In Situ Bioremediation: A Network Model of Diffusion and Flow in Granular Porous Media

S. K. Griffiths, R. H. Nilson, R. W. Bradshaw


DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.
DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.
IN SITU BIOREMEDIATION: A NETWORK MODEL OF DIFFUSION AND FLOW IN GRANULAR POROUS MEDIA

S. K. Griffiths, R. H. Nilson and R. W. Bradshaw
Sandia National Laboratories
Livermore, California 94551-0969

In situ bioremediation is a potentially expedient, permanent and cost-effective means of waste site decontamination. However, permeability reductions due to the transport and deposition of native fines or due to excessive microorganism populations may severely inhibit the injection of supplemental oxygen in the contamination zone. To help understand this phenomenon, we have developed a micro-mechanical network model of flow, diffusion and particle transport in granular porous materials. The model differs from most similar models in that the network is defined by particle positions in a numerically-generated particle array. The model is thus widely applicable to computing effective transport properties for both ordered and realistic random porous media. A laboratory-scale apparatus to measure permeability reductions has also been designed, built and tested.

INTRODUCTION

In situ bioremediation is a promising new technology for groundwater and soils decontamination, offering many potential benefits over both excavation and flushing [1,2,3]. Rather than extracting contaminants for subsequent treatment or permanent burial, this technique relies on microbial metabolism to decompose contaminants in place, yielding biomass and harmless byproducts [4,5]. Nearly all organic contaminants can be metabolized in this manner, including the recalcitrant pesticides, PCBs, and chlorinated dioxins, as well as the many common hydrocarbons [6,7,8].
Current estimates for the costs of waste site bioremediation range from $50-$80 per ton, as compared with $200 per ton or more for landfill disposal, and $250-$600 per ton for incineration [9]. Despite the apparent economic benefits, however, bioremediation is often not the preferred method for site restoration. The reason for this is in part because the technology is relatively new and in part because reliable predictive capabilities do not yet exist. Site managers cannot be assured a priori of the cost, duration and efficacy of bioremediation and so often opt for more direct though more costly alternatives. Improved understanding of bioremediation processes and improved capabilities for accurately predicting the duration and extent of cleanup would significantly increase the utility of this technology.

Successful application of bioremediation usually depends on injecting air, oxygen, oxygenated water or hydrogen peroxide to provide supplemental oxygen within the contamination zone [4,7]. Only in rare instances is bioremediation based on anaerobic decomposition [10,11]. Effective oxygenation of the contamination zone requires that soil permeabilities remain relatively high during the course of treatment. The treatment processes, however, may induce dramatic reductions in formation permeabilities. This phenomenon, known as formation blocking, usually results from one of three mechanisms: pore bubble nucleation due to the accumulation of gaseous decomposition products; pore plugging due to the long range transport and concentration of native fines; or pore plugging by the biomass resulting from excessive microorganism populations [12,13]. The two latter phenomena are the topic of our current study.

State-of-the-art modeling of bioremediation processes is generally based on three-dimensional, multicomponent, multiphase codes employing well established relative permeability transport algorithms [14,15]. These codes have been developed over nearly two decades to solve the somewhat simpler problem of subsurface contaminant transport [16]. The primary shortcoming of these codes in modeling bioremediation is that they neglect micro-scale phenomena in order to address the multidimensional macro-scale processes of the entire waste site [17,18]. Convective transport of particles and microorganisms is generally neglected, as is the effect of biomass on formation permeability. Similarly, most laboratory studies of biodegradation and bioremediation have concentrated on identifying specialized microorganisms and chemical environments which accelerate contaminant decomposition. While these studies consistently show the importance of supplemental oxygen to support the microorganism population, only a few have focused on the transport processes necessary to provide this oxygen in the contamination zone [12,14,19,20].

In the present study, we have developed a micro-mechanical network model intended to describe several micro-scale or sub-grid processes that are not explicitly treated in large-scale continuum models. A source listing of the program is given in Appendix A. One unique aspect of this model is its emphasis on particle transport and its effect on permeability. A primary goal of this work is to improve the
predictive capability of bioremediation modeling by helping to develop improved continuum correlations relating the macroscopic permeability and effective diffusivity to the evolved state of an interconnected pore structure. In addition, this research provides new capabilities for modeling fluid, species and particle motion in both ordered and random porous materials, with potential application to a wide range of micro-scale fluid transport and filtration problems.

In the experimental part of the present study, we have designed and built a laboratory-scale apparatus for measuring permeability reductions due to fines transport and deposition. Using this capability, we have made preliminary measurements of permeability reductions in glass spheres and in several common sands. The goal of this work was to provide permeability data on well characterized porous materials that would serve as a benchmark for the mathematical model.

NETWORK MODEL

Previous continuum models of plugging processes have failed to predict accurately the permeability reductions observed in laboratory and field experiments. This is true not only for bioremediation, but for more conventional filtration processes as well. Likewise, there has been very limited success with analytical models intended to describe plugging processes in terms of the reduction in throat size in a bundle of parallel capillary tubes. These classes of models fail to account for the interconnection among flow passages of differing sizes and the redistribution of flow that occurs as plugging proceeds.

To overcome these shortcomings we have developed a micro-mechanical network model that describes the internal structure of a random porous material as a system of interconnected tubes and nodes [21,22,23]. Tube networks such as these are generally constructed by placing tubes directly in the computational domain using a Poisson distribution or similar random process. Our approach differs from this in that the tube and node network is constructed about a collection of particles [24,25,26,27].

The computational domain is first symbolically packed with spherical particles from a specified size distribution. The size distribution is obtained by sampling spike, normal or log-normal distributions using a uniform random number generator. By this technique, a repeatable pseudo-random distribution of particle sizes of a specified mean and variance may be generated very quickly.

Once a particle size distribution is generated, the particles are assembled into a porous structure. To perform this task we have developed several packing algorithms. These appear in lines pages 3 through 19 of Appendix A. The first method employs a random number generator to place particles in a specified box, subject to
the constraint that no particles may inter-penetrate. This constraint may be further tightened to prevent the placement of a particle within a specified distance of any other. Particles are added in this manner to the box until a specified fractional density is obtained. This method is reasonably fast, but fractional densities greater than about 40% cannot be obtained by this technique. This is because granular materials having greater densities are not at all random.

To obtain high fractional densities, we have developed a second packing algorithm in which particles are added randomly to the top of the box. These particles are then moved downward (as though by gravity) until they arrive at the upper surface of those particles previously placed. When a falling particle first contacts another, the region around that fixed particle is examined numerically to determine the minimum energy state in which the new particle can be placed. The processes of adding individual particles is repeated until the box is full. This algorithm is quite slow since the size and location of many other particles must be checked as each new particle is packed. Despite the large effort required by this method, and the seemingly high degree of consolidation that should be obtained, fractional densities obtained by this method rarely exceed about 65%, depending on the size distribution of the particle set.

To obtain still high packing densities, we have developed a third algorithm in which all particles are initially placed by deterministic methods to obtain the desired density. Large particles are placed first in a regular packing. Smaller and still smaller particles are added later within the voids formed by those already placed. The resulting structure is highly ordered, and does not provide a good representation of a random material. To achieve greater disorder, each particle in the final set is sent on a random, noninterfering walk through the box. This is an extremely time consuming process, since the current position of each particle must be checked against all others in the box to ensure that inter-penetration does not occur. For the large computational price, however, fractional densities as high as the size distribution will permit may be obtained by this method.

Once the size and position of each particle has been determined, the network model is constructed. For this we have employed a subdivision of the space via a Voronoi tessellation [28]. This tessellation assigns each region of the domain to the closest particle. The Voronoi tessellation results in a polygon bounding each particle. The sides of these polygons form the skeleton of the network of channels surrounding the particles. A sample calculation showing the network at this stage is shown in Fig. 1. Note that each particle is bounded by several network channels. The most frequent number of bounding channels is five, though in large particle arrays as few as three and as many as twelve bounding channels may be seen. Also note that network channels always form intersections of three, and that these intersections define a unique node location at the end of each channel.
Figure 1. Schematic of particle array and corresponding Voronoi tessellation. The sides of the tessellation polygons form the network of channels through which fluid transport occurs.

Figure 2. Schematic of particle array and channel network. Channel intersections define the nodes of the computational domain. Although these channels are displayed as having parallel walls, the governing transport equations take into account that channels conform to particle surfaces.
After the Voronoi tessellation is computed and the skeleton of the network is known, the nodes identified by pore tube intersections are numbered and cross correlated. This cross-correlation table provides a numerical map of connected node pairs and corresponding tube numbers, making the network a useful computational domain for solving the transport equations. Following construction of the network skeleton, channel sizes for each segment of the network are computed using the known center positions and radii of the two particles defining each channel segment. The source listing of the tessellation algorithm and method of network construction appears in pages 19 through 35 of Appendix A.

A sample of a completed tube network is shown in Fig. 2. This network represents the pore structure in a two-dimensional slice through a three-dimensional particle array. In the sample shown, and in Fig. 1, the particles are uniform spheres. They appear to have nonuniform sizes only because of varying positions in the out-of-plane direction. The volume within this collection of tubes is the domain of the network model on which the conservation equations are solved.

CONSERVATION EQUATIONS

The transport equations governing fluid motion and species transport in a network model are much simpler than those usually encountered in multidimensional transport in porous materials. The reason for this is that all complexities of the pore geometry are explicitly described by the network itself. Phenomena such as reduced diffusivities due to the presence of a solids fraction and both longitudinal and transverse dispersion arise naturally as a consequence of the network geometry. As a result, these porous media transport phenomena, described by empirical correlations in conventional continuum models, are addressed in a network model on a first-principles basis, or nearly so.

For the isothermal transport of a species \( i \) through the tube network, conservation of mass is given simply by

\[
V_k \frac{d}{dt} (\rho_k \bar{f}_{i,k}) = \sum_{j=1}^{3} q_{j,k} \bar{f}_{j,k} - S_k
\]

(1)

where \( V_k \) is the portion of tube volume associated with node \( k \), \( \rho_k \) is the fluid density at node \( k \), and \( \bar{f}_{i,k} \) is the local species fraction. The summation on the right of Eq. (1) is a sum over the species transport rates from the three tubes connected to node \( k \), and \( \bar{f}_{j,k} \) is the mean species concentration in each tube. The final term on the right of Eq. (1) accounts for any sources or sinks due to surface or homogeneous reactions.
Momentum equations for the network model are likewise much simpler than those for continuum models of porous materials. Here, for low Reynolds number flows, the mean fluid velocity is related to the pressure gradient by conventional tube or channel correlations.

\[ q_{j,k} = -\frac{\delta_{3,j,k}}{24} \Delta p_{j,k} \quad \text{or} \quad q_{j,k} = -\rho \frac{\delta_{4,j,k}}{64} \frac{\Delta p_{j,k}}{\ell_{j,k}} \]  

\( \Delta p_{j,k} \) is the pressure drop across the tube joining the \( j \) and \( k \) nodes, \( \ell \) is the tube length, and \( \delta_{j,k} \) is the equivalent tube aperture yielding the correct mean flow rate. Equivalent apertures are discussed further in the following section. The first of these relations applies to a two-dimensional channel, appropriate for flow in two-dimensional geometries, while the second applies to a circular tube, appropriate for three-dimensional particle arrays.

To solve the governing transport equations, pressures are imposed on two external surfaces of the network domain. The other two boundaries are made impermeable by closing any tubes crossing these planes. Pressures for the tube network are then computed by solving Eq. (1) for each interior node. This system of coupled node equations is solved by a time marching algorithm. Using an appropriate equation-of-state relating the fluid pressure, temperature and density, either liquid or gas flows may be treated in this way. The algorithms for computing the pressure field are given in pages 35 through 48 of Appendix A.

Once the internal pressures and tube mass flow rates are computed, the permeability of the network is calculated by summing the flow contributions from each tube to obtain the total flow rate through the entire network. From this total flow rate, the fluid properties, size of the domain, and the specified boundary pressures, the macro-scale permeability of the medium can be computed. By letting tube diameters diminish as biomass accumulates or as channels become blocked by particulate fines, changes in the pore structure and the effect of the evolving structure on the flow rate can also be computed.

Finally, we note that Eq. (1) does not contain any contribution due to diffusive transport. Although this additional transport mechanism can be included in this manner, we have found that describing diffusive processes via tracer particles is generally a more direct means of utilizing the full capabilities of the network model. Tracer particle dynamics are addressed in a later section.
NETWORK SUB-SCALE MODELS

Although a network model resolves transport phenomena down to the particle scale, many important processes take place on still smaller scales. Diffusion and viscous dissipation, for example, both involve processes occurring at the molecular scale. Rather than explicitly modeling the details of molecular interactions, these sub-scale phenomena are treated here in the conventional continuum fashion. Further, by averaging the governing continuum equations over the channel cross-sections and then integrating over their lengths, closed-form expressions for effective transport properties of each tube segment are obtained. These permit computation of the full pressure and concentration fields using state variables only at the network nodes.

To account for sub-scale processes in the network model, we define four equivalent apertures relating the tube volume, effective permeability, diffusivity and fluid transit time to the tube length. The simplest of these is the equivalent aperture for tube volume, $\delta_V$, defined by

$$ V = \ell \delta_V $$

where $\ell$ is the tube length, and $V$ is its volume. It is straightforward to show that in this case the equivalent aperture is given by

$$ \delta_V = \frac{2}{\ell} \int_0^{\ell/2} \delta dz $$

where $\delta$ is the local tube aperture.

The second equivalent aperture is that of the permeability. In this case, the equivalent aperture for two-dimensional flow is defined by

$$ q = \frac{\delta^3}{12} \frac{dp}{dz} = -\frac{\delta_k}{12} \frac{\Delta p}{\ell} $$

where $q$ is the constant volumetric flow rate through the tube, $p$ is the local pressure and $\Delta p$ is the total pressure drop along the tube. Note that we make no distinction here between compressible and incompressible flow. The reason for this is that each tube is very short, thus the ratio of the pressure drop to the mean pressure is always very small. In light of this, the effects of compressibility may be neglected on this scale. To solve for $\delta_k$ requires that the local pressure is integrated over the tube length to obtain the total pressure drop. Substituting that result into Eq. (5) then yields

$$ \frac{1}{\delta_k^3} = \frac{2}{\ell} \int_0^{\ell/2} \frac{dz}{\delta^3} $$

where $\delta$ is the local tube aperture.
for the equivalent aperture providing the correct viscous forces within each tube.

The third equivalent aperture is that for the diffusivity. This is defined such that the constant diffusive transport rate, \( f \), through a tube is correctly specified by

\[
f = \delta D \frac{dc}{dz} = \delta D \Delta c / \ell
\]  

(7)

where \( D \) is the coefficient of binary diffusion, \( c \) is the local concentration of the diffusing species and \( \Delta c \) is the total variation of the concentration along the tube length. Now integrating Eq. (7) over the tube length to obtain the total variation in the concentration, and substituting that result back into Eq. (7) gives

\[
\frac{1}{\delta_D} = 2 \int_0^{\ell/2} \frac{dz}{\delta}
\]  

(8)

for the equivalent tube aperture based on diffusion.

The final aperture to be defined is that for the mean fluid velocity or transit time. This yields the correct time a tracer particle is resident within a tube and so is useful in tracking particle motion within the network. In this case, the equivalent aperture is defined by

\[
t = 2 \int_0^{\ell/2} \frac{dz}{u} = \frac{\ell}{\bar{u}}
\]  

(9)
where \( t \) is the tube transit time, and \( u \) and \( \bar{u} \) are the local and mean fluid speeds, respectively. These are given by

\[
u = \frac{\varepsilon^2 \Delta p}{12 dz} \quad \text{and} \quad \bar{u} = \frac{\varepsilon^2 \Delta p}{12 \ell}
\]

Substituting Eqs. (10a) and (10b) into Eq. (9) and rearranging slightly yields

\[
\frac{1}{\varepsilon_u^2} = \frac{2}{\ell} \int_0^{l/2} \frac{\varepsilon}{\varepsilon_k^2} dz = \frac{\varepsilon_V}{\varepsilon_k^2}
\]

Thus the equivalent aperture for mean fluid speed and transit time can be written simply in terms of those for the permeability, \( \varepsilon_k \), and the tube volume, \( \varepsilon_V \).

To apply these definitions of the four equivalent apertures, we must now take into account a specific particle geometry. Consider two two-dimensional particles having centers \((x_1, y_1)\) and \((x_2, y_2)\), and radii \(R_1\) and \(R_2\). As shown in Fig. 3, the local aperture between these particles is given by

\[
\delta = \delta_0 + R_1 + R_2 - \sqrt{R_1^2 - z^2} - \sqrt{R_2^2 - z^2}
\]

where the minimum aperture, occurring along the line joining the two centers, is

\[
\delta_0 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 - R_1 - R_2}
\]

and \(z\) is the distance measured in a direction orthogonal to the line of centers. Now taking

\[
R_2 = R_1 (1 + \epsilon) \quad \text{and} \quad z = R_1 \sin \theta
\]

and substituting these results into Eq. (12), the local aperture can be expressed in terms of only \(R_1\), \(\epsilon\) and \(\theta\).

\[
\delta = \delta_0 + R_1 \left[ 2 + \epsilon - \cos \theta - (1 + \epsilon) \left[ 1 - \frac{\sin^2 \theta}{(1 + \epsilon)^2} \right]^{1/2} \right]
\]

\[
\approx \delta_0 + R_1 [1 - 2 \cos \theta + \epsilon(1 - \sec \theta)]
\]

Note that the latter of these relations is based on a one-term expansion for small values of \(\epsilon\).

Substituting Eq. (12) into Eq. (2) and performing the indicated integration yields

\[
\frac{\varepsilon_V}{R_1} = 2\kappa + \epsilon - \frac{1}{2} [(1 + \epsilon)^2 - \sin^2 \theta]^{1/2}
\]

\[
-\frac{R_1}{\ell} \frac{(1 + \epsilon)^2 \sin^{-1} \left( \frac{\sin \theta}{1 + \epsilon} \right)}{1 - \frac{1}{2} (1 - \sin^2 \theta)^{1/2}} - \frac{R_1}{\ell} \theta
\]
for the equivalent volume aperture. The new parameter, \( \kappa \), appearing here is given by

\[
\kappa = 1 + \frac{\delta_0}{2R_1}
\]  

(17)

and in this case \( \theta \) is evaluated at the angular limit of integration

\[
\theta = \sin^{-1} \left( \frac{\ell}{2R_1} \right)
\]  

(18)

corresponding to the end of the tube segment.

Similarly, the equivalent aperture yielding the correct viscous drag is obtained by substituting Eq. (15b) into Eq. (6) and integrating the result. Using a tedious though straightforward change of variables, this gives

\[
\frac{\delta^2_{D}}{\ell R_1^2} = \frac{4}{3} \frac{\kappa^2 - 1}{f(\kappa, \theta)} \left[ 1 + \frac{3\epsilon I(\kappa, \theta)(\kappa^2 - 1)}{2f(\kappa, \theta)} \right]
\]  

(19)

where

\[
I(\kappa, \theta) = \frac{1}{3(\kappa + 1)} \left[ \frac{\sin \theta}{(\kappa - \cos \theta)^3} - \frac{3h(\kappa, \theta)}{2(\kappa^2 - 1)} + \frac{2f(\kappa, \theta)}{2(\kappa^2 - 1)} \right]
\]  

(20)

where the functions \( h(\kappa, \theta) \) and \( f(\kappa, \theta) \) are

\[
h(\kappa, \theta) = \frac{\sin \theta}{(\kappa - \cos \theta)^2} + \frac{3\kappa \sin \theta}{(\kappa^2 - 1)(\kappa - \cos \theta)}
\]

\[
+ \frac{2 - 2\kappa^2 + 1}{(\kappa^2 - 1)^{3/2}} \tan^{-1} \left[ \frac{\kappa + 1}{\kappa - 1} \tan \left( \frac{\theta}{2} \right) \right]
\]  

(21)

and

\[
f(\kappa, \theta) = \frac{(2 + \kappa^2) \sin \theta}{2(\kappa^2 - 1)(\kappa - \cos \theta)} + \frac{\kappa \sin \theta}{2(\kappa - \cos \theta)^2}
\]

\[
+ \frac{3\kappa}{(\kappa^2 - 1)^{3/2}} \tan^{-1} \left[ \frac{(\kappa + 1)^{1/2}}{(\kappa - 1)^{1/2}} \tan \left( \frac{\theta}{2} \right) \right]
\]  

(22)

respectively.

Finally, we consider the equivalent aperture appropriate for computing diffusive transport through the tube network. Now substituting Eq. (15b) into Eq. (8) and again performing the integration gives

\[
\frac{\delta_D}{\ell} = \frac{1}{g(\kappa, \theta)} \left[ 1 + \frac{\epsilon \psi(\kappa, \theta)}{2(\kappa + 1)g(\kappa, \theta)} \right]
\]  

(23)

13
where
\[ g(\kappa, \theta) = -\theta + \frac{2\kappa}{(\kappa^2 - 1)^{1/2}} \tan^{-1} \left[ \frac{(\kappa + 1)^{1/2}}{\kappa - 1} \tan \left( \frac{\theta}{2} \right) \right] \] (24)

and
\[ \psi(\kappa, \theta) = \frac{\sin \theta}{\kappa - \cos \theta} - \frac{2}{(\kappa^2 - 1)^{1/2}} \tan^{-1} \left[ \frac{(\kappa + 1)^{1/2}}{\kappa - 1} \tan \left( \frac{\theta}{2} \right) \right] \] (25)

For the special case in which all particles within the particle array are the same size, the results above become greatly simplified. Rewriting Eq. (16) for \( \epsilon = 0 \) yields
\[ \frac{\delta \nu}{R} = 2\kappa - (1 - \sin^2 \theta)^{1/2} - 2 \frac{R}{\ell} \theta \] (26)
for the equivalent volume aperture. Similarly, Eq. (19) in this limit reduces to
\[ \frac{\delta^2}{\ell R^2} = \frac{4 (\kappa^2 - 1)}{f(\kappa, \theta)} \] (27)
for the equivalent aperture for viscous fluid flow. Finally, the equivalent aperture for diffusion, given by Eq. (23) above, becomes
\[ \frac{\delta D}{\ell} = \frac{1}{g(\kappa, \theta)} \] (28)
for this case of uniform particle size. As before, the angular limit of integration, \( \theta \), is taken as that corresponding to the tube length.

