Record 1: Optional text Record 2: Maximum permitted number time points, Factor for conversion to SI-time units (seconds) FORMAT(I5,F15.0) Record 3: Time points (free format)
15-20	A6	SFRIN1	Subroutine for input of boundary and initial conditions for two-phase flow
25-30	A6	SFRUT1	Subroutine for print out of two-phase flow solution values
15-20	A6	HFINS1	Main control subroutine for input and initialization

Data set 1x----------Time stepping input data (see previous data set)

Data set for time step control data according to the previous format specification for CHKST1 or CHKST4

Data set 2----------Maximum CPU-time and input/output unit number (GFSBX1)

Cols	Format	Identifier	Explanation
1-5	F5.0	RUNMAX	Maximum permitted CPU-time (seconds). This to avoid an abnormal end of the execution by exceeding the maximum permitted CPU-time as specified on the JOB-card. The time-taking is performed in subroutines *GFSBX1*, *HFELFX* and *HFELT1*. The following computer residential subroutines are used: XTIME0 - to initialize the clock XTIME - to measure the time elapsed after the initialization of the clock

New record----------Specification of input and output data sets

1-5	I5	INUN(1)	Logical unit number for input to HFINPT, HFNDIN and HFELIN Default: INUN(1)=5
6-10	I5	INUN(2)	Logical unit number for input to subroutines HFSRC1, HFBCX1 and SELNDS Default: INUN(2)=5

11-15	I5	INUN(3)	Logical unit number for data from a previous execution to be read in. This is to continue the execution using solution values from the most recent time step in a previous execution as initial values. This is carried out by subroutines HFNDUV and HFNDU1. If no data are to be input in this way, then set: INUN(3)=0
16-20	I5	INUN(4)	Output unit used for storage of of input to subroutines HFINPT, HFNDIN and HFELIN Default: INUN(4)=21
21-25	I5	INUN(5)	Output unit for solution values of pressure and temperatures at each time step. This is carried out by subroutine HFSAVD. This output may later be used as input using INUN(3) Default: INUN(5)=22
26-30	I5	INUN(6)	Input unit for unsaturated data. Default: INUN(6)=55
31-35	I5	INUN(7)	Input unit for stress data. Default: INUN(7)=56
36-40	I5	INUN(8)	Input unit for element mesh. Default: INUN(8)= 5
41-45	I5	INUN(9)	Input unit for drain data Default: INUN(9)=57

New record----------Intermediate data sets

Cols	Format	Identifier	Description
1-5	I5	IUNIT(1)	Logical unit for storage of pressure equations
6-10	I5	IUNIT(2)	Logical unit for storage the right hand sides of pressure equations
11-15	I5	IUNIT(3)	Logical unit for storage of the temperature equations
16-20	I5	IUNIT(4)	Logical unit for storage of the right hand sides of the temperature equations
21-25	I5	IUNIT(5)	Logical unit for storage of the element matrices for the pressure equations
26-30	I5	IUNIT(6)	Logical unit for storage of the element matrices for the temperature equations
31-35	I5	IUNIT(7)	Currently not used
36-40	I5	IUNIT(8)	Currently not used
41-45	I5	IUNIT(9)	Logical unit for intermediate storage of frontal equations Default unit number: IUNIT(9) = 03

46-50	I5	IUNIT(10)	Logical unit for intermediate storage of the right hand sides of the frontal equations Default unit number: IUNIT(9) = 04

Note: A blank or a zero field will cause a default setting. A negative unit number indicates that the data set never should be opened.

Data set 3----------Problem identification

Cols	Format	Identifier	Explanation
1-80	20A4	TEX	Title of the problem.
1-80	20A4	TEX3	Supplementary title information. This information is only for printout and is is not kept.

Data set 4----------Problem identification of a previous run

Cols	Format	Identifier	Explanation
1-80	20A4	TEX3	Title of the previous execution.

Data set 5----------Solution parameters

Cols	Format	Identifier	Explanation
1-5	I5	IOPSLV	Solution parameter to specify the set of equations to be solved = 0 Pressure, Fluid and Rock temperatures to be solved for = 1 Pressure to be solved for = 2 Fluid temperature to be solved for = 3 Pressure and fluid temperature to be solved for = 4 Rock temperature to be solved for = 5 Fluid and rock temperatures to be solved for = 6 Displacements to be solved for = 7 Two-phase flow
6-10	I5	NDIM	Number of spatial dimensions. (1,2 or 3)
		IAXSYM	Coordinate system control parameter = 0 Cartesian coordinate system = 1 axi-symmetric coordinate system
		IUNSAT	> 0 Unsaturated flow
		IMATM	= 0 Element oriented Requires that element permeability HCEL be set = 1 Nodal oriented Requires that nodal permeability array PERM and nodal porosity array PORSTY be set

		IDISCR	Control parameter for mode of discrete modelling > Discrete modelling
41-45	I5	IOPTST	Control parameter to select option in subroutine *HFTEST*. This parameter is used for various ad hoc settings such as Hydrocoin level 1, Case 1,3 and 4, Level 2, Case 1.
		ICRAY	= 1 When running on Cray
		IFNPG1	= 1 Nodal permeability, fluid density, and viscosity are constant with time
		IWTBL	An integer to indicate if the position of the water table is to be located (only applicable to 2-D unsaturated flow problems) IWTBL > 1 water table to be located
31-35	I5	IPLOT	Plot parameter (=0 If no plotting is to be performed, =1 If plotting is to be performed).

Time step control parameters ----------

1- 5	I5	MXSTEP	Maximum permitted number of time steps.
6-10	I5	MXITER	Maximum permitted number of iterations per time step.
21-30	20A4	DT	Time step.
31-35	20A4	DTFAC	Time step multiplier In the present version of the program (subroutine *GFSBX1*) a special subroutine (*CHKSTP*) is called to set the value of the time step multiplier *DTFAC* for each time step. Note also that the maximum permitted time step DTMAX is given by DTMAX=DT*DTMXFC where DTMXFC is a multiplier being defined in subroutine *HFINPT*. Default: DTFAC=1.0
61-65	I5	NPSTPS	Number of time steps to be advanced before solving for pressure Default: NPSTPS=1
66-70	I5	NTSTPS	Number of time steps to be advanced before solving for temperature Default: NTSTPS=1
36-40	I5	MODPRT	Printout control parameter to suppress the printout of intermediate time steps, e.g. MODPRT=1 gives printout for every time step MODPRT=2 gives printout for time steps 1,3,5,7,... MODPRT=3 gives printout for time steps 1,4,7,10,... etc. Default: MODPRT=1

		NREADT	Restart time step in cases where the restart is to be made from a previous solution

Convergence criteria ----------

01-10	F10.0	PHERTL	Tolerance value for convergence in pressure solution Default: PHERTL=1.E-9
11-20	F10.0	TFERTL	Tolerance value for convergence in fluid temperature solution Default: TFERTL=1.E-9
21-30	F10.0	TRERTL	Tolerance value for convergence in rock temperature solution Default: TRERTL=1.E-9
31-35	I5	NGAUSS	Order of Gauss point scheme NGAUSS = 2, 3 or 4. Default: NGAUSS = 3

The following parameters are used in two-phase flow calculations only

	PHMXTL	Maximum permitted change in pressure during current time step
	PHMNTL	Maximum accepted change in pressure during current time step
	IDTRED	Maximum permitted number of time step reductions

Iteration coefficients ----------

1- 5	F5.0	W11	Weighting factor for the most recent iteration in a two point iteration scheme.
6-10	F5.0	W12	Weighting factor for the last iteration but one in a two point iteration scheme.
11-15	F5.0	W21	Weighting factor for the last iteration in a three point iteration scheme.
16-20	F5.0	W22	Weighting factor for the last iteration but one in a three point iteration scheme.
21-25	F5.0	W23	Weighting factor for the last iteration but two in a three point iteration scheme.

Note: HFWAV1: A(I)=W11*A(I) + W12*A1I(I)
HFWAV2: A(I)=W21*A(I) + W22*A1I(I) + W23*A2I(I)

Miscellaneous control parameters ----------

1-5	I5	IPORF	> 0 : POR = FPOR (IEL,XYZ,NP)
6-10	I5	IPERF	> 0 : CALL FPERM2(IEL,XYZ,HCN)
11-15	I5	IOPOR	Control parameter for porosity = 1 Nodal porosity array to be used
16-20	I5	IOPHC	Control parameter for spatial treatment of permeability IOPHC=0 permeability will be defined as an element property IOPHC=1 permeability will be defined as a nodal property

IOPHC=2 permeability values are to be associated with nodes through a function subroutine given by the user

21-25	I5	IMDPTH	Control parameter to average permeability and porosity over each element
26-30	I5	IPRUNS	Control parameter for printout of unsaturated data in subroutine *HFINP4*.
31-35	I5	IPSINK	Control parameter to specify to which phase point sources are to be applied
36-40		IMSTRE	Element matrix control parameter = 0 Element matrix and the R.H.S to be written on separate records = 1 Element matrix and the R.H.S to be written on the same record
41-45		IFMODE	Front solution control parameter = 0 Direct elimination of element matrix systems --- Example of usage of direct elimination: IFMODE = 0 IUNIT(9) = -1 : No auxiliary storage In HFMAIN: Set MXNNX=MXFROX+MXNPX JCL : //GO.FT03F001 DD DUMMY --- Example of usage of direct elimination: and using auxiliary storage: IUNIT(9) = 03 or 00 : Data set for temporary storage of front equations In HFMAIN: Set MXNNX=MXFROX JCL : //GO.FT03F001 DD DSN=&SCRACTCH // ... etc. --- = 1 Element matrix systems to be written to disk before elimination will take place --- Example of usage of non-direct elimination: IFMODE = 1 IUNIT(9) = 03 or 00 : Data set for temporary storage of front equations In HFMAIN: Set MXNNX=MXFROX JCL : //GO.FT03F001 DD DSN=&SCRACTCH // ... etc. ---
46-50		IPRCHK	Front solution printout parameter
51-55		IFRM	Read/write control parameter in front solver
56-60		MISCHK	General printout control parameter
61-65		IDNFMS	Mass source specification parameter = 0 Mass source input = 1 Volumetric source input

Data set 6----------Miscellaneous control parameters

Cols	Format	Identifier	Explanation
1-40	20I2	ICP1	Array of control parameters

ICP1(1) - Control parameter to select equation solver (GFSBX1, HFELFX, HFELT1)
= 0 Band matrix solver to be used
= 1 Frontal solver to be used

ICP1(2) - Solution control parameter (GFSBX1, HFINPT)
= 0 Non-steady state solutions to be obtained
= 1 Steady state solutions to be obtained

ICP1(3) - Control parameter for the storage of solution data on disc for subsequent use (GFSBX1, HFINPT)
= 0 Solution data are not to be stored
= 1 Solution data are to be stored

As a provisional arrangement this parameter is currently also set in subr. GFSBX1 prior to the call on HFINPT. The purpose of this is to indicate whether the input data are to be ouput on a disc file as well.

ICP1(4) - Printout control parameter (GFSBX1)
= 0 No action to be taken
= 1 Initial values to be printed

ICP1(5) - Printout control parameter (GFSBX1)
= 0 No action to be taken
= 1 Results from each time step to be printed

ICP1(6) - Printout control parameter (GFSBX1)

= 1 Results of each iteration within a time step to be printed

ICP1(7) - Printout control parameter (GFSBX1)
= 0 No action to be taken
= 1 Element incidences to be printed

ICP1(8) - Printout control parameter (EVDRV1)
= 0 No action to be taken
= 1 Calculations of derivatives at nodal points to be printed

Subroutine EVDRV1 is called from subroutine HFNFLX during the evaluation of fluid velocities

ICP1(9) - Control parameter for the storage of the basic integration matrices (HFELT1)
= 0 No action to be taken
= 1 Basic integration matrices to be stored on disc for subsequent use

ICP1(10) - Gravity control parameter
= 0 No action to be taken
= 1 Gravity effects not to be considered (HFINPT)

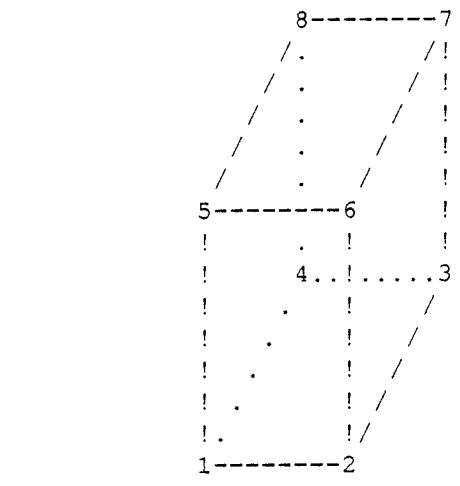
ICP1(11) - Printout control parameter (PHSLVF, PHSLVB)
= 0 No printout to be performed
= 1 Solution values to be printed
= 2 Message on global element assembly to be output
= 3 Message on the application of the boundary conditions to the global matrix to be output
= 4 Global matrix to be printed
= 5 Global matrix to be printed after the boundary conditions have been applied

ICP1(12) - Printout control parameter (TMSLVF, TMSLVB)
= 0 No printout to be performed
= 1 Solution values to be printed
= 2 Message on global element assembly to be output
= 3 Message on the application of the boundary conditions to the global matrix to be output
= 4 Global matrix to be printed
= 5 Global matrix to be printed after the boundary conditions have been applied

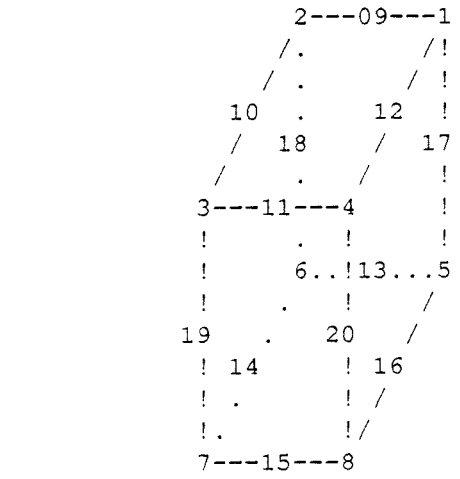
ICP1(13) - Control parameter to indicate that the first time step *DT* is to be divided into ICP1(13) sub time steps
= 0 No action to be taken
> 0 Set DT=DT/ICP1(13)

ICP1(14) - Control parameter to specify type of hexahedral element to be input (GFSBX1, HFIELN, HFLGA1, HFLGA2). Currently this option is set in subroutine *HFTEST* using the parameter *IEL21*.

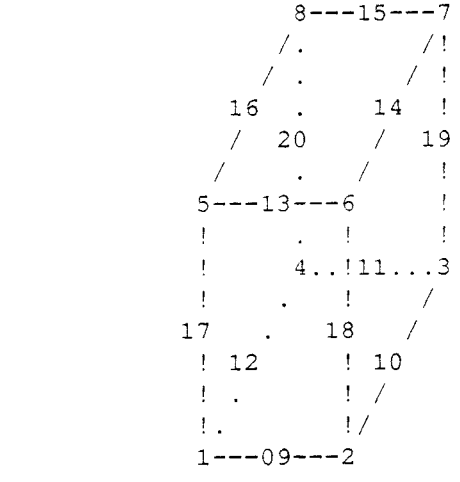
IEL21 = 0 Standard 8 node hexahedral element to be used



IEL21 = 1 Curvilinear isoparametric hexahedral element (8-21 nodes) defined as shown below



IEL21 = 2 Curvilinear isoparameteric hexahedral element (8-21 nodes) defined as shown below



This option causes a call on subroutine *HFIELN* from subroutine *GFSBX1* to redefine the element incidences according to the previous definition.

ICP1(15) - Printout control parameter (CHKFRN)
= 0 No action to be taken
= 1 Printout the frontwidth at each element assembly

ICP1(16) - Printout control parameter (HFELFX, HFELT1)
= 0 No action to be taken
= 1 Messages to be output

ICP1(17) - Input control parameter (GFSBX1)
= 0 No action to be taken
= 1 Essential boundary conditions on the pressure head to be input by calling subroutine HFBCX1

ICP1(18) - Output control parameter (HFNFLU)
= 0 No action to be taken
= 1 Flow velocities at nodal points to be computed and output
Note: This option is currently not used in the present version of the model program (*GFSBX1*).

ICP1(19) - Output control parameter (VELXY3)
= 0 No action to be taken
= 1 Flow velocities to be evaluated at interior points of elements, e.g. Gauss' points
= 2 Flow velocities to be printed (VELXY3)

ICP1(20) - Point heat source/sink control parameter (HFELT1, HFSCR1)
= 0 Heat source/sink to be applied to both fluid and rock media
= 1 Heat source/sink to be applied to the fluid medium only
= 2 Heat source/sink to be applied to the rock medium only

Continued on new record----------

Cols Format Identifier Explanation

1-40 20I2 ICP1 Array of control parameters

ICP1(1) - Control parameter to select equation solver (GFSBX1,

ICP1(21) - Permeability control parameter (*GFSBX1*)
= 0 No action to be taken
= 1 Permeability matrix HCEL to be initialized by the value of the reference permeability as given by HCREF.

ICP1(22) - Printout control parameter (HFWFUN)
= 0 No printout to be performed
= 1 Heat transfer function values to be printed
= 2 Heat transfer function values to be printed. This is a special option for 1-D analysis using 2-D elements

ICP1(23) - Printout control parameter (HFWFN1)
= 0 No printout to be performed
= 1 Results to be printout
= 2 Results of the summations also to be included
= 3 Full printout to be performed

ICP1(24) - Newton-Raphson iteration is to be performed
= 0 No action to be taken (Direct iteration)
= 1 Newton-Raphson iteration

ICP1(25) - Printout control parameter (HFLMU1)
= 0 No action to be taken
= 1 Element matrices formed by subroutine HFLMU1 to be output
Note: Subroutine HFLMU1 is called from subroutine TMSLVB

ICP1(26) - Printout control parameter (HFELT1)
= 0 No action to be taken
= 1 Information of the material properties of the elements to be output by calling subroutine HFELU1

ICP1(27) - Printout control parameter (HFLGA1, HFLGA2)
= 0 No action to be taken
= 1 Basis function values to be printed out

ICP1(28) - Printout control parameter (INTEG7, INTEG8)
= 0 No action to be taken
= 1 Fluid density, hydraulic conductivity and hydraulic gradient to be printed for each Gauss point

ICP1(29) - This option parameter is currently not used
= 0 No action to be taken

ICP1(30) - This option parameter is currently not used
= 0 No action to be taken

ICP1(31) - Printout control parameter (GFSBX1).
= 0 No action to be taken
= 1 Fluid temperature records to be output
Each record contains the temperature at each time step for a node
Note: This option is of interest when the transient heat transfer function is to be applied

ICP1(32) - Heat transfer function control parameter (GFSBX1, HFINPT, SOLVUT, HFPLT1A, INTEG9))
= 0 No action to be taken
= 1 Temperature history to be updated
= 2 Quasi-steady state heat transfer function to be applied
= 3 Transient heat transfer function to be applied

ICP1(33) - Input control parameter (HFTEST)
= 0 No action to be taken
= 1 See subroutine *HFTEST*
Note: Subroutine *HFTEST* has been used for

experimental purposes to specify non-standard input, to override various input data temporarily, etc.

ICP1(34) - Control parameter for pressure equation (GFSBX1)
= 0 No action to be taken
= 1 Pressure equation to be solved once and for all
This option is usually applied to cases with steady state pressure conditions.

ICP1(35) - Output control parameter for material and energy balances (GFSBX1)
= 0 Results to be output of the last time step only
= 1 Results to be output at each time step

ICP1(36) - 1-D Analysis control parameter (HFPLT1A)
= 0 No action to be taken
= 1 1-D analytical solution to be obtained and plotted

ICP1(37) - Time step control parameter (GFSBX1)
= 0 Time step is to be reduced if the iteration errors are too large
= 1 Turn off time step reduction

ICP1(38) - Solution control parameter for temperature (HFINPT, HFPLT1A, HFMTF1, TFEBL2)
= 0 No action to be taken
= 1 Fluid and rock media to be treated as a single equivalent medium

ICP1(39) - Heat transfer function parameter to alternate between the transient and quasi-steady state heat transfer function.
= 0 No action to be taken
= 1 To first apply quasi-steady state heat transfer function and then to iterate using transient heat transfer function

ICP1(40) - Solution control parameter for temperature equations when applying quasi-steady state heat transfer function.
Note: This parameter is used only for testing (*GFSBX1*).
= 0 No action to be taken
= 1 Switch from implicit to explicit solution for the first iteration

Physical properties of the fluid ----------

Cols	Format	Identifier	Explanation
1-10	F10.0	CF.	Compressibility of the fluid
11-20	F10.0	CR	Compressibility of the rock
21-30	F10.0	DNFREF	Reference value of fluid density
31-40	F10.0	VSCREF	Reference value of the dynamic viscosity of the fluid
41-50	F10.0	PREF	Reference value of pressure

51-60 F10.0 TREF Reference value of temperature

Acceleration of gravity ----------

Cols	Format	Identifier	Explanation
1-10	F10.0	G	Acceleration of gravity The gravity component is stored in vector GRAVTY(3) according to: GRAVTY(G,0,0) for NDIM=1 GRAVTY(0,G,0) for NDIM=2 GRAVTY(0,0,G) for NDIM=3
11-20	F10.0	SPRIME	Imposed value of the derivative of pressure versus saturation (Usage is controlled by ISPRIM below)
21-30			Coefficient for the fluid transfer between blocks and fractures
31-35		ISPRIM	Control parameter for usage of SPRIME

Reference values of porosity and permeability - Anisotropy factors-

1-10	F10.0	POREF	Default value of porosity or the reference value of porosity
1-10	F10.0	PRMREF	Default value of intrinsic permeability or the reference value of permeability
36-40	F5.3	FHCX	Multiplier for Anisotropy in the permeability in the x-direction. Default: FHCX=1.0
41-45	F5.3	FHCY	Multiplier for anisotropy in the permeability in the y-direction. Default: FHCY=1.0
46-50	F5.3	FHCZ	Multiplier for anisotropy in the permeability in the z-direction. Default: FHCZ=1.0

Note: Currently the multipliers for anisotropy are applied in function subroutine *FUNHC1*

Data set 9----------Input of porosity and permeability function coefficients

Cols	Format	Identifier	Explanation
11-20	F10.0	POR1	Reference value of porosity. Default: POR1=POR
1-10	F10.0	CPOR	Coefficient used in function *FPOR2* to specify the variation in porosity with depth. Default: CPOR=0.0
31-40	F10.0	PERM1	Reference value of permeability. Default: PERM1=PRMREF
21-30	F10.0	CPERM	Coefficient used in function *FPERM2* to specify the variation in permeability with depth. Default: CPERM=0.0

41-50	F10.0	FRPERM	Coefficient to specify permeability of a fracture zone versus the surrounding rock. The permeability of the fracture zone is obtained by multiplying FRPERM by the rock permeability. Default: FRPERM=1.0

Reference coordinates and unidirectional slope of flow domain -----

61-65	F5.0	XREF	Reference coordinate in the x-direction
66-70	F5.0	YREF	Ditto in the y-direction
71-75	F5.0	ZREF	Ditto in the z-direction
71-75	F5.3	SLOPE	Unidirectional (fractional) slope. E.g. 0.005 means a slope of 5 in 1000. The value of the slope is tranferred to subroutines *MTOP2D* and *BND2DN* in 2-D, or *MTOP3D* and *BND3DN* in 3-D, see also Data set No 17 in the sequal (IOP=8 and IOP=9).

Thermal properties of the fluid ----------

Cols	Format	Identifier	Explanation
31-40	F10.0	CTF	Coefficient of thermal volume expansion
1-10	F10.0	SCF	Specific heat capacity of the fluid.
1-10	F10.0	TCF	Thermal conductivity of the fluid Default: TCF(I,J)=TCF for I.EQ.J TCF(I,J)=0 for I.NE.J Note: If anisotropy in the thermal conductivity applies then subroutine *HFINPT* should be modified.

Thermal properties of the rock ----------

51-60	F10.0	DR	Rock density
11-20	F10.0	SCR	Specific heat capacity of the rock.
11-20	F10.0	TCR	Thermal conductivity of the rock Default: TCR(I,J)=TCR for I.EQ.J TCR(I,J)=0 for I.NE.J
31-40	E10.3	GEOGRD	Geothermal gradient (fractional). The value of this parameter is transferred to subroutine *PHYDRO*, which is used to set hydrostatic boundary conditions.
21-30	F10.0	HT	Heat transfer coefficient for the quasi-steady state heat transfer function.

$$u = h * (T' - T'')$$

A default value of h is computed as:

$$h = \frac{3 \lambda c s}{\delta a^2} (1 - \phi)$$

where c is the contact area coefficient, s the shape factor and δ is a coefficient being dependent on the assumed function for the heating of the blocks.

Heat transfer function parameters ----------

Cols	Format	Identifier	Explanation
1-5	F5.0	RCHAR	Characteristic block radius Default: RCHAR=1.0
6-10	F5.0	CAREA	Contact area coefficient Default: CAREA=1.0
11-15	F5.0	WSHF	Block shape factor Default: WSHF=1.0
16-20	F5.0	CDR	Coefficient between zero and one to be applied to the quasi-steady state heat transfer function. For T=at CDR=0.20, which is used as default value Default: CDR=0.20
21-25	I5	NSUM	Maximum permitted number of summation terms for the heat transfer function evaluation Default: NSUM=25
26-35	F10.0	EXTOL	Tolerance value to be used as the criterion to stop summation of terms in infinite series at the heat transfer function evaluation Default: EXTOL = 10^{-30}
51-80			This data field is currently not used.
41-50	E10.3	A1	Coefficient in interpolation formula for the radioactive decay of the heat source (a_1 in the formula below).
51-60	E10.3	AL1	Coefficient in interpolation formula for the radioactive decay of the heat source (a_1 in the formula below).
61-70	E10.3	A2	Coefficient in interpolation formula for the radioactive decay of the heat source (a_2 in the formula below).
71-80	E10.3	AL2	Coefficient in interpolation formula for the radioactive decay of the heat source The coefficients for the radioactive decay are stored in the common block HFREF3 and thereby transferred to the function subroutine *DECAY*. ($a_2$in the formula below).

The following formula is used:

$$\frac{Q(t)}{Q(0)} = a_1 e^{-\alpha_1 t} + a_2 e^{-\alpha_2 t}$$

Selection of functions for physical properties:
Fluid density, Fluid viscosity, Gas viscosity,
Permeability with depth and Porosity with depth functions

5-10 A6	FUNDNF	Function for fluid density
		FNDNF1 gives the reference density.
		FNDNF3 approximates the fluid density by a fourth degree polynomial.
		FNDNF4 approximates the fluid density by a linear function. This function was used in Hydrocoin Level 1, Case 4.
		FNDNF6 approximates the fluid density by an exponential function This function was used for Hydrocoin Level 2, Case 1.
15-10 A6	FUNVSC	Function for dynamic viscosity of the fluid
		FNVSF1 gives the reference value as been input to HFINPT.
		FNVSF2 approximates the dynamic viscosity using a power function.
25-30 A6	VISGAS	Function for dynamic viscosity of the gas
		VISGA1 uses reference value
		VISGA2 approximates the dynamic viscosity of the gas as function of pressure and temperature.
35-40 A6	FUNHCF	Function for permeability versus depth
		FNHC1 function returning a constant value.
		FNHC2 exponential function.
		FNHC3 function applied to a specified region.
45-50 A6	FUNPOR	Function for porosity versus depth
		FNPOR1 constant permeability.
		FNPOR2 exponential function.
		FNPOR3 function applied to a specified region.

3.2.2 User controlled input

The user controlled input was used to govern various initialization and input of mass or heat sources, unsaturated properties, grid modification, fracture distribution data for gas migration, etc. The motivation for the user controlled input was to facilitate the implementation of problems specific input settings to cause the least interference with the actual model programs.

The user controlled input was organized in such a way that a general subroutine call is made to a subroutine called PRGSEL. This subroutine contains most work arrays in its argument list. Most common blocks are included in it for possible modification of parameter data. The subroutine PRGSEL is called just before the beginning of the solution phase.

Therefore, practically any parameter or field data could be modified or input using the user controlled input. Subroutine PRGSEL contains a sequence of IF...THEN...ELSEIF tests for the various subroutine call. When a call on a requested subroutine is found among the input stream then the call will be executed.

3.2.2.1 General description of the layout used for the user controlled input (*PRGSEL*)

General explanation

Command:
SUBROU IU JU + Optional text

IU is the input file number from which data are to be read. If 00 is set instead of IU then this means that there is no input to be specifically read as a result of the current command. This means that data are taken from common areas or specified directly in the subroutine or function used to execute the command.

JU is the output file number for possible echoing of the input. This is currently used for very few commands and could for the time being be seen as a dummy feature that will be implemented later on.

Purpose:
Short comment on the general purpose of current command
Plus more or less details about input etc.

The following call is made:
Generally the operations associated with current command are performed by some subroutine or function. Therefore, to facilitate the user to replace the command by an alternative subroutine or function the imposed call is given.

```
CALL SUBROU( argument list )
```

3.2.2.2 List of user controlled input subroutines

P R G S E L	- Subroutine to perform user controlled input to *G W H R T*
AMINMX	- Find coordinate extremes in x- resp. y-direction
DATAIO	- Call subroutine to read data from disc file
DATAUO	- Write data to disc file
DRAINI	- Input drain data
GELIN1	- Input mesh parameter data and intial conditions for gas migration.
GFELHC	- Input parameters for parallel plate model conceptualization and set element or nodal permeability arrays
GFINJ1	- Set full gas displacement at "injection" nodes
GFRIN1	- Input parameter data to set boundary and initial conditions
GFSEDT	- Input fracture distribution data
GWSRCI	- Input point mass source/sinks to be applied to frontal equations
HDR4BC	- Set the initial pressure distribution hydrostatic (Hydrocoin Level 1, Case 4)
HFAX3A	- Transfer nodal data from work arrays -HH- and -LBHH- being read by subr. HFNDIN* to nodal arrays. This is for cases where the initial or boundary conditions have been set in grid data
HFBCX1	- Miscellaneous operations for input of boundary conditions, initial values, mesh modification, etc.
HFBELT	- Find nodes on top boundary and set prescribed flux condition
HFBFLW	- Input boundaries with prescribed flux conditions
HFELHC	- Set element permeability (ref.value + region specification)
HFELIF	- Input element incidences (discrete system)
HFELIN	- Input element incidences

HFELMP - Specify (unsaturated) material properties for elements or nodes

HFELM1 - Set element permeability according to values of -IELMAT-

HFELPM - Specify (unsaturated) material properties for elements or nodes

HFELP1 - Set element porosity

HFIELN - Redefine matrix of element incidences if 8-21 node hexahedral element is to be used

HFINFR - Input fracture elements

HFINP3 - Input of material properties for unsaturated conditions

HFINP4 - Set the initial pressure distribution (hydrostatic)

HFIN4P - Set the initial pressure distribution hydrostatic for unsaturated flow conditions

HFINP5 - Input data for stress solution and input of initial and boundary conditions

HFMSRC - Input concentrated mass sources

HFNDIF - Input nodal points for discrete system of fractures and blocks

HFNDIN - Input nodal points

HFNFRC - Invoke subroutine to set nodal permeability for nodes associated with fractures

HFPERM - Set nodal permeabilities

HFPRMA - Set mid-side properties (permeability) as averages of adjacent corner nodes

HFSRC1 - Input concentrated and/or distributed heat sources

HYDSBS - Input subroutine for pressure, permeability, porosity, etc.

MVSDE1 - Modify element mesh according to previous specification

NODSL2 - Specify area within which nodes are to be selected for printout

NODSL4 - Specify and set zones of specific material properties

PHXMIN - Set prescribed pressure on left hand boundary

PHXMAX - Set prescribed pressure on right hand boundary

PHYMIN - Set prescribed pressure on bottom boundary

PHYMAX - Set prescribed pressure on top boundary

PHYDRO - Sets hydrostatic boundary conditions for temperature dependent fluid density and a given temperature distribution

TFXMIN - Set prescribed temperature on left hand boundary

TFXMAX - Set prescribed temperature on right hand boundary

TFYMIN - Set prescribed temperature on bottom boundary

TFYMAX - Set prescribed temperature on top boundary

SELNDS - Input node numbers for selected printout of of nodal data

SFRIN1 - Set boundary and initial conditions for 1-d gas flow test

SHWTIN - Input the time dependent energy output for Hydrocoin Level 2, Case 2

```
      ------------------------------------------------------------------
      P R G S E L  - Subroutine to perform user controlled input
                      to  G W H R T*
                      called from: HFINS1
      ------------------------------------------------------------------
      SUBROUTINE PRGSEL(PH,LBPH,PH1I,PH2I,PHPTS,PHDT,SW,DU,LBDU,
     1                  TF,LBTF,TF1I,TF2I,TFPTS,TFDT,TF0,
     2                  TR,LBTR,TR1I,TR2I,TRPTS,TRDT,
     3                  PHDX,LBPHDX,PHDY,LBPHDY,PHDZ,LBPHDZ,
     4                  PERM,PORSTY,VISC,DENF,NODES,XN,YN,ZN,
     5                  IELS,IELNOD,IELTYP,HCEL,IELMAT,
     6                  NDFRO,WFRONT,CFRON,LBHH,HH,NJN,
     7                  EQUAT,PVCOL,VECRV,GLOAD,NACVA,MXFRON,
     8                  TIMEP,TMPF,TIRA,HFWA,HFWA1,HFWA2,DHS)

      IMPLICIT        REAL (A-H,O-Z)
      DIMENSION       PH(MXNP),LBPH(MXNP),PH1I(MXNP),PH2I(MXNP),
     >                PHPTS(MXNP),
     1                PHDT(MXNP),SW(MXNP),DU(MXNP,3),LBDU(MXNP,3),
     2                TF(MXNP),LBTF(MXNP),TF1I(MXNP),TF2I(MXNP),
     >                TFPTS(MXNP),
     3                TFDT(MXNP),
     4                TR(MXNP),LBTR(MXNP),TR1I(MXNP),TR2I(MXNP),TRPTS(MXNP),
     5                TRDT(MXNP),TF0(MXNP),
     6                PHDX(MXNP),PHDY(MXNP),PHDZ(MXNP),
     7                LBPHDX(1),LBPHDY(1),LBPHDZ(1),
     8                DHS(MXNP),
     9                NODES(MXNP),XN(MXNP),YN(MXNP),ZN(MXNP),NJN(MXNP)
C
      DIMENSION       TIRA(MXNP),HFWA(MXNP),HFWA1(MXNP),HFWA2(MXNP)
      DIMENSION       HH(MXLBHH),LBHH(MXLBHH)
      DIMENSION       PERM(MXNP),PORSTY(MXNP),VISC(MXNP),DENF(MXNP)
      DIMENSION       IELS(MXNE),IELNOD(MXNE,MX21),IELTYP(MXNE),
     1                HCEL(MXNE,3),
     1                IELMAT(MXNE,2),NDFRO(MXNE)
      DIMENSION       WFRONT(MXNN,MXBAND),CFRON(MXNN),
     1                EQUAT(MXFRON),PVCOL(MXFRON),
     2                VECRV(MXFRON),GLOAD(MXFRON),
     2                NACVA(MXFRON)
      DIMENSION       TIMEP(MXNSTP),TMPF(MXNSTP)
C
      COMMON/CNTRL1/ ICP1(40)
      COMMON/CNTRL2/ MXSTEP,MXITER,DTFAC,PHERTL,TFERTL,TRERTL
      COMMON/CNTRL3/ NSTEP,ITER,TIME,DT,PHERMX,TFERMX,TRERMX
      COMMON/CNTRL4/ PHMXDT,TFMXDT,TRMXDT
      COMMON/CNTRL5/ NPSTPS,NTSTPS
      COMMON/COEFF2/ SCF,SCR,HT
      COMMON/DTMAX1/ DTMAX
      COMMON/EQSOLV/ ICPSLV(10)
      COMMON/ERRMXN/ MXEDTN(3),MXE1IN(3),MXE2IN(3)
      COMMON/FECOM1/ MXNE,NE,MXNP,NP
      COMMON/FECOM2/ MXBAND,MBAND,MXNN,NN
      COMMON/FECOM4/ NNODE,NDOF,NEK,NDIM
C-----
      COMMON/GFELX1/ NELG,NELG1,NELG2,NELW,NELW1,NELW2,
     1               NPG,NPG1,NPG2,NPW,NPW1,NPW2,
     2               NELI,NELGWI,NPGWI1,NPGWI2
C
```

```
      COMMON/GFELX2/ XLGAS,XLWAT,XLGWI,XLTOT,DLSG
C-----
      COMMON/HFPLOT/ IPLOT
      COMMON/HFREF1/ PREF,TREF,HCREF,PRMREF,POREF,DNFREF,VSCREF
      COMMON/IAXSYM/ IAXSYM
      COMMON/ICPDSC/ ICPDSC(20)
      COMMON/ICPLOT/ ICPLOT(10)
      COMMON/IDTAIO/ IDTAIO
      COMMON/IDTAUO/ IDTAUO
      COMMON/IDISCR/ IDISCR
      COMMON/INEWTN/ INEWTN
      COMMON/INUN10/ INUN(10)
      COMMON/IOPHC / IOPHC
      COMMON/IOPSLV/ IOPSLV
      COMMON/ITERX / ITERX
      COMMON/IUFRON/ IU1,IU2,IU3,IU4
      COMMON/IUNITS/ IUNIT(10)
      COMMON/LHEX21/ IEL21
      COMMON/MISCHK/ MISCHK
      COMMON/MODPRT/ MODPRT,MPRT
      COMMON/MXLBHH/ MXLBHH
      COMMON/MXNPEL/ MX21
      COMMON/MXNSTP/ MXNSTP
      COMMON/NODPRT/ NDSPRT(100),MXNPRT,NPRT
      COMMON/PLOTXY/ XMIN,XMAX,YMIN,YMAX,SX,SY,XORG,YORG,DX,DY
      COMMON/PLOTTR/ IOPEN,ICLOS
      COMMON/RUNMAX/ RUNMAX
      COMMON/TIME1 / TIME1
      COMMON/TITLE / TEX(20)
      COMMON/UNSAM5/ IPRUNS
C
      REAL           XMIN,XMAX,YMIN,YMAX,SX,SY,XORG,YORG,DX,DY
      CHARACTER*4    TEX
      CHARACTER*6    SFRINX
      CHARACTER*6    XPROG,END1,END2
C
      DATA           END1  /'END   '/,END2/'   END'/
C
      IUN = INUN(1)
      WRITE(6,901) IUN
  901 FORMAT(/6X,'PRGSEL---Now begin labelled input - IUN=',I3)
C-----------------------------------------------------------------------
   10 READ(IUN,810,END=120) XPROG,INUNX,IUTX
  810 FORMAT(A6,2I4)
  811 FORMAT(A6,I4,: I4)
C
      WRITE(6,910) XPROG,INUNX,IUTX
  910 FORMAT(/6X,A6,2I4)
      WRITE(6,911) XPROG,INUNX,IUTX
  911 FORMAT(1H+,5X,A6,2I4,'  **PRGSEL***')
C
C---B e g i n   n e s t i n g   o f   I F   T H E N   t e s t--------------
C
      IF(XPROG.EQ.END1 .OR. XPROG.EQ.END2) THEN
        GOTO 110
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'AMINMX') THEN
        CALL AMINMX(XN,NP,XMIN,XMAX,IXMIN,IXMAX)
```

```
        CALL AMINMX(YN,NP,YMIN,YMAX,IYMIN,IYMAX)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'DATAIO') THEN
        IF(INUNX.LT.1 .OR. INUNX.GT.99) THEN
          WRITE(6,949) INUNX
  949     FORMAT(/6X,'PRGSEL---Error: Unit number out of range -',
     1                        ' INUNX =',I3)
        ELSE
C---Call subroutine to read data from disc file
          CALL       DATAIO(PH,TF,TR,DU,PERM,PORSTY,
     1                      XN,YN,ZN,NODES,
     2                      IELNOD,IELTYP,HCEL,IELMAT,IUNIX)
        ENDIF
C
        IDTAIO = 1
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'DATAUO') THEN
        READ(IUN,851          ) ICPDSC
  851   FORMAT(20I3)
        WRITE(6,951)            ICPDSC
  951   FORMAT(/6X,'PRGSEL---ICPDSC: ',20I3)
        WRITE(6,952)
  952   FORMAT(/6X,'PRGSEL---The following data arrays are to be',
     1                    ' written to disc file:')
C
        IF(ICPDSC( 1).GT.0) WRITE(6,*) '     Pressure           (PH)'
        IF(ICPDSC( 2).GT.0) WRITE(6,*) '     Fluid temperature  (TF)'
        IF(ICPDSC( 3).GT.0) WRITE(6,*) '     Rock temperature   (TR)'
        IF(ICPDSC( 4).GT.0) WRITE(6,*) '     Rock displacments  (DU)'
        IF(ICPDSC( 5).GT.0) WRITE(6,*) '     Permeability       (PERM)'
        IF(ICPDSC( 6).GT.0) WRITE(6,*) '     Porosity          ',
     1                                 '(PORSTY)'
        IF(ICPDSC( 7).GT.0) WRITE(6,*) '     X-coordinate       (XN)'
        IF(ICPDSC( 8).GT.0) WRITE(6,*) '     Y-coordinate       (YN)'
        IF(ICPDSC( 9).GT.0) WRITE(6,*) '     Z-coordinate       (ZN)'
        IF(ICPDSC(10).GT.0) WRITE(6,*) '     Nodes             ',
     1                                 '(NODES)'
        IF(ICPDSC(11).GT.0) WRITE(6,*) '     IELNOD            ',
     1                                 '(IELNOD)'
        IF(ICPDSC(12).GT.0) WRITE(6,*) '     IELTYP            ',
     1                                 '(IELTYP)'
        IF(ICPDSC(13).GT.0) WRITE(6,*) '     Element permeability',
     1                                 ' (HCEL)'
        IF(ICPDSC(14).GT.0) WRITE(6,*) '     Element material no ',
     1                                 ' (IELMAT)'
C
        IDTAUO = 1
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'DRAINI') THEN
        INUN9=INUN(9)
        IF(INUNX.GT.0) INUN9=INUNX
C---Input drain data
        CALL DRAINI(IELNOD,IELTYP,XN,YN,ZN,INUN9)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'GELIN1') THEN
C
C---Input mesh parameter data and intial gas migration conditions
        CALL GELIN1(IELNOD,IELTYP,IELMAT,XN,YN,PH,LBPH,
```

```
     1               TR,LBTR,ZN,TF0,INUNX)
C
        Here
          XNNEW -> XN, PNEW -> PH,
          XNOLD -> ZN, POLD -> TF0
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'GFELHC') THEN
C---Input parameters for parallel plate model conceptualization
      and set element or nodal permeability arrays
        CALL GFELHC(HCEL,PERM,INUNX)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'GFINJ1') THEN
C---Set full gas displacement at "injection" nodes
        CALL GFINJ1(IELNOD,NODES,XN,YN,ZN,PH,TR)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'GFRIN1') THEN
C---Input parameter data to set boundary and initial conditions
        CALL GFRIN1(NODES,PH,LBPH,TR,LBTR,XN,YN,ZN,INUNX)
C
        IF(MISCHK.GT.0) THEN
          CALL GFPRT1(XN,YN,ZN,    PH,1,NP,'     PH')
          CALL GFPRT1(XN,YN,ZN,    TR,1,NP,'     TR')
        ENDIF
C
        WRITE(6,945) NPG,NPW,NPW1
  945   FORMAT(/6X,'HFINS1---NPG=',I3,' NPW=',I3,' NPW1=',I3)
C
C---Store arrays of gas and water saturated nodes into work array
        CALL HFAX1B(HH(   1),  LBHH (1),TR(   1),LBTR(   1),NPG)
        CALL HFAX1B(HH(NPW1),  LBHH(NPW1),PH(NPW1),LBPH(NPW1),NPW)
        CALL GFPRT1(XN,YN,ZN,    HH,1,NP,'     HH')
C
        Here
          PG    -> TR, LBTR -> LBPG
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'GFSEDT') THEN
C---Input fracture distribution data
        CALL GFSEDT(INUNX)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'GWSRCI') THEN
C---Input point mass source/sinks to be applied to frontal equations
        CALL GWSRCI(XN,YN,ZN,INUNX)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HDR4BC') THEN
C---Set the initial pressure distribution (hydrostatic)
        CALL HDR4BC(PH,LBPH,XN,YN,ZN)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HFAX3A') THEN
C---Transfer nodal data from work arrays -HH- and -LBHH- being read
    by subr.  HFNDIN* to nodal arrays. This is for cases where the
    initial or boundary conditions have been set in grid data
        CALL HFAX3A(HH,LBHH,PH,LBPH,TF,LBTF,TR,LBTR,NP)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HFBCX1') THEN
C---Test for input of boundary conditions or initial values
        CALL HFBCX1(PH,LBPH,TF,LBTF,TR,LBTR,DU,LBDU,
     1              XN,YN,ZN,IELNOD,INUN(2))
C-----------------------------------------------------------------------
```

```
      ELSE IF(XPROG.EQ.'HFBELT') THEN
        READ(INUNX,848,END=130) FILTEX,QFLX,ICPF
  848   FORMAT(A10,3F10.0,10I2)
        WRITE(6,948) FILTEX,QFLX,ICPF
  948   FORMAT(/6X,'HFINS1---FILTEX: ',A10,'  QFLX: ',1P,3E10.3/
     1          6X,9X,'ICPF: ',10I2)
C
C---Find nodes on top boundary and set prescribed flux condition
CC      CALL HFBELT(IELNOD,IELTYP,XN,YN,ZN,
C    1              IELNBL,JLB,MXNBEL,QFLX)    ** set work arrays  **
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HFBFLW') THEN
C---Input boundaries with prescribed flux conditions
        CALL HFBFLW(IELNOD, INUN(1), 2, IEND2)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HFELHC') THEN
C---Set element permeability (ref.value + region specification)
        CALL HFELHC(IELNOD,IELTYP,HCEL,IELMAT,XN,YN,ZN,INUNX)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HFELIF') THEN
        INUN8=INUN(8)
        IF(INUNX.GT.0) INUN8=INUNX
C---Input element incidences (discrete system)
        CALL HFELIF(IELS,IELNOD,IELTYP,HCEL,INUN8)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HFELIN') THEN
        INUN8=INUN(8)
        IF(INUNX.GT.0) INUN8=INUNX
C---Input element incidences
        CALL HFELIN(IELS,IELNOD,IELTYP,HCEL,INUN8)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HFELMP') THEN
        INUN6=INUN(6)
        IF(INUNX.GT.0) INUN6=INUNX
C---Specify (unsaturated) material properties for elements or nodes
        CALL HFELMP(IELNOD,IELTYP,HCEL,IELMAT,NODES,XN,YN,ZN,INUN6)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HFELM1') THEN
C---Set element permeability according to values of -ielmat-
        CALL HFELM1(IELNOD,IELTYP,HCEL,IELMAT,XN,YN,ZN,INUNX)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HFELPM') THEN
        INUN6=INUN(6)
        IF(INUNX.GT.0) INUN6=INUNX
C---Specify (unsaturated) material properties for elements or nodes
        CALL HFELPM(IELNOD,IELTYP,HCEL,IELMAT,NODES,XN,YN,ZN,INUN6)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HFELP1') THEN
C---Set element porosity
        CALL HFELP1(IELNOD,IELTYP,HCEL,XN,YN,ZN)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HFIELN') THEN
C---redefine matrix of element incidences if 8-21 node hexahedral
c   element is to be used
        IF(IEL21.GT.1) CALL HFIELN(IELNOD,IELTYP)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HFINFR') THEN
C---Input fracture elements
```

```
        CALL HFINFR(INUN(1),INUN(4))
C-----------------------------------------------------------------------
      ELSE
      END IF
C
C---B e g i n   n e s t i n g  o f  I F  T H E N  t e s t--- 2-nd ------
C
      IF(XPROG.EQ.END1 .OR. XPROG.EQ.END2) THEN
        GOTO 110
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HFINP3') THEN
        INUN6=INUN(6)
        IF(INUNX.GT.0) INUN6=INUNX
C---Input of material properties for unsaturated conditions
        CALL HFINP3(INUN6, 0)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HFINP4') THEN
        INUN6=INUN(6)
        IF(INUNX.GT.0) INUN6=INUNX
C---Set the initial pressure distribution (hydrostatic)
        CALL HFINP4(PH,LBPH,PERM,XN,YN,ZN,INUN6)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HFIN4P') THEN
        INUN6=INUN(6)
        IF(INUNX.GT.0) INUN6=INUNX
C---Set the initial pressure distribution (hydrostatic)
        CALL HFIN4P(PH,LBPH,PERM,XN,YN,ZN,INUN6)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HFINP5') THEN
        INUN7=INUN(7)
        IF(INUNX.GT.0) INUN7=INUNX
C---Input data for stress solution and input of initial
c   and boundary conditions
        CALL HFINP5(IELNOD,IELTYP, XN,YN,ZN,DU,LBDU,INUN7, 0)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HFMSRC') THEN
        IF(IUTX.GT.0) WRITE(INUN(4),811) XPROGX,INUN(4)
C---Input concentrated mass sources
        IF(IUTX.LE.0) CALL HFMSRC(INUN(1),      0, 1)
        IF(IUTX.GT.0) CALL HFMSRC(INUN(1),INUN(4), 1)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HFNDIF') THEN
        INUN8=INUN(8)
        IF(INUNX.GT.0) INUN8=INUNX
C---Input nodal points (discrete system)
        CALL HFNDIF(NODES,XN,YN,ZN,LBHH,HH,3,INUN8)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HFNDIN') THEN
        INUN8=INUN(8)
        IF(INUNX.GT.0) INUN8=INUNX
C---Input nodal points
        CALL HFNDIN(NODES,XN,YN,ZN,LBHH,HH,3,INUN8)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HFNFRC') THEN
C---Invoke subroutine to set nodal permeability for nodes associated
c   with fractures
        CALL HFNFRC(XN,YN,ZN,PERM,NODES,INUNX)
CC<<<<  WRITE(6,920) XPROG
```

```
  920   FORMAT(/6X,'PRGSEL---',A6,': currently inactive')
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HFPERM') THEN
C---Set nodal permeabilities
        CALL HFPERM(XN,YN,ZN,PERM, INUN(10))
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HFPRMA') THEN
        IF(IMATM.EQ.1) THEN
C---Set mid-side properties (permeability) as averages
c   of ajcacent corner nodes
          CALL HFPRMA(IELNOD,PERM)
        ENDIF
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HFSRC1') THEN
C---Input concentrated and/or distributed heat sources
        CALL HFSRC1(XN,YN,ZN,DHS,IELNOD,IELTYP,INUN(2), 0)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'HYDSBS') THEN
C---Input subroutine for pressure, permeability, porosity, etc.
        CALL HYDSBS(PH,LBPH,SW,
     1              PERM,PORSTY,VISC,DENF,NODES,
     2              XN,YN,ZN,IELNOD,IELTYP,HCEL,IELMAT,
     3              INUNX)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'MVSDE1') THEN
        READ(INUNX,*,END=130) SLOPE,CRDREF,DISX
        WRITE(6,947) SLOPE,CRDREF,DISX
  947   FORMAT(6X,'HFINS1---SLOPE=',F8.3,' CRDREF=',F8.3,
     1                                 ' DISX=',F11.3)
C---modify element mesh according to previous specification
        CALL MVSDE1(XN,YN,NP,SLOPE,CRDREF,DISX)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'NODSL2') THEN
C---Specify area within which nodes are to be selected for printout
        CALL NODSL2(XN,YN,ZN,NP, NDSPRT,MXNPRT,NPRT, INUN(2))
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'NODSL4') THEN
        INUN8=INUN(8)
        IF(INUNX.GT.0) INUN8=INUNX
C---Specify and set zones of specific material properties
        CALL NODSL4(IELNOD,IELTYP,IELMAT,NODES,XN,YN,ZN,INUN8)
C-----------------------------------------------------------------------
      ELSE
      END IF
C
C---B e g i n   n e s t i n g  o f  I F  T H E N  t e s t--- 3-rd ------
C
      IF(XPROG.EQ.END1 .OR. XPROG.EQ.END2) THEN
        GOTO 110
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'PHXMIN') THEN
C---Set prescribed pressure on left hand boundary
        CALL PHXMIN(PH,LBPH,XN,YN,'PH')
C
      ELSE IF(XPROG.EQ.'PHXMAX') THEN
C---Set prescribed pressure on right hand boundary
        CALL PHXMAX(PH,LBPH,XN,YN,'PH')
C
```

```
      ELSE IF(XPROG.EQ.'PHYMIN') THEN
C---Set prescribed pressure on bottom boundary
        CALL PHYMIN(PH,LBPH,XN,YN,'PH')
C
      ELSE IF(XPROG.EQ.'PHYMAX') THEN
C---Set prescribed pressure on top boundary
        CALL PHYMAX(PH,LBPH,XN,YN,'PH')
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'PHYDRO') THEN
C---Call subroutine to input data for hydrostatic boundary conditions
        CALL PHYDRO(XN,YN,ZN,PH,LBPH,TF,LBTF,TR,LBTR,INUNX)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'SELNDS') THEN
C---Input node numbers for selected printout of nodal data
        CALL SELNDS(INUN(2))
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'SFRIN1') THEN
        INUN6=INUN(6)
        IF(INUNX.GT.0) INUN6=INUNX
C---Set boundary and initial conditions for 1-d gas flow test
        CALL SFRIN1(NODES,PH,LBPH,TR,LBTR,XN,YN,ZN,INUN6, 0)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'SHWTIN') THEN
C---Input the time dependent energy output for hydrocoin l2,c2
        CALL SHWTIN(INUNX)
C-----------------------------------------------------------------------
      ELSE IF(XPROG.EQ.'TFXMIN') THEN
C---Set prescribed temperature on left hand boundary
        CALL PHXMIN(TF,LBTF,XN,YN,'TF')
C
      ELSE IF(XPROG.EQ.'TFXMAX') THEN
C---Set prescribed temperature on right hand boundary
        CALL PHXMAX(TF,LBTF,XN,YN,'TF')
C
      ELSE IF(XPROG.EQ.'TFYMIN') THEN
C---Set prescribed temperature on bottom boundary
        CALL PHYMIN(TF,LBTF,XN,YN,'TF')
C
      ELSE IF(XPROG.EQ.'TFYMAX') THEN
C---Set prescribed temperature on top boundary
        CALL PHYMAX(TF,LBTF,XN,YN,'TF')
C-----------------------------------------------------------------------
      ELSE
        WRITE(6,960)
  960   FORMAT(/6X,'PRGSEL---Error: Invalid label')
      END IF
C-----------------------------------------------------------------------
      GOTO 10
C-----------------------------------------------------------------------
  110 WRITE(6,980)
  980 FORMAT(/6X,'PRGSEL---Now read end label')
      RETURN
C-----------------------------------------------------------------------
  120 WRITE(6,990) IUN
  990 FORMAT(6X,'PRGSEL---Now read to end of file=',I3)
      RETURN
```

```
130 WRITE(6,990)  INUNX
    RETURN
    END
```