\section*{Effective Transport Properties}

The sub-grid models described above are intended for use in a network model generally capable of describing transport in disordered porous materials. We can, however, also apply these sub-grid relations directly to regular particle arrays. For such regular arrays, the geometry of the corresponding pore network is easily specified. In addition, the permeability and diffusivity have been computed by direct solution of the Navier-Stokes and diffusion equations for several regular patterns of two and three-dimensional particles [29,30,31]. Comparing the present results with
Figure 4. Unit cell of uniform hexagonal array of cylindrical particles. Particle size and minimum aperture determine porosity, diffusivity and permeability.

these numerical solutions provides a useful check on the accuracy of the network approach and on the range of applicability of correlations derived from the equations outlined earlier.

To this end, we now consider the simple problem of flow and diffusion through a regular hexagonal array of circular cylinders. As shown in Fig. 4, the particle size and geometry of the unit cell uniquely determine all transport properties of the array. From the definitions of the equivalent apertures, the effective permeability of this array is given by

\[ k = \frac{1}{12} \frac{\delta_k^2}{h \sqrt{\tau}} \]  

where \( h = \sqrt{3} S / 2 \) is the height of the unit cell, and \( \tau = 4/3 \) is the tortuosity for flow from left to right in the geometry shown. Again, \( \delta_k \) is the equivalent aperture for viscous flow. Using Eq. (27), this result may be expressed as

\[ k = \frac{1}{12} \frac{(\kappa^2 - 1)}{f(\kappa, \theta)} \frac{\ell d^2}{h \sqrt{\tau}} \]  

where \( f(\kappa, \theta) \) is given by Eq. (22). Again from the geometry we obtain

\[ \frac{\ell}{h} = \frac{2}{3} \quad \text{and} \quad \kappa = 1 + \frac{\delta_0}{d} = \sqrt{\frac{1 - \phi_0}{1 - \phi}} \]  

(31a,b)
where \( \phi \) is the porosity (void volume fraction), and \( \phi_0 = 1 - \pi/2\sqrt{3} \approx 0.093 \) is the porosity at the percolation threshold. The percolation threshold is the condition at which interconnected pores just marginally span the material sample of interest. Also from geometry, the angular limit of integration is

\[
\theta = \sin^{-1}\left( \frac{\ell}{d} \right) \quad \text{where} \quad \frac{\ell}{d} = \frac{\kappa}{\sqrt{3}} \quad (32a,b)
\]

Now defining a reference permeability as the area of the unit cell,

\[
k_r = \frac{\pi d^2}{4(1 - \phi)} \quad (33)
\]

the normalized permeability may be expressed as

\[
k^* = \frac{k}{k_r} = \frac{\phi - \phi_0}{3\sqrt{3}\pi f(\kappa, \theta)} \quad (34)
\]

We note that exactly this result is also obtained for vertical flow through the unit cell of Fig. 4. The application of equivalent apertures to that problem is somewhat more difficult, however, since the unit cell for flow in that direction involves the confluence of two tubes into one.

A comparison between these results and an exact analytical solution to the problem [32] is shown in Fig. 5. For this simplified geometry of a regular hexagonal array, Eq. (34) agrees to within 15% for all porosities above the percolation threshold and below \( \phi = 0.9 \). Based on this agreement, we conclude that the expressions describing the equivalent aperture for viscous flow in two-dimensional channels should be suitable for network modeling of flow through disordered materials. Similar agreement has been obtained between the equivalent aperture formulas and exact solutions for two-dimensional square arrays and for the more complex problem of three-dimensional flow through regular arrays of spheres.

We now consider the use of equivalent apertures for computing the effective diffusivity of a regular hexagonal array. From Eqs. (8) and (28), the effective diffusivity may be written as

\[
D^* = \frac{D_e}{D} = \frac{\delta_D}{h\sqrt{\tau}} = \frac{\ell}{h g(\kappa, \theta) \sqrt{\tau}} \quad (35)
\]

where \( D_e \) is the apparent diffusivity in the porous array, and \( D \) is the coefficient of diffusion. This expression can be evaluated directly using the values of \( \ell/h, \kappa, \theta \) and \( \tau \) given above, along with Eq. (24) for \( g(\kappa, \theta) \). Again, although this result applies to diffusion from left to right in the geometry shown in Fig. 4, the same
result is obtained for diffusion in the vertical direction, and for that matter, in any arbitrary direction through the unit cell.

A comparison between this result and analytical solutions to the diffusion problem [33] is also shown in Fig. 5. In this case, Eq. (35) agrees to within 10% for all porosities above the percolation threshold and below $\phi = 0.8$. Again, a similar approach has been applied to other geometries with comparable agreement between the results of the network model and direct numerical solutions.

Although the diffusivity and permeability of a well ordered medium can be described analytically, numerical simulations are usually needed to determine the transport properties of random materials. To deduce the permeability of a particular network model, it is only necessary to set pressure boundary conditions on a pair of opposing faces and calculate the resulting flow rate. By seeding tracer particles into the flow and observing their motion it is then possible to observe the process of hydrodynamic dispersion that results from differences in fluid speed along different streamlines. The dispersivity of a given medium is influenced by the distribution of tube sizes and lengths, the degree of connectivity, and the degree of mixing that occurs at tube junctions. Dispersion also occurs on the scale of a single flow tube owing to differences in fluid speed between wall and center streamlines. All of these factors are accounted for explicitly in the network model by means of tracer particles.
PARTICLE TRANSPORT

Tracer particles may serve either as fictitious markers of the fluid motion or as physical particles of finite size. In the first capacity they are used to represent diffusive and dispersive contributions to the transport of reactive species carried by the mean flow [34,35,36]. In the latter role, physical particles of a finite size can be advected through the network to describe the advection, deposition, and accumulation of particles in the void space [37,38,39]. Blockage or partial blockage of individual tubes is straightforward to compute since the full geometry of the tube network is known and may be evolved in time. Thus, the reduction in permeability due to the relocation and deposition of biomass and native fines can be calculated from fundamental considerations.

To track the motion of tracer particles requires a knowledge not only of their streamline positions within tubes but also of their trajectories at nodal interconnections. Two basic configurations of nodes are possible: (1) a node having one in-flow and two out-flow tubes; and (2) a node having two in-flow and one out-flow tubes. These are the only possible configurations for a network constructed via Voronoi tessellation. In assigning particles to outflow tubes it is simplest to assume that all junctions are well mixed and to assign probabilities to the outflow channels and their streamlines based on their respective flow rates. However, it is generally much more realistic to require that particles follow continuous paths that smoothly interconnect the incoming and outgoing streamlines. That substantially more difficult approach is implemented in the present network model.

To compute particle motion through a network node, we assume a parabolic fluid velocity profile over the cross-section of each of the three tubes forming the node. Given the spatial position of a particle as it exits a tube, the node entrance streamline based on the parabolic velocity distribution can be determined. The corresponding node exit streamline can then be computed based on the total node in-flow and out-flow and knowledge of whether the node currently possesses one or two in-flow tubes. Having computed the node exit streamline, the particle is placed at the correct radial position (in the correct exit tube, if more than one exit tube exists) again based on the parabolic velocity profile. Using this approach, real or tracer particles may be transported through the entire tube network following a single streamline through the repeated branching and confluence of tubes. For low Reynolds number creeping flows, typical of those in the applications of interest, turbulent transport of particles across streamlines does not exist. The only mechanism for this process is particle diffusion. It is this diffusion, along with the velocity profile within each tube and the variation in mean tube velocities, that gives rise to both longitudinal and transverse dispersion in flows in porous material.

Dispersion in a porous material is a complex process involving simultaneous flow and diffusion [40,41]. To model this process, we first compute the mean flow...
field for the entire tube network. This yields the local fluid speed at each radial and longitudinal position within each network tube. We then inject a large number of particles at the inflow boundary and track their progress through the network for a fixed time. Particles are partitioned among tubes on the inlet boundary based on the total inlet flow rate and the contribution to this total from each inlet tube. This ensures that particles enter the network inlet tubes in correct proportions. Once an inlet tube is selected, the particle is placed at a radial position using a probability distribution that reflects the parabolic velocity profile. This ensures that particles entering a given tube are correctly distributed in accordance with local fluid speeds. Under these procedures, particles enter the network as though they were supplied from a reservoir of fluid containing a uniform particle concentration.

As the injected particles enter the tube network, they are advected by the local mean flow and diffuse about this mean speed. The diffusive portion of this transport is described by a noninterfering random walk [21,34]. At each time step, an advective displacement is computed from the local fluid speed and the size of the time step, \( \delta t \). The time step may be constant or may be obtained by sampling a uniform random distribution. For the same time period, a random diffusive displacement, \( \delta \ell_D \), is computed from the time step size and the specified diffusivity, \( D \). This is given by

\[
\delta \ell_D = \xi \sqrt{2D \delta t}
\]  

(36)

where \( \xi \) is a distribution function that may be unity, random or Gaussian. All three give very similar results, provided that the step size is adjusted such that the mean step size is consistent with Eq. (36). The direction of the diffusive step is then computed by sampling a uniform random distribution, and the combined advective and diffusive steps are taken. If the resulting particle position is outside the tube network, the step is recomputed.

Since all of the tracer particles advance with different speeds, they tend to spread apart as they traverse the medium. To quantify this longitudinal dispersion, each of the particles is transported through the network for a fixed period, \( t \), and the final position of each is noted. Following the injection and transport of a large number of particles, the cumulative distribution of final particle positions is fit with an error function to obtain the mean and variance of the particle positions. The error function is used because it is the solution to the continuum dispersion equations. The dispersivity, \( D^* \), is then be calculated from the variance, \( \sigma^2 \) of the particle positions by

\[
D^* = \frac{\phi \sigma^2}{D \delta t}
\]  

(37)

where \( \phi \) is the porosity of the medium. The dispersivity is a property of the material, and so should be independent of the time interval, \( t \), provided that the interval is large enough to sample a statistically significant portion of the network.
The procedure above is used to compute longitudinal dispersion and the longitudinal dispersivity. Transverse dispersivities are computed by a similar method, except that particles are injected into only a single entrance tube. In this case, the particles are tracked and the final transverse position of each is noted. These positions are fit using a Gaussian distribution, and the transverse dispersivity is computed from the variance, again using Eq. (37).

A random walk is also used, in the absence a net fluid motion, to compute the effective diffusivity in the network model. As in the earlier simulations of hydrodynamic dispersion, particles are introduced into tubes along a vertical or horizontal line through the network. In this case, however, the starting line is generally centered within the medium since there will be no net displacement of the particle front. From their initial positions, the particles are again sent on random walks. After a specified time period, the effective diffusivity of the network is computed by matching a Gaussian distribution to the computed spatial distribution of the final tracer positions.

A number of options may be exercised in computing diffusivities. The computational algorithm permits either a random walk on the tube network or directly on the void volume defined by the particles comprising the granular material. Also, when computing diffusivities, the distinct contributions of ordinary and Knudsen diffusion may be determined by varying the mean diffusive step size relative to the characteristic pore diameter. Knudsen diffusion becomes dominant when the mean free path (diffusive step size) of the tracer particles become comparable to that of the pore size. In this regime, most diffusive steps result in collisions with a particle of the porous structure. Although Knudsen diffusion is not usually important in bioremediation applications, this capability of the network model has been employed in computing effective diffusivities for porous fiber preforms used in composites manufacturing by low-pressure chemical vapor infiltration.

Particle transport simulations are particularly advantageous in computing the transport of reactive chemical species [35,36]. Using minor variations on the methods above, species concentrations can be computed from particle concentrations during continuous particle injection. Particle lifetimes defined by reaction probabilities, can be used to account for both homogeneous and surface reactions. Surface reaction rates are especially easy to compute by this technique since the surface impingement rate, defined by a particle displacement to a region outside the tube network, is already monitored as a necessary part of the particle advective and diffusive motion. Details of the mathematical methods used in particle transport appear in pages 48 through 58 of Appendix A.
SAMPLE CALCULATIONS

To demonstrate the unique capabilities of a network model, we now present the results of calculations for several sample problems. These sample problems include the pressure field for an incompressible flow, permeability reductions due to uniform film growth on particles of a granular medium, particle-scale fingering and the associated longitudinal dispersion, and the effects of the Peclet number on transverse dispersion.

The first step in solving all transport problems using a network model is to compute the pressure field. This is done by solving the coupled continuity equations for all network nodes by means of a relaxation technique, subject to the desired boundary conditions. The result is a two or three-dimensional spatial distribution of pressures. A sample pressure field is shown in Fig. 6. The horizontal axis in this plot is the spatial position along the direction of flow. The vertical axis is the normalized pressure, where the normalization is such that the inlet pressure is unity and the exit value is zero. Boundary conditions for this sample problem are fixed pressures on the inlet and exit and impermeable boundaries on the top and bottom of the domain. The inlet and exit conditions are imposed by identifying those nodes lying on these boundaries and assigning the appropriate value, which is then held fixed through computation. Conditions on the impermeable boundaries are imposed by identifying those tubes crossing the top or bottom of the domain, and assigning to these tubes an effective aperture of zero. In this manner, no flow may cross the top or bottom boundaries. These particular boundary conditions are frequently used because they permit a direct calculation of the directional permeability once the pressure and flow fields have been determined.

This plot shows one particularly interesting feature of a typical pressure field. Although the mean gradient of the pressure is always negative, corresponding to flow from the inlet toward the exit, local pressure gradients are sometimes positive. That is, on the scale of particles within a porous granular material, local fluid velocities may oppose the mean flow direction. This condition arises naturally in disordered materials and is important because such local variations in both the magnitude and direction of fluid speeds contribute significantly to the very large apparent diffusivities associated with longitudinal dispersion.

Another interesting feature of Fig. 6 is the degree to which the mean node pressures deviate from the linear gradient obtained from a continuum model. The gradient of mean node pressures is about 20% above the linear value near the inlet and about 20% below at the exit. This deviation from the continuum result arises because the permeability of the network is locally lower than the average value near the inlet and locally higher near the exit. Since the total flow rate through the network is the same at all axial positions, low local permeabilities give high pressure gradients, while high permeabilities give relatively lower gradients of the mean node
Figure 6. Normalized pressures at nodes inside a tube network. Dashed curves indicate interconnected nodes. The solid diagonal line shows the linear pressure gradient that satisfies the continuum equations for low Reynolds number flow of an incompressible fluid through a homogeneous porous medium.

pressure. This behavior is a result of the inherent nonuniformity of disordered porous materials. On larger domains, this effect is still more pronounced. This is important to the problem of bioremediation, since low permeabilities are associated with large specific surface area, and large surface areas lead to high contaminant retention. Thus regions most likely to require decontamination are also the most difficult to supply with the oxygen and nutrients needed for rapid treatment by this method.

We now consider the problem of the evolving pore structure and associated reductions in permeability due to accumulation of solids on particles of a granular porous solid. To generate the pore networks for this type of problem, we first generate a collection of particles by one of the three methods previously described. Then, a region of uniform thickness around each particle is excluded from the void space to account for that portion of the initial void occupied by the deposited material. This numerical process mimics that of biofilm growth on the surfaces of granular solids when the particle size is much larger than that of the microbe. Finally, the tube network is constructed about the particles and accumulated mass to obtain a network representation of the remaining void volume.

Three sample networks constructed in this way are shown if Fig. 7. Figure 7A represents the initial formation, while Figs. 7B and 7C represent the same collection
of particles with 33% and 66% of the initial void filled. The initial network (7A) at a porosity of 0.6 consists of 223 tubes and has a normalized permeability of about $k^* = 1.1 \times 10^{-3}$. When the porosity is reduced to 0.4, the resulting network (7B) still contains 166 tubes, but the normalized permeability has dropped by over an order of magnitude to only $k^* = 8.0 \times 10^{-5}$. This large reduction in permeability is due only in a small part to the fact that the remaining tubes have reduced diameters. The more important reason for this large effect is that the network void volume is losing connectivity. In the initial configuration, at a porosity of 0.6, over 97% of all the nodes are connected to two other nodes. At a porosity 0.4, only 37% of the nodes are still connected to two other nodes, over 50% are connected only to one, and about 10% are no longer connected to the network. Finally, at a porosity of 0.2, only 14% of the nodes remain doubly connected, 55% possess a single connection, and over 25% have become altogether isolated. It is this dramatic decrease in the number of interconnected pores that accounts for most of the large drop in permeability.

Despite the fact that the network still retains a large number of node connections at a porosity of 0.2, the permeability of the network shown in Fig. 7C is zero. At this porosity, the network is just below the percolation threshold, as a continuous path connecting the entrance and exit planes no longer exists.

To illustrate the use of tracer particles in computing effective transport properties, we now consider the problems of longitudinal and lateral dispersion. Dispersion in a porous material is a complex process involving both flow and diffusion. Variations in local fluid velocities yield varying particle speeds as tracer particles traverse the pore network. Diffusion is important to this process because only by diffusion can particles move from one streamline to another. High diffusion coefficients give rise to a wider sampling of fast and slow streamlines in large and small pores, yielding smaller variations in average fluid speeds. Thus, contrary to intuition and to many statements made in the literature, an increase in molecular diffusivity leads to a decrease in longitudinal dispersivity.

Fig. 8 shows the instantaneous fluid interface during intrusion of a fluid into the pore network of a three-dimensional random packing of polydisperse particles. The interface is tracked across the network by a large number of tracer particles injected into the boundary between the two fluids. As the invading fluid fills progressively more of the pore volume, the interface roughens due to the varying mean fluid speeds along the various paths. This roughness of the advancing front is equivalent to a diffusion process in which the two fluids intermix along the plane of the intrusion front. This is not a true diffusion, however, and is referred to instead as dispersion.

A pronounced feature of Fig. 8 is the finger-like structure of the fluid interface. Such structures are widely known to occur in flows in porous media when the viscosity of the invading fluid is lower than that of the fluid initially occupying the pore volume [42,43]. In that case, roughness of the interface results from an inherent
Figure 7. Evolution of pore structure due to uniform accumulation on particles. 7A shows original pore structure, 7B shows 10% accumulation by total volume, and 7C shows 20% accumulation. The network of 7B has a permeability more than an order of magnitude below that of the original. The portion of the network shown in 7C is just below the percolation threshold.
instability, and the resulting structures are known as Saffman-Taylor fingers [44]. The fingers in Fig. 8 do not have this origin. Instead, these fingers result only from the statistical nature of the distribution of pore sizes and connectivity of the pore network. For a given network, the formation of these fingers is almost entirely deterministic and depends only on the magnitude of the Peclet number indicating the relative importance of advective and diffusive transport.

Quantitative values of the dispersivity are extracted by analyzing the spatial distribution of the tracer particles used to map the intrusion interface. This is illustrated in Fig. 9. Here the final position of all tracer particles is shown on a slice through the particle array. The jagged solid curve is the cumulative distribution of these positions starting from the right boundary of the domain. The dashed curve is an error function, fit to the cumulative distribution in a least-squares sense by selecting the best values of the mean and standard deviation. The error function is used for this purpose because it is a solution to the continuum equations describing diffusion about the fluid interface. The dispersivity can be computed directly from the mean and variance obtained from this fit by the relation

$$D^* = \frac{\sigma^2}{2t} = \bar{u} \sigma^2 / 2\bar{x},$$

where $\bar{u}$ is the mean fluid speed, $\sigma^2$ is the variance, and $\bar{x}$ is the mean particle position. Note that the very good agreement between the cumulative distribution
Figure 9. Tracer particle final positions. Particles are injected into the left boundary and carried by the mean flow for a specified period. Fitting the cumulative distribution of their final positions (dots and solid curve) using an error function (dashed curve) yields the dispersivity of the network.

of the particles positions and the error function fit indicates that dispersion in the network does indeed mimic a diffusion process.

The last sample problem concerns lateral dispersion. This is an important process in both bioremediation and contaminant transport since it strongly influences the vertical and lateral extent of the plume formed as fluids are transported downstream of a source. As with longitudinal dispersion, lateral dispersion involves a coupling between advective and diffusive transport. To examine this process, we have again computed the moving interface between two fluids during fluid injection. This time, however, the fluid is injected into a single tube on the left boundary of the network, rather than along its entire length. This is illustrated in Fig. 10. As before, a large number of particles are initially placed at the fluid interface. These are then carried into the network by the mean flow.