3.2.2.3 Description of user controlled subroutines

********** A M I N M X **********

Command:
AMINMX 00 00 <Optional text>

Purpose:
To find the minimun resp. maximum coordinates of current element grid.
This command is usually used in conjunction with some other operation which requires that the extreme coordinate values be known.

The following call is made:
```
CALL AMINMX (XN,NP,XMIN,XMAX,IXMIN,IXMAX)
CALL AMINMX (YN,NP,YMIN,YMAX,IYMIN,IYMAX)
```

********** D A T A I O **********

Command:
DATAIO IU 00 <Optional text>

Purpose:
To impose a call to subroutine *DATAIO* to read solution data, coordinates of element grid, element incidences, etc. from disc file IU.

The following call is made:
```
CALL DATAIO (PH,TF,TR,DU,PERM,PORSTY,XN,YN,ZN,NODES,IELNOD,
             IELTYP,HCEL,IELMAT,IUNIX)
```

Control options are stored in the following common block:
COMMON/ICPDSC/ ICPDSC(20)

Control options for variables that may be written or read from disk

Option	Variable	
1	PH	- Pressure
2	TF	- Fluid temperature
3	TR	- Rock temperature Gas pressure
4	DU	- Rock displacement
5	PERM	- Permeability
6	PORSTY	- Porosity
7	XN	- X-coordinates
8	YN	- Y-coordinates
9	ZN	- Z-coordinates
10	NODES	- Node numbers Node material numbers
11	IELNOD	- Element connectivity matrix

12	IELTYP	- Element type array
13	HCEL	- Element permeability
14	IELMAT	- Element material numbers

Below follows the subroutine used:

```
      SUBROUTINE DATAIO(PH,TF,TR,DU,PERM,PORSTY,
     1                  XN,YN,ZN,NODES,
     2                  IELNOD,IELTYP,HCEL,IELMAT,INUN)

      IMPLICIT REAL(A-H,O-Z)

      DIMENSION       PH(*),TF(*),TR(*),DU(MXNP,*),
     1                PERM(*),PORSTY(*),
     2                XN(*),YN(*),ZN(*),NODES(*),
     3                IELNOD(MXNE,*),IELTYP(*),HCEL(MXNE,*),
     4                IELMAT(MXNE,*)

      COMMON/FECOM1/ MXNE,NE,MXNP,NP
      COMMON/FECOM4/ NNODE,NDOF,NEK,NDIM
      COMMON/ICPDSC/ ICPDSC(20)
      COMMON/IUNITS/ IUNIT(5)
      COMMON/MXNPEL/ MX21

      READ(IUNIT(1)) NSTEP,TIME,DT,NE,NP,MXNDE,NDOF,NDIM,ICPDSC

      WRITE(6,910) IUNIT(1),NSTEP,TIME,DT,NE,NP,MXNDE,NDOF,NDIM,ICPDSC
  910 FORMAT(/6X,'DATAIO---Now to read data on disc file:  ',I2/
     1        6X,9X,'NSTEP=',I3,' TIME=',1P,E10.3,' DT=',E10.3/
     2        6X,9X,'NE=',I4,' NP=',I4,' MX21=',I2,' NDOF=',I1,
     3             ' NDIM=',I1/
     4        6X,9X,'ICPDSC: ',20I2)

      IF(MXNDE.GT.MX21) THEN
        WRITE(6,915) MXNDE,MX21
  915   FORMAT(/6X,'DATAIO---Warning: MXNDE=',I2,' > MX21=',I2,' which'/
     1          6X,9X,' means that',
     2               ' the second dimension in IELNOD(MXNE,MX21)'/
     3          6X,9X,'may be too small for current input data.'/
     4          6X,9X,'Increase second dimension to be sure that',
     5             ' there is enough space.')
      ENDIF

      IF(NP.GT.MXNP) THEN
        WRITE(6,920) NP,MXNP
  920   FORMAT(/6X,'DATAIO---Insufficient length of nodal arrays:'/
     1          6X,9X,'NP=',I4,' > MXNP=',I4,' STOP')
        STOP
      ENDIF

      IF(NE.GT.MXNE) THEN
        WRITE(6,925) NE,MXNE
  925   FORMAT(/6X,'DATAIO---Insufficient length of element arrays:'/
     1          6X,9X,'NE=',I4,' > MXNE=',I4,' STOP')
        STOP
      ENDIF

      IF(ICPDSC(1).EQ.1)
     1  CALL READX(PH,NP,INUN,IENDX)

      IF(ICPDSC(2).EQ.1)
     1  CALL READX(TF,NP,INUN,IENDX)
```

```
      IF(ICPDSC(3).EQ.1)
     1  CALL READX(TR,NP,INUN,IENDX)

      IF(ICPDSC(4).EQ.1) THEN
        DO 10 IDIM=1,NDIM
          CALL READX(DU(1,IDIM),NP,INUN,IENDX)
10      CONTINUE
      ENDIF

      IF(ICPDSC(5).EQ.1)
     1  CALL READX(PERM,NP,INUN,IENDX)

      IF(ICPDSC(6).EQ.1)
     1  CALL READX(PORSTY,NP,INUN,IENDX)

      IF(ICPDSC(7).EQ.1)
     1  CALL READX(XN,NP,INUN,IENDX)

      IF(ICPDSC(8).EQ.1)
     1  CALL READX(YN,NP,INUN,IENDX)

      IF(ICPDSC(9).EQ.1)
     1  CALL READX(ZN,NP,INUN,IENDX)

      IF(ICPDSC(10).EQ.1)
     1  CALL READY(NODES,NP,INUN,IENDX)

      IF(ICPDSC(11).EQ.1) THEN
        DO 20 IX=1,MXNDE
          CALL READY(IELNOD(1,IX),NE,INUN,IENDX)
20      CONTINUE
      ENDIF

      IF(ICPDSC(12).EQ.1)
     1  CALL READY(IELTYP,NE,INUN,IENDX)

      IF(ICPDSC(13).EQ.1) THEN
        DO 30 IDIM=1,NDIM
          CALL READX(HCEL(1,IDIM),NE,INUN,IENDX)
30      CONTINUE
      ENDIF

      IF(ICPDSC(14).EQ.1) THEN
        CALL READY(IELMAT(1,1),NE,INUN,IENDX)
        CALL READY(IELMAT(1,2),NE,INUN,IENDX)
      ENDIF

      RETURN
      END
```

********** D A T A U O **********

Command:

DATAUO IU 00 <Optional text>

Purpose:

To write solution data, coordinates, element incidences, etc. to a disc file.

```
      COMMON/ICPDSC/ ICPDSC(20)
```

Control options for variables that may be written or read from disc:

Option	Variable	
1	PH	- Pressure
2	TF	- Fluid temperature
3	TR	- Rock temperature Gas pressure
4	DU	- Rock displacement
5	PERM	- Permeability
6	PORSTY	- Porosity
7	XN	- X-coordinates
8	YN	- Y-coordinates
9	ZN	- Z-coordinates
10	NODES	- Node numbers Node material numbers
11	IELNOD	- Element connectivity matrix
12	IELTYP	- Element type array
13	HCEL	- Element permeability
14	IELMAT	- Element material numbers

Below follows the subroutine used:

```
      SUBROUTINE DATAUO(PH,TF,TR,DU,PERM,PORSTY,
     1                  XN,YN,ZN,NODES,
     2                  IELNOD,IELTYP,HCEL,IELMAT,INUN)

      IMPLICIT DOUBLEPRECISION(A-H,O-Z)

      DIMENSION       PH(NP),TF(NP),TR(NP),DU(MXNP,*),
     1                PERM(NP),PORSTY(NP),
     2                XN(NP),YN(NP),ZN(NP),NODES(NP),
     3                IELNOD(MXNE,*),IELTYP(NE),HCEL(MXNE,*),
     4                IELMAT(MXNE,*)

      COMMON/FECOM1/ MXNE,NE,MXNP,NP
      COMMON/FECOM4/ NNODE,NDOF,NEK,NDIM
      COMMON/ICPDSC/ ICPDSC(20)
      COMMON/IUNITS/ IUNIT(5)
      COMMON/MXNPEL/ MX21

      WRITE(6,910) IUNIT(1),NSTEP,TIME,DT,NE,NP,MX21,NDOF,NDIM,ICPDSC
  910 FORMAT(/6X,'DATAUO---Now to write data on disc file: ',I2/
     1        6X,9X,'NSTEP=',I3,' TIME=',1P,E10.3,' DT=',E10.3/
     2        6X,9X,'NE=',I4,' NP=',I4,' MX21=',I2,' NDOF=',I1,
     3             ' NDIM=',I1/
     4        6X,9X,'ICPDSC: ',20I2)

      WRITE(IUNIT(1)) NSTEP,TIME,DT,NE,NP,MX21,NDOF,NDIM,ICPDSC

      IF(ICPDSC(1).EQ.1) WRITE(INUN) PH

      IF(ICPDSC(2).EQ.1) WRITE(INUN) TF

      IF(ICPDSC(3).EQ.1) WRITE(INUN) TR
```

```
      IF(ICPDSC(4).EQ.1) THEN
        DO 10 IDIM=1,NDIM
          CALL WRITEX(DU(1,IDIM),NP,INUN)
10    CONTINUE
      ENDIF

      IF(ICPDSC(5).EQ.1) WRITE(INUN) PERM

      IF(ICPDSC(6).EQ.1) WRITE(INUN) PORSTY

      IF(ICPDSC(7).EQ.1) WRITE(INUN) XN

      IF(ICPDSC(8).EQ.1) WRITE(INUN) YN

      IF(ICPDSC(9).EQ.1) WRITE(INUN) ZN

      IF(ICPDSC(10).EQ.1) THEN
        CALL WRITEY(NODES,NP,INUN)
      ENDIF

      IF(ICPDSC(11).EQ.1) THEN
        DO 20 IX=1,MX21
          CALL WRITEY(IELNOD(1,IX),NE,INUN)
20    CONTINUE
      ENDIF

      IF(ICPDSC(12).EQ.1) WRITE(INUN) IELTYP

      IF(ICPDSC(13).EQ.1) THEN
        DO 30 IDIM=1,NDIM
          CALL WRITEX(HCEL(1,IDIM),NE,INUN)
30    CONTINUE
      ENDIF

      IF(ICPDSC(14).EQ.1) THEN
        CALL WRITEY(IELMAT(1,1),NE,INUN)
        CALL WRITEY(IELMAT(1,2),NE,INUN)
      ENDIF

      RETURN
      END
```

********** D R A I N I **********

Command:

```
  DRAINI    IU 00 <Optional text>
```

Purpose:

Input drain data

The following call is made:

```
    CALL    DRAINI (IELNOD,IELTYP,XN,YN,ZN,INUN9)
```

********** G E L I N 1 **********

Command:

```
  GELIN1    IU 00 <Optional text>
```

Purpose:

The following call is made:
GELIN1 (IELNOD,IELTYO,IELMAT,XN,YN,PH,LBPH,TR,LBTR,ZN,TFO,INUNX)

********** G F E L H C **********

Command:
GFELHC IU 00 <Optional text>

Purpose:
Input parameters for parallel plate model conceptualizations and set element or nodal permeability arrays.

The following call is made:
CALL GFELHC (HCEL,PERM,INUNX)

********** G F I N J 1 **********

Command:
GFINJ1 00 00 <Optional text>

Purpose:
Set gas displacement at "injection" nodes a ccording to a given saturation (currently S=0.8).

The following call is made:
CALL GFINJ1 (IELNOD,NODES,XN,YN,ZN,PH,TR)

The subroutine checks the capillary pressure corresponding to the specified saturation.
The gas pressure at injection nodes are adjusted such that

$$p_g = p_w + p_c$$

where p_g is the gas pressure, p_w is water pressure and p_c is the capillary pressure.

********** G F R I N 1 **********

Command:
GFRIN1 IU 00 <Optional text>

Purpose:
To input parameter data for setting boundary and initial conditions for "SFR-Gas flow problem".

The following call is made:
```
CALL GELIN1(IELNOD,IELTYP,IELMAT,XN,YN,PH,LBPH,
            TR,LBTR,ZN,TF0,INUNX)
```

Below follows the subroutine used:

```
SUBROUTINE GFRIN1(NODES,PH,LBPH,PG,LBPG,XN,YN,ZN,IUN,IUT)
IMPLICIT DOUBLE PRECISION(A-H,U-Z)
DIMENSION      NODES(MXNP),PH(MXNP),LBPH(MXNP),
1              PG(MXNP),LBPG(MXNP),
2              XN(MXNP),YN(MXNP),ZN(MXNP)
```

```
      COMMON/FECOM1/ MXNE,NE,MXNP,NP
      COMMON/FECOM4/ NNODE,NDOF,NEK,NDIM
      COMMON/GAS3   / SLIR(3),SGR(3)
      COMMON/GRVTY1/ GRVTY1(3)
      COMMON/HFREF1/ PREF,TREF,HCREF,PRMREF,POREF,DNFREF,VSCREF
      COMMON/HGCUSH/ HG,HS

      COMMON/IOPBC1/ IOPBC1(10)

      READ(IUN,    )  IPRT,YMN,PMN,YMX,PMX,HG,HS
      READ(IUN,802) (IOPBC1(I),I=1,10)
  802 FORMAT(10I2)
      WRITE(6,925) IPRT,YMN,PMN,YMX,PMX,HG,HS,
     1               (IOPBC1(I),I=1,10)
  925 FORMAT(/6X,'IPRT=',I1/
     2        6X,9X,'Parameters transferred to *PRMSE1*'/
     3        6X,9X,'YMN=',F8.2,' PMN=',F7.2,' YMX=',F8.2,' PMX=',F7.2/
     4        6X,9X,'Thickness of Gas Cushion (HG) =',F6.1,' Metres'/
     5        6X,9X,'Depth to Sea Bottom      (HS) =',F6.1,' Metres'/
     6        6X,9X,'IOPBC1:',10I2)

      Check if gravity is to be accounted for and set control parameter

      IF(GRVTY1(1).NE.0..OR.GRVTY1(2).NE.0.OR.GRVTY1(3).NE.0.) THEN
        IOPG=1
      ELSE
        IOPG=0
      ENDIF

      WRITE(6,910) IOPBC1,IOPG
  910 FORMAT(1H1//6X,'GFRIN1---Boundary and Initial Conditions for',
     1                                    ' "SFR Gas Flow Problem"'/
     2           /6X,9X,'IOPBC1:',10I2,' IOPG=',I1)

    Find extreme values of coordinates of the flow domain

      CALL DMINMX(XN,NP,XMIN,XMAX,IXMIN,IXMAX)
      CALL DMINMX(YN,NP,YMIN,YMAX,IYMIN,IYMAX)

      PA   = PREF
      G    = 9.81
      DENF = DNFREF

    Check thickness of flow domain

      IF(IOPG.EQ.0) THEN
        H=0.0
      ELSE IF(NDIM.EQ.1) THEN
        H=XMAX-XMIN
      ELSE IF(NDIM.EQ.2) THEN
        H=YMAX-YMIN
      ENDIF

    set initial hydrostatic pressure distribution in the water
    as well as the gas phase

      IF(NDIM.EQ.1) THEN
        CALL GFSET1(XN,PH,PG,G,HS,XMAX)
      ELSE IF(NDIM.EQ.2) THEN
        CALL GFSET1(YN,PH,PG,G,HS,YMAX)
      ENDIF
   Begin loops to find nodes on bottom and top and to set the
    boundary conditions at the nodes found
      TOL=0.001
```

```
  Deal with the bottom boundary (icpbc(3)=1)-------------------------
    IF(IOPBC1(3).EQ.1) THEN
    IF(IOPG.EQ.0) THEN
         COEFF= PA + DENF*G*(      HG)
      ELSE IF(IOPG.EQ.1) THEN
        COEFF= PA + DENF*G*(H+HS+HG)
          ENDIF

  Gas cushion pressure
      COEFF=DENF*G*HG
      WRITE(6,920) COEFF,HG
920   FORMAT(/6X,'GFRIN1---Pressure of Gas Cushion=',1P,E10.3/
   1          6X,9X,       'Height of Gas Cushion  =',E10.3)
   Begin loop to set b.c. at bottom boundary
      DO 30 I=1,NP
        IF(NDIM.EQ.1 .AND. XN(I).GT.XMIN+TOL) GOTO 30
        IF(NDIM.EQ.2 .AND. YN(I).GT.YMIN+TOL) GOTO 30
  Add gas cushion pressure at bottom node
        PG(I)=PG(I) + COEFF
        LBPG(I)=1
  Set prescribed water pressure according to capillary curve
  at irreducible water saturation (slir)
        CALL HFNSW5(NODES(I),PC,SLIR(1))
        PH(I)=PG(I)-PC
        LBPH(I)=1
        WRITE(6,930) I,LBPG(I),I,PG(I),I,LBPH(I),I,PH(I)
930     FORMAT(6X,'GFRIN1---Bottom Node: LBPG(',I3,')=',I1,
   1                          ' PG(',I3,')=',1P,E10.3:/
   2           6X,9X,11X,'  LBPH(',I3,')=',I1,' PH(',I3,')=',1P,E10.3)

        WRITE(6,931) NODES(I),SLIR,PC
931     FORMAT(6X,9X,'Material No:',I3,
   1                 ' Irred. water sat.=',3F5.2,' PC=',1P,E10.3)
 30   CONTINUE
    ENDIF

    IF(IOPBC1(4).EQ.1) THEN
  Deal with the top boundary (ICPBC(4)=1)----------------------------
      IF(IOPG.EQ.0) THEN
        COEFF= PA
      ELSE IF(IOPG.EQ.1) THEN
  Atmoshperic pr. + sea
        COEFF= PA + DENF*G*HS
      ENDIF
  Begin loop to set B C. on top boundary-----------------------------
      DO 50 K=1,NP
        I=K
        IF(NDIM.EQ.1 .AND. XN(I).LT.XMAX-TOL) GOTO 40
        IF(NDIM.EQ.2 .AND. YN(I).LT.YMAX-TOL) GOTO 40
  Set top node
        PH(I)   = COEFF
        LBPH(I) = 1
        PG(I)   = PH(I)
        LBPG(I) = 1
        WRITE(6,940) I,LBPG(I),I,PG(I),I,LBPH(I),I,PH(I)
940     FORMAT(6X,'GFRIN1---Top    Node: LBPG(',I3,')=',I1,
   1                           ' PG(',I3,')=',1P,E10.3:/
   2           6X,9X,11X,'  LBPH(',I3,')=',I1,' PH(',I3,')=',1P,E10.3)
 40   CONTINUE
 50   CONTINUE
```

```
      ENDIF
-----------------------------------------------------------------------
      WRITE(6,950)
  950 FORMAT(/6X,'GFRIN1---Boundary and initial conditions for',
     1                                  ' 1-D gas flow problem'/)
      WRITE(6,960) PA,H,HS,HG,DENF,G
  960 FORMAT(6X,9X,'Parameter values:'//
     1       6X,9X,'Atmospheric Reference Pressure   :',1P,E10.3/
     2       6X,9X,'Thickness of Aquifer (Rock)      :',1P,E10.3/
     3       6X,9X,'Depth to See Bottom              :',1P,E10.3/
     4       6X,9X,'Thickness of Gas Cushion         :',1P,E10.3/
     5       6X,9X,'Fluid Density                    :',1P,E10.3/
     6       6X,9X,'Acceleration of Gravity          :',1P,E10.3)

      WRITE(6,990)
  990 FORMAT(/6X,'GFRIN1---Now to RETURN')

      RETURN
      END
```

Description of parameters

IOPBC1 - Array for control option of B.C. settings
IOPBC1(3)=1 set B.C. on bottom boundary
IOPBC1(4)=1 set B.C. on top boundary

IOPG - Gravity control paramter
IOPG=0 no gravity
IOPG=1 gravity

PA - Atmospheric reference pressure

H - Thickness of aquifer (rock)

HS - Depth to see bottom

HG - Thickness of gas cushion

DENF - Fluid density

G - Acceleration of gravity

Note that HG,HS and iopbc1 are currently input by subroutine *hfinp4*

********** G F S E D T **********

Command:
GFSEDT IU JU <Optional text>

Purpose:
Input fracture distribution data for parallel plate or capillary tube conceptualization of the rock mass.

The following call is made:
CALL GFSEDT (INUX)

```
Record 1  Optional text
        READ(INUN,801,END=30) TEX
    801 FORMAT(18A4)

Record 2  Cavern width and fracture spacing
        READ(INUN, * ,END=30) CAVWID,FRSP,NFRI,IOPTRN

          CAVWID is cavern width
          FRSP   is the fracture spacing
          NFPI   is the number of fracture classes
          IOPTRN is an integer control parameter to indicate if
                    subsequent input parameter are to be
                    transformed
                    from bulk parameters values into equivalent frac-
                    ture parameter values

                  = 0 Indicates that fracture distribution data areis
                    to be input directly without any transformation

                  = 1 Equivalent fracture parameters to be calculated
                    for capillary tube conceptualization of the
                    rock mass

                  = 2 Equivalent fracture parameters to be calculated
                    for parallel palts conceptualization of the
                    rock mass

          The subsequent input is dependent on
          the input value of IOPTRN.

          For IOPTRN = 0 :

            DO 10 I=1,NFRI
              READ(INUN, * ,END=40) IFRI,FRWIDI(I),
        1                                PFRI(I),CFRI(I),QFACI(I)
       10   CONTINUE
-------------------------------------------------------------------

          For IOPTRN = 1 OR 2:
              IF(IOPTRN.EQ.1 .OR. IOPTRN.EQ.2) THEN
Record 3  (free format)

          READ(INUN, * ) (RPERM(I),XLCAVI(I),I=1,NFRI)

          For IOPTRN.EQ.1 :
          Equivalent parameters for capillary tube
          conceptualization to be computed
            DO 20 I=1,NFRI
              CALL       PPROP1(RPERM(I),FRSP,CAVWID,XLCAVI(I),
        1                       XNF,FRWID,FRPERM,PORI,
        2                       FRWIDI(I),PFRI(I),CFRI(I))
              QFACI(I) = CFRI(I)*PI*FRWIDI(I)**2/4.
             .
             .
             .
       20   CONTINUE

       --------------------

          For IOPTRN = 2 :
```

```
      Equivalent parameters for parallel plate
      conceptualization to be computed
        DO 22 I=1,NFRI
          CALL       PPROP1(RPERM(I),FRSP,CAVWID,XLCAVI(I),
     1                      CFRI(I),FRWIDI(I),PFRI(I),PORI,
     2                      DTUBE,FRD,XNDF)
          QFACI(I) = CFRI(I)*CAVWID*FRWIDI(I)
   22   CONTINUE
```

QFACI -Array for multiplication factors to compute the total gas flow for repective fracture class

$$Q_{fac(i)} = n_{f(i)} * L_c * b$$

$$Q_{tot} = \sum_{i=1}^{n} Q_{fr(i)} * Q_{fac(i)}$$

where

n	is total number of fracture categories
$n_{fac(i)}$	is the number of fractures in fracture class(i)
L_c	is the cavern width
b	is fracture width
$Q_{fr(i)}$	is the flux per unit area of fracture class(i)
$Q_{fac(i)}$	is the multiplication factor to relate flux per unit area to the total flux for fracture category

---

********** G W S R C I **********

Command:
GWSRCI IU 00 Input mass sources

Purpose:
To impose a call to subroutine *GWSRCI* to input mass sources.

The input is stored in the two arrays as indicated below and transferred through a common block to the subroutine *GWSRC1*. This subroutine is called upon by the front solver each time a nodal variable is to be eliminated to check if there is any mass source associated with the node.

Free format:
Record 1 NLOAD is the number of mass sources to be input

Record 2 READ(INUN, *) (IWLOAD(I),GWLOAD(I),I=1,NLOAD)

IWLOAD(I) is node number
GWLOAD(I) is the mass source strength at the corresponding node

The folLowing subroutine call is made:

```
   CALL GWSRCI(XN,YN,ZN,IU)
```

********** H D R 4 B C **********

Command:
HDR4BC IU 00 <Optinal text>

Purpose:
Set the inital pressure distrbution (hydrostatic)

The flow domain is the area located between two circles one is an inner circle with radius RMIN and the other is an outer circle with the radius RMAX.

For IOPR=1 :
Nodes along the inner radius of the flow domain are to be prescribed.
For IOPR=2
Nodes along the outer radius of the flow domain are to be prescribed.

The following call is made:

```
CALL HDR4BC (PH,LBPH,XN,YN,ZN)
```

********** H F A X 3 A **********

Command:
HFAX3A 00 00 + Optional text

Purpose:
To transfer the boundary conditions as input together with the coordinates of the nodal points and stored in the -HH- and -LBHH- into the work arrays for pressure -PH- and encoded boundary conditions -LBPH-, for fluid temperatures -TF- and encoded boundary conditions -LBTF- and rock temperatures -TR- and encoded boundary conditions -LBTR. The array for rock temperature is used for gas pressure if gas migration is solved for.

The following call is made:

```
CALL HFAX3A(HH,LB,PH,LBPH,TF,LBTF,TR,LBTR,NP)
```

The following program statements are executed for this call:

```
ENTRY HFAX3A(HH,LB,PH,LBPH,TF,LBTF,TR,LBTR,NP)
    DO 100 I=1,NP
      LBPH(I) = LB((I-1)*3+1)
      PH(I)   = HH((I-1)*3+1)
      LBTF(I) = LB((I-1)*3+2)
      TF(I)   = HH((I-1)*3+2)
      LBTR(I) = LB((I-1)*3+3)
      TR(I)   = HH((I-1)*3+3)
100 CONTINUE
```

********** H F B C X 1 **********

Command:
HFBCX1 IU JU Optional text

Purpose:
To modify element mesh, to set boundary and initial conditions, etc.

The following call is made:

```
CALL HFBCX1(PH,LBPH,TF,LBTF,TR,LBTR,DU,LBDU,
            XN,YN,ZN,IELNOD,INUN(2))
```

Below follows a description of the input data:

Data set 1----------Miscellaneous operations to modify the element mesh, specify boundary conditions,set initial values etc. (subroutine HFBCX1*)

Cols	Format	Identifier	Explanation
1-2	A2	PRM(1)	Current operation to be applied to pressure by putting 'PH' in the data field. Otherwise leave this data field blank
3-4	A2	PRM(2)	Current operation to be applied to fluid temperature by putting 'TF' in the data field. Otherwise leave this data field blank
5-6	A2	PRM(3)	Current operation to be applied to the rock temperature by putting 'TR' in the data field. Otherwise leave this data field blank
7-8	I2	IOP	Control parameter (see explanation below)
9-10	I2	ICODE	Code = 0 Initial values to be input = 1 Prescribed values to be input
11-15	F5.0	VARY(1)	Value (see explanation in the sequel)
16-20	F5.0	VARY(2)	Value (see explanation in the sequel)
21-25	F5.0	VARY(3)	Value (see explanation in the sequel)
26-30	I5	NODES(1)	Number of 1:st node
.	.	.	.
.	.	.	.
.	.	.	.
76-80	I5	NODES(11)	Number of 11:th node

Note: The last record of the data set should be a record with -1 in cols 1-2

Below follows a complementary description of some of the parameters in data set 17:

IOP=1 - Indicates that subroutine HFBCI1 is to be called. This subroutine performs the following operations:

```
LB(NODES(JN)) = ICODE
HH(NODES(JN)) = VARY(1)
```

where LB represents either LBPH, LBTF or LBTR and HH represents either PH, TF or TR. JN is an index running from 1 to 11.

IOP=2 - Indicates that subroutine HFBCI2 is to be called. This subroutine is used to set boundary conditions or initial values for an element mesh generated by subroutine L1DGEN.
** This option has become obsolescent ***

IOP=3 - Indicates that subroutine HFBCI3 is to be called. This subroutine is used to set a linear variation of pressure, fluid temperature or rock temperature. The following operations are performed:

```
JN      = NODES(I)
LB(JN)  = ICODE
HH(JN)  = VARY(1) + VARY(2)*(YN(JN)-VARY(3))
```

where I is an index running from 1 to 11, LB represents LBPH, LBTF or LBTR, HH represents PH, TF or TR, and YN is the coordinate in y-direction (YN may be replaced by XN or ZN depending on the orientation of the coordinate system).
Currently this option is used to specify the natural geothermal gradient for 2-D runs

IOP=4 - Indicates that subroutine HFBCI4 is to be called. This subroutine is used to set a linear variation in pressure, fluid temperature or rock temperature. The following operation is performed:

```
HH(I) = VARY(1) + VARY(2)*(YN(I) - VARY(3))
```

where I is an index running over all nodal points
Currently this option is used to set the initial temperature distribution according to the natural geothermal gradient for 2-D runs

Note that when using this option all nodal points will be affected while in the previous case (IOP=3) only the nodes as specified in the list (NODES) will be affected. Note also that no codes (ICODE) are assigned when using this option

IOP=5 - Indicates that subroutine MODFY1 is to be called. This subroutine is used to scale the nodal coordinates in the following way:

```
XN(I) = XN(I)*VARY(1)
YN(I) = YN(I)*VARY(2)
ZN(I) = ZN(I)*VARY(3)
```

IOP=6 - Indicates that subroutine HFSET2 is to be called. This subroutine is used to smooth out an initial peak in the following way:

```
HH(I) = VARY(2)                                              +
 (VARY(1)-VARY(2))*EXP(-VARY(3)/XL*XN(I))
```

IOP=7 - Indicates that HFBCI5 is to be called. This subroutine is used to specify nodal values for a given sequence of nodes

```
HH(I) = VARY(1)
LB(I) = ICODE
```

where I is an index running from NODES(1) to NODES(2).

IOP=8 - Indicates that subroutine MTOP2D (in 2-D) or MTOP3D (in 3-D) is to be called. This subroutine is used to apply a unidirectional slope to a grid mesh whose top boundary is horizontal.

IOP=9 - Indicates that subroutine BND2DN (2-D version) or BND3DN (3-D version) is to be called. This subroutine is used to locate and specify values of the top and/or the bottom boundary nodes. The top boundary may be horizontal or linearly sloping.

Subroutine BND2DN
Subroutine to identify and set values for the top and bottom boundary nodes. The top boundary is assumed to be horizontal or linearly sloping.

Variables:

VARY(1) - Value which specifies the slope of the top boundary (=SLOPE), i.e.
Y = VARY(1)/(XMAX-XMIN)*
(X-XMIN) + YMAX

VARY(2) - Value to be assigned to the nodes on the top boundary.

VARY(3) - Value to be assigned to all nodes on the bottom boundary.

ICODE - Code to set type code for nodes on a boundary
= 0 Free variable
= 1 Prescribed variable
Note: Currently ICODE is used according to the following
= 1 Locate and prescribe values on top boundary
= 2 Locate and prescribe values on bottom boundary
= 3 Locate and prescribe values on both top and bottom boundaries

---

********** H F B E L T **********

Command:
HFBELT

Purpose:
Find nodes on top boundary and set prescribed flux condition

Two auxiliary subroutines NODREF to setup nodal definitions for the element sides and ELMREF are called to find all element sides located along the exterior boundary. As a result of these two subroutines one obtains an array containing the element number and the element side number for the respective element side. This array is then used to identify the nodal points along the exterior boundary Currently, the nodes along the top boundary are located by checking the coordinates along the exterior boundary against the maximum coordinates of the element mesh. However, more complex criteria could easily be incorporated to select nodes along a boundary of a flow domain of more complex boundary shapes. Currently, two-d elements are assumed to be 8-node quadrilaterals and 3-d elements are assumed to be 8-node hexahedral elements.

The following call is made:
CALL HFBELT (IELNOD,IELTYP,XN,YN,ZN,IELNBL,JLB,MXNBEL,QFLX)

where

JBL - Array for elements along the boundary
JLB(NBEL,1) is the element number
JBL(NBEL,2) is the side number of current element

MXNBEL - Maximum permitted number of elements along the exterior boundary that can be treated

NBEL - Current number of elements along the boundary

```
     IELNBL  - A work matrix whose first index must be at least
                 equal
                 to the total number of elements, and whose second
                 index must be at least equal to 8 for
                 quadri-lateral
                 elements and 26 for hexahedral elements
     JLFLX   - An array for elements along the boundary subject
                to
                prescribed flux
     JLSIDE  - A concurrent array for element side numbers of the
                elemente considered in the previous array
```

Currently a simple routine is used for selecting boundary sides:

```
     N O D T S T   - Subroutine to check if an element side
                     is to be included
     ************************************************************

     SUBROUTINE NODTST(X,Y,Z,*)
     IMPLICIT DOUBLE PRECISION
     COMMON/EXTRE2/ XMN,XMX,YMN,YMX,ZMN,ZMX
     IPRT=0
     IF(IPRT.GT.0) WRITE(6,900) X,Y,Z
 900 FORMAT(/6X,'NODTST---X=',F8.3,' Y=',F8.3,
1                              ' Z=',F8.3,' Now entered')
     IF(IPRT.GT.0) WRITE(6,910) XMN,XMX,YMN,YMX,ZMN,ZMX
 910 FORMAT(6X,9X,'XMN=',F8.3,' XMX=',F8.3,
    1             ' YMN=',F8.3,' YMX=',F8.3/
    2       6X,9X,'ZMN=',F8.3,' ZMX=',F8.3)
     IF(X.LT.XMN .OR. X.GT.XMX) RETURN1
     IF(Y.LT.YMN .OR. Y.GT.YMX) RETURN1
     IF(Z.LT.ZMN .OR. Z.GT.ZMX) RETURN1
     IF(IPRT.GT.0) WRITE(6,930)
 930 FORMAT(6X,9X,'Point was included')
     RETURN
     END
```

********** H F B F L W **********

Command:

```
  HFBFLW  IU JU Optional text
```

Purpose:

To specify element boundaries subject to prescribed flux etc.