The results shown in Fig. 10 are for a special case in which there is no diffusion of the tracer particles between streamlines. In this case, lateral spreading of the plume is limited to a few pore diameters above and below the point of injection. The reason for this is that all of the particles are confined to the streamline on which they were injected. For low Reynolds numbers there is relatively little mixing of streamlines, even in highly disordered materials, so there is no mechanism to produce significant lateral spreading of the plume. As with longitudinal dispersion, a quantitative
Figure 10. Particle-scale lateral dispersion in a pore network. At high Peclet numbers, lateral dispersion is limited because species do not move readily across streamlines within the pore volume. Results shown are for asymptotically large Peclet number.

Figure 11. Lateral dispersion at low Peclet number. Large coefficient of diffusion allows tracer particles to cross streamlines and follow mean flow into an increasing lateral extent of the network. Results shown are for $\text{Pe} = 1$. 
estimate of the lateral dispersivity can be obtained by fitting the distribution of final particle positions within a slab vertical slab, using in this case a Gaussian profile.

In contrast to longitudinal dispersion, lateral dispersion becomes more pronounced when particles are allowed to diffuse between streamlines. This is illustrated in Fig. 11. Here, the intrusion interface is tracked through the same network used for Fig. 10. In this case, however, the coefficient of diffusion is set to a value to give a Peclet number based on the pore diameter of \( \text{Pe} = \frac{\rho u d}{D} = 1 \). The result is a dramatic increase in the extent of the lateral spread of the plume. The top and bottom boundaries of the plume now grow away from the centerline in proportion to the square-root of the longitudinal distance from the injection point. Note that the larger longitudinal extent of the plume is due to longitudinal dispersion, which is also present here but was absent in the results of Fig. 10.

It is important to recognize that the large lateral extent of the plume in Fig. 11 does not result directly from diffusion. For the diffusion coefficient used to obtain \( \text{Pe} = 1 \), the maximum lateral extent of the plume would be only about half as large if diffusion were the only mechanism for lateral transport.

LABORATORY APPARATUS

The basic experimental approach for investigating deposition of fine suspended solid particles during fluid flow through a porous media was to determine the increases in the local hydraulic gradients within the porous media caused by the deposition of particles. These particles accumulate on the surfaces of the porous material over a considerable depth compared to the dimensions of a typical pore. Thus, pressure gradients are distributed over a relatively large distance within the porous media rather than being localized, e.g., at the surface of a filtering element or plate. The experimental protocol was to measure and record the local pressure distributions within the porous filtration media as a function of time. These data were then used to calculate the hydraulic gradient as a function of position in the packed bed. The experimental variables included the dimensions of the packing material used to create the porous media, the flow rate of the suspension, and the concentration of particulate material. Another objective of this work was to determine the local loading of deposited suspension that caused the observed increases in hydraulic gradients.

A schematic diagram of the experimental apparatus is shown in Fig. 12. An aqueous suspension was pumped from a continuously stirred reservoir to a cylindrical packed column that was operated in the upflow mode. Tubing ports along the axis of the column were connected to a differential pressure gauge to measure the
Figure 12. Diagram of experimental apparatus. Suspension is pumped through bottom of column against constant pressure of the reservoir at the top. Pressures are measured along the length of the column.

local hydraulic pressure relative to the prevailing atmospheric pressure in the laboratory. These pressure data were recorded and stored digitally during the course of an experiment. The liquid exited from the column by overflowing from a small tank that served to provide a constant hydrostatic head during the experiment. Photographs of the experimental apparatus are shown in Figs. 13 and 14.

The granular material that filtered the suspension was packed into cylindrical columns fabricated from acrylic tubing. The columns had an inside diameter of 50 mm (2.0 in.) and were approximately 0.51 m long. The packing material rested on a stainless steel wire mesh that was supported by a narrow ring of acrylic plastic held in place by a flange at the bottom (inlet end) of the column. This arrangement made virtually all of the column cross-section available for flow.

The detailed view of a packed column in Fig. 14 shows the arrangement for monitoring pressure during the flow experiments. A series of ports were drilled in the side wall of the column to allow access for measuring pressure. The pressure ports were spaced 10 mm apart near the inlet end of the columns, 20 mm apart in the center section, and 40 mm apart near the outlet (top end). The ports were spaced more closely near the inlet end as pressure gradients were expected to be largest there. The pressure ports consisted of stainless steel syringe tubing with Luer-Lok adapters that were used to connect the ports to the pressure-sensing device. The openings of the syringe tubes were aligned along the axis of the column. A fine piece
Figure 13. Photograph of the apparatus used to study flow in porous media, showing the packed bed column, reservoir, pump, data acquisition computer and pressure measuring system.

Figure 14. Photograph of a packed bed column as used for filtration experiments showing arrangement of the pressure monitoring ports. The white material at the bottom of the column is a deposit resulting from deposition of fines from a suspension.
<table>
<thead>
<tr>
<th>Packing Material</th>
<th>Diameter (μm)</th>
<th>Mesh Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-170 Spheres*</td>
<td>300-450</td>
<td>40-50</td>
</tr>
<tr>
<td>A-055 Spheres*</td>
<td>500-600</td>
<td>30-35</td>
</tr>
<tr>
<td>Ottawa Sand**</td>
<td>600-850</td>
<td>20-30</td>
</tr>
<tr>
<td>A-150 Spheres*</td>
<td>1400-1700</td>
<td>12-14</td>
</tr>
</tbody>
</table>

* Potter's Industries, Inc., ρ = 2.45 – 2.50 gm/cc  
** Fisher Scientific Co., ρ = 2.62 gm/cc

Table 1. Dimensions of the packing materials used for the filtration experiments.

of stainless steel wire was inserted into each of the syringe tubes to prevent intrusion and clogging by the packing material. As each monitoring port tube was necessarily filled with water to equalize the hydrostatic head before flow was initiated, the time response to changes in pressure were slowed somewhat by the reduction in cross-sectional area due to the inserted wire. However, pressure changes typically occurred quite slowly during these experiments, and the configuration used here responded to step pressure changes within 10 to 15 seconds.

Two types of materials were used to pack the filtration beds. The sizes of these packings are summarized in Table 1, where the particle size refers to the mean diameter. The primary packing material consisted of several grades of solid spherical beads of soda-lime glass (Potters Industries, Inc., Valley Forge, PA) that are classified into relatively narrow size ranges. Three sizes of spheres were used to provide a range of particle diameter that varied by a factor of about four, as indicated in Table 1. The second type of material was Ottawa sand (Fisher Scientific Corp., Pittsburgh, PA). It is a naturally-occurring silica sand that has been classified into a narrow size range of 600 to 850 μm (diameter), which corresponds to 20-30 mesh fraction. The individual grains of sand are generally ellipsoidal in shape. The complete specifications of Ottawa sand are given by ASTM-C 778-80a.

The columns were packed by gradually adding weighed amounts of a given packing material to a column filled with water to avoid trapping air. As the beads settled, the column was agitated by tapping to ensure a uniform packing distribution. The uniformity of packing was subsequently confirmed by measuring the hydraulic gradient during flow of deionized and filtered water and to verify that the gradient was uniform.
A peristaltic pump was used to supply a constant flow rate of suspension during these experiments. The suspension was pumped using a Masterflex Model 7523 peristaltic pump (Cole-Parmer Instrument Co., Vernon Hills, IL). Flow rates up to 270 milliliters per minute (ml/min) could be achieved. The flow rate was determined from the pump rotation speed, rather than in-line measurement, to avoid interference between the suspended particles and a flow sensor. The pumping rate was set by a digital speed control and was reproducible to ±2%. Flow rates were calibrated by collecting known volumes of liquid during measured time intervals over the entire range of flow rates to be investigated using a variety of tubing sizes compatible with the pump heads.

Pulsations from the peristaltic pump were reduced, but not completely eliminated, by an in-line reservoir "pulse dampener," Cole-Parmer Instrument Co., Vernon Hills, IL) that established a trapped liquid/gas interface between the pump and the inlet to the column. This device required approximately 20 ml of liquid holdup and did not cause any loss of the suspended material in these tests. The amplitude of pulsations from the pump were also reduced quite effectively by using the smallest tubing diameter in the pump head that was capable of providing the desired flow rate. As smaller tubing required greater rotational speeds to achieve a given flow rate, the relatively high frequency of the pulses was found to reduce the peak-to-peak amplitude of pressure variations. The data acquisition system, described below, further reduced the influence of pulsations by using an averaging procedure to collect data. Experiments with clean water confirmed that hydraulic gradients could be measured accurately in the presence of larger pressure pulsations than actually occurred during filtration experiments.

The pressure measurement system was designed to determine relatively small hydraulic gradients and thus precise measurements of local pressures throughout the column were necessary. The measurement technique was based on eliminating the hydrostatic component of pressure, thereby increasing the resolution with which pressure changes due to fluid flow and, subsequently, the effect of particle deposition, could be determined. Measurements were made with thin-film pressure transducers (Omega Engineering Inc., Stamford, CT, Models PX-162 and PX-164) and water-filled manometers. The differential pressure transducers had ranges from 0 to 5 in. (water) to 0 to 27 in. (water) and provided electrical output signals that could be interfaced with the digital data acquisition system. The liquid manometers were used for baseline measurements of the hydraulic gradients during flow of clean water, rather than flow of suspensions.

Each pressure port was periodically connected to the pressure transducer by a computer-controlled multi-position valve (Valco Instruments, Houston, TX). The valve ports were connected sequentially, starting each cycle with the ports at the upper end of the column, having the lowest differential pressure, and stepping to the inlet port, having the largest differential pressure. This procedure minimized the
offset inherent in the movement of the free surface of liquid between the manometer tubing and the valve/transducer interface. The first position of the valve was open to the atmosphere so that each cycle of pressure readings could re-establish an atmospheric (zero) reference pressure. Pressure readings were corrected for the slight displacement of the water-air interface in the manometer tubing by a calculation feature in the Workbench software. In addition, a time-averaging scheme in this software was used to minimize small fluctuations in pressure caused by the pump prior to capturing the data. The valve cycle was set to wait for one minute at each port, resulting in a total cycle period of 16 minutes.

Data acquisition and recording were performed using the software application, Workbench (Omega Engineering Corp., Stamford, CT), running on an Apple Macintosh SE computer. This software also provided the stepping control of the multi-position valve and the synchronizing signal to identify which location in the column corresponded to the pressure data that were recorded. Fig. 15 shows a typical display of the pressure data captured during several complete sequences of valve operation during a flow experiment. The stepped line indicates the values of pressure at various locations in the column. The axial separation of pressure monitoring ports along the column is not necessarily equal between all of these steps. It is evident that the small pressure pulses created by the pump are very well damped.

Aqueous suspensions of particles of kaolin clay in a buffer solution were prepared for the filtration experiments. The constituents were kaolin powder, (Mallinkrodt
Chemical Co., St. Louis, MO, food grade) which is hydrated aluminum silicate having the approximate formula $\text{H}_2\text{Al}_2\text{Si}_2\text{O}_8 - X \text{H}_2\text{O}$. Aluminum sulfate, $\text{Al}_2(\text{SO}_4)_3 - X \text{H}_2\text{O}$, or "alum" (Mallinkrodt Chemical Co., St. Louis, MO, reagent grade) was used as the coagulant for the suspended clay particles. Sodium bicarbonate and potassium chloride were added to buffer the suspension to near a neutral $\text{pH}$ and to provide ionic strength. The standard suspension was prepared in deionized water to produce final concentrations of 30 mg/l kaolin, 10 mg/l alum, 50 mg/l KCl and 42 mg/l NaHCO$_3$ [45]. Typically, concentrated solutions or suspensions of the individual constituents were gradually added to deionized water before an experiment and stirred strongly by mechanical or ultrasonic agitation for several minutes. The suspension in the reservoir was continuously stirred during the flow tests, however, the suspended particles required several hours to settle out when stagnant. Several filtration experiments were conducted using suspensions in which the concentration of the constituents was doubled.

Flow experiments were initiated using clean water, pumped at the same flow rate intended for the suspension. This enabled us to collect data for the hydraulic gradient of the clean packing and to verify that all the data channels were functioning properly. Flow of suspension into the column was started at the beginning of the switching cycle of the multi-position valve. The test was continued, replenishing the suspension reservoir as needed, for up to 9 hours.

EXPERIMENTAL RESULTS

Hydraulic gradients were determined using columns loaded with the various packing materials in order to verify that our measurements corresponded to those reported by other workers and to ensure that the experimental system was functioning properly. Flow rates were varied over a wide range to ensure the consistency of the measured permeabilities. The data for hydraulic gradients measured for flow of water in clean columns is presented in Table 2. The hydraulic gradient is given in units of hydrostatic head (inches of water) per unit length of packing (inches) and is nominally dimensionless [48]. The filtration, or Darcy, velocity was calculated based on the volumetric flow rate and the void fraction of a given packed column.

These data are also plotted in Fig. 16, in which the measurements done in this work, represented by solid symbols, are compared with data from the literature, which are shown as open symbols. A direct comparison can be made of our data for a large packing material (1400-1700 $\mu$m) and the data of Darby, et al [47] using the same glass beads. These data agree very well. Similarly, Hunt's data [45] for a slightly smaller packing (1190-1400 $\mu$m) display a somewhat greater hydraulic gradient at a given filtration velocity compared to the largest packing, as expected. Extrapolating Hunt's data for a smaller packing (495-589 $\mu$m) and the data from this work for Ottawa sand (600-850 $\mu$m) also yields good agreement. Fig. 16 illustrates
that the hydraulic gradients determined in this work were considerably smaller than those reported in the literature. The experimental apparatus used in our work is much more sensitive to small differential pressure changes during flow than the prior studies.

Several experiments were conducted to verify the ability of the experimental apparatus to measure the increases in local hydraulic gradients that result as fine particles deposited within the column. The parameters varied were the size of the packing material, flowrate, and concentration of the suspended particles. The smallest packing material, P-0170 glass beads (see Table 1), collected the suspended fines very effectively, resulting in relatively large pressure drops between the inlet region and first (1-cm) monitoring port. The standard suspension concentration tended to clog this packing material quite readily, even at flowrates of only 40 ml/min. The suspension was not distributed over a sufficient length of the column to permit meaningful determinations of the hydraulic gradient in this case. Conversely, the largest packing material, A-150 glass beads, did not collect enough suspended

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Diameter (μm)</th>
<th>Velocity (cm/s)</th>
<th>Gradient</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-0170 Beads</td>
<td>300-450</td>
<td>0.017</td>
<td>0.195</td>
</tr>
<tr>
<td></td>
<td>300-450</td>
<td>0.026</td>
<td>0.281</td>
</tr>
<tr>
<td></td>
<td>300-450</td>
<td>0.034</td>
<td>0.381</td>
</tr>
<tr>
<td></td>
<td>300-450</td>
<td>0.043</td>
<td>0.490</td>
</tr>
<tr>
<td>Ottawa Sand</td>
<td>600-850</td>
<td>0.034</td>
<td>0.110</td>
</tr>
<tr>
<td></td>
<td>600-850</td>
<td>0.051</td>
<td>0.161</td>
</tr>
<tr>
<td></td>
<td>600-850</td>
<td>0.068</td>
<td>0.202</td>
</tr>
<tr>
<td>A-150 Beads</td>
<td>1400-1700</td>
<td>0.034</td>
<td>0.026</td>
</tr>
<tr>
<td></td>
<td>1400-1700</td>
<td>0.051</td>
<td>0.037</td>
</tr>
<tr>
<td></td>
<td>1400-1700</td>
<td>0.068</td>
<td>0.047</td>
</tr>
<tr>
<td></td>
<td>1400-1700</td>
<td>0.085</td>
<td>0.057</td>
</tr>
<tr>
<td>Darby [47]</td>
<td>1400-1700</td>
<td>0.180</td>
<td>0.1</td>
</tr>
<tr>
<td>Darby [46]</td>
<td>500-600</td>
<td>0.170</td>
<td>0.57</td>
</tr>
<tr>
<td>Hunt [45]</td>
<td>495-589</td>
<td>0.145</td>
<td>0.38</td>
</tr>
<tr>
<td></td>
<td>495-589</td>
<td>0.279</td>
<td>0.73</td>
</tr>
<tr>
<td></td>
<td>1190-1400</td>
<td>0.145</td>
<td>0.12</td>
</tr>
<tr>
<td></td>
<td>1190-1400</td>
<td>0.279</td>
<td>0.23</td>
</tr>
<tr>
<td>Hunt [45]</td>
<td>1190-1400</td>
<td>0.554</td>
<td>0.46</td>
</tr>
</tbody>
</table>

Table 2. Hydraulic gradients measured for flow of water in clean packed columns.
finishes during experiments lasting up to 10 hours to permit accurate measurements of the increases in hydraulic gradients. Even at the highest practical flowrate of 200 ml/min using a double concentration of fines did not deposit sufficient particulate for useful measurements.

Flow experiments with A-055 glass beads and Ottawa sand provided good examples of the utility of the experimental apparatus to measure hydraulic gradients during deposition of fines. The plot in Fig. 17 shows the increases in hydraulic gradient resulting during flow of suspended fines in the A-055 material. The units of hydraulic gradient are cm of water (head) per cm of column length in this plot. The ordinate indicates the axial position above the screen that supports the packing material, in centimeters. The filtration velocity was 0.085 cm/sec, which corresponds to a volumetric flowrate of 100 ml/min, and the concentration of the suspension was twice that specified in procedural section above. The dotted line indicates the mean hydraulic gradient for flow of water in a clean packed bed of A-055 glass beads at this filtration velocity. The values of the hydraulic gradient obtained for the upper part of the column (more than about 18 cm from the support screen) are essentially those of a clean column regardless of time. This result is expected as visual inspection confirmed that very few, if any, suspended particles appeared at the exit end of the packed bed. This result is also consistent with published reports that indicate that fines collect preferentially in the inlet section of a packed bed.
Figure 17. Observed increases in hydraulic gradient resulting from deposition of fines in a packed bed.

The data points and associated lines in Fig. 17 indicate that the hydraulic gradient increased as the experiment progressed and material collected preferentially in the entry region of the bed. The hydraulic gradient decreased with distance further from the inlet area, as expected. The anomaly in the data at positions 4 and 5 cm from the inlet could have been caused by clogging of the monitoring ports. A localized nonuniformity in the packing of the column may also have contributed to these anomalous results. Regardless, the expected trends in the data with regard to time and position in the bed were observed. When related to a mass balance of the total flux of fine particles into the column, these data are also consistent with literature reports, e.g. [45], that relatively small amounts of deposited particles can increase local hydraulic gradients by factors of 5 or more compared to the permeability of a clean column.
SUMMARY

The subsurface transport of nutrients and microorganisms is a key factor in determining the applicability of in situ bioremediation to waste site decontamination. One of the most important and least understood aspects of this transport process is the dramatic reduction in formation permeability resulting from the growth and accumulation of biomass. These processes, which may reduce permeabilities by three to four orders of magnitude, are strongly dependent on microstructural features of the host medium. Such features, including the pore size distribution and the degree of pore connectivity, cannot be explicitly accounted for in traditional site-scale continuum models. Instead, these processes must be investigated on scales comparable to those of the interstitial pore diameters.

To better understand the processes of transport and permeability reduction, we have constructed mathematical and laboratory models applicable to intermediate scales. These scales are large enough to encompass hundreds or thousands of pores, but still small compared to field operations. The mathematical model depicts flow and transport through a network of interconnected passages representing the pore volume of a granular material. This network model is sufficiently general to accommodate a broad range of transport processes in fully disordered materials, yet simple enough to permit derivation of closed-form relationships describing the permeability and effective diffusivity of regular particle arrays.

In contrast to most previous network models that simply interconnect randomly-placed tubes, our computational domain is generated by packing spheres or cylinders of a specified size distribution. This approach provides a more realistic representation of granular materials, soils, and sedimentary geologic media. Such attention to microstructural detail is especially important in modeling deposition processes, since reductions in permeability are very sensitive to pore geometry. The packing may be regular or may be constructed by one of several methods using statistical means to obtain varying degrees of disorder and varying packing densities. After fixing the size and location of all particles, a Voronoi tessellation is used to define the centerlines and junctions of the channels forming the network. The aperture of each channel varies with axial position along its length in accordance with the geometry of the bounding solid surfaces. To speed numerical solutions, analytical integration along each channel axis is used to define effective apertures used in calculating the pore volume, fluid speed, transit time, and the cross-sectional area for diffusion.

Although the network model is mainly intended for numerical simulation of transport in random packings, it can also be used to derive analytical expressions relating effective transport properties to the pore geometry. Closed-form expressions were derived for the effective permeability and effective diffusivity for transport through a hexagonal array of circular cylinders. These analytical results are
in good agreement with published solutions to the multidimensional equations describing Stokes flow and binary diffusion through a unit cell of the medium. Similar agreement has been obtained in applying this simplified version of the network model to more complex two and three-dimensional ordered media. In all cases, the effective transport properties are related to the fundamental geometric parameters used to characterize all permeable materials, including particle size, porosity, tortuosity, and the percolation threshold. In addressing random media and the more complex processes like dispersion and plugging, it is generally necessary to perform numerical simulations using the full capabilities of the model.

An important and powerful feature of this network model is its capability to simulate particle transport. Fictitious tracer particles can be used to compute effective diffusivities and dispersivities of disordered materials. In addition, particles of finite size can be used to simulate the transport and deposition of fines and the effect of such deposition on permeability. To compute dispersivities, particles are injected on the inlet boundary and transported through the network at the local fluid speed. During each time step of this deterministic motion, the tracer particles are also displaced by a random motion, simulating the effects of diffusion between adjacent streamlines within the pore volume. After a specified interval, the spatial distribution of the particle positions is fit with an error function or Gaussian profile to obtain the mean and variance of the particle positions. From these values, both the longitudinal and lateral dispersivity can be computed. In contrast to most previous work in which streamlines are assumed well mixed at each branching and confluence of channels, the present algorithm maps each streamline through these junctions. In the absence of diffusion it is therefore possible to trace each streamline within the network between the inlet and outlet boundaries. This avoids the artificial dispersion that can sometimes arise in network models.