The following call is made:

```
   CALL HFBFLW(IELNOD, INUN(1), 2, IEND2)
```

Below follows a description of the input data for subroutine HFBFLW

Data set 20----------Input element sides subject to prescribed flux boundary conditions (subroutine HFBFLW*)

Cols	Format	Identifier	Explanation
1-5	I5	IEL	Element having a side subject to prescribed flux.

6-10	I5	ISIDE	Local number of the element side under consideration. The following definition applies:
11-15	I5	NDS4(1)	Node in the global numbering system specifying the 1:st corner of the element side under consideration.
16-20	I5	NDS4(2)	Node in the global numbering system specifying the 2:nd corner of the element side under consideration.
21-25	I5	NDS4(3)	Node in the global numbering system specifying the 3:rd corner of the element side under consideration.
26-30	I5	NDS4(4)	Node in the global numbering system specifying the 4:th corner of the element side under consideration.
31-40	F10.0	QFLX1	Flux component in the x-direction
41-50	F10.0	QFLX2	Flux component in the y-direction
41-50	F10.0	QFLX2	Flux component in the z-direction

Note: If no data are to be input to this data set insert a record containing -1 in columns 1-2.

Both IEL and ISIDE may be set equal to zero provided that the four corner nodes are given by NDS4. Alternatively, the array for the corner nodes may be zeroed provided that IEL and ISIDE are properly specified.

********** H F E L H C **********

Command:
HFELHC IU 00 <Optional text>

Purpose:
Set element permeability (ref.value + region specification)

The following call is made:
```
CALL HFELHC (IELNOD,IELTYP,HCEL,IELMAT,XN,YN,ZN,INUNX)
```

The following program statements will be executed for this call:

```
 READ(INUN,810,END=120) XF1,XF2,YF1,YF2,ZF1,ZF2,FRPERM,IMAT
810 FORMAT(7F10.0,I3)

 IF(FRPERM.EQ.0.) FRPERM=1.0
```
B e g i n m a i n l o o p o v e r e l e m e n t s----------
```
 DO 100 IEL=1,NE

   LTYPE=IELTYP(IEL)
   MXNDE=LNTYP(1,IELTYP(IEL))
```
Mean depth of current element
set parameter indicating the number of corner nodes of current element, 8 for the standard hexahedral element (LTYPE=11) and 6 for the prismatic element (LTYPE=12)
```
   NCRN=8
   IF(LTYPE.EQ.12) NCRN=6

   J=0
   XP=0.0
   YP=0.0
   ZP=0.0
```

```
   Begin loop over nodal points of current element--------------------
    this loop is used to set permeability as a function of the

       DO 40 JJ=1,MXNDE
         JN=ABS(IELNOD(IEL,JJ))
         IF(JN.EQ.0) GOTO 40
         J=J+1
         IF(J.GT.NCRN) NOD9(J-NCRN)=JJ
         XYZ(1,J)=XN(JN)
         XYZ(2,J)=YN(JN)
         XYZ(3,J)=ZN(JN)
         XP=XP + XN(JN)
         YP=YP + YN(JN)
         ZP=ZP + ZN(JN)
 40    CONTINUE

       NNODE=J

       DEPTH=FDPTH2(XYZ)
       CFAC=FUNHC2(IEL,DEPTH)

       HCEL(IEL,1)=PRMREF*CFAC*FHCX
       HCEL(IEL,2)=PRMREF*CFAC*FHCY
       HCEL(IEL,3)=PRMREF*CFAC*FHCZ
-----------------------------------------------------------------------
    Check if the centroid (XP,YP,ZP) of current element is located
    within the interval XF1< X <XF2, YF1< Y <YF2, ZF1< Z <ZF2

       XP=XP/J
       YP=YP/J
       ZP=ZP/J

       IF(XP.LT.XF1 .OR. XP.GT.XF2 .OR. YP.LT.YF1
     1                               .OR. YP.GT.YF2) GOTO 80
       IF(NDIM.EQ.2) GOTO 70
       IF(ZP.LT.ZF1 .OR. ZP.GT.ZF2) GOTO 80

 70    CONTINUE

       HCEL(IEL,1)=HCEL(IEL,1)*FRPERM
       HCEL(IEL,2)=HCEL(IEL,2)*FRPERM
       HCEL(IEL,3)=HCEL(IEL,3)*FRPERM
       IF(IMAT.GT.0) IELMAT(IEL,1)=IMAT

       IF(IPRT.EQ.0) GOTO 80
       WRITE(6,945) IEL,HCEL(IEL,1),IEL,HCEL(IEL,2),IEL,HCEL(IEL,3)
945    FORMAT(/6X,'HFELHC-HCEL(',I3,',1)=',1P,E9.3,
     1          ' HCEL(',I3,',2)=',1P,E9.3,' HCEL(',I3,',3)=',1P,E9.3)

       WRITE(6,946) XP,YP,ZP,J,MXNDE
946    FORMAT( 6X,'HFELHC-XP=',F8.2,' YP=',F8.2,' ZP=',F8.2,
     1                               ' J=',I2,' MXNDE=',I2)
 80    CONTINUE

100 CONTINUE
-----------------------------------------------------------------------

 Where the interval is defined as:
     XF1 < X < XF2: range in X-direction
     YF1 < Y < YF2: range in Y-direction
     ZF1 < Z < ZF2: range in Z-direction
```

```
  FRPERM is a multiplication factor for permeability
  IMAT is the material number
-------------------------------------------------------------------

********************** H F E L I F ********************************

Command:
  HFELIF    IU 00 <Optional text>

Purpose:
  Input element incidences (discrete system)
  This is an alternative version of HFELIN.

The following call is made:
   CALL HFELIF (IELS,IELNOD,IELTYP,HCEL,INUNB)
-------------------------------------------------------------------
```

********** H F E L I N **********

Command:
HFELIN IU JU Optional text

Purpose:
To input element incidences

The following call is made:
CALL HFELIN(IELS,IELNOD,IELTYP,HCEL,INUN8)

Below follows a description of the input data for this subroutine

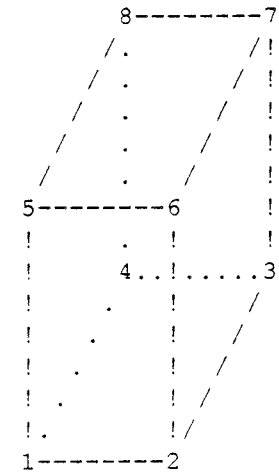
---

----------Element incidences (*HFELIN*, which is an alternative entry to subroutine *HFNDIN*)

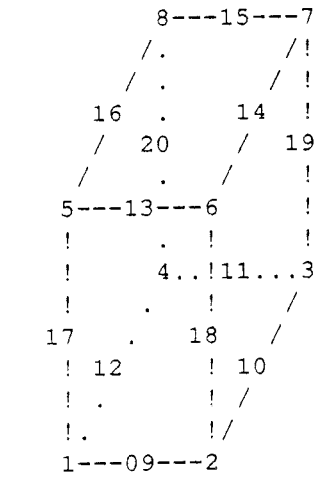
Cols	Format	Identifier	Explanation
1-5	I5	I1	Element number
6-10	I5	LTYPE	Element type

= 5 2-D eight node parabolic quadrilateral element

= 7 3-D eight node hexahedral element



=11 Curvilinear isoparametric hexahedral element (8-21 nodes)



Note: See also option ICP1(14) in data set no 5.

11-20	F10.0	HCELX	Permeability or, alternatively, hydraulic conductivity in the x-direction
21-30	F10.0	HCELY	Permeability or, alternatively, hydraulic conductivity in the y-direction
31-40	F10.0	HCELZ	Permeability or, alternatively, hydraulic conductivity in the z-direction
41-45	I5	NDS(1)	Nodal numbers (counter-clockwise ordering of the nodes)
.	.	.	.
.	.	.	.
.	.	.	.
76-80	I5	NDS(8)	Nodal numbers (counter-clockwise ordering of the nodes)

Note: The last record of the data set should be a record with -1 in cols 1-2

********** H F E L M P **********

Command:

HFELMP 00 00 + Optional text

Purpose:

To set material numbers for elements or nodes.

The following call is made:

```
CALL HFELMP(IELNOD,IELTYP,HCEL,IELMAT,NODES,
            XN,YN,ZN,
            IU)
```

For element oriented material properties (IMATM=0) then matrix IELMAT is set.
For node oriented material properties (IMATM=1) then array NODES is set.

Generally, all element or node material numbers are set equal to 1. Elements whose controids or nodes who are located within the interval:

XF1< X < XF2, YF1 < Y < YF2, ZF1 < X < ZF2

will currently be given material numbers equal to 2. The

previous interval specification is currently input in subroutine HFINPT* and transferred by the common block:

```
COMMON /XZERO / XF1,XF2,YF1,YF2,ZF1,ZF2
```

An alternative way of setting material no 2 is to specify the elements or nodes using the array LBSEEP in the common block:

```
COMMON /LBSEEP/ LBSEEP(100),MXSEEP,NSEEP
```

****************** H F E L M 1 **********************************

Command:

```
HFELM1   00 00 <Optional text>
```

Purpose:

Set element permeability according to values of -ielmat-

The following call is made:

```
CALL   HFELM1 (IELNOD,INLTYP,HCEL,IELMAT,XN,YNZN,INUNX)
```

This subroutine is used to set element permeability for the material number specified in the following input

```
READ(INUN,810,END=   ) IMAT,FRPERM
```

where IMAT is the material number to be set and PRMREF is a reference value of the permeability which must have been previously input in the parameter data input. Elements whose material numbers agree with IMAT will be given the element permeability as:

```
HCEL(IEL,1) = PERMV
HCEL(IEL,2) = PERMV
HCEL(IEL,3) = PERMV
```

Each element must be designated with a material number and the material numbers must have been previously stored in the matrix IELMAT(IEL,1) where IEL is an indes to denote the respective element. PERMV is defined as PRMREF*FRPERM

********************** H F E L P M ******************************

Command:

```
HFELPM
```

Purpose:

Specify (unsaturated) material properties for elements or nodes

The following call is made;

```
CALL   HFELPM (IELNOD,IELTYP,HCEL,IELMAT,NODES,XN,YN,ZN,INUN6)
```

******************** H E F L P 1 ********************************

Command:

```
HEFLP1   00 00 <Optional text>
```

Purpose:

Set element porosity.

The following call is made:

```
CALL   HFELP1 (IELNOD,IELTYP,HCEL,XN,YN,ZN,)
```

H F E L P 1 - Subroutine to set element porosity

Version 1, 1986-04-14

```
      SUBROUTINE HFELP1(IELNOD,IELTYP,HCEL,XN,YN,ZN)

      IMPLICIT REAL (A-H,O-Z)
      DIMENSION      IELNOD(MXNE,MX21),IELTYP(MXNE),HCEL(MXNE,3),
     1               XN(MXNP),YN(MXNP),ZN(MXNP),
     2               XYZ(3,21)

      COMMON/FECOM1/ MXNE,NE,MXNP,NP
      COMMON/FECOM4/ NNODE,NDOF,NEK,NDIM
      COMMON/FECOM6/ LNTYP(3,12)
      COMMON/HFREF1/ PREF,TREF,HCREF,PRMREF,POREF,DNFREF,VSCREF
      COMMON/IOPOR / IOPOR
      COMMON/MXNPEL/ MX21

      CHARACTER*6    POR2
      LOGICAL        LPORFN
      DATA           LPORFN /.FALSE./

      GOTO 10
-----------------------------------------------------------------------
      H F E L P 2   - ENTRY TO SELECTION OF POROSITY FUNCTION
-----------------------------------------------------------------------
      ENTRY   HFELP3(POR2,FPOR2)
      WRITE(6,920) POR2
  920 FORMAT( 6X,'HFELP3---Entry to *HFELP1* for porosity',
     1                                        ' function:  ',A6)
      LPORFN = .TRUE.
      RETURN
-----------------------------------------------------------------------
   10 CONTINUE
      IPRT=1
      IF(NDIM.GT.2) THEN
        WRITE(6,930) NDIM
  930   FORMAT(/6X,'HFELP1---Attention: NDIM=',I1,' This means that'/
     1          6X,9X,'HCEL must be dimensioned as HCEL(MXNE,4) in',
     2                ' the main program')
      ENDIF
     B E G I N   M A I N   L O O P   O V E R   E L E M E N T S----------
      DO 100 IEL=1,NE
        LTYPE=IELTYP(IEL)
        MXNDE=LNTYP(1,IELTYP(IEL))
        J=0
---BEGIN LOOP OVER NODAL POINTS OF CURRENT ELEMENT--------------------
    THIS LOOP IS USED TO SET POROSITY USING FUNCTION *FPOR2*
        DO 40 JJ=1,MXNDE
          JN=ABS(IELNOD(IEL,JJ))
          IF(JN.EQ.0) GOTO 40
          J=J+1
          XYZ(1,J)=XN(JN)
          XYZ(2,J)=YN(JN)
          XYZ(3,J)=ZN(JN)
   40   CONTINUE
        IF(LPORFN) THEN
          HCEL(IEL,NDIM+1) = FPOR2(IEL,XYZ,J)
        ELSE
          HCEL(IEL,NDIM+1) = POREF
        ENDIF
  100 CONTINUE
```

```
    WRITE(6,950) NE,POREF,LPORFN
950 FORMAT(/6X,'HFELP1---Now set element porosity using function',
   1                    ' *FPOR2*'/
   2         6X,9X,'NE=',I4,' POREF=',1P,E12.4,' LPORFN=',L2)

    IF(IPRT.GT.0) THEN
      WRITE(6,960) (IEL,HCEL(IEL,NDIM+1),IEL=1,NE)
960   FORMAT(6X,5(I4,1P,E10.3)/(6X,5(I4,E10.3)))
    ENDIF

    IF(IOPOR.NE.0) THEN
      WRITE(6,970) IOPOR
970   FORMAT(6X,'HFELP1---IOPOR=',I1,' was changed into zero')
      IOPOR = 0
    ENDIF

    RETURN
    END
```

********** H F I E L N **********

Command:

```
  HFIELN  00  00 + Optional text
```

Purpose:

To redefine the element incidences for hexahedral 8-21 nodes elements

The following call is made:

```
   CALL HFIELN (IELNOD,IELTYP)
```

********** H F I N F R **********

Command:

```
  HFINFR  IU JU Optional text
```

Purpose:

To input list of fracture elements

The following call is made:

```
   CALL HFINFR(INUN(1),INUN(4))
```

Below follows a description of the input data:

Data set 1 ----------Elements forming fracture zones (subr. HFINFR)

Cols	Format	Identifier	Explanation
1-5	I5	NLFRAC	Number of fracture elements to be specified on the ensuing record

Note: If no fracture elements are to be input, then set -1 in columns 1-5 and ommit the next record.

New record----------

1-5	I5	LFRAC(1)	Element number
6-10	I5	LFRAC(2)	Element number
.			
.			
.			
76-80	I5	LFRAC(16)	Elment number Etc.

Note: Subroutine HFINFR is called from subroutine HFSUB1

********** H F I N P 3 **********

Command:
HFINP3 IU JU Optional text

Purpose:
To input characteristic curves for capillary pressure and and saturation, relative permeability versus saturation and to compute curves for the derivative of saturation versus capillary pressure for respective material.

The following call is made:
CALL HFINP3(INUN6, 0)

New record----------Input of material properties for unsaturated conditions (subr. *HFINP3*)
Currently max. 3 different materials can be input and the maximum permitted number of data points for each material is set to 41.

(Format(3I5,I2,3I1,15A4))

Cols	Format	Identifier	Description
1-5	I5	LMAT	Material number (1,2 or 3)
6-10	I5	NTHX1	Number of data points to be input for the relationship between saturation (or moisture content) and the capillary pressure.
11-15	I5	NTHX2	Number of data points to be input for the relationship between the saturation (or moisture content) and the relative permeability.
16-17	I2	IUTRF	Control parameter to indicate if the input data are to be transformed. = -1 saturation is input = 0 moisture content is input = 1 moisture content is input and is to be transformed
18	I1	IGAS	Indicates that gas permeability is to be input (IGAS > 0)
19	I1	INOP	Indicates type fo input format = 0 2F10.0 = 1 7F10.0 = 2 Free format (not used yet)
20	I1	IPRT	Control parameter for printout. = 0 No printout = 1 Print all input = 2 Print all input plus the transformed data
21-75	15A4	TEX(15)	Alphanumeric information to identify the input data.
		ISPRIM	Parameter to control values of end points of capillary pressure derivative curves (This parameter is not yet input here but in subroutine *HFINPT*)

New record----------Saturation versus capillary pressure (subr. HFINP3*)

Cols	Format	Identifier	Description
1-10	F10.0	THX1(1,L)	Moisture content or saturation at data point No 1 for material No L.
11-20	F10.0	PHC(1,L)	Capillary pressure at data point No 1 for material No L.
21-30	F10.0	THX1(2,L)	Moisture content or saturation at data point No 2 for material No L.
31-40	F10.0	PHC(2,L)	Capillary pressure at data point No 2 for material No L.
41-50	F10.0	THX1(3,L)	Moisture content or saturation at data point No 3 for material No L.
51-50	F10.0	PHC(3,L)	Capillary pressure at data point No 3 for
51-60	F10.0	THX1(4,L)	Moisture content or saturation at data point No 4 for material No L.
61-70	F10.0	PHC(4,L)	Capillary pressure at data point No 4 for material No L.

.
to be continued on as many records as necessary.

New record----------Relative permeability versus capillary pressure (subr. *HFINP3*)

Cols	Format	Identifier	Description
1-10	F10.0	THX2(1,L)	Relative permeability at data point No 1 for material No L.
11-20	F10.0	HCRX(1,L)	Capillary pressure at data point No 1 for material No L.
21-30	F10.0	THX2(2,L)	Relative permeability at data point No 2 for material No L.
31-40	F10.0	HCRX(2,L)	Capillary pressure at data point No 2 for material No L.
41-50	F10.0	THX2(3,L)	Relative permeability at data point No 3 for material No L.
51-60	F10.0	HCRX(3,L)	Capillary pressure at data point No 3 for for material No L.
61-70	F10.0	THX2(4,L)	Relative permeability at data point No 4 for material No L.
71-80	F10.0	HCRX(4,L)	Capillary pressure at data point No 4 for material No L.

to be continued on as many records as necessary.

********** H F I N P 4 **********

Command:

```
HFINP4  IU 00 Optional text
```

Purpose:

To specify boundary and initial conditions for unsaturated flow test cases.

The following call is made:

```
HFINP4 (PH,LBPH,PERM,XN,YN,ZN,INUN)
       IF (IU.GT.0) THEN INUN6=IU ELSE INUN6=INUN(6) ENDIF
```

Below follows a description of the input data:

New record----------Boundary conditions for various test cases (subr. *HFINP4*)

Cols	Format	Identifier	Description
1-80	20A4	TEX(20)	Alphanumeric information.

New record----------Control parameters (subr. *HFINP4*)

Cols	Format	Identifier	Description
1-5	I5	IND	Control parameter. = 0 No action = 1 Hydrostatic pressure to be set for all nodes = 2 Vertical boundary with prescribed pressure to be set according to X=X2 P=(Z-WLV1)*DENF*G = 3 Vertical boundary with prescribed pressure to be set according to X=X1, Y1 < Y < Y1X P=(Z-WLV1)*DENF*G = 4 Vertical boundary with prescribed pressure to be set according to X1 < X < X2 Y=Y2 P=(Z-WTBL)*DENF*G = 5 Hydrostatic pressure to be set according to X=XMAX, YMIN < Y < YMAX Y=YMAX, XMIN < X < XMAX P=(Z-WTBL)*DENF*G
6-10	F10.0	TOL	A tolerance value.
11-15	F10.0	RWELL	Radius of well.
16-20	F5.0	DGAMMA	Pressure gradient (Pa/metre) to be applied to the initial flow field. A Positive value of DGAMMA indicates that there is an upward pressure gradient. Conversely, a negative value of DGAMMA indicates that there is a downward pressure gradient.
21-25	I5	IPRT	Printout control parameter = 0 No action = 1 Printout values of fluid density and acceleration of gravity = 2 Printout of coordinates and pressure values set at boundaries = 3 Printout of pressure values nodal codes for all nodes.

The following input is for gas migration calculation and may be discarded.

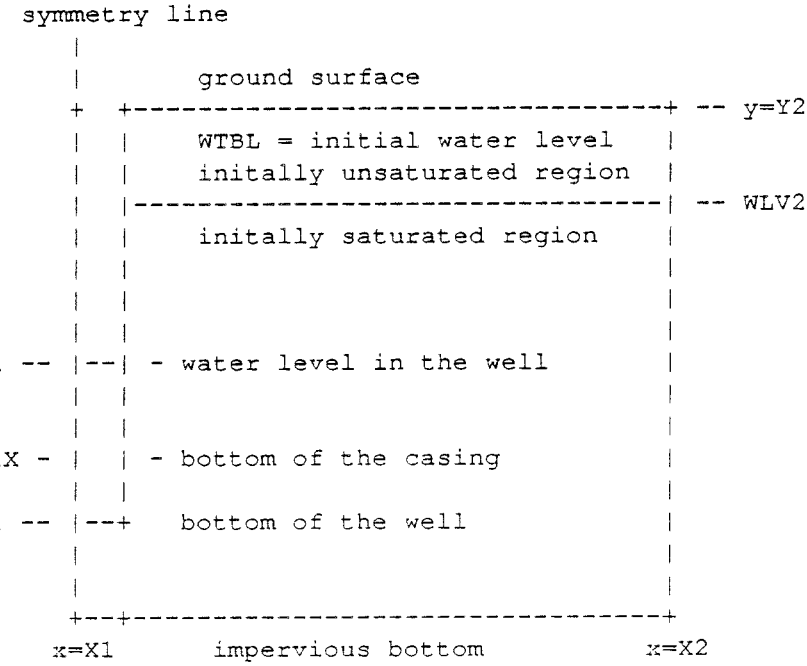
26-30	F5.0	YMN	Y-coordinate to specify bottom of region.
31-35	F5.0	PMN	Permeability at bottom of region.
36-40	F5.0	YMX	Y-coordinate to specify top of region.

41-45	F5.0	PMX	Permeability at top of region.
46-50	F5.0	HG	Height of gas cushion (metres).
51-55	F5.0	HS	Depth to sea bottom (metres).
56-75	10I2	IOPBC1(10)	Array for control parameters associated with boundary and conditions for gas migration calculation (see subroutine SFRIN1 in user controlled input).

New record----------Geometric properties of a test case for unsaturated flow a well (subr. *HFINP4*)

Cols	Format	Identifier	Description
1-10	F10.0	WTBL	Initial elevation of the water table. (see definition sketch below)
11-20	F10.0	WLV1	Level of the water table at X=X1. (see definition sketch below)
21-30	F10.0	WLV2	Level of the water tabel at X=XMAX. (see definition sketch below)
31-40	F10.0	X1	Currently set by the program as X1=XMIN.
41-50	F10.0	Y1	Level of the water table in the well. (see definition sketch below)
51-60	F10.0	Y1X	Level of the bottom of the casing in the well (see definition sketch below)

Definition sketch:



Note: The following record is currently not used

New record----------Selection of nodes for printout (*HFINP4* and *HFNDSL*)

Cols	Format	Identifier	Description
1-10	F10.0	XX1	All nodes located within the area defined as:
11-20	F10.0	XX2	XX1 < X < XX2 YY1 < Y < YY2 ZZ1 < Z < ZZ2
21-30	F10.0	YY1	are to be selected for printout. The selected node numbers are stored (added to if there already are any node numbers selected by some other program) in the common block /NODPRT/. As many records as desired may be input, but the maximum number of node numbers that can be stored in the common block is currently limited to 100.
31-40	F10.0	YY2	
41-50	F10.0	ZZ1	
51-60	F10.0	ZZ2	

********** H F I N 4 P **********

Command:
HFIN4P IU 00 <Optional text>

Purpose:
Set the inital pressure distribution (hydrostatic).
This is an alternative version of HFINP4

The following call is made:
```
CALL HFIN4P (PH,LBPH,PERM,XN,YN,ZN,INUN6)
```

********** H F I N P 5 **********

Command:
HFINP5 IU 00 Optional text

Purpose:
To input elastic rock properties

The following call is made:
```
HFINP5 (IELNOD,IELTYP,XN,YN,ZN,DU,LBDU,INUN7)

        IF (IU.GT.0) THEN
          INUN7=IU
        ELSE
          INUN7=INUN(7)
        ENDIF
```

Where IU is file number for the subsequent input
JU is file number for output.

New record----------Elastic properties of the rock medium (subr. *HFINP5*)

Cols	Format	Identifier	Description
1-10	F10.0	EM	Young's modulus.

11-20 F10.0 PR Poisson's ratio.

21-30 F10.0 ALFA Coefficient of thermal volume expansion of the rock mass.

********** H F M S R C **********

Command:
HFMSRC IU 00 <Optional text>

Purpose:
To input volumetric mass sources

The following call is made:
IF(IUTX.LE.0) CALL HFMSRC(INUN(1), 0, 1)
IF(IUTX.GT.0) CALL HFMSRC(INUN(1),INUN(4), 1)

Data set 1----------Mass source/sinks

Cols	Format	Identifier	Explanation
1-3	I3	LPSINK(1)	Element with the last appearance of the 1:st nodal source/sink.
4-5	I2	NPSINK(1)	Local node number in the element containing the 1:st nodal source/sink.
6-15	F10.0	WPSINK(1)	Strength of source/sink no 1.
16-18	I3	LPSINK(2)	Element with the last appearance of the 2:nd nodal source/sink.
19-20	I2	NPSINK(2)	Local node number in the element containing the 2:nd nodal source/sink.
21-30	F10.0	WPSINK(2)	Strength of source/sink no 2.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

Note 1: Currently it is only permitted to input 5 mass point sources

Note 2: Currently the values are multiplied by the fluid density DNFREF and divided by 86400 to obtain the fluxes in kg/sec, implying that the input is given in m^3/day (see subroutine *HFINPT*).

********** H F N D I F **********

Command:
HFNDIF

Purpose:
Input nodal points (discrete system)
This is an atternative version of HFNDIN

The following call is made:
CALL HFNDIF (NODES,XN,YN,ZN,LBHH,HH,3,INUN8)

********** H F N D I N **********

Command:
HFNDIN IU 00 <Optional text>

Purpose:
To input coordinates of nodes in element mesh

The following call is made:

```
CALL HFNDIN(NODES,XN,YN,ZN,LBHH,HH,3,INUN8)
```

Below follows a description of the input data:

----------Nodal coordinates and initial or prescribed values of pressure, fluid- and rock temperature (subr. HFNDIN)

Cols	Format	Identifier	Explanation
1-80	20A4	TEX(20)	Alphanumeric information to be used for the identification of current element mesh.
New record			
1-5	I5	MXNDE	Maximum number of nodes per element.
6-10	I5	NEIN	Number of elements to be input.
11-15	I5	NPIN	Number of nodes to be input.
16-20	I5	IEL21X	Type of hexahedral element. = 0 To indicate that 2-D or standard 3-D hexahedral elements are to be input = 1 See Data set 5: ICP1(14) = 2 "-
21-25	I5	IMTYPE	Control parameter to check if the element being input applies to current version (No 1) of the input subroutine (*HFNDIN*). = 0 or 1 Version No 1 = 2 Version No 2
26-30	I5	NDIM	Number of spatial dimensions.

Note: The previous items are usually written in connection with the creation of the element mesh. The purpose of this is to facilitate the checking of the element mesh. Both the previous records may be left blank, except for the parameter NDIM, which must be specified also here.

New record----------

Cols	Format	Identifier	Explanation
1-5	I5	NDS1	Node number
6-15	F10.0	X1	X-coordinate
16-25	F10.0	Y1	Y-coordinate
26-35	F10.0	Z1	Z-coordinate
36-40	I5	LB1(1)	Code (=0 initial value, =1 prescribed value)
41-50	F10.0	HH1(1)	Prescribed or initial value of pressure
51-55	I5	LB1(2)	Code (=0 initial value, =1 prescribed value)
56-65	F10.0	HH1(2)	Prescribed or initial value of fluid temperature.
66-70	I5	LB1(3)	Code (=0 initial value, =1 prescribed value).
71-80	F10.0	HH1(3)	Prescribed or initial value of the rock temperature.

Note: The last record of the data set should be a record with -1 in cols 1-2

```
   ********** H F N F R C **********

   Command:
     HFNFRC  IU  00 <Optional text>

   Purpose:
     To set nodal permeability

   Example of usage

     HFNFRC  05  00 Set nodal permeability for -Region A -
     X1RA  X2RA   Y1RA   Y2RA   Z1RA  Z2RA  FRPERM LMAT

     where the region is given by

     X1RA < X < X2RA, Y1RA < Y < Y2RA, Z1RA < Z < Z1RA
     FRPERM is a multiplication factor for current permeability value
     LMAT   is the material number to be given the elements or nodes
            located within the specified region.

   The following call is made:
      CALL HFNFRC(XN,YN,ZN,PERM,NODES,IU)

   ------------------------------------------------------------------
   The following program statements are executed for this command:

    READ(INUN,810,END=20) XF1,XF2,YF1,YF2,ZF1,ZF2,FRPERM,IMAT
810 FORMAT(7F10.0,I3)
    IF(FRPERM.EQ.0.) FRPERM=1.0
    DO 10 I=1,NP
      IF(XN(I).LT.XF1-TOL.OR. XN(I).GT.XF2+TOL) GOTO 10
      IF(YN(I).LT.YF1-TOL.OR. YN(I).GT.YF2+TOL) GOTO 10
      IF(ZN(I).LT.ZF1-TOL.OR. ZN(I).GT.ZF2+TOL) GOTO 10
      PERM(I)=PERM(I) * FRPERM
      IF(IMAT.GT.0) NODES(I)=IMAT
      NTOT=NTOT+1
      IF(IPRT.GT.0) THEN
        NC=NC+1
        JARY(NC)=I
        IF(NC.LT.10) GOTO 10

        WRITE(6,911) JARY
911     FORMAT(6X,10I5)

        NC=0
      ENDIF
 10 CONTINUE
  ------------------------------------------------------------------

   ********** H F P E R M **********
   Command:
     HFPERM   00 00 <Optional text>

   Purpose:
     Set nodal permeabilities

   The following program statements are executed for this command:
    IF(NDIM.EQ.1) THEN
      DO 20 I=1,NP
        PERM(I) = PERM1 * FUNHC2(IDUM,XN(I))
 20   CONTINUE
    ELSE IF(NDIM.EQ.2) THEN
      DO 30 I=1,NP
        PERM(I) = PERM1 * FUNHC2(IDUM,YN(I))
 30   CONTINUE
```

```
   ELSE IF(NDIM.EQ.3) THEN
     DO 40 I=1,NP
       PERM(I) = PERM1 * FUNHC2(IDUM,ZN(I))
40   CONTINUE
   ENDIF
```

where PERM1 is a reference value of permebility see input for parameter data.
PERM is the array for the nodal permeability values
FUNHC2 is a function to relate the permeability dependence versus depth. The function to be selected must be specified in the input of the parameter data.

The following call is made:

```
   CALL   HFPERM (XN,YN,ZN,PERM,INUN(10) )
```

********** H F P R M A **********

Command:

HFPRMA 00 00 + Optional text

Purpose:

To set mid-side nodal permeability as averages of adjacent nodes for respective element.

The following call is made:

```
   CALL HFPRMA(IELNOD,PERM)
```

********** H F S R C 1 **********

Command:

HFSRC1 IU JU Optional text

Purpose:

To input distributed or point heat sources

Note: INUN(2) is used as input file for these data.

The following call is made:

```
   CALL HFSRC1(XN,YN,ZN,DHS,IELNOD,IELTYP,INUN(2), 0)
```

Description of input data:

There are three modes of input. The input mode is selected by a control word as follows:

'HSDR' in column 1-4 on the first record indicates that the subsequent record contains a region specification for a distributed heat source

```
HSDR Optional text
(A4,1X,A67)
WDHS,XREP(1),XREP(2),XREP(3),XREP(4),XREP(5),XREP(6)
(7F10.0)
```

where WHDS is the heat emitted per unit volume and time
All nodes encounrered within the following region are condidered: XREP(1) <= X <= XREP(2)
XREP(3) <= Y <= XREP(4)
XREP(5) <= Z <= XREP(6)

'HSCR' in column 1-4 on the first record indicates that the subsequent record contains a region specification plus a total amount of heat emitted from this region. the program will then find all nodes located in the specified region and distribute the total load uniformly as concentrated heat sources.

HSCR Optional Text
(A4,1X,A67)
WDHS,XREP(1),XREP(2),XREP(3),XREP(4),XREP(5),XREP(6)
(7F10.0)
where WHDS is the heat emitted per unit volume and time
All nodes encountered within the following region are considered: XREP(1) <= X <= XREP(2)
XREP(3) <= Y <= XREP(4)
XREP(5) <= Z <= XREP(6)

'HSCN' in column 1-4 on the first record indicates that the subsequent record contains a list of nodes plus the heat source strength that should be applied to each of these nodes.

HSCR Optional Text
(A4,1X,A67)
NS, WS, NNDS (node list)
(I2,F8.2, 14I5/(16I5))
where WHDS is the heat emitted per unit volume and time at each node.

The input is terminated by putting '-1' or 'END' in columns 1-4

Input of distributed heat sources (subroutine *HFSCRX*)

Cols	Format	Identifier	Explanation
1-2	I2	IOP	Control option for heat source input = 0 Apply distributed heat source using the default coordinate specification of the heat source. Currently, the following is default (2-D): -500 < x < +500 -505 < y < -495 -E20 < z < +E20 =-1 No distributed heat sources to be applied (The rest of the information on this record will be disregarded) = 1 Apply distributed heat source using the coordinate specification as follows on current record (see explanation for XREP below)
3-10	F8.0	WDHS	Heat source strength. To be specified as heat flow rate, per unit area in 2-D, per unit volume in 3-D.

11-20	F10.0	XREP(1)	X-min	The location of the heat source:
21-30	F10.0	XREP(2)	X-max	
31-40	F10.0	XREP(3)	Y-min	x-min < x < x-max
41-50	F10.0	XREP(4)	Y-max	y-min < y < y-max
51-60	F10.0	XREP(5)	Z-min	z-min < z < z-max
61-70	F10.0	XREP(6)	Z-max	

Data set 16----------Input of concentrated heat sources (subroutine *HFSCR1*)

Cols	Format	Identifier	Explanation
1-2	I2	NS	Number of heat sources to be input by current record.
3-10	F8.0	WS	Strength of the heat source(s) being input by current record.
11-15	I5	NNDS(1)	Nodal source no 1.
16-20	I5	NNDS(2)	Nodal source no 2.
.	.	.	.
.	.	.	.
.	.	.	.
76-80	I5	NNDS(14)	Nodal source no 14.

Note: The last record of this data set must contain -1 in columns 1-2

********** H Y D S B S **********

Command:

```
HYDSBS  IU  00 <Optional text>
```

Purpose:

To impose a call to the general input subroutine *HYDSBS*

The following call is made:

```
CALL HYDSBS(PH,LBPH,SW,
            PERM,PORSTY,VISC,DENF,NODES,
            XN,YN,ZN,IELNOD,IELTYP,HCEL,IELMAT,
            IU)
```

********** M V S D E 1 **********

Command:

```
MVSDE1  IU 00 <Optional text>
```

Purpose:

Modify element mesh according to input specification

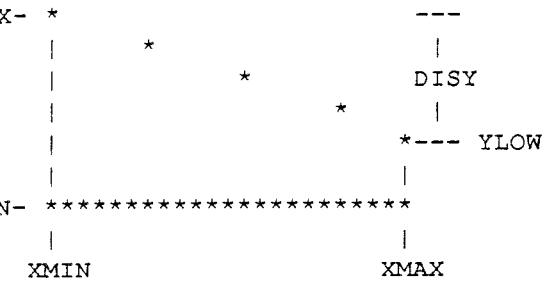
Description of parameters:

```
NDIM        - number of spatial dimensions
IOPSCN      - an integer to control the numbering
                if set equal to zero then the program will
                number the elements such that the band- or
                front-width will be minimized
-------------------------------------------------------------
                = 1   Y - X - Z
                = 2   X - Y - Z
                = 3   Z - X - Y
                = 4   Z - Y - X
-------------------------------------------------------------
IOP         - An integer to control input
```

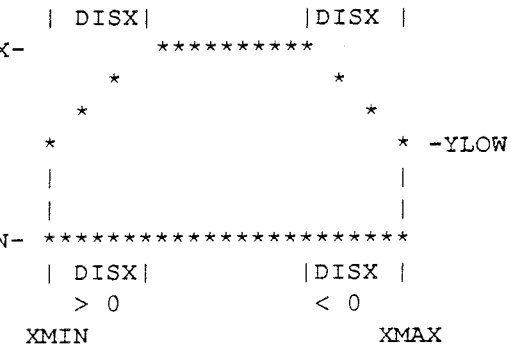
=0 equidistant mesh size in each direction
=1 mesh sizes to read by the program
=2 mesh sizes to be read by the program but polar coordinates are assumed
=3 mesh points to be given in absolute coordinates (free format)

IROT - An integer to control rotation of the generated mesh

ISLOPE - An integer to control if slope is to be applied
= 0 no slope to be applied
= 1 unidirectional slope to be applied to part of top boundary



= 2 unidirectional slope to be applied to part of top boundary



SLOPE - unidirectional slope to be applied to the top boundary of the mesh
DISX - Maximum displacement in the x-direction
CRDREF - an elevation below which no nodes may be displaced
INUN - logical unit number for output of the resulting element mesh
NLX - an integer to set number of grid cells in the x-direction
NLY - ditto in the y-direction
NLZ - ditto in the z-direction
DLX - array for the sizes of the grid cells in the x-direction
DLY - ditto in the y-direction
DLZ - ditto in the z-direction
SLOPE1.....Desired slope of the inclined part of one of the vertical boundaries
SLOPE=(YMAX-YLOW)/DISX
YLOW.....Lowest permissible level at which nodal points may be displaced

```
DISX........Maximum displacement in the x-direction
              computed for non-zero slope
              DISX=(YMAX-YLOW)/SLOPE
Either of SLOPE1 or DISX must be non-zero on input. If both
variables are set non-zero then SLOPE1 will be used as input

The following program statements is executed to the call on the
subroutine:
   READ(IU,*,END=...) CRDREF,DISX


The following call is made:
   CALL   MVSDE1 (XN,YN,NP,SLOPE,CRDREF,DISX)
```

```
****************** N O D S L 2 ***********************************

Command:
  NODSL2  IU  00 + Optional text

Purpose:
  To specify regions for nodal printout

Example of usage
  NODSL2  05  00 Specification of regions for printot
  X1R1  X2R1   Y1R1   Y2R1   Z1R1  Z2R1   0
  X1R2  X2R2   Y1R2   Y2R2   Z1R2  Z2R2   0
    .     .      .      .      .     .     .
    .     .      .      .      .     .     .
  X1R1  X2R1   Y1R1   Y2R1   Z1R1  Z2R1  -1

  where the regions are given by
  X1RI < X < X2RI, Y1RI < Y < Y2RI, Z1RI < Z < Z1RI

The following call is made:
   CALL NODSL2(XN,YN,ZN,NP,NDSPRT,MXNPRT,NPRT,INUN(2))
The following program statements are executed for this call:

 N O D S L 2   - SUBROUTINE TO LOCATE NODES
                 THIS VERSION READS IN AN INTERVAL
                 X1 < X < X2,   Y1 < Y < Y2,   Z1 < Z < Z2
                 AND FINDS ALL NODES LOCATED WITHIN THIS
                 INTERVAL


 SUBROUTINE NODSL2(XN,YN,ZN,NP, NDSPRT,MXNPRT,NPRT, INUN)

 IMPLICIT REAL (A-H,O-Z)

 DIMENSION XN(NP),YN(NP),ZN(NP)

 DIMENSION NDSPRT(MXNPRT)

 COMMON/FECOM1/ MXNE,NE,MXNP,NP
 COMMON/FECOM4/ NNODE,NDOF,NEK,NDIM

 TOL=0.001

 WRITE(6,905) NP,INUN,MXNPRT,NPRT,TOL
905 FORMAT(/6X,'NODSL2---NP=',I4,' INUN=',I2,' MXNPRT=',I3,
1                  ' NPRT=',I3,' tolerance (TOL)=',1PE10.3)

  5 READ(INUN,810,END=50) X1,X2,Y1,Y2,Z1,Z2,IMAT
810 FORMAT(6F10.0,I5)

 NPRT1=NPRT
```

```
      XMN=X1-TOL
      XMX=X2+TOL
      YMN=Y1-TOL
      YMX=Y2+TOL
      ZMN=Z1-TOL
      ZMX=Z2+TOL

      WRITE(6,910) X1,X2,Y1,Y2,Z1,Z2,IMAT,NPRT,MXNPRT
  910 FORMAT(/6X,'NODSL2---X1=',F8.2,' < X < X2=',F8.2/
     >         6X,9X,'Y1=',F8.2,' < Y < Y2=',F8.2/
     1         6X,9X,'Z1=',F8.2,' < Z < Z2=',F8.2/
     2         6X,9X,'IMAT=',I3,' NPRT=',I3,'  MXNPRT=',I3)

      DO 20 I=1,NP
      IF(XN(I).LT.XMN .OR. XN(I).GT.XMX .OR.
     1   YN(I).LT.YMN .OR. YN(I).GT.YMX) GOTO 20

      IF(NDIM.LT.3) GOTO 10
      IF(ZN(I).LT.ZMN .OR. ZN(I).GT.ZMX) GOTO 20

   10 NPRT=NPRT+1
      IF(NPRT.GT.MXNPRT) GOTO 20
      NDSPRT(NPRT)=I
   20 CONTINUE

      IF(NPRT.EQ.NPRT1) GOTO 40
      NPRT1=NPRT1+1

      IF(NPRT.GT.MXNPRT) THEN

        WRITE(6,920) NPRT,MXNPRT
  920   FORMAT(/6X,'NODSL2---NPRT=',I3,' > MXNPRT=',I3)

        NPRT=MXNPRT

      ENDIF

      NPRTX=NPRT-NPRT1+1

      WRITE(6,930) NPRTX
  930 FORMAT(/6X,'NODSL2---NPRTX=',I3,' nodes selected by ',
     1                    'current specification:')

      IF(NPRT1.LE.MXNPRT) WRITE(6,931) (NDSPRT(I),I=NPRT1,NPRT)
  931 FORMAT(6X,10I5)

      IF(IMAT.LT.0) GOTO 60

      GOTO 5

   40 WRITE(6,940)
  940 FORMAT(/6X,'NODSL2---No nodes selected by current region',
     1                     ' specification')
      IF(IMAT.LT.0) THEN
        GOTO 60
      ELSE
        GOTO 5
      ENDIF

   50 WRITE(6,945) INUN
  945 FORMAT(6X,'NODSL2---Now read to End of File - INUN=',I5)

   60 WRITE(6,950)
  950 FORMAT(6X,'NODSL2',66('-'))
```

```
      RETURN
      END
C-----------------------------------------------------------------------
C***************** N O D S L 4 ***********************************
C
C Command:
C   NODSL4    IU 00 <Optional text>
C
C Purpose:
C   Specify and set zones of specific material properties.
C
C The following call is made:
C    CALL    NODSL4 (IELNOD,IELTYP,IELMAT,NODES,XN,YN,ZN,INUN8)
C
C The following program statements are executed for this call:
C
C  N O D S L 4  - Subroutine to indentfy elements and their related *
C                 nodes for given specifications
C                 this version is used to assign material
C                 properties to elements and nodes
C
      SUBROUTINE NODSL4(IELNOD,IELTYP,IELMAT,NODES,XN,YN,ZN, INUN)
C
      IMPLICIT REAL (A-H,O-Z)
      DIMENSION IELNOD(MXNE,MX21),IELTYP(MXNE),
     >          IELMAT(MXNE,2),
     1          NODES(MXNP),XN(MXNP),YN(MXNP),ZN(MXNP)
C
      DIMENSION JNSKIP(125)
C
      COMMON/FECOM1/ MXNE,NE,MXNP,NP
      COMMON/FECOM6/ LNTYP(3,12)
      COMMON/MXNPEL/ MX21
C
      MXSKIP=125
      NSKIP=0
C
   10 READ(INUN,810,END=15) XX1,XX2,YY1,YY2,ZZ1,ZZ2,IMAT
  810 FORMAT(6F10.0,I5)
C
      WRITE(6,910) XX1,XX2,YY1,YY2,ZZ1,ZZ2,IMAT
  910 FORMAT(/6X,'NODSL4---XX1=',F8.2,' XX2=',F8.2/6X,9X,'YY1=',F8.2,
     1                    ' YY2=',F8.2/6X,9X,'ZZ1=',F8.2,' ZZ2=',F8.2/
     2         6X,9X,'material number:',I3)
C
      CALL NODSLC(XN,YN,ZN,NP,XX1,XX2,YY1,YY2,ZZ1,ZZ2,
     1            JNSKIP,MXSKIP,NSKIP)
C
      IF(IMAT.GT.0) GOTO 10
C
      GOTO 16
   15 WRITE(6,935) INUN
  935 FORMAT(6X,'NODSL4---Now read to End of File - INUN=',I5)
   16 CONTINUE
C
      NEX=0
C
      DO 50 IEL=1,NE
      NNODE=LNTYP(1,IELTYP(IEL))
      NCHK=0
C
      DO 30 J=1,NNODE
      DO 20 K=1,NSKIP
      IF(IELNOD(IEL,J).NE.JNSKIP(K)) GOTO 20
      NCHK=NCHK+1
   20 CONTINUE
   30 CONTINUE
```

```
      IF(NCHK.EQ.NNODE) THEN

        NEX=NEX+1
        IELMAT(IEL,1)=ABS(IMAT)
      ENDIF

   50 CONTINUE

      NPX=0
      DO 80 I=1,NP
      DO 60 J=1,NSKIP
      IF(JNSKIP(J).EQ.I) GOTO 70
   60 CONTINUE

      GOTO 80

   70 NPX=NPX+1
      NODES(I)=ABS(IMAT)

   80 CONTINUE

      WRITE(6,940) NP,NPX,(J,JNSKIP(J),J=1,NPX)
  940 FORMAT(/6X,'NODSL4---Total number of nodes (NP) =',I4/
     1         6X,9X,       'Number of nodes selected   =',I4/
     2         6X,9X,       'The following nodes were selected :'/
     3        (6X,5(I5,I5,4X)))

      WRITE(6,950) NE,NEX
  950 FORMAT(/6X,'NODSL4---Total number of elements (NE) =',I4/
     1         6X,9X,       'number of elements selected   =',I4)

      RETURN
      END
```

********** P H X M A X **********

Command:
  PHXMAX

Purpose:
  Set prescribed pressure on right boundary.