Using this network model, permeability reductions during bioremediation can be simulated by computing a sequence of solutions for a fixed particle array in which particles of the array are coated by a layer of increasing thickness. In sample calculations presented here, we found that a reduction in porosity from 60% to 40% as a result of layer growth led to more than an order of magnitude reduction in permeability. Further reduction in the porosity to 20% led to a network below the percolation threshold. Such a dramatic reduction in permeability is not a direct consequence of narrowing in the larger channels, but rather results from a loss of interconnection within the pore network by the occlusion of many smaller passages.

Other sample calculations of flow in random granular materials revealed several interesting phenomena. In a rectangular domain having an imposed one-dimensional pressure gradient, we found regions in which the local pressure gradient was reversed and the direction of local flow opposed that of the mean pressure gradient. We also saw that the magnitude of the associated lateral pressure gradients were nearly 20% of that in the direction of mean flow. These deviations from the one-dimensional
continuum behavior are a consequence of disorder and heterogeneity. Such large variations in the pressure and velocity fields contribute substantially to macro-scale dispersion.

Experimental studies were conducted to investigate permeability reductions by the accumulation of fines. The apparatus developed for this purpose demonstrated the capability to make measurements of small, local pressure variations within a packed column at relatively low flow rates. This apparatus also enabled improved spatial resolution of permeability reductions within a column, as compared with previous systems described in the literature. Experiments with flow of clean water verified that our results agreed well with published data. Similarly, an experimental capability to measure reductions in permeability due to the deposition of fines during the flow of suspensions was demonstrated. In this series of experiments, permeability reductions were measured over a range of conditions in which the flow rate, properties of the fines suspension, and size and size distribution of the granular packing materials were varied.

One important goal in this program was to compare the measured and calculated permeability reductions for both fines deposition and biofilm growth. Out of this comparison, we intended to benchmark the simulations and develop constitutive relations describing permeability reductions. This goal was not achieved. Due to problems in writing SOPs for biologically active agents in the wake of the DOE Tiger Team visit and due to the relocation of personnel and the apparatus to the new IMTL building, the schedule for experimental work slipped significantly. This did not allow time, within the span of the program, to achieve these last goals. We have attempted to complete this portion of the work since that time, but this also has not been successful.

Despite these problems, we have already made use of several of the techniques and capabilities developed here in areas unrelated to bioremediation. The permeability and effective diffusivity of ordered and nearly-ordered carbon fiber arrays have been computed in support of an ARPA-funded program on the rapid densification of carbon-carbon composites. This manufacturing process shares with bioremediation the problem of formation blocking, leading to long processing times and very high cost. The Voronoi tessellation algorithm developed for generating our tube network has also been used to generate a structural skeleton for modeling the solid mechanics of propellant fragmentation in an MOU-funded study of propellant recovery by temperature cycling and for modeling the solid mechanics of foams in a Sandia CRADA with Dow Chemical Corporation. Finally, the network model is now being considered as a platform for modeling the transport and adsorption in micro-porous materials used as gas separation and storage devices.
ACKNOWLEDGMENT

The authors wish to thank R. S. Larson for his assistance in integrating several of the expressions used in computing effective transport properties. The assistance of B. C. Long in constructing and operating the experimental apparatus is also gratefully acknowledged.

REFERENCES


APPENDIX A - SOURCE LISTING OF PROGRAM BIOREM

BIOREM MAIN

00001 C  DIMENSION X0(500), Y0(500), SL(500)
00002 C  DIMENSION XX(131), YY(131), DDX(500), DDDY(500)
00003 C  DIMENSION XS(500),YS(500), IFRM(12,500), LIFRM(12,500)
00004 C  DIMENSION RADI(500), DIAM(500), HITE(500), DIAMO(500), HITEO(500)
00005 C  REAL LBON(12,500), LSID(12,500), AREA(500)
00006 C  DIMENSION XSID(12,500), YSID(12,500), IDONE(500)
00007 C  DIMENSION FERI(500), APRX(500), APRY(500)
00008 C  REAL LOADR, LOADL, LOADT, LOADE, LOADD, LOADY, LOADX, LOADY
00009 C  INTEGER FSID(20), JB(200), NSID(500), NORM(4)
00010 C  INTEGER IDNOD(12,500), NODAR(11,1500), IDTUB(12,500), TUENOD(4,1500)
00011 C  REAL XNOD(1500), YNOD(1500), TUBAR(15,1500), FLO(3), APROE(1500)
00012 C  REAL XBA1(2), YBA1(2), XBA2(2), YBA2(2), XB(500), YB(500)
00013 C  REAL FREQ(500), XNPAR(4,1500)
00014 C  REAL DNEIGH(18,500), DBOX(500)
00015 C  REAL DEER(1500), DDIF(1500), DVOL(1500), ARTIM(1500)
00016 C  REAL XX(3,3), RHS(3), XTU(1500), YTU(1500), RTU(1500), QHS(3)
00017 C  REAL XHES(3)
00018 C  INTEGER DETA, DEX, DEY, DETH
00019 C  INTEGER INEIGH(18,500), JBLOK(1500)
00020 C  REAL*8 PRES(1500), PRESO(1500), DPRES(1500), DOTM, FSUM
00021 C  REAL DOTM, DV(1), DXY(1)
00022 C  REAL DXX(1), DYY(1)
00023 C  REAL XNPAR(4,1500)
00024 C  REAL DNEIGH(18,500), DBOX(500)
00025 C  INTEGER IDNOD(12,500), NODAR(11,1500), IDTUB(12,500), TUENOD(4,1500)
00026 C  REAL XNOD(1500), YNOD(1500), TUBAR(15,1500), FLO(3), APROE(1500)
00027 C  REAL XBA1(2), YBA1(2), XBA2(2), YBA2(2), XB(500), YB(500)
00028 C  REAL FREQ(500), XNPAR(4,1500)
00029 C  REAL DNEIGH(18,500), DBOX(500)
00030 C  REAL DEER(1500), DDIF(1500), DVOL(1500), ARTIM(1500)
00031 C  REAL XX(3,3), RHS(3), XTU(1500), YTU(1500), RTU(1500), QHS(3)
00032 C  INTEGER DETA, DEX, DEY, DETH
00033 C  INTEGER INEIGH(18,500), JBLOK(1500)
00034 C  REAL*8 PRES(1500), PRESO(1500), DPRES(1500), DOTM, FSUM
00035 C  REAL DOTM, DV(1), DXY(1)
00036 C  REAL DXX(1), DYY(1)
00037 C  REAL DNEIGH(18,500), DBOX(500)
00038 C  INTEGER IDNOD(12,500), NODAR(11,1500), IDTUB(12,500), TUENOD(4,1500)
00039 C  REAL*8 PRES(1500), PRESO(1500), DPRES(1500), DOTM, FSUM
00040 C  REAL DOTM, DV(1), DXY(1)
00041 C  REAL DXX(1), DYY(1)
00042 C  REAL DNEIGH(18,500), DBOX(500)
00043 C  INTEGER IDNOD(12,500), NODAR(11,1500), IDTUB(12,500), TUENOD(4,1500)
00044 C  REAL*8 PRES(1500), PRESO(1500), DPRES(1500), DOTM, FSUM
00045 C  REAL DOTM, DV(1), DXY(1)
00046 C  REAL DXX(1), DYY(1)
00047 C  REAL DNEIGH(18,500), DBOX(500)
00048 C  INTEGER IDNOD(12,500), NODAR(11,1500), IDTUB(12,500), TUENOD(4,1500)
00049 C  REAL*8 PRES(1500), PRESO(1500), DPRES(1500), DOTM, FSUM
00050 C  REAL DOTM, DV(1), DXY(1)
00051 C  REAL DXX(1), DYY(1)
00052 C  REAL DNEIGH(18,500), DBOX(500)
00053 C  INTEGER IDNOD(12,500), NODAR(11,1500), IDTUB(12,500), TUENOD(4,1500)
00054 C  REAL*8 PRES(1500), PRESO(1500), DPRES(1500), DOTM, FSUM
00055 C  REAL DOTM, DV(1), DXY(1)
00056 C  REAL DXX(1), DYY(1)
00057 C  REAL DNEIGH(18,500), DBOX(500)

A1
4:  Y LOC OF TUBE END 1
5:  X LOC OF TUBE END 2
6:  Y LOC OF TUBE END 2
7:  X SHIFT IN TUBE POSITION
8:  Y SHIFT IN TUBE POSITION
9:  BOUNDARY NUMBER OF TUBE IDT
10: FLUID VELOCITY
11: MASS FLOW RATE
12: TUBE VOLUME

XNOD (IDN) AND YNOD (IDN)
1: X AND Y COORDINATES OF NODE IDN

IDNOD (L,ID)
1: NODE NUMBER ASSIGNED WITH SIDE L OF SEED ID

IDTHUB (L,ID)
1: TUBE NUMBER OF SIDE L OF SEED ID

TUBNOD (K,IDT)
1: NODE NUMBER AT TUBE END 1
2: NODE NUMBER AT TUBE END 2
3: SEED NUMBER OF 1ST PARTICLE
4: SEED NUMBER OF 2ND PARTICLE

PI = 4.*ATAN(1.)
TWOPI = 2.*PI
RT3 = SQRT(3.)
RT2 = SQRT(2.)
ISED = 111311
PRINT*,4.*ATAN(1.),4.*ATAN(-1.)

ICONI' = 0

Do 1480 ID=1,500
D3
L=1,12
IDNOD(L,ID) = O

Do 1404 IDN=1,1500
NODAR(1,IDN) = O

GMN=0.
C MAX=0.
00116  CMAX=0.
00117  CMAX=0.
00118  CMAX=0.
00119  CMAX=0.
00120  CMAX=0.
00121  CMAX=0.
00122  CMAX=0.
00123  CMAX=0.
00124  CMAX=0.
00125  CMAX=0.
00126  CMAX=0.
00127  CMAX=0.
00128  CMAX=0.
00129  CMAX=0.
00130  CMAX=0.
00131  CMAX=0.
00132  CMAX=0.
00133  CMAX=0.
00134  CMAX=0.
00135  CMAX=0.
00136  CMAX=0.
00137  CMAX=0.
00138  CMAX=0.
00139  CMAX=0.
00140  CMAX=0.
00141  CMAX=0.
00142  CMAX=0.
00143  CMAX=0.
00144  CMAX=0.
00145  CMAX=0.
00146  CMAX=0.
00147  CMAX=0.
00148  CMAX=0.
00149  CMAX=0.
00150  CMAX=0.
00151  CMAX=0.
00152  CMAX=0.
00153  CMAX=0.
00154  CMAX=0.
00155  CMAX=0.
00156  CMAX=0.
00157  CMAX=0.
00158  CMAX=0.
00159  CMAX=0.
00160  CMAX=0.
00161  CMAX=0.
00162  CMAX=0.
00163  CMAX=0.
00164  CMAX=0.
00165  CMAX=0.
00166  CMAX=0.
00167  CMAX=0.
00168  CMAX=0.
00169  CMAX=0.
00170  CMAX=0.
00171  CMAX=0.

BIOREMSMAIN

00115  CMAX=0.
00116  CMAX=0.
00117  CMAX=0.
00118  CMAX=0.
00119  CMAX=0.
00120  CMAX=0.
00121  CMAX=0.
00122  CMAX=0.
00123  CMAX=0.
00124  CMAX=0.
00125  CMAX=0.
00126  CMAX=0.
00127  CMAX=0.
00128  CMAX=0.
00129  CMAX=0.
00130  CMAX=0.
00131  CMAX=0.
00132  CMAX=0.
00133  CMAX=0.
00134  CMAX=0.
00135  CMAX=0.
00136  CMAX=0.
00137  CMAX=0.
00138  CMAX=0.
00139  CMAX=0.
00140  CMAX=0.
00141  CMAX=0.
00142  CMAX=0.
00143  CMAX=0.
00144  CMAX=0.
00145  CMAX=0.
00146  CMAX=0.
00147  CMAX=0.
00148  CMAX=0.
00149  CMAX=0.
00150  CMAX=0.
00151  CMAX=0.
00152  CMAX=0.
00153  CMAX=0.
00154  CMAX=0.
00155  CMAX=0.
00156  CMAX=0.
00157  CMAX=0.
00158  CMAX=0.
00159  CMAX=0.
00160  CMAX=0.
00161  CMAX=0.
00162  CMAX=0.
00163  CMAX=0.
00164  CMAX=0.
00165  CMAX=0.
00166  CMAX=0.
00167  CMAX=0.
00168  CMAX=0.
00169  CMAX=0.
00170  CMAX=0.
00171  CMAX=0.

BIOREMSMAIN

A3
C

DELDEL = 0.

IREDU = 1

PRINT*, 'INPUT IREDU,DELDEL', IREDU, DELDEL

READ*, IREDU, DELDEL

SWELL = 0.

PRINT*, 'INPUT SWELLING FRACTION', SWELL

READ*, SWELL

IK=0

IF (IK.EQ.0) GO TO 1519

C

ELSEIF (IMETH .EQ. 99) THEN

DO 693 ID = 1,NPAR

READ(99,*), XS(ID), YS(ID), HITE(ID), DIAM(ID), RADI(ID)

693 CONTINUE

ELSEIF (IMETH .EQ. 1) THEN

FAC=1

NSEED=150

PRINT*, 'INPUT NSEED, ISED', NSEED, ISED

READ*, NSEED, ISED

ISEED=ISED

DSAT = 1.

PRINT*, 'INPLPT DSAT', DSAT

READ*, DSAT

DSMIN = ABS(DSAT) * 0.80 / SQRT(1. * NSEED)

DEF = DMIN

POINO = 0.4

ALOGSIG = 1. E-8

PRINT*, 'INPUT POINO, SIG', POINO, ALOGSIG

READ*, POINO, ALOGSIG

DELDEL = 0.

IREDU = 1

PRINT*, 'INPUT IREDU,DELDEL', IREDU, DELDEL

READ*, IREDU, DELDEL

SWELL = 0.

PRINT*, 'INPUT SWELLING FRACTION', SWELL

READ*, SWELL

IK=0

PRINT*, 'INPUT OK: 0 = NO 1 = YES', IOK

READ*, IOK

IF (IK.EQ.0) GO TO 1519

KSD = 0

SUMP = 0.

NBSUM = 0

IF (DSAT.EQ.1) NSEED=50*NSEED

DO 1010 I=1, NSEED

1010 IDBC(I) = 0

ITRY = 0

DONE(I) = 0

ALOGMU = ALOG(DSMIN/2.)
00229        ZZZ=RAN(ISED)
00230        CALL ERRINV(ALOGR)
00231        DIAM(I)=2.*EXP(ALOGR)
00232 C
00233  1013 XS(I)=XMAX*RAN(ISED)
00234         YS(I)=YMAX*RAN(ISED)
00235         HITE(I)=DIAM(I)/2. *(1.-2.*RAN(ISED))
00236         HMNN = MAX (HITE(I), DMIN/50.)
00237         IF (HITE(I) .LT. 0.) THEN
00238             HITE(I) = MIN (HITE(I), -HMNN)
00239             ELSE
00240                 HITE(I) = MAX (HITE(I), HMNN)
00241             ENDIF
00242 C
00243         RADI(I)=SQRT( (DIAM(I)/2.)**2 - HITE(I)**2 )
00244         IF (DSAT.GT.1.E-6) THEN
00245             RPART=RADI(I)
00246             DELSO=RPART-XS(I)
00247             DELSO=MAX(DELSO,XS(I)+RPART-XMAX)
00248             DELSO=MAX(DELSO,RPART-YS(I))
00249             DELSO=MAX(DELSO,YS(I)+RPART-YMAX)
00250             IF (DELSO.GT.0.0) THEN
00251                 ITRY=ITRY+1
00252                 IF (ITRY.GT.MAX(100,200*KSEED)) GO TO 1014
00253                 GO TO 1013
00254             ENDIF
00255             EPSR2 = (DSMIN /100.)***2
00256             DO 1012 J=1,I-1
00257                 DELX2=(XS(I)-XS(J))**2
00258                     DELY2=(YS(I)-YS(J))**2
00259                 DELH2=(HITE(I)-HITE(J))**2
00260                 DELR2=((DIAM(I)+DIAM(J))/2.)**2
00261                 IF (DELX2+DELY2+DELH2 .LT. DELFQ+EPS2 ) THEN
00262                     ITRY=ITRY+1
00263                     IF (ITRY.GT.MAX(100,200*KSEED)) GO TO 1014
00264                 GO TO 1013
00265                 ENDIF
00266  1012 CONTINUE
00267 C
00268         EPS2 = (DSMIN /100.)***2
00269         DO 1012 J=1,I-1
00270                 DELX2=(XS(I)-XS(J))**2
00271                 DELY2=(YS(I)-YS(J))**2
00272                 DELH2=(HITE(I)-HITE(J))**2
00273                 DELR2=((DIAM(I)+DIAM(J))/2.)**2
00274                 IF (DELX2+DELY2+DELH2 .LT. DELFQ+EPS2 ) THEN
00275                     ITRY=ITRY+1
00276                 GO TO 1013
00277         ENDIF
00278         SUMP=SUMP+3.14*RADI(I)**2
00279 C
00280         SUMP=SUMP+3.14*RADI(I)**2
00281         PORO=1.-SUMP/(XMAX*YMAX)
00282         IF (PORO.LT.POROIN) GO TO 1014
00283 C
00284  1016 KSEED=KSEED+1
00285  1010 CONTINUE
00286  1014 NSEED=KSEED
00287 C
00288         ELSEIF (IMETH.EQ.2) THEN
00289             NSEED = 150
00290             PRINT*,"INPUT NSEED, ISED",NSEED,ISED
00291         ENDIF
00292         NSEED = ISED
00293         DSMIN = SQRT(XMAX**YMAX) *0.80/SQRT(1.*NSEED)
00294         DREF = DSMIN
00295         ALOGSIG = 1.E-8
00296         PRINT*,"INPUT SIG",ALOGSIG
00297         BIOREM$MAIN
00298
00299        READ*,ALOGSIG

A5
DELDIA = 0.
IREDU = 1
PRINT*, 'INPUT IREDU, DELDIA', IREDU, DELDIA
READ*, IREDU, DELDIA
SWELL = 0.
PRINT*, 'INPUT SWELLING FRACTION', SWELL
READ*, SWELL
IOK = 0
PRINT*, 'INPUT OK: 0 = NO 1 = YES', IOK
READ*, IOK
IF (IOK.EQ.0) GO TO 1519
C
KSEED = 0
SUMP = 0.
NBSUM = 0
NSEED = 10 *NSEED
NDIVX = 20 *XMAX /DSMIN
NDIVY = 20 *YMAX /DSMIN
NDIV = MAX (NDIVX, NDIVY)
NACROS = XMAX /DSMIN
EPS = 1.1 *XMAX / (NDIV + 1.)
PRINT*, NDIVX, NDIVY, EPS
YTOP = 0.
DO 2010 I = 1, NSEED
IDE = 0.
DONE(I) = 0
C
ALOGMU = ALOG (DSMIN/2.)
ZZZ = RAN (ISED)
CALL ERRINV (ALOGR)
DIAM(I) = 2. * EXP (ALOGR)
HITE(I) = DIAM(I) / 2. * (1. - 2. * RAN (ISED))
RADI(I) = SQRT ( (DIAM(I)/2.)**2 - HITE(I)**2 )
BOTF = 2. * RAN (ISED)
YMIN = YMAX
RPART = IFILL * RADI(I)
DO 2022 II = 1, NDIVX
XST = (XMAX * II) / (NDIVX + 1.)
DELEBO = XST - RPART
DELEBO = MIN (DELEBO, XMAX - XST - RPART)
IF (DELEBO .LT. EPS) GO TO 2022
DO 2030 JJ = 1, NDIVY
YST = YMAX * (1. - (1. * JJ) / (NDIVY + 1.))
IF (YST .GT. YTOP + DEMFXO * DDMIN) GO TO 2023
DELEBO = YST - RPART
IF (IFILL .EQ. 0) DELEBO = YST - RADI(I) * BOTF
IF (DELEBO .LT. EPS) THEN
YMIN = YST
XS(I) = XST + 1.E-2 * RADI(I)
YS(I) = YST + 1.E-2 * RADI(I)
GO TO 2022
ENDIF
JJJMKX = MIN (I - 1, 4 * NACROS)
DO 2012 JJJ = 1, JJJMKX