The following call is made:
```
    CALL   PHXMAX (PH,LBPH,XN,YN,'PH')
```

********** P H X M I N **********

Command:
  PHXMIN

Purpose:
  Set prescribed pressure on left hand boundary.

The following call is made:
```
   CALL   PHXMIN (PH,LBPH,XN,YN,'PH')
```

********** P H Y D R O **********

Command:
  PHYDRO IU JU Optional text

Purpose:
  To select node numbers for printout
  etc.

The following call is made:
CALL PHYDRO(XN,YN,ZN,PH,LBPH,TF,LBTF,TR,LBTR,INUNX)

****

Data set ----------Input node numbers subject to hydrostatic conditions (subroutine *PHYDRO*)

Cols	Format	Identifier	Explanation
1-5	I5	NI	Number of nodes to be specified on the ensuing record
6-10	I5	I1	Node at which the groundwater table is situated
11-15	I5	IPRT	Printout control parameter (=0 if no printout is desired, =1 if printout is desired)

New record----------

Cols	Format	Identifier	Description
1-5	I5	NODES(1)	1:st node in the vertical line of nodes ordered from the top to the bottom of the flow domain.
6-10	I5	NODES(2)	2:nd node.
.	.	.	.
.	.	.	.
76-80	I5	NODES(16)	16:th node.

Note: If no data are to be input to this data set insert a record containing -1 in columns 1-2.

********** P H Y M A X **********

Command:
PHYMAX

Purpose:
Set prescribed pressure on top boundary

The following call is made:
CALL PHYMAX (PH,LBPH,XN,YN,'PH')

********** P H Y M I N **********

Command:
PHYMIN

Purpose:
Set prescribed pressure on bottom boundary

The following call is made:
CALL PHYMIN (PH,LBPH,XN,YN,'PH')

********** T F X M A X **********

Command:
TFXMAX

Purpose:
Set prescribed temparature on right hand boundary

The following call is made:
CALL PHXMAX (TF,LBFT,XN,YN,'TF')

The following subroutine is used to execute the call:

```
*****************************************************************
 P H X M A X  - Subroutine to set prescribed pressure at the      *
                the vertical boundary x=XMAX, YMIN<= y <= YMAX    *
*****************************************************************
C
      SUBROUTINE PHXMAX(PH,LBPH,XN,YN,T2)
C
      IMPLICIT REAL (A-H,O-Z)
      DIMENSION      PH(MXNP),LBPH(MXNP),XN(MXNP),YN(MXNP)
      COMMON/FECOM1/ MXNE,NE,MXNP,NP
      COMMON/PBOUND/ PHBC(6),LBPHBC(6)
      COMMON/PLOTXY/ XMIN,XMAX,YMIN,YMAX,SX,SY,XORG,YORG,DX,DY
      REAL           XMIN,XMAX,YMIN,YMAX,SX,SY,XORG,YORG,DX,DY
      CHARACTER*2 T2
C
      IPRT=1
C
      IF(IPRT.GT.1) WRITE(6,903) (XN(I),I=1,NP)
  903 FORMAT(/6X,'PHXMAX---XN:'/(6X,8F8.1))
C
      IF(IPRT.GT.1) WRITE(6,904) (YN(I),I=1,NP)
  904 FORMAT(/6X,9X,9X,   'YN:'/(6X,8F8.1))
C
      IF(IPRT.GT.0) WRITE(6,915)
  915 FORMAT(/6X,'PHXMAX---Right vertical boundary')
C
      TOL   = 1.E-5
      XMX   = XMAX-TOL
      NPMAX = 0
C
      DO 20 I=1,NP
        IF(XN(I).LT.XMX) GOTO 20
CC<<<<  PH(I)   = PHBC(2)
CC<<<<  LBPH(I) = LBPHBC(2)
C-----
        LBPH(I) = 1
C-----
        NPMAX   = NPMAX+1
        IF(IPRT.GT.0) WRITE(6,920) NPMAX,I,XN(I),T2,I,PH(I),T2,I,LBPH(I)
  920   FORMAT( 6X,'PHXMAX---NPMAX=',I3,'  XN(',I4,')=',F7.2,
     1             ' ',A2,'(',I4,')=',1P,E10.3,' LB',A2,'(',I4,')=',I1)
   20 CONTINUE
C
      WRITE(6,930) XMAX,PHBC(2),NPMAX
  930 FORMAT(/6X,'PHXMAX---XMAX=',F7.2,' PHBC(2)=',F7.2,' NPMAX=',I4)
C
      RETURN
      END
```

```
********************* T F X M I N *********************************
```

Command:

TFXMIN

Purpose:

Set prescribed temperature on left hand boundary

The following call is made:

```
    CALL   PHXMIN (TF,LBFT,XN,YN,'TF')
```

The following subroutine is used to execute the call:

```
*********************************************************************
 P H X M I N  - Subroutine to set prescribed pressure at the       *
                the vertical boundary x=XMIN, YMIN<= y <= YMAX     *
*********************************************************************
C
      SUBROUTINE PHXMIN(PH,LBPH,XN,YN,T2)
C
      IMPLICIT REAL (A-H,O-Z)
      DIMENSION      PH(MXNP),LBPH(MXNP),XN(MXNP),YN(MXNP)
      COMMON/FECOM1/ MXNE,NE,MXNP,NP
      COMMON/PBOUND/ PHBC(6),LBPHBC(6)
      COMMON/PLOTXY/ XMIN,XMAX,YMIN,YMAX,SX,SY,XORG,YORG,DX,DY
      REAL           XMIN,XMAX,YMIN,YMAX,SX,SY,XORG,YORG,DX,DY
      CHARACTER*2    T2
C
      IPRT=1
C
      IF(IPRT.GT.1) WRITE(6,903) (XN(I),I=1,NP)
  903 FORMAT(/6X,'PHXMIN---XN:'/(6X,8F8.1))
C
      IF(IPRT.GT.1) WRITE(6,904) (YN(I),I=1,NP)
  904 FORMAT(/6X,9X,9X,   'YN:'/(6X,8F8.1))
C
      IF(IPRT.GT.0) WRITE(6,905)
  905 FORMAT(/6X,'PHXMIN---Left vertical boundary')
C
      TOL   = 1.E-5
      XMN   = XMIN+TOL
      NPMIN = 0
C
      DO 20 I=1,NP
        IF(XN(I).GT.XMN) GOTO 20
CC<<<<  PH(I)   = PHBC(1)
CC<<<<  LBPH(I) = LBPHBC(1)
C-----
        LBPH(I) = 1
C-----
        NPMIN   = NPMIN+1
        IF(IPRT.GT.0) WRITE(6,910) NPMIN,I,XN(I),T2,I,PH(I),T2,I,LBPH(I)
  910   FORMAT( 6X,'PHXMIN---NPMIN=',I3,'  XN(',I4,')=',F7.2,
     1             ' ',A2,'(',I4,')=',1P,E10.3,' LB',A2,'(',I4,')=',I1)
   20 CONTINUE
C
      WRITE(6,930) XMIN,PHBC(1),NPMIN
  930 FORMAT(/6X,'PHXMIN---XMIN=',F7.2,' PHBC(1)=',F7.2,' NPMIN=',I4)
C
      RETURN
      END
```

********** T F Y M A X **********

Command:

```
  TFYMAX
```

```
      Purpose:
        Set prescribed tempareture on top hand boundary

      The following call is made:
          CALL   PHYMAX (TF,LBFT,XN,YN,'TF')

      ********************************************************************
       P H Y M A X  - Subroutine to set prescribed pressure at then      *
                      horizontal boundary  XMIN<= x <= XMAX, y=YMAX      *
      ********************************************************************
C
       SUBROUTINE PHYMAX(PH,LBPH,XN,YN,T2)
C
       IMPLICIT REAL (A-H,O-Z)
       DIMENSION      PH(MXNP),LBPH(MXNP),XN(MXNP),YN(MXNP)
       COMMON/FECOM1/ MXNE,NE,MXNP,NP
       COMMON/PBOUND/ PHBC(6),LBPHBC(6)
       COMMON/PLOTXY/ XMIN,XMAX,YMIN,YMAX,SX,SY,XORG,YORG,DX,DY
       REAL           XMIN,XMAX,YMIN,YMAX,SX,SY,XORG,YORG,DX,DY
       CHARACTER*2    T2
C
       IPRT=1
C
       IF(IPRT.GT.1) WRITE(6,903) (XN(I),I=1,NP)
  903  FORMAT(/6X,'PHYMAX---XN:'/(6X,8F8.1))
C
       IF(IPRT.GT.1) WRITE(6,904) (YN(I),I=1,NP)
  904  FORMAT(/6X,9X,9X,   'YN:'/(6X,8F8.1))
C
       IF(IPRT.GT.0) WRITE(6,915)
  915  FORMAT(/6X,'PHYMAX---Upper horizontal boundary')
C
       TOL   = 1.E-5
       YMX   = YMAX-TOL
       NPMAX = 0
C
       DO 20 I=1,NP
         IF(YN(I).LT.YMX) GOTO 20
CC<<<<   PH(I)   = PHBC(4)
CC<<<<   LBPH(I) = LBPHBC(4)
C-----
         LBPH(I) = 1
C-----
         NPMAX   = NPMAX+1
         IF(IPRT.GT.0) WRITE(6,920) NPMAX,I,YN(I),T2,I,PH(I),T2,I,LBPH(I)
  920    FORMAT( 6X,'PHYMAX---NPMAX=',I3,'  YN(',I4,')=',F7.2,
     1             ' ',A2,'(',I4,')=',1P,E10.3,' LB',A2,'(',I4,')=',I1)
   20 CONTINUE
C
       WRITE(6,930) YMAX,PHBC(4),NPMAX
  930  FORMAT(/6X,'PHYMAX---YMAX=',F7.2,' PHBC(4)=',F7.2,' NPMAX=',I4)
C
       RETURN
       END
      ------------------------------------------------------------------

      ****************** T F Y M I N ***********************************

      Command:
        TFYMIN
```

```
      Purpose:
        Set prescribed temperature on bottom boundary

      The following call is made:
          CALL   PHYMIN (TF,LBFT,XN,YN,'TF')

      ***********************************************************************
       P H Y M I N  - Subroutine to set prescribed pressure at the       *
                      horizontal boundary XMIN<= x <= XMAX, y=YMIN       *
      ***********************************************************************
C
       SUBROUTINE PHYMIN(PH,LBPH,XN,YN,T2)
C
       IMPLICIT REAL (A-H,O-Z)
       DIMENSION      PH(MXNP),LBPH(MXNP),XN(MXNP),YN(MXNP)
       COMMON/FECOM1/ MXNE,NE,MXNP,NP
       COMMON/PBOUND/ PHBC(6),LBPHBC(6)
       COMMON/PLOTXY/ XMIN,XMAX,YMIN,YMAX,SX,SY,XORG,YORG,DX,DY
       REAL           XMIN,XMAX,YMIN,YMAX,SX,SY,XORG,YORG,DX,DY
       CHARACTER*2    T2
C
       IPRT=1
C
       IF(IPRT.GT.1) WRITE(6,903) (XN(I),I=1,NP)
  903  FORMAT(/6X,'PHYMIN---XN:'/(6X,8F8.1))
C
       IF(IPRT.GT.1) WRITE(6,904) (YN(I),I=1,NP)
  904  FORMAT(/6X,9X,9X,   'YN:'/(6X,8F8.1))
C
       IF(IPRT.GT.0) WRITE(6,905)
  905  FORMAT(/6X,'PHYMIN---Lower horizontal boundary')
C
       TOL   = 1.E-5
       YMN   = YMIN+TOL
       NPMIN = 0
C
       DO 20 I=1,NP
         IF(YN(I).GT.YMN) GOTO 20
CC<<<<   PH(I)   = PHBC(3)
CC<<<<   LBPH(I) = LBPHBC(3)
C-----
         LBPH(I) = 1
C-----
         NPMIN   = NPMIN+1
         IF(IPRT.GT.0) WRITE(6,910) NPMIN,I,YN(I),T2,I,PH(I),T2,I,LBPH(I)
  910    FORMAT( 6X,'PHYMIN---NPMIN=',I3,'  YN(',I4,')=',F7.2,
     1             ' ',A2,'(',I4,')=',1P,E10.3,' LB',A2,'(',I4,')=',I1)
   20 CONTINUE
C
       WRITE(6,930) YMIN,PHBC(3),NPMIN
  930  FORMAT(/6X,'PHYMIN---YMIN=',F7.2,' PHBC(3)=',F7.2,' NPMIN=',I4)
C
       RETURN
       END
      ----------------------------------------------------------------------

      ******************** S E L N D S ********************************

      Command:
        SELNDS  IU JU Optional text
```

Purpose:
To select node numbers for printout
etc.

The following call is made:
CALL SELNDS(INUN(2))

Data set 18----------Input node numbers to be printed out including pressure, fluid velocities, temperature and the strength of a distributed heat source (subroutine *SELNDS*)

Cols	Format	Identifier	Explanation
1-5	I5	IN16(1)	Node number
6-10	I5	IN16(2)	Node number
.	.	.	.
.	.	.	.
.	.	.	.
76-80	I5	IN16(16)	Node number

Note: The last record in this data set must contain -1 in columns 1 to 2.

********** S F R I N 1 **********

Command:
SFRIN1 IU 00 <Optional text>

Purpose:
Set boundary and initial conditions for 1-D gas flow test

The following call is made:
CALL SFRIN1 (NODES,PH,LBPH,TR,LBTR,XN,YN,ZN,INUN6,0)

********** S H W T I N **********

Command:
SHWTIN

Purpose:
Input the time dependent energy output for hydrocoin LEVEL2,CASE1

The following call is made:
CALL SHWTIN (INUNX)

This subroutine reads:
I. Optional text card
II. Number of heat source time break points to be read in by the program.
III. The time (days) and the energy output (kW) for each time break point. The data values are stored in the following common block:

where TPOWER(150) is the array for the time break points and
QPOWER(150) is the energy output to be applied at the corresponding time break point.
NTPOW is the number of data pairs to be read in.

Note: The heat source values being input will only be applied to the heat flow equation is function DECAY3 is specified in the input for function selection.

3.3 DESCRIPTION OF AUXILIARY FILES

Specification of disc storage used for the element matrices

The computer model presented has been developed on an AMDAHL 470/7A computer being compatible with IBM computers. This means that the job control language (JCL) referred to in the sequel is not generally applicable. In fact it may not have to considered at all when running the model program on other computers such as Prime, Cray, etc. since these computers have more tractable methods for handling permanent as well as temporary datasets. Thus in addition to the Amdahl computer as mentioned above the model program has been run on a Prime 9950 computer and a Cray1 computer.

The datasets that have to be specified in one way or the other are

Input:

- INUN(1) for parameter data
- INUN(3) if a previous solution is to be restarted and continued
- INUN(8) for input of the element grid (coordinates of nodal points and element incidences)

Output:

- INUN(4) Echo file for input of parameter data and element grid data
- INUN(5) Output file for solution values

These output files are only needed for post-processing.

Temporary datasets:

- INUN(8) to hold the frontal equations

This is in most cases the only temporary dataset that the user may have to consider is INUN(8) which is used to hold the frontal equations for problems that are too large to allow for incore solution of the matrix problem.

Notation used for dataset definitions

- NE - Number of elements
- NP - Number of nodal points
- NEK - Number of variables in each element
- IDB - Control parameter (=1 for single precision, =2 double precision)
- NW - Number of words per record
- NRBL - Number of records per block
- NRED - Number of physical records for each logical record

3.3.1 Description of Input/Output data sets

Variable	Default	Description
INUN(1)	5	Input file. Parameter values (subr. * HFINPT*), nodal data (subr. *HFNDIN*) and element incidences (subr. *HFELIN*)

INUN(2)	5	Input file. Heat sources, concentrated or distributed (subr. *HFSRC1*), boundary and initial conditions (subr. *HFBCX1*), and nodes selected for printout (subr. *SELNDS*)
INUM(3)	20	Input file. Solution values from the preceding execution to restart the previous job. This data file is usually created by the use of subroutine *HFSAVD*. The input data are read into the model by subroutine *HFNDUV*.
INUN(4)	21	Output data set used for storage of the parameter values, nodal data and element incidences. The parameter data are read by subroutine *HFINPT*, the nodal data by *HFNDIN* and the element incidences by subroutine *HFELIN*. Alternative entries to the latter two subroutines, i.e. *HFNDUT* and *HFELUT*, are used to output the nodal data and the element incidences. RECFM = FB LRECL = 80 BLKSIZE = LRECL*NRBL SPACE = (BLKSIZE,((NP+NE+25)/NRBL,5))
INUN(5)	22	Output data set for the storage of the solution values. The solution values are written to the present data set using subroutine *HFSAVD*. This data file may the be used for various post-processing or to restart the model execution. RECFM = VSB LRECL = NP*4*IDB/NRED + 4 BLKSIZE = LRECL+4 SPACE = (BLKSIZE,(3*NRED,1))
INUN(6)	55	Input data set used for unsaturated data. RECFM = FB LRECL = 80 BLKSIZE = LRECL*NRBL SPACE = (BLKSIZE,((NP+NE+25)/NRBL,5))
INUN(7)	56	Input data set used for stress data. RECFM = FB LRECL = 80 BLKSIZE = LRECL*NRBL SPACE = (BLKSIZE,((NP+NE+25)/NRBL,5))
INUN(8)	5	Input data set used for element mesh data. RECFM = FB LRECL = 80 BLKSIZE = LRECL*NRBL SPACE = (BLKSIZE,((NP+NE+25)/NRBL,5))

3.3.2 Description of intermediate data sets

The logical unit numbers for the temporary datasets are stored in the following common block:

COMMON/UNITS / IUNIT(10)

Variable	Default	Description
IUNIT(1)	11	Element matrices for the pressure equation RECFM = VSB LRECL = 4+4 + NEK*NEK*IDB +4 BLKSIZE = LRECL*NRBL + 4

		SPACE = (BLKSIZE,(NE/NRBL,5))
IUNIT(2)	12	Right hand sides to the element matrices for the pressure equation RECFM = VSB LRECL = NEK*4*IDB + 4 BLKSIZE = LRECL*NRBL + 4 SPACE = (BLKSIZE,(NE/NRBL,2))
IUNIT(3)	13	Element matrices for the temperature equations RECFM = VSB LRECL = 4+4 + NEK*NEK*IDB +4 BLKSIZE = LRECL*NRBL + 4 SPACE = (BLKSIZE,(NE/NRBL,5))
IUNIT(4)	14	Right hand sides to the element matrices for the temperature equations RECFM = VSB LRECL = NEK*4*IDB + 4 BLKSIZE = LRECL*NRBL + 4 SPACE = (BLKSIZE,(NE/NRBL,2))
Variable	Default	Description
IUNIT(5)	15	Basic integration matrices RECFM = VSB LRECL = 4+4 + NEK*NEK*IDB +4 BLKSIZE = LRECL*NRBL + 4 SPACE = (BLKSIZE,(NE/NRBL,5))
IUNIT(6)	16	Basic element matrices RECFM = VSB LRECL = (4+4 + NEK*NEK*IDB +4 BLKSIZE = LRECL*NRBL + 4 SPACE = (BLKSIZE,(NE/NRBL,5))
IUNIT(8)	03	Frontal equations RECFM = VSB LRECL = MXFRON*4*IDB + 4*IDB + 4+4+4 BLKSIZE = LRECL*NRBL + 4 SPACE = (BLKSIZE,(NP/NRBL,5))
IUNIT(9)	04	Right hand sides to frontal equations RECFM = VSB LRECL = 4*IDB + 4 BLKSIZE = LRECL*NRBL + 4 SPACE = (BLKSIZE,(NP/NRBL,5))

The above given default values of the logical unit numbers of the previous datasets are currently set in a BLOCK DATA attached to the MAIN program in the program listings.