BIOREMSMAIN

J = I - 1 - JJJ + 1
DELEX2 = (XST - XS(J))**2

A6
00345  DELX2 = (YST-YS(J))*2
00346  DELH2 = (HITE(I)-HITE(J))*2
00347  DELR2 = ((DIAM(I)+DIAM(J))/2.)*2
00348  DIST = SQRT (DELX2+DELY2+DELH2) - SQRT (DELR2)
00349  IF (DIST .LT. EPS) THEN
00350       IF (YST .LT. YMIN) THEN
00351          YMIN = YST
00352          XS(I) = XST + 1.E-2*RADI(I)
00353          YS(I) = YST + 1.E-2*RADI(I)
00354       ENDIF
00355  GO TO 2022
00356  ENDIF
00357  2012 CONTINUE
00358  2023 CONTINUE
00359  2022 CONTINUE
00360  YTOP = YS(I)
00361 C
00362   PRINT1901,I,XS(I),YS(I),RADI(I)
00363   DELBO = YMAX-YS(I)-RPART
00364   IF (DELBO .LT. EPS) GO TO 2014
00365   KSEED = KSEED + 1
00366   SUMP = SUMP + PI*RADI(I)*2
00367   PORO = 1.-SUMP/(XMAX*YMAX)
00368 C
00369  2010 CONTINUE
00370  2014 NSEED = KSEED
00371   PRINT*,NSEED
00372   PRINT*,PORO
00373   READ*, DDXX
00374 C
00375 ELSEIF (IMETH.EQ.5) THEN
00376    NSEED = 150
00377    PRINT*, 'INPUT ISED',ISED
00378    READ*, ISED
00379    ISED0 = ISED
00380    PRINT*, 'INPUT FIBER DIAM AND POROISTY',DIAMF,EPSO
00381    READ*, DIAMF,EPSO
00382    IRAG = 1
00383    PRINT*, 'DO YOU WANT A RAGGED BOTTOM: 0 = NO 1 = YES',IRAG
00384    READ*, IRAG
00385    DMIN = DIAMF
00386    DREF = DMIN
00387    DELDIA = 0.
00388    IREDU = 1
00389    PRINT*, 'INPUT IREDU,DELDIA', IREDU,DELDIA
00390    READ*, IREDU,DELDIA
00391    SWELL = 0.
00392    PRINT*, 'INPUT SWELLING FRACTION', SWELL
00393    READ*,SWELL
00394    IOK=0
00395    ACON = 1
00396    PRINT*, 'INPUT RANDOM VARIATION IN SPACING ( 0 TO 1) ',ACON
00397    READ*, ACON
00398    PRINT*, 'INPUT OK: 0 = NO 1 = YES',IOK
00399    READ*, IOK

BIOREMSMAIN
00400   IF (IOK.EQ.0) GO TO 1519
00401 C
00402   KSEED=0
00403  SEMP=O.
00404  NBSUM=O
00405  NSEED = 2 * XMAX *MAX / (PI/4. *DIAMF**2) *(1.-EPSO)
00406  NDIVX = 20 *MAX /DIAMF
00407  NDIVX = 20 *MAX /DIAMF
00408  NDIV = MAX (NDIVX, NDIVY)
00409  NACROS = XMAX /DSMIN
00410  TOL = 1.1 *MAX / (NDIVX+1.)
00411  RHO = 1. - EPSO
00412  DMIN = DIAMF /DSQR(2.*RT3/PI*RHO)
00413  PRINT*,NDIV,NDIV,TOL
00414  YTOP = 0.
00415  DO 5010 I = 1,NSEED
00416      IDBC(I) = 0
00417      DONE(I) = 0
00418      C
00419      DIAM(I)=DIAMF
00420      C
00421      HITE(I)=0.
00422      RADI(I)=SQRT( (DIAM(I)/2.)**2 - HITE(I)**2 )
00423      BOTF = 2.* RAN(ISED)
00424      SPAC = (1. + ACON*(1.-BOTF)) *(DSMIN - DIAMF)
00425      SPAC = MAX (0., SPAC)
00426      SPACM = SPAC + TOL
00427      RANX = (1. - BOTF) *SPACM /100.
00428      RANY = (1. - 2.*RAN(ISED)) *SPACM /100.
00429      C
00430      YMIN = MAX
00431      JMIND = 0
00432      RPART = IFILL *RADI(I)
00433      DO 5022 II = 1,NDIVX
00434      XST = (XMAX *II) / (NDIVX+1.)
00435      DELBO = XST - RPART
00436      DELBO = MIN (DELBO,MAX-XST-RPART)
00437      IF (DELBO .LT. TOL) GO TO 5022
00438      DO 5023 JJ = 1,NDIVY
00439      YST1 = YMAX *(1.- (1.*JJ) / (NDIVY+1.))
00440      NUMY = (YTOP+DSMIN+SPACM) *(NDIVY+1.) /YMAX + 1
00441      YST2 = YMAX *(NUMY-JJ+1.) / (NDIVY+1.)
00442      YST = MIN (YST1, YST2)
00443      DELBO = YST - RPART
00444      IF (IFILL .EQ. 0) DELBO = YST - RADI(I) *IRAG *BOTF
00445      IF ( (DELEBO .LT. TOL) THEN
00446      JYY = JJ
00447      YMIN = YST
00448      XS(I) = XST + RANX
00449      YS(I) = YST + RANY
00450      GO TO 5022
00451      ENDIF
00452      JJJJMX = MIN (I-1, 2 *NACROS)
00453      DO 5012 JJJJ = 1,JJJJMX
00454         J = I-1 - JJJJ + 1
00455         DELX2 = (XST-XS(J))**2
00456         DELY2 = (YST-YS(J))**2
00457  00457  00457
00458  BIOREMSMAIN
00459
00460
00460  00460  00460  00460  00460  00460

A8
IF (YST .LT. YMIN) THEN
    JJY = JJ
    YMIN = YST
    XS(I) = XST + RANX
    YS(I) = YST + RANY
    SPAC1 = DIST
    JMIN1 = J
ENDIF
GO TO 5022
ENDIF
GO TO 5022

**CONTINUE**
5012 CONTINUE
5023 CONTINUE
5022 CONTINUE

**FIND SECOND CLOSEST PARTICLE BELOW**

SMIN = YMAX
JMIN2 = 0
DO 5044 J = 1, I-1
    IF (J .EQ. JMIN1) GO TO 5044
    IF (YST(J) - YST .LT. 3.*TOL .AND. JMIN1.GT.0) GO TO 5044
    DELX2 = (XS(I) - XS(J))**2
    DELH2 = (HITE(I) - HITE(J))**2
    DELR2 = ((DIAM(I) + DIAM(J))/2.)***2
    DIST = SQRT (DELX2 + DELY2 + DELH2) - SQRT (DELR2)
    IF (DIST .LT. SMIN) THEN
        SMIN = DIST
        JMIN2 = J
    ENDIF
5044 CONTINUE
SPAC2 = SMIN

**MOVE FIBER TO EQUILIBRIUM POSITION**

BCON = TOL /10.

IF (JMIN1.EQ.0 .AND. JMIN2.EQ.0) THEN
    CC1 = 0.
    CC2 = 0.
    AA1 = 1.
    AA2 = 0.
    BB1 = 0.
    BB2 = 1.
ELSEIF (JMIN1.EQ.0 .AND. JMIN2.GT.0) THEN
    CC1 = (DIAM(I) + DIAM(JMIN2)/2. + SPAC + BCON)**2
    1 = (XS(I) - XS(JMIN2))**2 - (YS(I) - YS(JMIN2))**2
    CC2 = 0.
    AA1 = 2.* (XS(I) - XS(JMIN2))
    AA2 = YS(I) - YS(JMIN2)
    BB1 = 2.* (YS(I) - YS(JMIN2))
ELSEIF (JMIN1.GT.0 .AND. JMIN2.EQ.0) THEN
    CC1 = (DIAM(I)/2. + DIAM(JMIN1)/2. + SPAC + BCON)**2
    1 = (XS(I) - XS(JMIN1))**2 - (YS(I) - YS(JMIN1))**2
    CC2 = 0.
    AA1 = 2.* (XS(I) - XS(JMIN1))
    AA2 = YS(I) - YS(JMIN1)
    BB1 = 2.* (YS(I) - YS(JMIN1))

BIOREM$MAIN

BB2 = - (XS(I) - XS(JMIN2))
CC2 = 0. \\
AA1 = 2. *(XS(I)-XS(JMIN1)) \\
AA2 = 1. *(YS(I)-YS(JMIN1)) \\
BB1 = 2. *(YS(I)-YS(JMIN1)) \\
BB2 = -2. *(XS(I)-XS(JMIN1)) \\
ELSEIF (JMIN1.GT.0 .AND. JMIN2.GT.0) THEN \\
CC1 = (DIAM(I)/2. + DIAM(JMIN1)/2. + SPAC + BCON)**2 \\
CC2 = (DIAM(I)/2. + DIAM(JMIN2)/2. + SPAC + BCON)**2 \\
AA1 = 2. *(XS(I)-XS(JMIN1)) \\
AA2 = 2. *(XS(I)-XS(JMIN2)) \\
BB1 = 2. *(YS(I)-YS(JMIN1)) \\
BB2 = 2. *(YS(I)-YS(JMIN2)) \\
ENDIF \\
DLTX = (CC1*BB2-CC2*BB1) / (AA1*BB2-AA2*BB1) \\
DLTY = (CC2-AA2*DLTX) /BB2 \\
RLIM = (2.*TOL)**2 \\
IF (DLTX**2+DLTY**2 .GT. RLIM) THEN \\
DLTX = 0. \\
DLTY = 0. \\
ENDIF \\
XS(I) = XS(I) + DLTX \\
YS(I) = YS(I) + DLTY \\
XS(I) = MAX (XS(I), TOL+RANx) \\
XS(I) = MIN (XS(I), XMAX-TOL-RANx) \\
YS(I) = MAX (YS(I), TOL+RANy) \\
YS(I) = MIN (YS(I), YMAX-TOL-RANY) \\
YTOP = MAX (YS(I), YTOP) \\
DELT = MAX (YS(I), YTOP) \\
DELBO = YMAX-YS(I)-RPAR \\
IF (DELBO .LT. TOL) GO TO 5014 \\
PRINT1903, I, JMIN1, JMIN2, XS(I), YS(I), RADI(I), SPAC, DLTX, DLTY \\
KSEED = KSEED + 1 \\
SUMF = SUMF + PI *RADI(I)**2 \\
POR = 1. -SUMF/(XMAX*YMAX) \\
CONTINUE \\
ELSEIF (IMETH.EQ.3) THEN \\
NSEED = 150 \\
PRINT*, 'INPUT NSEED, ISED', NSEED, ISED \\
READ*, NSEED, ISED \\
ISED0 = ISED \\
BIOREG$MAIN \\
DSMIN = SQRT(XMAX*YMAX) *0.80/SQRT(1.*NSEED) \\
DREF = DMIN \\
ALOSIG = 1.8-8 \\
PRINT*, 'INPUT SIG', ALOSIG \\
READ*, ALOSIG \\
DELIA = 0. \\
A10
IREW = 1
PRINT*, 'INPUT IREW, DELDIA', IREW, DELDIA
READ*, IREW, DELDIA
SWELL = 0.
PRINT*, 'INPUT SWELLING FRACTION', SWELL
READ*, SWELL
IOK = 0
PRINT*, 'INPUT OK: 0 = NO 1 = YES', IOK
READ*, IOK
IF (IOK .EQ. 0) GO TO 1519
C
KSEED = 0
SUMP = 0.
NBSUM = 0
NSEG = 10 * NSEG
NDIVX = 10 * XMAX / DMIN
NDIVY = 10 * YMAX / DMIN
NDIVZ = 10
NDIV = MAX (NDIVX, NDIVY)
NACROS = XMAX / DMIN
EPS = 1.1 * XMAX / (NDIVX + 1.)
PRINT*, NDIVX, NDIVY, EPS
YTOP = 0.
DO 3010 I = 1, NSEG
IDEC(I) = 0
DONE(I) = 0
C
A = ASEG = (DMIN / 2.)
ZZZ = RAN (ISED)
CALL ERRINV (ALGCR)
DIAM(I) = 2. * EXP (ALGCR)
BOTF = 2. * RAN (ISED)
C
YMIN = YMAX
DO 3022 II = 1, NDIVX
XST = (XMAX * II) / (NDIVX + 1.)
DELEO = XST - RPART
DELEO = MIN (DELEO, XMAX - XST - RPART)
IF (DELEO .LT. EPS) GO TO 3022
DO 3021 KK = 1, NDIVZ
HIT = DIAM(I) / 2. * (1. - 2. * (1. * KK) / (NDIVZ + 1.))
IF (1. * I .LT. NACROS) HIT = DIAM(I) / 2. * (1. - 2. * RAN (ISED))
RAD = SQRT ( (DIAM(I) / 2.) ** 2 - HIT ** 2)
RPART = IFILL * RAD
DO 3023 JJ = 1, NDIVY
YST = YMAX * (1. - (1. * JJ) / (NDIVY + 1.))
IF (YST .GT. YTOP + DEMAXO * DMIN) GO TO 3023
DELEO = YST - RPART
IF (IFILL .EQ. 0) DELEO = YST - RAD * BOTF
IF (DELEO .LT. EPS) THEN
YMIN = YST
ENDIF
JJJSKX = MIN (I - 1, 3 * NACROS)
BIOREMSMAIN
XS(I) = XST + 1.E-2 * RAD
YS(I) = YST + 1.E-2 * RAD
HITL(I) = HIT
RADI(I) = RAD
GO TO 3022
ENDIF
JN = MIN (I - 1, 3 * NACROS)
DO 3012 JJJJ = I, JJJJMX
J = I - JJJJ + 1
DELF2 = (XST-XS(J))**2
DELF2 = (YST-YS(J))**2
DELF2 = (HIT-HITE(J))**2
DELF2 = ((DIAM(I)+DIAM(J))/2.)**2
DIST = SQRT (DELF2+DELF2+DELF2) - SQRT (DELF2)
IF (DIST .LT. EPS) THEN
  IF (YST .LT. EPS(I)) THEN
    YMIN = YST
    RADI(I) = RAD
    XS(I) = XST + 1.E-2 *RADI(I)
    YS(I) = YST + 1.E-2 *RADI(I)
    HITE(I) = HIT
  ENDIF
ENDIF
GO TO 3021
CONTINUE
CONTINUE
CONTINUE
CONTINUE
CONTINUE
MP = YS(I)
PRIW1901, I, XS(I), YS(I), 2.*HITE(I)/DIAM(I), RAD1(I)
IF (DELBO .LT. WS) GO TO 3014
DELBO = YMAX-YS(I) -RPART
SEED = KSEED + 1
SUMP = SUMP + PI *RADI(I)**2
FORO = 1. -SUMP/(xMAX*YMAX)
CONTINUE
NSEED = KSEED
PRIW*, NSEED
PRIW*, FORO
READ*, DDXX
ELSEIF (IMETH .EQ. 4) THEN
IPAK = 1
PRINT*, 'INPUT IPAK: 1 = HEX 2 = STAGGERED', IPAK
READ*, IPAK
EPS0 = 0.3
DIAMF = 0.1
PRINT*, 'INPUT FIBER DIAMETER AND FRICTION', DIAMF, EPS0
READ*, DIAMF, EPS0
DEF = DIAMF
 DELDIA = 0.
 IREDU = 1
 PRINT*, 'INPUT IREDU, DELDIA', IREDU, DELDIA
 READ*, IREDU, DELDIA
ACON = 0
BIORENSMAIN
PRINT*, 'INPUT RANDOM VARIATION IN SPACING ( 0 TO 1 ) ', ACON
READ*, ACON
ISED = 1234321
PRINT*, 'INPUT SEED FOR RANDOM DISPLACEMENT', ISED
READ*, ISED
RHOS = 1. - EPS0
DXXX = DIAMF *SQRT(PI /2. /RT3 /RHOS)
IF (IPAK .EQ. 2) DXXX = DIAMF *SQRT(PI /2. /RHOS)
NX = NINT (XMAX /DXXX) + 1
XMAX = (NX-1) *DXXX
DYDX = RT3
IF (IPAK .EQ. 2) DYDX = 1.
PAC = DYDX /2.
DYYY = FAC *DXXX
NY = NINT (YMAX /DYYY) + 1
YMAX = (NY-1) *DYYY
SPAC = DXXX - DIAMF
IF (IPAK .EQ. 2) SPAC = DXXX /RT2 - DIAMF
SPAC = SPAC
DISD = 0.000 *50.
PRINT*, 'INPUT FACTOR FOR RANDOMIZATION', DISD
READ*, DISD
IPLTWK = 0
IF (DISD .GT. 0.) THEN
    PRINT*, 'PLOT RANDOM MOTION ? 0 = NO 1 = YES', IPLTWK
    READ*, IPLTWK
    ENDIF
SWELL = 0.
PRINT*, 'INPUT SWELLING FRACTION', SWELL
READ*, SWELL
IOK=0
PRINT*, 'INPUT OK: 0 = NO 1 = YES', IOK
READ*, IOK
IF (IOK.FQ.0) GO TO 1519
KSEED=0
EPS = 1.E-3 *SQRT(XMAX*YMAX)
EPS = MIN (EPS, SPAC/100.)
ID = 0
DO 1025 I=1,NY
   JD=1-MOD(I,2)
   DO 1015 J=1,NX-JD
      ID = ID + 1
      THET = 2.*PI *RAN(ISED)
      DELS = SPAC *(0.5-RAN(ISED))
      XS(ID)=(J-1)*DXXX+JD*DXXX/2. + ACON *DELS *COS(THET)
      YS(ID)=DYYY*(I-1) + ACON *DELS *SIN(THET)
      HITE(ID) = 0.
      DIAM(ID) = DIAMF
      RADI(ID) = DIAMF /2.
      DBOX(ID) = EPS
      WORK(1,ID) = XS(ID)
      WORK(2,ID) = YS(ID)
  1015 CONTINUE
  1025 CONTINUE
BIOREM$MAIN

KSEED = ID
NSEED = KSEED
NPAR = NSEED
ENDIF
C--- ROUGH POROSITY, SPAC, ETC. ---
C
NPAR = NSEED
VSUM = 0.
RBAR = 0.
DBAR = 0.
DO 887 ID = 1, NPAR
RBAR = RBAR + RAD(I)
DBAR = DBAR + DIAM(I)
ARID = PI * RAD(I)**2
VSUM = VSUM + ARID
887 CONTINUE
RBAR = RBAR /NPAR
DBAR = DBAR /NPAR
PORO = 1. - VSUM / (XMAX*YMAX)
RHOS = 1. - PORO
HLEG = SQRT (PI /2. /RT3 /RHOS) *DBAR
SPAC = HLEG - DBAR
SPAC = MAX (SPAC, DBAR /100.)
DIAMF = DBAR
DREF = DBAR
C--- RANDOMIZE SEED LOCATIONS ---
C
DISD = 0.000 *50.
PRINT*, 'INPUT FACTOR FOR RANDOMIZATION',DISD
READ*,DISD
IPLTWK = 0
IF (DISD .GT. 0.) THEN
  PRINT*, 'PLOT RANDOM MOTION ? 0 = NO 1 = YES',IPLTWK
  READ*,IPLTWK
ENDIF
NBORS = 6
NDARR = 18
CALL NEAREST (NDARR,NBORS,NSEED,XS,YS,X0,Y0,DNEIGH,INEIGH)
DELS = SPAC /10.
DELSO = DELS
KCYMX = DISD *DIAMF /DELS
IF (KCYMX .GT. 0) THEN
  IF (IPLTWK .GT. 0) THEN
XORIG = 0.
YORIG = 0.
RATXY = XMAX /YMAX
IF (RATXY.GT.1.0) THEN
XAXIS = 7.5
YAXIS=XAXIS *0.95775 /RATXY
ELSE
XAXIS = 7.5 *RATXY
YAXIS=XAXIS *0.95775 /RATXY
ENDIF
ENDIF
XSTP = (XMAX - XORIG) /1.
YSTP = (YMAX - YORIG) /1.
BIOREMSMAIN
CALL AREA2D(XAXIS,YAXIS)
CALL GRAF(XORIG,XSTP,XMAX,YORIG,YSTP,YMAX)
CALL FRAME
CALL MARKER(15)
CALL CURVE(XS,YS,NSEED,-1)
ENDIF
SLMAX = 0.
ITRMYX = 100
DO 1091 KCY = 1, KCYMX
JMOV = 0
DO 1092 I = 1, NSEED
   ITRY = 0
   DELS = DELS0
   TOL = 2. * PI * RAN(ISED)
   DELX = DELS * COS(THET)
   DELY = DELS * SIN(THET)
   XNEW = XS(I) + DELX
   YNEW = YS(I) + DELY
   X = DBOX(I)
   IF (XNEW .LT. TOL) YNEW = 2.*TOL - XNEW
   IF (XNEW .GT. XMAX-TOL) YNEW = 2.*TOL - YNEW
   IF (XNEW .LT. TOL) YNEW = 2.*TOL - YNEW
   DO 1093 JJ = 1, NBORS
      J = INEIGH(JJ, I)
      DIST = SQRT((XNEW-XS(J))**2 + (YNEW-YS(J))**2)
      IF (DIST.LT. RADI(I)+RADI(J)) THEN
         IF (ITRY.GT. ITRYMX) GO TO 1092
         ITRY = ITRY + 1
         GO TO 651
      ENDIF
   CONTINUE
   JMOV = JMOV + 1
   XS(I) = XNEW
   YS(I) = YNEW
   SL(I) = SQRT((XS(I)-X0(I))**2 + (YS(I)-Y0(I))**2)
   SLMAX = MAX(SL(I), SLMAX)
   CONTINUE
   IF (MOD(KCY-1, MAX(l,KCYMX/500)) .EQ. 0) THEN
      CALL SPACEF (57, NSEED, XS, YS, XMAX, YMAX, SDEVX, SDEVY)
      WRITE(16,900) KCY*DELS/DIAMF, SDEVX, SDEVY
   ENDIF
   IF (SLMAX.GT. DIAMF/3.) THEN
      DISMX = 0.
      DISY = 0.
      DISX = 0.
      DO 1855 ID = 1, NSEED
         DSID = SQRT((XS(ID)-WORK(1,ID))**2 + (YS(ID)-WORK(2,ID))**2)
         DSID = SQRT(DSID)
         DISMX = MAX(DISMX, DSID)
         DISY = DSBR + DSID
         DISX = MAX(DISX, ABS(XS(ID)-WORK(1,ID)))
         DISY = MAX(DISY, ABS(YS(ID)-WORK(2,ID)))
      CONTINUE
   ENDIF
   IF (IPLIKW .EQ. 1) THEN
      XB(1) = X0(ID)
      XB(2) = XS(ID)
      YB(1) = Y0(ID)
      YB(2) = YS(ID)
      CALL CURVE(XB, YB, 2, 0)
   ENDIF
CONTINUE
   DISBR = DISBR / NSEED
   CALL SPACEF (87, NSEED, XS, YS, XMAX, YMAX, SDEVX, SDEVY)
   WRITE(17,900) KCY*DELS/DIAMF, DISMX/DIAMF, DISBR/DIAMF,
SDEV, SDEVY, DISMX/SPAC, DISBR/SPAC

00867  1

00868  C

SIMAX = 0.
CALL NERNEB (NDARR, NBORS, NSEED, XS, X0, Y0, DNEIGH, INEIGH)

00871  C

00872  IF (IPL'IWK .EQ. 0) THEN

00873  CBAR = 0.

00874  NSUM = 0

00875  DO 1101 ID = 1, NSEED

00876  DBXX = MIN (XS(ID), XMAX-XS(ID)) /DIAMF

00877  DBXY = MIN (YS(ID), YMAX-YS(ID)) /DIAMF

00878  IF (DBXX.LT.2.0 .OR. DBXY.LT.2.0) GO TO 1101

00879  DO 1101 J = 1, NBORS

00880  NSUM = NSUM + 1

00881  CBAR = CBAR + DNEIGH(J,ID)

00882      1101 CONTINUE

00883  CBAR = CBAR /NSUM

00884  SDEV = 0.

00885  DO 1102 ID = 1, NSEED

00886  DBXX = MIN (XS(ID), XMAX-XS(ID)) /DIAMF

00887  DBXY = MIN (YS(ID), YMAX-YS(ID)) /DIAMF

00888  IF (DBXX.LT.2.0 .OR. DBXY.LT.2.0) GO TO 1102

00889  DO 1102 J = 1, NBORS

00890  SDEV = SDEV +(DNEIGH(J,ID)-CBAR)**2

00891      1102 CONTINUE

00892  SDEV = SQRT (SDEV / (NSUM-1.))

00893  PRINT901,JMOV,CBAR, SDEV, SDEVX, SDEVY, DISX, DISY

00894  ENDIF

00895  ENDIF

00896      1091 CONTINUE

00897  C

00898  IF (IPL'IWK .EQ. 1) CALL ENDPL(0)

00899  ICRAN = 1

00900  PRINT*, 'MORE RANDOMIZING? 0 = NO  1 = YES', ICRAN

00901  READ*, ICRAN

00902  IF (ICRAN .EQ. 1) THEN

00903  CALL SPACEF (87, NSEED, XS, YS, XMAX, YMAX, SDEVX, SDEVY)

00904  PRINT900, SXEX, SDEV, SDEVY

00905  PRINT*, 'INPUT FACTOR FOR RANDOMIZATION', DISD

00906  READ*, DISD

00907  PRINT*, 'PLOT RANDOM MOTION? 0 = NO  1 = YES', IPL'IWK

00908  READ*, IPL'IWK

00909  GO TO 1784

00910  ENDIF

00911  ENDIF

00912  C

00913  C--- TAKE SUBSET OF PARTICLES

00914  C

00915  ISUB = 0

00916  PRINT*, 'TAKE SUBSET OF PARTICLES? 0 = NO  1 = YES', ISUB

00917  READ*, ISUB

00918  IF (ISUB .EQ. 1) THEN

00919  XNEW = 0.

00920  YNEW = 0.

00921  XMAXN = XMAX

00922  YMAXN = YMAX

00923  PRINT*, 'INPUT XORIG, XMAX, YORIG, YMAX',

00924  1  XNEW, XMAXN, YNEW, YMAXN

00925  A16
READ*, XNEW, XMAX, YNEW, YMAX
XMAX = XMAX - XNEW
YMAX = YMAX - YNEW
DO 519 ID = 1, NSEED
   XS(ID) = XS(ID) - XNEW
   YS(ID) = YS(ID) - YNEW
519 CONTINUE
ENDIF
C --- DROP PARTICLES OUTSIDE BOX FROM SEED SET ---
C
NSUM = 0
NOUT = 0
SMEAS = SQRT (XMAX*YMAX)
EPS = 1.E-3 * SMEAS
DO 1188 ID = 1, NSEED
   DBXX = MIN (XS(ID), XMAX-XS(ID))
   DBXY = MIN (YS(ID), YMAX-YS(ID))
   IF (DBXX.LT.0.0 .AND. DBXY.LT.0.0) DBZZ = -SQRT(DBXX**2+DBXY**2)
   IF (DBZZ .GT. 0.0) THEN
      NSUM = NSUM + 1
      X0 (NSUM) = XS(ID)
      Y0 (NSUM) = YS(ID)
      DIAM0 (NSUM) = DIAM(ID)
      HITEO (NSUM) = HITE(ID)
   ELSEIF (DBZZ .GT. EPS-RADI(ID)) THEN
      NOUT = NOUT + 1
      WORK(1,NOUT) = XS(ID)
      WORK(2,NOUT) = YS(ID)
      WORK(3,NOUT) = DIAM(ID)
      WORK(4,NOUT) = HITE(ID)
   ENDIF
1188 CONTINUE
DO 1189 ID = 1, NSEED
   XS(ID+NSUM) = WORK(1,ID)
   YS(ID+NSUM) = WORK(2,ID)
   DIAM(ID+NSUM) = WORK(3,ID)
   HITE(ID+NSUM) = WORK(4,ID)
1189 CONTINUE
NPAR = NSEED
C --- ASSIGN RADII AND SWELL PARTICLES ---
DIAMF0 = DIAMF
DIAMF = 0.
DO 1163 ID = 1, NPAR
   DIAM0(ID) = DIAM(ID)
   DIAM(ID) = DIAM(ID) *(1.+SWELL)
   RAD2 = MAX (1.E-20*XMAX, (DIAM(ID)/2.)**2-HITE(ID)**2)
RADI(ID) = SQRT (RAD2)
DIAMF = DIAMF + DIAM(ID)

CONTINUE
DIAMF = DIAMF /NSEED

C--- MONTE CARLO POROSITY ---

C
ISAD = 11554321
NSHOT = 5000
NVOID = 0

DO 492 J = 1, NSHOT
XXX = XMAX *RAN(ISAD)
YYY = YMAX *RAN(ISAD)

DO 493 ID = 1, NSEED
DELS2 = (XS(ID)-XXX)**2 + (YS(ID)-YYY)**2

IF (DELS2 .LT. RADI(ID)**2) GO TO 492

CONTINUE
NVOID = NVOID + 1

CONTINUE
EPSMC = (1. *NVOID) /NSHOT

SUM POROSITY ---
SUMV = 0.
DIAMF = 0.

DO 677 ID = 1, NSEED
DIAM0(ID) = DIAM(ID)
DIAMF = DIAMF + DIAM(ID)
DELBX = MIN (XS(ID), XMAX-XS(ID))
DELB = MIN (DELBX, DELBY)
DELB = MIN (ABSD, RADI(ID))
VID = PI *RADI(ID)**2

IF (ABSD .LE. RADI(ID)) THEN
THET = ACOS (ABSD /RADI(ID))
VPIE = THET *RADI(ID)**2
HGT = SQRT (RADI(ID)**2 - ABSD**2)
VTRI = HGT *ABSD
DELV = VPIE - VTRI
IF (DELV .GT. 0.) THEN
VID = VID - DELV
ELSE
VID = DELV
ENDIF

BiCREM$MAIN
ENDIF
SUMV = SUMV + VID

CONTINUE
PORO = 1. - SUMV /XMAX /YMAX
DIAMF = DIAMF /NSEED
PRINT980, 'PORO', PORO

C--- ROTATE SEEDS BY PI/2 IF REQUESTED
C
IF (IROTA .EQ. 1) THEN
DO 1044 ID = 1, NSEED
YTMP = XMAX - XS(ID)
XS(ID) = YS(ID)
YS(ID) = YTMP

END
01041 1044  CONTINUE
01042     XIMP = YMAX
01043     YMAX = XMAX
01044     XMAX = XIMP
01045     ENDIF
01046 C--- INTERMEDIATE DUMP ---
01047 C
01048 C
01049     WRITE (99,*) NPAR, XMAX, YMAX, EPSO, DIAMF
01050     DO 692 ID = 1, NPAR
01051          WRITE(99,*) XS(ID), YS(ID), HITE(ID), DIAM(ID), RADI(ID)
01052              692 CONTINUE
01053 C
01054 C--- INITIALIZE PLOT ---
01055 C
01056     XORIG = 0.
01057     YORIG = 0.
01058     RATXY = XMAX / YMAX
01059     IF (RATXY.GT.1.0) THEN
01060          XAXIS = 7.5
01061          XAXIS = XAXIS *0.95775 / RATXY
01062 ELSE
01063          XAXIS = 7.5 * RATXY
01064          XAXIS = XAXIS *0.95775 / RATXY
01065     ENDIF
01066     XSTP = (XMAX - XORIG) / 1.
01067     YSTP = (YMAX - YORIG) / 1.
01068 C
01069     CALL AREA2D(XAXIS, YAXIS)
01070     CALL GRAF(XORIG, XSTP, XMAX, YORIG, YSTP, YMAX)
01071     CALL FRAME
01072 C
01073     PRINT*, ' '
01074     PRINT*, ' '
01075     PRINT*, ' '
01076     PRINT*, ' '
01077     PRINT981, 'NPAR', NPAR
01078     PRINT980, 'XMAX', XMAX
01079     PRINT980, 'YMAX', YMAX
01080     PRINT980, 'EPSO', EPSO
01081     PRINT980, 'PORO', PORO
01082     PRINT980, 'EPSC', EPSC
01083     PRINT980, 'SPAC', SPAC

BIOREM$MAIN

01084     PRINT980, 'DREF', DREF
01085 C--- PLOT SEEDS
01086 C
01087 C
01088     IF (IPLTS.EQ.1) THEN
01089          CALL SCIPIC(0.5)
01090          CALL MARKER(18)
01091          CALL CURVE(XS, YS, NSEED, -1)
01092          CALL RESET('MARKER')
01093          CALL SCIPIC(1.)
01094     ENDIF
01095 C
01096     RJBAR=0.
01097     ABAR=0.
01098     NSUM=0.
01099  NSUMOLD=0
01100  JLMAX=0
01101  JLMIN=200
01102  SUMLB=0.
01103  NBSUM=0
01104  DO 1621 K=1,20
01105  1621  PSID(K)=0
01106  C
01107  DO 1100  I=1,NSEED
01108  C
01109  C--- COMPUTE LINE CONSTANTS FOR Ith SEED ---
01110  C
01111  JL=0
01112  DEMX=DEMAXO*1.6/SQRT(1.*NSEED)
01113  DO 1030  J=1,NSEED
01114  IF (I.EQ.J) GO TO 1030
01115  DELS = SQRT((XS(I)-XS(J))**2 + (YS(I)-YS(J))**2)
01116  IF (DELS.GT.DEMX) GO TO 1030
01117  JL = JL + 1
01118  X0(JL) = (XS(I)+XS(J)) /2.
01119  Y0(JL) = (YS(I)+YS(J)) /2.
01120  DELX = XS(J) - XS(I)
01121  DELY = YS(J) - YS(I)
01122  DEL = SIGN(MAX(ABS(DELY),1.E-8),DELY)
01123  SL(JL) = -DELS /DELY
01124  JB(JL) = J
01125  1030  CONTINUE
01126  NLIN = JL
01127  C
01128  C--- ADD BOUNDARY DOMAIN LINES ---
01129  C
01130  EPS = 0.
01131  RRR = 0.5 + EPS
01132  DO 1040  J = 1, NBOX
01133  C
01135  X000 = (0.5 + RRR *COS(THET)) *XMAX
01136  Y000 = (0.5 + RRR *SIN(THET)) *YMAX
01137  SLOO = -1. /SIGN (MAX (ABS(TAN(THET)),1.E-10), TAN(THET))
01138  C
01139  C--- REF: PURCEL PAGE 50
01140  C

BIOREMSMAIN

01141  BBB = 1.
01142  AAA = - SLOO
01143  CCC = SLOO *X000 - Y000
01144  DELS = ABS(AAA*XS(I)+BBB*YS(I)+CCC)/SQRT(AAA**2+BBB**2)
01145  IF (DELS.GT.DEMX) GO TO 1040
01146  JL = JL + 1
01147  JB(JL) = - J
01148  X0(JL) = X000
01149  Y0(JL) = Y000
01150  SL(JL) = SLOO
01151  1040  CONTINUE
01152  JLMAX = MAX (JL, JLMAX)
01153  JLMIN = MIN (JLMIN, JL)
01154  C
01155  C--- LOCATE NEAREST (PARTICLE) BOUNDING LINE ---
01156  C
01157      DMIN = 1.8E8
01158      EPS = 1.E-4  *SMEAS
01159      DO 1050 J = 1, NLIN
01160      C
01161      C----- NEW STUFF ----- C
01162      C
01163      IF (XO(J) .GT.XMAX-EPS .OR. XO(J) .LT.EPS) GO TO 1050
01164      IF (YO(J) .GT.YMAX-EPS .OR. YO(J) .LT.EPS) GO TO 1050
01165      IF (JB(J) .GT. NSEED-NOGT) GO TO 1050
01166      C
01167      DELS=SQRT((XS(I)-XO(J))**2+(YS(I)-YO(J))**2)
01168      IF (DELS.LT.EPS) THEN
01169          DMIN=DELS
01170          JNN=J
01171      ENDIF
01172      1050      CONTINUE
01173      C
01174      JC=-1
01175      JN=JMN
01176      XX(1)=XO(JN)
01177      YY(1)=YO(JN)
01178      C
01179      C----- START WALK AROUND CELL BOUNDARY -----
01180      C
01181      KSAV=1
01182      NSYD=0
01183      MSID=0
01184      RAVE=0.
01185      RMIN=1.
01186      RMAX=0.
01187      ASYM=0.
01188      EPS = 1.E-2  *SMEAS
01189      DO 1110 K=1,20
01190      C
01191      JO=JC
01192      JC=JN
01193      C
01194      DMIN=SQRT(2.)
01195      XVJO=XX(K)
01196      YVJO=YY(K)
01197      DO 1120 J=1,JL

BIOREMSMAIN

01198      C
01199      IF (J.EQ.JC) GO TO 1120
01200      IF (J.EQ.JO) GO TO 1120
01201      C
01202      DSLO = SL(JC) - SL(J)
01203      DSLO = SIGN(MAX(ABS(DSLO),1.E-6), DSLO)
01204      XVJ=( YO(JC)-SL(JC)*XO(JC) )/DSLO
01205      DSLO = SL(JC) - SL(J) *
01206      1
01207      DSLO = SIGN(MAX(ABS(DSLO),1.E-6), DSLO)
01208      YVJ=YO(J)+SL(J)*(XVJ-XO(J))
01209      IF (ABS(SL(J)) .LT.ABS(SL(JC))) THEN
01210      ENDIF
01211      C
01212      C----- NEW STUFF ----- C
01213      IF (XVJ.GT.XMAX+EPS .OR. XVJ.LT.-EPS) GO TO 1120
01214      IF (YVJ.GT.YMAX+EPS .OR. YVJ.LT.-EPS) GO TO 1120
C      -- CHECK FOR COUNTER-CLOCKWISE PATH --
C      NOTE:  PUT THIS FIX IN BONZO
C
IF (JC .LE. NLIN) THEN
  DIREC1 = (X(0,JJC)-X(I)) *(Y(JJC)-Y(I))
  DIREC2 = (Y(I)-Y(JJC))*(X(JJC)-X(I))
ELSE
  DIREC1 = X(JJC) - X(I)
  DIREC2 = Y(JJC) - Y(I)
END IF
IF (JB(JJC).LT.-2) THEN
  DIREC1 = - DIREC1
  DIREC2 = - DIREC2
ENDIF
DIREC = DIREC1
IF (ABS(DIREC2).GT.ABS(DIREC1)) DIREC = DIREC2
IF (DIREC. LT. 0.0) GO TO 1120

C
C      BONDMAIN
C
IF (JC.GT.NLIN) THEN
  LBSUM = LBSUM + LBB
ELSE
  LBB = SQRT((XS(I)-X(JJC))**2+(YS(I)-Y(JJC))**2)
  LBSUM = LBSUM + LBB
ENDIF

C      COMPUTE AREA OF TRIANGULAR SEGMENT
C
IF (K.GT.1) THEN
  DEH = SQRT((XS(I)-X(JC))**2+(YS(I)-Y(JC))**2)
  DELB = SQRT((XX(K+1)-XX(K))**2+(YY(K+1)-YY(K))**2)
  IF (JC.GT.NLIN) THEN
    BBB = 1.
    AAA = -SL(JC)
  ELSE
    IF (ABS(DIREC2).GT.ABS(DIREC1)) DIREC = DIREC2
    IF (DIREC. LT. 0.0) GO TO 1120
  ENDIF
ENDIF

CONTINUE
KSAV = KSAV + 1

IF (K.GT.1) THEN
  LSID(K-1,I) = SQRT((XX(K+1)-XX(K))**2+(YY(K+1)-YY(K))**2)
  LBSUM = LBSUM + LSID(K-1,I)
ENDIF

C      COMPUTE BOND AND SIDE LENGTH
C
IF (K.GT.1) THEN
  LSID(K-1,I) = SQRT((XX(K+1)-XX(K))**2+(YY(K+1)-YY(K))**2)
  LBSUM = LBSUM + LSID(K-1,I)
ENDIF

01215 C
01216 C      --- CHECK FOR COUNTER-CLOCKWISE PATH ----
01217 C      NOTE:  PUT THIS FIX IN BONZO
01218 C
01219 C
01220 IF (JC .LE. NLIN) THEN
01221 DIREC1 = (X(0,JJC)-X(I)) *(Y(JJC)-Y(I))
01222 DIREC2 = (Y(I)-Y(JJC))*(X(JJC)-X(I))
01223 ELSE
01224 DIREC1 = X(JJC) - X(I)
01225 DIREC2 = Y(JJC) - Y(I)
01226 IF (JB(JJC).LT.-2) THEN
01227 DIREC1 = - DIREC1
01228 DIREC2 = - DIREC2
01229 ENDIF
01230 DIREC = DIREC1
01231 IF (ABS(DIREC2).GT.ABS(DIREC1)) DIREC = DIREC2
01232 IF (DIREC. LT. 0.0) GO TO 1120
01233 C
01234 DUV=SQRT((X(JJC)-X(I))**2+(Y(JJC)-Y(I))**2)
01235 IF (DUV.LT.DMIN) THEN
01236 DMIN=DUV
01237 DMN=DMN+DMN
01238 DMIN=DMIN
01239 DMIN=DMIN
01240 DMIN=DMIN
01241 DMIN=DMIN
01242 DMIN=DMIN
01243 DMIN=DMIN
01244 DMIN=DMIN
01245 DMIN=DMIN
01246 ENDIF
01247 1120 CONTINUE
01248 KSAV = KSAV + 1
01249 C
01250 C      --- COMPUTE BOND AND SIDE LENGTH ----
01251 C
01252 IF (K.GT.1) THEN
01253 LSID(K-1,I) = SQRT((XX(K+1)-XX(K))**2+(YY(K+1)-YY(K))**2)
01254 LBSUM = LBSUM + LSID(K-1,I)
01255 ELSE
01256 LBB = SQRT((XS(I)-X(JC))**2+(YS(I)-Y(JC))**2)
01257 LBSUM = LBSUM + LBB
01258 ENDIF
01259 ELSE
01260 LSID(K-1,I) = SQRT((XS(I)-XS(L))**2+(YS(I)-YS(L))**2)
01261 SIMLB = SIMLB + LSID(K-1,I)
01262 ENDIF
01263 ELSE
01264 C
01265 C      COMPUTE AREA OF TRIANGULAR SEGMENT
01266 C
01267 IF (K.GT.1) THEN
01268 DELH = SQRT((XS(I)-X(JC))**2+(YS(I)-Y(JC))**2)
01269 DELB = SQRT((XX(K+1)-XX(K))**2+(YY(K+1)-YY(K))**2)
01270 IF (JC.GT.NLIN) THEN
01271 BBB = 1.
01272 AAA = -SL(JC)
01273 ENDIF
01274 ENDIF
01275 ENDIF
01276 ENDIF
01277 ENDIF
CCC=SL(JC)*X0(JC)-Y0(JC)
DELH=ABS((AAA*XS(I)+BBB*YS(I))+CCC)/SQRT(AAA**2+BBB**2)
ENDIF
ASUM=ASUM+0.5*DELH*DELB
ENDIF
C
IF (K.GT.3 .AND. JC.EQ.JMN) GO TO 1130
C
C--- COMPUTE CORRESPONDENCE AND FLAG BOUNDARY SEEDS
C
IF (JN.GT.NLIN) THEN
IPRM(K,I)=JB(JN)
ENDIF
IF (IDBC(I).EQ.O) THEN
IDBC(I)=IPRM(K,I)
LBC(I)=K
IDEC(I)=IPRM(K, I)
ELSEIF (IDBC(I) .EQ.-1 .OR. IDBC(I) .EQ.-3) THEN
ENDIF
IDP=JB(JN)
IPRM(K,I)=IDP
IF (IDP.LT.1) THEN
CONTINUE
ENDIF
ENDIF
IF (JN.LE.NLIN) THEN
xsJ=XS(JB(JN))
YSJ=YS(JB(JN))
DDSJ=SQRT((XS(1)-W)**2+(YS(I)-YSJ)**2)
RAVE=RAVE+DDSJ
RMIN=MIN(RMIN, DDSJ)
CMIN=MIN(CMIN, DDSJ)
CMAX=IGX(CMAX, DDSJ)
MsID=MsID+l
ENDIF
NSYD=NSYD+l

c
IF (IPLTB.EQ.1 .AND. JN.LE.NLIN) THEN
XB(l)=XS(I)
YB(l)=YS(I)
XB(2)=xs(JB(JN))
YB(2)=YS(JB(JN))
cALL cuRVE(XB,YB,2,1)
ENDIF
CONTINUE