3.3.3 Specification of data sets used by the frontal solvers

The logical unit numbers of the datasets connected with the frontal solvers are stored in the following common block:

COMMON/IUFRON/ IU1,IU2,IU3,IU4

Variable	Default	Description
IU1	01	Data set containing the element matrices
IU2	02	Data set for the right hand side vectors to the element matrices
IU3	03	Data set for intermediate storage of the fron-

		tal equations
IU4	04	Data set for intermediate storage of the right hand sides of the frontal equations

The above given default values of the logical unit numbers of the datasets are currently set in a BLOCK DATA attached to subroutine * HFRON2 * in the program listings.

Note: If the logical unit numbers in the common block /UNITS/ differ from the ones in the common block /IUFRON/ the former ones will override the latter ones.

3.3.4 Examples of DCB specification

Example of DCB specification for temporary data sets

```
                  NE = 592, NP = 2901, NEK = 20, MXFRON = 226

11  DCB=(RECFM=VSB,LRECL=3212,BLKSIZE=9640,       NRBL  =   3
    SPACE=(9640,(198,5))
12  DCB=(RECFM=VSB,LRECL=164,BLKSIZE=4104,        NBRL  =  25
    SPACE=(4104,(24,2))
03  DCB=(RECFM=VSB,LRECL,1828,BLKSIZE=9144,       NRBL  =   5
    SPACE=(9144,(581,2))
04  DCB=(RECFM=VSB,LRECL=12,BLKSIZE=2404,         NRBL  = 200
    SPACE=(2404,(15,1))
```

Example of DCB for the output data sets

```
                                                  NP    = 2901

21  DCB=(RECFM=FB,LRECL=80,BLKSIZE=2000,          NRBL  =  25
    SPACE=(2000,(141,5))
22  DCB=(RECFM=VSB,LRECL=7740,BLKSIZE=7740,       NRED  =   3
    SPACE=(7740,(9,1))
```

3.3.5 Block data and common

Block data attached to MAIN

```
     COMMON/IUNITS/ IUNIT(10)
     COMMON/HFWIUN/ IUNIT1,IUNIT2
     DATA           IUNIT/11,12,13,14,15,16, 4*0/
     DATA           IUNIT1,IUNIT2/31,32/
```

Block data attached to subroutines *PHSLVF* and *TMSLVF*

```
     COMMON/IUNITS/ IUNIT(10)
     COMMON/IUFRON/ IU1,IU2,IU3,IU4
     DATA           IU1,IU2,IU3,IU4/01,02,03,04/
     IU1 = IUNIT(1)
     IU2 = IUNIT(2)
```

In subroutine *HFELFX*

```
     COMMON/IUNITS/ IUNIT(10)
     IUN1 = IUNIT(1)
     IUN2 = IUNIT(2)
```

In subroutine *HFELT1*

```
      COMMON/IUNITS/ IUNIT(10)
      IUN3 = IUNIT(3)
      IUN4 = IUNIT(4)
      IUN5 = IUNIT(5)
      IUN6 = IUNIT(6)
```

Examples of JCL and input data setups

In the following examples the model has been run in batch mode in three steps: compile, link and go. Below follows a typical JCL set-up, in which part of the model program is input and compiled as a source module and part of the model program is invoked from a disc library, of course implying that these subroutines have been compiled and stored on this disc library prior to the execution.

3.3.6 Restart

A previous solution is continued by specifying the number of the data file where the solution values have been stored. The program reads this data file until the end and uses the data from the last time step as initial values for the continued solution. (see 3.2 Input data preperation) It is also possible to perform the restart at another time step than the final one by specifying a restart time step (NREADT)

Input of solution values from a previous execution (subr. *HFNDUV*). These data are unformatted.

NSTEP	Time step number.
TIME	Time.
DT	Time increment.
ICPSLV(10)	Control parameters to indicate the kind of data that have been stored on the file and that may be input. ICPSLV(1)=1 : pressure values to be input ICPSLV(2)=1 : fluid temperature values to be input ICPSLV(3)=1 : rock temperature values to be input ICPSLV(4)=1 : rock displacemnts to be input
IOPHCX	Control parameter to indicate if nodal permeabilities have been stored. IOPHCX=1 : nodal permeabilities are to be input
NPX	Number of nodal values.
NDIMX	Number of spatial dimensions.

****

```
      H F N D U V - SUBROUTINE FOR UNFORMATTED INPUT OF NODAL VALUES
                    THIS SUBROUTINE READS DATA UNITL THE END OF THE
                    INPUT FILE IS REACHED
      ****************************************************************
      NREADT        - An integer to indicate how many time steps that
                      should be read by this subroutine. If a value
                      of zero is set for this parameter the file will
                      be read until the end
      ****************************************************************
      SUBROUTINE HFNDUV(PH,TF,TR,DU,PERM,NSTEP,TIME,DT,IUN)
      IMPLICIT REAL (A-H,O-Z)
      DIMENSION       PH(MXNP),TF(MXNP),TR(MXNP),DU(MXNP,3),PERM(MXNP)
      COMMON/EQSOLV/ ICPSLV(10)
      COMMON/FECOM1/ MXNE,NE,MXNP,NP
      COMMON/IOPHC / IOPHC
      COMMON/MATMD1/ IMATM,IPORF,IPERF
      COMMON/NREADT/ NREADT
      IF(IUN.LT.1) THEN
        WRITE(6,960) IUN
  960   FORMAT(/6X,'HFNDUV---No input data read by this subroutine',
     1                    '---IUN=',I3)
        WRITE(6,999) NSTEP,TIME,DT
  999   FORMAT(/6X,'HFNDUV---NSTEP=',I3,
     1                     ' TIME=',1P,E15.7,' DT=',E15.7)
        RETURN
      ENDIF
      REWIND IUN
      IEND = 0
   20 CALL HFNDUX(PH,TF,TR,DU,PERM,NSTEP,TIME,DT,IUN,IEND,*40)
CC<<<<WRITE(6,920) NSTEP,TIME,DT,ICPSLV,IOPHC,NP,NDIM
  920 FORMAT( 6X,'HFNDUV---NSTEP=',I3,' TIME=',1P,E10.3,
     1                                 ' DT=',1P,E10.3/
     2         6X,9X,'ICPSLV : ',10I2,' IOPHC=',I2/
     3         6X,9X,'NP=',I4,'  NDIM=',I1)
C---CHECK IF THE READING HAS COME TO THE LAST TIME STEP (NREADT)
      IF(NSTEP.EQ.NREADT .AND. NREADT.GT.0) GOTO 60
      IF(IEND.EQ.0) GOTO 20
   40 WRITE(6,940) IUN
  940 FORMAT(/6X,'HFNDUV---Now read nodal values from unit:',I3)
      RETURN
   60 WRITE(6,970) NREADT,IUN
  970 FORMAT(/6X,'HFNDUV---Now read',I3,' (NREADT) time steps',
     1                  ' on file IUN=',I3)
      NP1 = 000
      NP2 = 000
      WRITE(6,980) (I,PH(I),I=NP1,NP2)
  980 FORMAT(/6X,'HFNDUV---PH:'/(2X,1P,5(I5,E10.3)))
      RETURN
      END
```

```
      ****************************************************************
      H F N D U X  - SUBROUTINE FOR UNFORMATTED INPUT OF NODAL VALUES
                     THIS SUBROUTINE READS DATA FOR ONE TIME STEP
                     AT EACH CALL UPON THE SUBROUTINE
      ****************************************************************
      IEND           - An integer which on output is set equal to one
                       if the input file is read to end of file
      ****************************************************************
      SUBROUTINE HFNDUX(PH,TF,TR,DU,PERM,NSTEP,TIME,DT,IUN,IEND,*)
      IMPLICIT REAL (A-H,O-Z)
      DIMENSION      PH(MXNP),TF(MXNP),TR(MXNP),DU(MXNP,3),PERM(MXNP)
      COMMON/EQSOLV/ ICPSLV(10)
      COMMON/FECOM1/ MXNE,NE,MXNP,NP
      COMMON/FECOM4/ NNODE,NDOF,NEK,NDIM
      COMMON/IOPHC / IOPHC
      COMMON/MATMD1/ IMATM,IPORF,IPERF
      IF(IUN.LT.1) GOTO 30
      IF(IEND.NE.0) GOTO 10
      READ(IUN,END=10) NSTEP,TIME,DT,ICPSLV,IOPHCX,NPX,NDIMX
      WRITE(6,920) NSTEP,TIME,DT,IOPHCX,NPX,NDIMX
  920 FORMAT( 6X,'HFNDUX---NSTEP=',I3,' TIME=',1P,E9.3,
     1                                 ' DT=',1P,E9.3,
     2             ' IOPHC=',I1,' NP=',I4,' NDIM=',I1)
      IF(IOPHCX.NE.IOPHC) THEN
        WRITE(6,930) IOPHCX,IOPHC
  930   FORMAT(/6X,'HFNDUX---IOPHCX=',I2,' .NE. IOPHC=',I2/
     1          6X,9X,'IOPHC=IOPHCX')
        IOPHC=IOPHCX
      ENDIF
      CALL HFNDU1(PH,TF,TR,DU,PERM,IUN,IEND)
      IF(IEND.EQ.0) GOTO 20
   10 IEND=1
      WRITE(6,935) NSTEP,TIME,DT,ICPSLV,IOPHCX,NPX,NDIMX
  935 FORMAT(6X,'HFNDUX',66('-'),
     1      /6X,9X,      'NSTEP=',I3,' TIME=',1P,E10.3,
     2                           ' DT=',1P,E10.3/
     3       6X,9X,'ICPSLV : ',10I2/
     4       6X,9X,'IOPHCX=',I2,'  NPX=',I4,'  NDIMX=',I1)
      WRITE(6,940) IUN,IEND
  940 FORMAT(/6X,'HFNDUX---Now read to End of File:',I3,' IEND=',I2/
     1        6X,'HFNDUX',66('-'))
      RETURN1
   20 CONTINUE
CC<<<<WRITE(6,960) IUN
  960 FORMAT( 6X,'HFNDUX---Now read nodal values on file:',I3)
      RETURN
```

```
   30 IEND=9
      WRITE(6,970) IUN,IEND
  970 FORMAT(/6X,'HFNDUX---No data have been input by this ',
     1                'subroutine --- IUN=',I2,' IEND=',I1)

      RETURN
      END
      ******************************************************************

      H F N D U 1  - SUBROUTINE FOR UNFORMATTED INPUT/OUTPUT OF NODAL
                     DATA

      ******************************************************************

      SUBROUTINE HFNDU1(PH,TF,TR,DU,PERM,IUN,IEND)

      IMPLICIT REAL (A-H,O-Z)
      DIMENSION      PH(NP),TF(NP),TR(NP),DU(MXNP,3),PERM(NP)

      COMMON/EQSOLV/ ICPSLV(10)
      COMMON/FECOM1/ MXNE,NE,MXNP,NP
      COMMON/FECOM4/ NNODE,NDOF,NEK,NDIM
      COMMON/IOPHC / IOPHC
      COMMON/MATMD1/ IMATM,IPORF,IPERF
      COMMON/MISCHK/ MISCHK
C-----
      IF(MISCHK.NE.0) WRITE(6,901) MXNE,NE,MXNP,NP
  901 FORMAT(6X,'HFNDU1---MXNE=',I4,' NE=',I4,' MXNP=',I4,' NP=',I4)
C-----

      IF(IEND.NE.0) GOTO 80
      IF(IUN.EQ.0) GOTO 60

      IF(ICPSLV(1).GT.0) READ(IUN,END=60) PH
      IF(ICPSLV(2).GT.0) READ(IUN,END=60) TF
      IF(ICPSLV(3).GT.0) READ(IUN,END=60) TR

      IF(ICPSLV(4).GT.0) THEN
        WRITE(6,902) NDIM,NP
  902   FORMAT(/6X,'HFNDU1---NDIM=',I1,' NP=',I5)
        DO 10 K=1,NDIM
        READ(IUN) (DU(J,K),J=1,NP)
CC<<<<  WRITE(6,905) (DU(J,K),J=1,NP)
  905   FORMAT(6X,1P,6E12.3)
   10   CONTINUE
      ENDIF

      IF(IOPHC.EQ.1) READ(IUN) PERM
      IF(IOPHC.EQ.1) WRITE(6,906) (PERM(J),J=1,10)
  906 FORMAT(/6X,'HFNDU1---PERM:'/
     1        (6X,1P,6E10.3))
      RETURN
   60 IEND=1
   80 WRITE(6,910) IEND,IUN
  910 FORMAT(/6X,'HFNDU1---IEND=',I2,' IUN=',I2)
      RETURN
C-----------------------------------------------------------------------
      H F N D U 2  - ALTERNATIVE ENTRY TO OUTPUT NODAL DATA TO A
                     DISC FILE
C-----------------------------------------------------------------------
      ENTRY       HFNDU2(PH,TF,TR,DU,PERM,IUN)
      IF(ICPSLV(1).GT.0) WRITE(IUN) PH
```

```
      IF(ICPSLV(2).GT.0) WRITE(IUN) TF
      IF(ICPSLV(3).GT.0) WRITE(IUN) TR
      IF(ICPSLV(4).GT.0) THEN
        DO 110 K=1,NDIM
  110   WRITE(IUN) (DU(J,K),J=1,NP)
      ENDIF

      IF(IOPHC.EQ.1) WRITE (IUN) PERM

      RETURN
      END

      ******************************************************************
      H F S A V D  - SUBROUTINE TO STORE RESULTS ON A FILE
      ******************************************************************

      SUBROUTINE HFSAVD(PH,TF,TR,DU,PERM,NSTEP,TIME,DT,IUN)

      IMPLICIT REAL (A-H,O-Z)
      DIMENSION      PH(MXNP),TF(MXNP),TR(MXNP),DU(MXNP,3),PERM(MXNP)

      COMMON/EQSOLV/ ICPSLV(10)
      COMMON/FECOM1/ MXNE,NE,MXNP,NP
      COMMON/FECOM4/ NNODE,NDOF,NEK,NDIM
      COMMON/IOPHC / IOPHC
      COMMON/MATMD1/ IMATM,IPORF,IPERF

      IF(IUN.EQ.0) GOTO 40

      WRITE(IUN) NSTEP,TIME,DT,ICPSLV,IOPHC,NP,NDIM

      CALL HFNDU2(PH,TF,TR,DU,PERM,IUN)

      WRITE(6,910) NSTEP,IUN,TIME,DT,NP,NDIM
  910 FORMAT( 6X,'HFSAVD---Now written results of time step:',I3,
     1                                          ' on unit:',I3/
     2        6X,9X,'TIME=',1P,E15.7,' DT=',1P,E15.7,
     3                      '  NP=',I4,'  NDIM=',I1)

      WRITE(6,911) ICPSLV,IOPHC
  911 FORMAT( 6X,9X,'ICPSLV=',10I2,' IOPHC=',I2)
      RETURN
   40 WRITE(6,950) IUN
  950 FORMAT(/6X,'HFSAVD---IUN=',I2,' No data were stored')
      RETURN
      END
```

REFERENCES

References to studies where GWHRT model has been used:

Groundwater and heat flow

1. Thunvik,R. and Braester, C., 1980, Hydrothermal conditions around a radioactive waste repository, Part 1 - A mathematical model for the flow of groundwater and heat in fracture rock, part 2 - Numerical solutions, Part 3 - Numerical solutions for anisotropy, SKBF-KBS-TR:80-19.

2. Thunvik, R., Braester, C., 1982.
Hydrothermal Conditions around a Radioactive Waste Repository, Proc. of the Fifth International Symposium on Radioactive Waste Management, Berlin (West), June 1982.

3. Thunvik, R., Braester, C., 1984.
Radioactive Waste Repositories in Hard Rock Aquifers - Hydrodynamic Aspects, Nuclear Technology, November 1984, pp. 255-267.

Unsaturated flow

4. Thunvik, Roger, 1984, Calculations of fluxes through a repository caused by a local well, SKBF.KBS-TR:83-50.

Gas migration

6. Thunvik, R., Braester, C., 1986,
"Calculation of gas migration in fractured rock", SFR-progress report 86-04.

7. Thunvik, R., Braester, C., 1987,
"Calculation of gas migration in fractured rock",
SKB-Technical report 87-18, Swedish Nuclear Fuel and Waste Management Company.

8. Braester, C, Thunvik, R. 1987,
"Gas migration from low level radioactive waste repositories" (Accepted for publication in Nuclear Technology).

Stochastic analysis

9. Andersson, J., Thunvik, R., 1986,
"Prediction of mass transport in discrete fracture networks with the aid of geometrical data", Water Resources Research Vol. 22, No 13, Pages 1941-1950, December 1986.

Hydrocoin

1. Thunvik, Roger, 1987, "Calculations on Hydrocoin level 1 using the GWHRT flow model
Case 1 - Transient flow of water from a borehole penetrating a confined aquifer

Case 3 - Saturated-unsaturated flow through a layered sequence of sedimentary rocks
Case 4 - Transient thermal convection in a saturated medium
SKB Technical Report 87-03,
Swedish Nuclear Fuel and Waste Management Co,
Stockholm, Sweden.

2. Thunvik, Roger, 1987, "Calculations on Hydrocoin level 2, case 1 using the GWHRT flow model - Thermal convection and conduction around a field heat transfer experiment", SKB Technical Report 87-04, Swedish Nuclear Fuel and Waste Management Co, Stockholm, Sweden.

Specification of level 1, case 1

1. Hodginson, D. and Barker, J., 1985, Specification of a test problem for HYDROCOIN Level 1 Case 1: Transient flow from a borehole in a fractured permeable medium, AERE - R 11574, HARWELL.

Specification of level 1, case 3

1. Grundfeldt, Bertil, 1984, Proposal for a test problem for Hydrocoin Level 1, Case 3, saturated-unsaturated flow through a layered sequence of sedimentary rocks, Kemakta Consultants Co.

Specification of level 1, case 4

1. Hodginson, D., 1985, Specification of a test problem for Hydrocoin Level 1 Case 4: Transient thermal convection in a saturated permeable medium, AERE -R - 11566, DOE/RW/84.198.

Specification of level 2, case 1

2. Hodginson, D. and Herbert Alan, 1984, Specification of a test problem for Hydrocoin Level 2 Case 1: Thermal convection and conduction around a field heat transfer experiment, AERE -R - 11627, DOE/RW/84, January 1985.

1. Hodginson, D. and Herbert Alan, 1985, Specification of a test problem for Hydrocoin Level 2 Case 1: Thermal convection and conduction around a field heat transfer experiment, AERE -R - 11627, DOE/RW/85.028.

List of SKB reports

Annual Reports

1977–78
TR 121
KBS Technical Reports 1 – 120.
Summaries. Stockholm, May 1979.

1979
TR 79–28
The KBS Annual Report 1979.
KBS Technical Reports 79-01 – 79-27.
Summaries. Stockholm, March 1980.

1980
TR 80–26
The KBS Annual Report 1980.
KBS Technical Reports 80-01 – 80-25.
Summaries. Stockholm, March 1981.

1981
TR 81–17
The KBS Annual Report 1981.
KBS Technical Reports 81-01 – 81-16.
Summaries. Stockholm, April 1982.

1982
TR 82–28
The KBS Annual Report 1982.
KBS Technical Reports 82-01 – 82-27.
Summaries. Stockholm, July 1983.

1983
TR 83–77
The KBS Annual Report 1983.
KBS Technical Reports 83-01 – 83-76
Summaries. Stockholm, June 1984.

1984
TR 85–01
Annual Research and Development Report 1984
Including Summaries of Technical Reports Issued during 1984. (Technical Reports 84-01–84-19)
Stockholm June 1985.

1985
TR 85-20
Annual Research and Development Report 1985
Including Summaries of Technical Reports Issued during 1985. (Technical Reports 85-01-85-19)
Stockholm May 1986.

1986
TR 86-31
SKB Annual Report 1986
Including Summaries of Technical Reports Issued during 1986
Stockholm, May 1987

1987
TR 87-33
SKB Annual Report 1987
Including Summaries of Technical Reports Issued during 1987
Stockholm, May 1988

Technical Reports

1988
TR 88-01
Preliminary investigations of deep ground water microbiology in Swedish granitic rocks
Karsten Pedersen
University of Göteborg
December 1987

TR 88-02
Migration of the fission products strontium, technetium, iodine, cesium and the actinides neptunium, plutonium, americium in granitic rock
Thomas Ittner[1], Börje Torstenfelt[1], Bert Allard[2]
[1]Chalmers University of Technology
[2]University of Linköping
January 1988

TR 88-03
Flow and solute transport in a single fracture. A two-dimensional statistical model
Luis Moreno[1], Yvonne Tsang[2], Chin Fu Tsang[2], Ivars Neretnieks[1]
[1]Royal Institute of Technology, Stockholm, Sweden
[2]Lawrence Berkeley Laboratory, Berkeley, CA, USA
January 1988

TR 88-04
Ion binding by humic and fulvic acids: A computational procedure based on functional site heterogeneity and the physical chemistry of polyelectrolyte solutions
J A Marinsky, M M Reddy, J Ephraim, A Mathuthu
US Geological Survey, Lakewood, CA, USA
Linköping University, Linköping
State University of New York at Buffalo, Buffalo, NY, USA
April 1987

TR 88-05
Description of geophysical data on the SKB database GEOTAB
Stefan Sehlstedt
Swedish Geological Co, Luleå
February 1988

TR 88-06
Description of geological data in SKBs database GEOTAB
Tomas Stark
Swedish Geological Co, Luleå
April 1988

TR 88-07
Tectonic studies in the Lansjärv region
Herbert Henkel
Swedish Geological Survey, Uppsala
October 1987

TR 88-08
Diffusion in the matrix of granitic rock. Field test in the Stripa mine. Final report.
Lars Birgersson, Ivars Neretnieks
Royal Institute of Technology, Stockholm
April 1988

TR 88-09
The kinetics of pitting corrosion of carbon steel. Progress report to June 1987
G P Marsh, K J Taylor, Z Sooi
Materials Development Division
Harwell Laboratory
February 1988