RMAX=MAX(RMAX, DDSJ)
C
CHMN=MIN(CHMN, DDSJ)
CMAX=MAX(CMAX, DDSJ)
MSID=MSID+1
ENDIF

C
NSYD=NSYD+l
C
C--- PLOT BONDS ---
C
IF (IPLTB.EQ.1 .AND. JN.LE.NLIN) THEN
XB(I)=XS(I)
YB(I)=YS(I)
XB(2)=xs(JB(JN))
YB(2)=YS(JB(JN))
cALL cuRVE(XB,YB,2,1)
ENDIF
CONTINUE

A23
CONTINUE
RAVE=RAVE/\text{MAX}(\text{MSID}, 1)

\text{--- COMPUTE SIDE AND AREA DIAGNOSTICS ---}

\text{NSUM=NSUM+1}
\text{RJBAR=RJBAR+NSYD}
\text{FSID(\text{NSYD})=FSID(\text{NSYD})+1}
\text{GAVE=GAVE+RAVE}
\text{\text{GMIN=GMIN+RMIN}}
\text{\text{GMAX=GMAX+RMAX}}
\text{\text{ARBAR=ARBAR+ASUM}}

\text{--- PLOT CELL BOUNDARIES ---}
\text{IF (\text{IPLTC}.EQ.1) THEN}
\hspace{1cm} \text{IF (YS(I) .LT. 0.5) THEN}
\hspace{2cm} \text{CALL CURVE(XX,YY,KSAV,0)}
\hspace{2cm} \text{CALL RESET ('DASH')}\text{ENDIF}
\hspace{1cm} \text{ENDIF}
\hspace{1cm} \text{AREA(I)=ASUM}
\hspace{1cm} \text{NSID(I)=NSYD}

\text{--- NUMBER NODES, ASSIGN X-Y COORDINATES, NODE CORRESPONDANCE AND FLAG BOUNDARY NODES ---}
\text{IDN=0}
\text{DO 1400 \text{ID}=1,NSEED}
\hspace{1cm} \text{DO 1400 \text{L}=1,NSID(ID)}
\hspace{2cm} \text{IF (IDNOD(L,ID),EQ.0) THEN}
\hspace{3cm} \text{IDNOD(L,ID)=IDN+1}
\hspace{3cm} \text{XNOD(IDN)=XSID(L,ID)}
\hspace{3cm} \text{YNOD(IDN)=YSID(L,ID)}
\hspace{1cm} \text{ENDIF}
\hspace{1cm} \text{ENDIF}
\hspace{1cm} \text{IF (IDP.GT.0) THEN}
\hspace{2cm} \text{LP=LPFRM(L,IM)}
\hspace{2cm} \text{IDNOD(LP,IDP)=IDN}
\hspace{1cm} \text{ENDIF}
\hspace{1cm} \text{LM1=MOD(L+NSID(IDC)-2,NSID(IDC))+1}
\hspace{1cm} \text{IDP=IPRM(IM,ID)}
\hspace{1cm} \text{IF (IDP.GT.0) THEN}
\hspace{2cm} \text{LP=LPFRM(LM1,ID)}
\hspace{2cm} \text{IDNOD(LP,IDP)=IDN}
\hspace{1cm} \text{ENDIF}
\hspace{1cm} \text{IF (IDP.EQ.-IDBIN .OR. IDM.EQ.-IDBIN)}

\text{--- BIOREMSMAIN ---}
\text{IDNOD(L,ID)=IDN}
\text{IDP=IPRM(L,ID)}
\text{IF (IDP.GT.0) THEN}
\hspace{1cm} \text{LP=LPFRM(L,ID)}
\hspace{1cm} \text{LPP1=MOD(LP,NSID(IDC))+1}
\hspace{1cm} \text{IDNOD(LPP1,IDP)=IDN}
\hspace{1cm} \text{ENDIF}
\hspace{1cm} \text{LM1=MOD(L+NSID(IDC)-2,NSID(IDC))+1}
\hspace{1cm} \text{IDP=IPRM(IM1,ID)}
\hspace{1cm} \text{IF (IDP.GT.0) THEN}
\hspace{2cm} \text{LP=LPFRM(LM1,ID)}
\hspace{2cm} \text{IDNOD(LP,IDP)=IDN}
\hspace{1cm} \text{ENDIF}
\text{NODAR(8,IDP)=0}
\text{IDP=IPRM(L,ID)}
\text{IF (IDP.GT.0) NODAR(8,IDN)=IDP}
\text{IF (IDP.GT.0) NODAR(8,IDN)=IDP1}
\text{IF (IDP.EQ.-IDBIN .OR. IDP1.EQ.-IDBIN)
01389      1      NODAR (8, IDN) = -IDBIN
01390      IF (IDP .EQ. -IDBOU .OR. IDPM .EQ. -IDBOU )
01391      1      NODAR (8, IDN) = -IDBOU
01392      C
01393      ENDIF
01394      1400 CONTINUE
01395      NNOD = IDN
01396      C
01397      C NUMBER TUBES, IDENTIFY TUBE NODES, LENGTHS AND DIAMETERS
01398      C ASSIGN TUBE ENDPOINTS AND XY SHIFTS
01399      C
01400      1610 IDT = 0
01401      DO 1402 ID = 1, NSEED
01402      DO 1402 L = 1, NSID (ID)
01403            IDP = IFRM (L, ID)
01404            IF (IDP .GT. ID .OR. IDP .LT. 0) THEN
01405                IDT = IDT + 1
01406                TUBNOD (1, IDT) = INOD (L, ID)
01407                TUBNOD (3, IDT) = ID
01408                TUBNOD (4, IDT) = IDP
01409                LP1 = MOD (L, NSID (ID)) + 1
01410                TUBNOD (2, IDT) = INOD (LP1, ID)
01411                TUBAR (1, IDT) = MAX (LSID (L, ID), 1.E-6*XMAX)
01412                TUBAR (9, IDT) = 1.*IDP
01413                IF (IDP .GT. 0) THEN
01414                  LP = LPRM (L, ID)
01415                  IDTUB (LP, IDP) = IDT
01416                  DEL = SQRT ( (XS (ID) - XS (IDP))**2 + (YS (ID) - YS (IDP))**2 )
01417                  IF (IREDU .EQ. 1) THEN
01418                    TUBAR (2, IDT) = DEL - RADI (ID) - RADI (IDP) - DELDIA
01419                    ELSEIF (IREDU .EQ. 2) THEN
01420                      TUBDI = DEL - RADI (ID) - RADI (IDP)
01421                      TUBAR (2, IDT) = TUBDI *(1. - DELDIA)
01422                    ELSEIF (IREDU .EQ. 3) THEN
01423                      TUBDI = DEL - RADI (ID) - RADI (IDP)
01424                      TUBAR (2, IDT) = TUBDI / (1. + TUBDI*DELDIA)
01425                  ELSEIF (IREDU .EQ. 1) THEN
01426                  TUBAR (2, IDT) = TUBDI / (1. + DELDIA*TUBDI)
01427                  ENDIF
01428                  ELSE
01429                    IF (IDP .EQ. -1) DEL = YS (ID)
01430                    IF (IDP .EQ. -2) DEL = XMAX - XS (ID)
01431                    IF (IDP .EQ. -3) DEL = YMAX - YS (ID)
01432                    IF (IDP .EQ. -4) DEL = XS (ID)
01433                    IF (IREDU .EQ. 1) THEN
01434                      TUBAR (2, IDT) = DEL - RADI (ID) - DELDIA/2.
01435                    ELSEIF (IREDU .EQ. 2) THEN
01436                      TUBDI = DEL - RADI (ID)
01437                      TUBAR (2, IDT) = TUBDI *(1. - DELDIA)
01438                    ELSEIF (IREDU .EQ. 3) THEN
01439                      TUBDI = DEL - RADI (ID)
01440                      TUBAR (2, IDT) = TUBDI / (1. + TUBDI*DELDIA)
01441                  ENDIF
01442                  ENDIF
01443      KBLOC (IDT) = 1
01444      IF (TUBAR (2, IDT) .LT. 1.E-8*XMAX) THEN
01445        KBLOC (IDT) = 0
01446      TUBAR (2, IDT) = 1.E-8*XMAX

A25
01447 C ENDIF
01448 IF (IDP.GT.0) THEN
01449 DEL=SQRT( (XS(ID)-XS(IDP))**2 + (YS(ID)-YS(IDP))**2 )
01450 DELTA = (RADI(ID)-RADI(IDP))/2.
01451 DXDL= (XS(IDP)-XS(ID))/DEL
01452 DYDL= (YS(IDP)-YS(ID))/DEL
01453 DELTAX = DELTA *DXDL
01454 DELTAY = DELTA *DYDL
01455 ELSE
01456 DELTAX = 0.
01457 DELTAY = 0.
01458 IF (IDP .EQ. -1) DELTAY = TUBAR(2,IDT) /2.
01459 IF (IDP .EQ. -2) DELTAY = - TUBAR(2,IDT) /2.
01460 IF (IDP .EQ. -3) DELTAY = - TUBAR(2,IDT) /2.
01461 IF (IDP .EQ. -4) DELTAX = TUBAR(2,IDT) /2.
01462 ENDIF
01463 TUBAR(3,IDT) = XSID(L,ID) + DELTAX
01464 TUBAR(4,IDT) = YSID(L,ID) + DELTAY
01465 TUBAR(5,IDT) = XSID(L+1,ID) + DELTAX
01466 TUBAR(6,IDT) = YSID(L+1,ID) + DELTAY
01467 TUBAR(7,IDT) = DELTAX
01468 TUBAR(8,IDT) = DELTAY
01469 CONTINUE
01470 C ENDIF
01471 1402 CONTINUE
01472 NTUB = IDT
01473 C--- IDENTIFY NODE TRIOS AND ASSOCIATED TUBE NUMBERS ---
01474 C SEQUENCE OF TRIOS IS IN ANTI-CLOCKWISE DIRECTION
01475 C
01476 DO 1406 ID=1,NSEED
01477 DO 1406 L=1,NSID(ID)
01478 IF (IDP.GT.ID .OR. IDP.LT.0) THEN
01479 ID = IDNOD(L,ID)
01480 IF (IDP.GT.0) THEN
01481 LP1 = LPRM(L,ID)
01482 IF (IDP.GT.ID .OR. IDP.LT.0) THEN
01483 IF (NODAR(1,IDN).EQ.0) THEN
01484 LP1 = MOD(L,NSID(ID))+1
01485 LM1 = MOD(L+NSID(ID)-2,NSID(ID))+1
01486 NODAR(2,IDN) = IDNOD(LP1,ID)
01487 NODAR(3,IDN) = IDNOD(LM1,ID)
01488 NODAR(5,IDN) = IDTUB(L,ID)
01489 NODAR(6,IDN) = IDTUB(LM1,ID)
01490 NODAR(9,IDN) = ID
01491 NODAR(10,IDN) = IPRM(L,ID)
01492 NODAR(11,IDN) = IPRM(LM1,ID)
01493 NODAR(1,IDN) = 2
01494 IDP = IPRM(L,ID)
01495 IF (IDP.GT.0) THEN
01496 LP = LPRM(L,ID)
01497 LPP1 = MOD(LP,NSID(IDP))+1
01498 IF (IDNOD(L,IDN).EQ.IDNOD(LPP1,IDP)) THEN
01499 LPP2 = MOD(LPP1,NSID(IDP))+1
01500 NODAR(4,IDN) = IDNOD(LPP2,IDP)
01501 NODAR(7,IDN) = IDTUB(LPP1,IDP)
01502 NODAR(1,IDN) = 3
01503 GO TO 1406
01504 ENDIF

BIOREM5MAIN

A26
ENDIF
01508 IDP = IPRM(LML,ID)
01509 NODAR(10,IDN) = IDP
01510 IF (IDP.GT.0) THEN
01511 LP = LPRM(LML,ID)
01512 IF (IDNOD(L,ID).EQ.IDNOD(LP,IDP)) THEN
01513 NODAR(4,IDN) = IDNOD(LP,IDP)
01514 NODAR(7,IDN) = IDTUB(LP,IDP)
01515 ENDIF
01516 ENDIF
01517 ENDIF
01518 ENDIF
01519 1406 CONTINUE
01520 C
01521 C--- TABULATE AND STORE BOUNDARY NODES AND CROSSING TUBES ---
01522 C NNBC(J) IS THE NUMBER OF NODES/TUBES ON BOUNDARY J; NBID(I,J) IS
01523 C THE NODE NODE NUMBER OF THE JTH NODE ON BOUNDARY I; TBID(I,J) IS
01524 C THE TUBE NUMBER OF THE JTH TUBE CROSSING BOUNDARY I.
01525 C
01526 DO 1447 IDN = 1, NNOD
01527 IDB = - NODAR(8,IDN)
01528 IF (IDB.GT.0) THEN
01529 NNBC(IDB) = NNBC(IDB) + 1
01530 NBID(IDB,NNBC(IDB)) = IDN
01531 DO 1447 J = 1, NODAR(1,IDN)
01532 IF (IDB,IDN) .NE. NODAR(8,IDNJ)) THEN
01533 TBID(IDB,NNBC(IDB)) = NODAR(4+J,IDN)
01534 GO TO 1447
01535 C
01540 ENDIF
01541 1448 CONTINUE
01542 ENDIF
01543 1447 CONTINUE
01544 C
01545 C--- PLOTT PARTICLES
01546 C
01547 IF (IPLTP.EQ.1) THEN
01548 DO 1811 ID = 1, NSEED
01549 XCEN = XS(ID) *XAXIS /XMAX
01550 YCEN = YS(ID) *YAXIS /YMAX
01551 RADIN = 0.95 *RADI(ID) *XAXIS /XMAX
01552 CALL BLCIR (XCEN, YCEN, RADIN, 0)
01553 1811 CONTINUE
01554 ENDIF
01555 C
01556 IF (IPLTP.NE.0) THEN
01557 IF (IPLTP .EQ. 1) THEN
01558 CALL MARKER(16)
01559 CALL BLSYM
01560 ELSEIF (IPLTP .EQ. -1) THEN
01561 CALL MARKER(-1)
01562 ENDIF

A27
CALL BLREC(-0.2*XAXIS,-0.2*YAXIS,0.2*XAXIS,1.4*YAXIS,0)
CALL BLREC(-0.2*XAXIS,-0.2*YAXIS,1.4*XAXIS,0.2*YAXIS,0)
CALL BLREC(1.0*XAXIS,-0.2*YAXIS,0.2*XAXIS,1.4*YAXIS,0)
CALL BLREC(-0.2*XAXIS,1.0*YAXIS,1.4*XAXIS,0.2*YAXIS,0)

DO 1801 ID=1,NSEED
  \( X_B(1)=X_S(ID) \)
  \( Y_B(1)=Y_S(ID) \)
  \( FAC=2.*RADIUS(ID)/XMAX*XAXIS/0.082 \)
  CALL SCLPIC(FAC)
  CALL CURVE(XB,YB,1,1)
1801 CONTINUE

CALL SCLPIC(1.)
CALL RESER('BLSYM')
ENDIF
C
C--- COMPUTE MAX NODE DIAMETERS ---
C
DO 3018 IDN=1,NNOD
  XTU(IDN) = XNOD(IDN)
  YTU(IDN) = YNOD(IDN)
  RTU(IDN) = XMAX /100.
  RMIN = XMAX
  IF (NODAR(1,IDN) .LT. 3) GO TO 3018
DO 3019 J = 1, NODAR(1,IDN)
  IDT = NODAR (4+J, IDN)
  ID = TUBNOD (3, IDT)
  IDP = TUBNOD (4, IDT)
  IF (ID .GT. 0 .AND. IDP .GT. 0) THEN
    XAA(J,1) = -2. *(%1 - XS(IDP))
    XAA(J,2) = -2. *(YS(ID) - YS(IDP))
    XAA(J,3) = -2. *(RADIUS(ID) - RADIUS(IDP))
    RHS(J) = RADIUS(ID)**2 - RADIUS(IDP)**2
    QHS(J) = RADIUS(ID) - RADIUS(IDP)
  ELSE
    XAA(J,1) = 0.
    RHS(J) = 0.
    QHS(J) = 0.
    IDSAV = ID
    ID = MAX (ID, IDP)
    IDP = MIN (IDSAV, IDP)
  ENDIF
3019 CONTINUE

BIORESMAIN

DIS = SQRT ((XS(ID) -XS(IDP))**2 + (YS(ID) -YS(IDP))**2)
RTRY = DIS - RADIUS(ID) - RADIUS(IDP)
RMIN = MIN (RMIN, RTRY)
ELSE
  XAA(J,1) = 0.
  XAA(J,2) = 0.
  XAA(J,3) = 0.
  RHS(J) = 0.
  QHS(J) = 0.
  IDSAV = ID
  ID = MAX (ID, IDP)
  IDP = MIN (IDSAV, IDP)
ENDIF

A28
WT = 1.
WIR = 0.3
RTU(IDN) = 1.1 *RMIN
RNEW = RTU(IDN)
ITSK = 0
DO 3077 ITS = 1, 50
ITSK = ITSK + 1
C
RTU(IDN) = (1.-WIR) *RTU(IDN) + WIR *RNEW
DETA = XAA(1,1) *XAA(2,2) - XAA(1,2) *XAA(2,1)
XHS(1) = RHS(1) + 2.* QHS(1) *RTU(IDN)
XHS(2) = RHS(2) + 2.* QHS(2) *RTU(IDN)
DETY = XAA(1,1) *XHS(2) - XAA(1,2) *XAA(2,1)
XNEW = DETY /DETA
YNEW = DETY /DETA
XTU(IDN) = (1.-WT) *XTU(IDN) + WT *XNEW
YTU(IDN) = (1.-WT) *YTU(IDN) + WT *YNEW
DELM = 0.
ERR = 0.
DO 3076 J = 1, NODAR(1,IDN)
ID = NODAR(J+8, IDN)
IF (ID .LT. 1) GO TO 3076
RTUF = (XS(ID)-XTU(IDN))**2 + (YS(ID)-YTU(IDN))**2
RXX = SQRT (RTUF) - RADI(ID)
RES(J) = SQRT(RTU(IDN) - RADI(ID) - RTU(IDN))
ERR = MAX (ERR, ABS(RES(J)))
IF (ABS(RXX-RTU(IDN)) .GT. DELM) THEN
DELM = ABS (RXX-RTU(IDN))
RNEW = RXX
ENDIF
3076 CONTINUE
C
DO 3078 J = 1, NODAR(1,IDN)
ID = NODAR(J+8, IDN)
IF (ID .LT. 1) GO TO 3078
RTUF = (XS(ID)-XTU(IDN))**2 + (YS(ID)-YTU(IDN))**2
RXX = SQRT (RTUF) - RADI(ID)
RES(J) = SQRT(RTU(IDN) - RADI(ID) - RTU(IDN))
ERR = MAX (ERR, ABS(RES(J)))
IF (ABS(RXX-RTU(IDN)) .GT. DELM) THEN
DELM = ABS (RXX-RTU(IDN))
RNEW = RXX
ENDIF
3078 CONTINUE
RTU(IDN) = MAX (RTU(IDN), 1.E-30)
C
DO 3018 J = 1, NODAR(1,IDN)
IDT = NODAR (4+J, IDN)
ID = TUBNOD (3, IDT)
IDP = TUBNOD (4, IDT)
IF (ID.GT.0 .AND. IDP.GT.0 ) THEN
BASE2 = (XS(ID)-XS(IDP))**2 + (YS(ID)-YS(IDP))**2
BASE = SQRT (BASE2)
RLEG = BASE /2.
XMD = (XS(ID) + XS(IDP)) /2.
YMD = (YS(ID) + YS(IDP)) /2.
HIGHT2 = (XNOD(IDN) -XMD)**2 + (YNOD(IDN) -YMD)**2
HIGHT = SQRT (HIGHT2)
3018 CONTINUE
C
C--- COMPUTE EFFECTIVE PORE RADII ---
C
PORSUM = 0.
DO 3017 IID = 1, NNOD
AFORE(IID) = 0.
DO 3016 J = 1, NODAR(1,IDN)
IDT = NODAR (4+J, IDN)
ID = TUBNOD (3, IDT)
IDP = TUBNOD (4, IDT)
IF (ID.GT.0 .AND. IDP.GT.0 ) THEN
BASE2 = (XS(ID)-XS(IDP))**2 + (YS(ID)-YS(IDP))**2
BASE = SQRT (BASE2)
RLEG = BASE /2.
XMD = (XS(ID) + XS(IDP)) /2.
YMD = (YS(ID) + YS(IDP)) /2.
HIGHT2 = (XNOD(IDN) -XMD)**2 + (YNOD(IDN) -YMD)**2
HIGHT = SQRT (HIGHT2)
3018 CONTINUE
C--- COMPUTE EFFECTIVE PORE RADII ---
3017 CONTINUE
3016 CONTINUE
3015 CONTINUE
3014 CONTINUE
3013 CONTINUE
3012 CONTINUE
3011 CONTINUE
3010 CONTINUE
3009 CONTINUE
3008 CONTINUE
3007 CONTINUE
3006 CONTINUE
3005 CONTINUE
3004 CONTINUE
3003 CONTINUE
3002 CONTINUE
3001 CONTINUE
3000 CONTINUE
2999 CONTINUE
2998 CONTINUE
2997 CONTINUE
2996 CONTINUE
2995 CONTINUE
2994 CONTINUE
2993 CONTINUE
2992 CONTINUE
2991 CONTINUE
2990 CONTINUE
2989 CONTINUE
2988 CONTINUE
2987 CONTINUE
2986 CONTINUE
2985 CONTINUE
2984 CONTINUE
2983 CONTINUE
2982 CONTINUE
2981 CONTINUE
2980 CONTINUE
2979 CONTINUE
2978 CONTINUE
2977 CONTINUE
2976 CONTINUE
2975 CONTINUE
2974 CONTINUE
2973 CONTINUE
2972 CONTINUE
2971 CONTINUE
2970 CONTINUE
2969 CONTINUE
2968 CONTINUE
2967 CONTINUE
2966 CONTINUE
2965 CONTINUE
2964 CONTINUE
2963 CONTINUE
C
C--- COMPUTE EFFECTIVE PORE RADII ---
C
PORSUM = 0.
DO 3017 IID = 1, NNOD
AFORE(IID) = 0.
DO 3016 J = 1, NODAR(1,IDN)
IDT = NODAR (4+J, IDN)
ID = TUBNOD (3, IDT)
IDP = TUBNOD (4, IDT)
IF (ID.GT.0 .AND. IDP.GT.0 ) THEN
BASE2 = (XS(ID)-XS(IDP))**2 + (YS(ID)-YS(IDP))**2
BASE = SQRT (BASE2)
RLEG = BASE /2.
XMD = (XS(ID) + XS(IDP)) /2.
YMD = (YS(ID) + YS(IDP)) /2.
HIGHT2 = (XNOD(IDN) -XMD)**2 + (YNOD(IDN) -YMD)**2
HIGHT = SQRT (HIGHT2)
3018 CONTINUE
C--- COMPUTE EFFECTIVE PORE RADII ---
AFORE(IID) = 0.
DTHETA = ATAN (HIGHT /RLM)
IF (RADI(ID)+RADI(IDP) .LE. BASE) THEN
APART = DTHETA /2. *(RADI(ID)**2 + RADI(IDP)**2)
ELSE
RLAP = (RADI(ID) + RADI(IDP) - BASE) /2.
PHI1 = ACOS ((RADI(ID)-RLAP) /RADI(ID))
HLAP = RADI(ID) *SIN (PHI1)
PHI2 = ACOS ((RADI(IDP)-RLAP) /RADI(IDP))
ATRI = (RADI(ID)-RLAP) *HLAP /2.
APART = (DTHETA-PHI1) /2. *RADI(ID)**2 + ATRI
ENDIF
APART = APART + (DTHETA-PHI2) /2. *RADI(IDP)**2 + ATRI
ENDIF
DAREA = MAX (0., HIGHT *RLEG - APART)
ELSE
IDSAV = ID
ID = MAX (ID, IDP)
IDP = MIN (IDSAV, IDP)
HYPOT2 = (XS(ID)-XNOD(IDN))**2 + (YS(ID)-YNOD(IDN))**2
HYPOT = MAX (1.E-10*XMAX, SQRT (HYPOT2))
IF (IDP/2*2 .EQ. IDP) THEN
RLAP = HIGHT
PHI = ACOS ((RADI(ID)-RLAP) /RADI(ID))
ELSE
HLAP = RADI(ID) *SIN (PHI)
ATRI = (RADI(ID)-RLAP) *HLAP /2.
APART = APART + (DTHETA-PHI) /2. *RADI(ID)**2
ENDIF
DAREA = MAX (0., HIGHT *RLEG /2. - APART)
ENDIF
APORE(IDN) = APORE(IDN) + DAREA
CONTINUE
END
C--- PLOTT NODES ---
CONTINUE
C--- PLOTT NODES ---
C IF (IPLTN.NE.0) THEN
DO 1809 IDN = 1,NNOD
XSHIF = 0.
YSHIF = 0.
PAC = 1.
1809 IF (NODAR(8,IDN).EQ.0) THEN
SIZ = 0.
DSHIF = 0.
DO 1808 J=1,NODAR(1,IDN)
IDF = NODAR(4+J,IDN)
XSHIF = XSHIF + TUBAR(7,IDT)
YSHIF = YSHIF + TUBAR(8,IDT)
1808 IF (NODAR(8,IDN).EQ.0) THEN
SIZ = 0.
DSHIF = 0.
CONTINUE
C
01737      DSHIF = DSHIF + SQRT(TUBAR(7,IDT)**2 + TUBAR(8,IDT)**2)
01738      SIZ = SIZ + TUBAR(2,IDT)
01739  1808 CONTINUE
01740      XSHIF = XSHIF /NODAR(1,IDN)
01741      YSHIF = YSHIF /NODAR(1,IDN)
01742      SIZ = SIZ /NODAR(1,IDN)
01743      FAC = 0.8 *MIN(DSHIF,SIZ) /XMAX *XAXIS /0.082
01744      ENDIF
01745      FAC = MAX (FAC, .01)
01746      XB(1) = XNOD(IDN) + XSHIF
01747      YB(1) = YNOD(IDN) + YSHIF
01748      IF (IPLTN.NE.1) FAC = 1.
01750      CALL SCLPIC(FAC)
01751      C
01752      XB(1) = XIU(IDN)
01753      YB(1) = YIU(IDN)
01754      FAC = 2. *XIU(IDN) /XMAX *XAXIS /0.082
01755      FAC = MAX (FAC, 0.01)
01756      FAC = MIN (FAC, 0.2 *XMAX /XMAX *XAXIS /0.082 )
01757      CALL SCLPIC(FAC)
01758      CALL MARKER(15)
01759      C
01760      CALL SCLPIC(0.5)
01761      CALL CURVE(XB,YB,1,1)
01762  1809 CONTINUE
01763      CALL SCLPIC(1.)
01764      ENDIF
01765      C
01766      CALL CURVE(XB,YB,2,0)
01767      C

BIOREM$MAIN

01768      IF (IPLTN.NE.0) THEN
01769      CALL BLREC(-0.2*XAXIS,-0.2*YAXIS,0.2*XAXIS,1.4*YAXIS,0)
01770      CALL BLREC(-0.2*XAXIS,-0.2*YAXIS,1.4*XAXIS,0.2*YAXIS,0)
01771      CALL BLREC( 1.0*XAXIS,-0.2*YAXIS,0.2*XAXIS,1.4*YAXIS,0)
01772      CALL BLREC(-0.2*XAXIS, 1.0*XAXIS,1.4*XAXIS,0.2*YAXIS,0)
01773      DO 1803 IDT=1,NTUB
01774      XB(1) = TUBAR(3,IDT)
01775      YB(1) = TUBAR(4,IDT)
01776      XB(2) = TUBAR(5,IDT)
01777      YB(2) = TUBAR(6,IDT)
01778      WID = MIN (TUBAR(2,IDT), 3.*TUBAR(1,IDT))
01779      IF (TUBAR(2,IDT) .LT. 1.1E-2 *SPAC) WID = 0.
01780      WID = 0.98 *MAX (WID, 0.)
01781      WID = WID /XMAX *XAXIS
01782      IF (IPLTN.EQ.1) CALL THKCRV(WID)
01783      CALL CURVE(XB,YB,2,0)
01784  1803 CONTINUE
01785      CALL RESET('THKCRV')
01786      ENDIF
01787      C
01788      C--- PLOTT TUBE/NODE TRIOS ---
01789      C
01790      IPLTTRI = 0
01791      IF (IPLTTRI.NE.0) THEN
01792      DO 1793 IDN = 1, NNOD
01793      XB(1) = XNOD(IDN)
01794      YB(1) = YNOD(IDN)

A31
ICH = NODAR(1, IDN)
DO 1793 J = 1, ICH
   ID = NODAR(J+1, IDN)
   XB(2) = XNOD(ID)
   YB(2) = YNOD(ID)
   CALL CURVE(XB, YB, 2, 0)
1793 CONTINUE
CALL RESET('DASH')
ENDIF
C SHADE BACKGROUND
C IF (ISHAD.NE.0) THEN
   XBAl(1) = 0.
   YBAl(1) = 0.
   XBAl(2) = XMAX
   YBAl(2) = 0.
   XBA2(1) = 0.
   YBA2(1) = YMFX
   XBA2(2) = XMAX
   YBA2(2) = YMFX
   IF (XBAl(2).LT.0.9) THEN
      CALL SHDPAT (17)
      IF (ISHAD.LT. 0) CALL SHDPAT(16)
      CALL SHDCRV(xBA1, YBAl, 2, xBA2, YBA2, 2)
   ENDIF
   CALL ENDPL(0)
   PRIwT*, ISEED0
C CHECK CONSISTANCY
C IFAS = 1
DO 502 IDN = 1, NNOD
   DO 503 L = 1, NODAR(1, IDN)
      NUM = NODAR(L+1, IDN)
      IDT = NODAR(L+4, IDN)
      L2OK = 0
      NOK = 0
      DO 504 L2 = 1, NODAR(1, NUM)
         IF (NODAR(L2+1, NUM) .EQ. IDN) THEN
            L2OK = L2
            NOK = NOK + 1
         ENDIF
      504 CONTINUE
      IF (NOK .EQ. 0) THEN
         IFAS = 0
         PRINT*, 'NO NODE CORRESPONDANCE'
         PRINT9104, IDN, NUM, L, IDT, XNOD(IDN), YNOD(IDN)
         DO 465 IKL = 1, NODAR(1, NUM)
            N2 = NODAR(IKL+1, NUM)
         465 CONTINUE
ELSEIF (NOK .EQ. 1) THEN
   IPAS = 0
   PRINT*, 'NO TUBE CORRESPONDANCE'
   PRINT1910, IDN, NODAR(1, IDN), L, IDT,
   1 NUM, NODAR(1, NUM), L2CK, IDT2
ENDIF
ELSE
   IPAS = 0
   PRINT*, 'MULTIPLE NODE CORRESPONDANCE'
ENDIF
CONTINUE
IF (IPAS .EQ. 1) PRINT*,'CORRESPONDANCE IS OK'
PRINT*, 'INPUT ICONT 1 = CONT 2 = RESTART'
IF (ICONT.EQ.2) GO TO 1600
READ*, ICONT

COMPUTE TUBE DIAGNOSTICS
---
SUMT = 0.
NSUM = 0
DMAX = 0.
DO 1771 IDT = 1, NTUB
   IDBOUN = MIN (TUBNO(3,IDT), TUBNO(4,IDT))
   IF (IDBOUN .GT. 0) THEN
      NSUM = NSUM + 1
   ENDIF
1771 CONTINUE
DTBAR = SUMT /NSUM
SDEVT = 0.
IFRE = NTUB /20
DO 1773 IBIN = 1, IFRE
   C FREQ(IBIN) = 0.
   DrMIN = 0.
   DTMFX = 2.*DTBAR
   DO 1772 IM' = 1, NTUB
      IDBOUN = MIN (TUBNO(3,IDT), TUBNO(4,IDT))
      IF (IDBOUN .GT. 0) THEN
         SDEVT = SDEVT + (TUBBAR(2,IDT) - DTMIN)**2
         FREQ(IBIN) = FREQ(IBIN) + 1.
      ENDIF
1772 CONTINUE
SDEVT = SQRT(SDEVT /NSUM)
PRINT980, 'SDEV', SDEVT
PRINT981, 'NB0U', NTUB-NSUM
PRINT981, 'DBAR', DTMFX
WRITE(15,901) IFRE
A33
**COMPUTE BOUNDARY LENGTHS**

```fortran
DO 1776 J = 1, NBOX
  LBOUN(J) = 0.
  DO 1777 I = 1, NSEED
    IBOU = -IDBC(I)
    IF (IBOU .GT. 0) THEN
      LB = LEC(I)
      LBOUN(IBOU) = LBOUN(IBOU) + LSID(LB, I)
    ENDIF
  1777 CONTINUE
  LBOUN(J) = LBOUN(J) + LSID(J, I)
  PERI(I) = PERI(I) + LSID(L, I)
  IDP = IFRM(L, I)
  IF (IDP .LT. 0) THEN
    THEN = PI/2. + (-IDP - 1) * 2. * PI/NBOX
    DXDL = COS(THET)
    DYDL = SIN(THET)
    DXXX = XS(IDP) - XS(I)
    DYYY = YS(IDP) - YS(I)
    DLLL = SQRT(DXXX**2 + DYYY**2)
    DXDL = DXXX / DLLL
    DYDL = DYYY / DLLL
  ELSE
    PERI(I) = PERI(I) + LSID(L, I) * ABS(DXDL) / 2.
    APRX(I) = APRX(I) + LSID(L, I) * ABS(DYDL) / 2.
  ENDIF
  CONTINUE
```

**COMPUTE PERIMETER AND PROJECTED AREAS**

```fortran
DO 1778 I = 1, NSEED
  PERI(I) = PERI(I) + LSID(L, I)
  IDP = IFRM(L, I)
  IF (IDP .LT. 0) THEN
    THEN = PI/2. + (-IDP - 1) * 2. * PI/NBOX
    DXDL = COS(THET)
    DYDL = SIN(THET)
    DXXX = XS(IDP) - XS(I)
    DYYY = YS(IDP) - YS(I)
    DLLL = SQRT(DXXX**2 + DYYY**2)
    DXDL = DXXX / DLLL
    DYDL = DYYY / DLLL
  ELSE
    PERI(I) = PERI(I) + LSID(L, I) * ABS(DXDL) / 2.
    APRX(I) = APRX(I) + LSID(L, I) * ABS(DYDL) / 2.
  ENDIF
  CONTINUE
```

**BIOREMSMAIN**

```fortran
DO 1785 I = 1, NSEED
  IF (IDBC(I) .LT. -NBOX) PRINT902, I, IDBC(I), XS(I), YS(I)
  CONTINUE
```

```fortran
CKSUM = 0.
```

```fortran
CKSUM = CKSUM + (1.*K)*(1.*FSID(K))/NSUM
```

```fortran
PRINT2, FSUM, (1.*FSUM) / NSUM / (1./IFRE)
```

```fortran
CWRITE(15, 900) (IBIN-0.5) / IFRE, FDIS, FSUM
```

```fortran
PRINT901, IFRE, FDIS, FSUM
```

```fortran
DO 1631 K = 1, 20
  PRINT2, FSUM, (1.*FSUM) / NSUM
  PRINT2, 1631
```

```fortran
PRINT2, K, FSID(K), (1.*FSID(K)) / NSUM
```

```fortran
PRINT902, JMIN, JMAX
```

```fortran
CAVE = GAVE / NSUM * SQRT(1.*NSEED)
```

```fortran
GMIN = GMIN / NSUM * SQRT(1.*NSEED)
```

```fortran
GMAX = GMAX / NSUM * SQRT(1.*NSEED)
```

```fortran
CMIN = CMIN * SQRT(1.*NSEED)
```

```fortran
```
C\text{MAX} = \text{CMAX} \times \sqrt{1. \times \text{NSEEK}}
01970 C
01971 \text{PRINT900,CMIN}
01972 \text{PRINT900,CMIN}
01973 \text{PRINT900,SAVE}
01974 \text{PRINT900,CMAX}
01975 \text{PRINT900,CMAX}
01976 \text{READ*}
01977 \text{CONTINUE}
01978 C
01979 C--- \text{PRINT NODE/TUBE CORRESPONDANCE STUFF} ---
01980 C
01981 \text{DO 1733 IDN = 1, 0 *NNOD}
01982 \text{ICH = NODAR(1,IDN)}
01983 \text{DO 1733 J = 1, ICH}
01984 \text{IDT = NODAR(4+J,IDN)}
01985 \text{PRINT904,IDN,NODAR(1+J,IDN),TUBNOD(1,IDT),TUBNOD(2,IDT)}
01986 \text{1733 CONTINUE}
01987 C
01988 C--- \begin{code} BEGIN CALCULATIONS OF PRESSURE FIELD \end{code} ---
01989 C
01990 \text{PIN = 1000.}
01991 \text{PEX = 1.}
01992 \text{TMP = 298.}
01993 \text{RGAS = 287.}
01994 \text{PRINT*, 'INPUT PIN,PEX',PIN,PEX}
01995 \text{READ*,PIN,PEX}

\text{BIOREM} \text{MAIN}
01996 \text{NSOR = 1000}
01997 \text{WGT = 0.5}
01998 \text{PRINT*, 'INPUT NSOR, WGT',NSOR,WGT}
01999 \text{READ*,NSOR,WGT}
02000 \text{ABSERR = 1.E-20}
02001 \text{RELERR = 1.E-8}
02002 \text{PRINT*, 'INPUT ABSERR,RELERR',ABSERR,RELERR}
02003 \text{READ*,ABSERR,RELERR}
02004 \text{IPLTPR = 0}
02005 \text{IPLTFP = 0}
02006 \text{IPLTPFY = 0}
02007 \text{IF (IDBIN .NE. IDBEV) THEN}
02008 \text{IPLTPF = 0}
02009 \text{IPLTPFY = 0}
02010 \text{ENDIF}
02011 \text{PRINT*, 'INPUT IPLTPR,IPLTFP,IPLTPFY',IPLTPR,IPLTFP,IPLTPFY}
02012 \text{READ*,IPLTPR,IPLTFP,IPLTPFY}
02013 \text{ICOMP = 0}
02014 \text{PRINT*, 'INPUT ICOMP: 0 = INCOMP 1 = COMP',ICOMP}
02015 \text{READ*,ICOMP}
02016 \text{KPAR = 300}
02017 \text{DISTR = 0.5}
02018 \text{PRINT*, 'INPUT NUMBER OF TRACER PARTICLES AND TIME',KPAR,DISTR}
02019 \text{READ*,KPAR,DISTR}
02020 \text{PECL = 1.E10}
02021 \text{ISED = 12344321}
02022 \text{IF (KPAR .GT. 0) THEN}
02023 \text{PRINT*, 'INPUT PECLET NUMBER',PECL}
02024 \text{READ*,PECL}
02025 \text{IMICF = 0}
02026 \text{PRINT*, 'SPATIAL DIST OR MICRO-FINGERS?', 0 = DIST 1 = MF',IMICF}
READ*,IMICF
IPARA = 1
PRINT*, 'PARABOLIC VEL PROFILE ? 0 = NO 1 = YES', IPARA
READ*,IPARA
IMIX = 0
PRINT*, 'MIX NODE STREAMLINES ? 0 = NO 1 = YES', IMIX
READ*,IMIX
IUNI = 0
PRINT*, 'UNIFORM PARTICLE DIST IN INLET TUBE ? 0 = NO 1 = YES',
1
PRINT*,IUNI
IRANS = 0
PRINT*, 'RANDOMIZE STREAMLINE AT EVERY STEP ? 0 = NO 1 = YES',
1
READ*,IRANS
IZZZ = 0
PRINT*, 'EXIT TUBE ON ENTRANCE STREAMLINE ? 0 = NO 1 = YES',
1
READ*,IZZZ
IPLTR = 0
IF (IMICF .NE. 1) THEN
PRINT*, 'PARTICLE TRACES ? 0 = NO 1 = YES',IPLTR
READ*,IPLTR
ENDIF
PRINT*, 'INPUT SEED FOR TRACER INJECTION', ISED
READ*,ISED
IPRINI = 1
IF (IMETH .LT. 0) IPRINI = 0
IF (IcoWr .NE. 0) IPRINI = 0
PRINT*, 'DO YOU WANT PRESSURES (RE)INITIALIZED', IPRINI
READ*,IPRINI
XXXO = 0.5
ECGE = 0
PRINT*, 'AVERAGES OR LOCAL APERTURE: 1 = AVE 2 = LOC ?', KMETHOD
READ*,KMETHOD
IGEO = 2
PRINT*, 'INPUT IGEO', IGEO
READ*,IGEO
IOK = 0
PRINT*, 'INPUT OK', IOK
READ*,IOK
IF (IOK.EQ.0) GO TO 1620
C--- INITIALIZE PRESSURES ---
IF (IPRINI .NE. 0) THEN
DO 440 IDN = 1, NNOD
THET = XNOD(IDN) / XMAX
IF (IDBIN .NE. IDBEV) THET = XNOD(IDN) / XMAX
IF ((IDBIN.EQ.2) .OR. (IDBIN.EQ.3)) THET = 1. - THET
PRES(IDN) = PIN
IF (IcoWr.EQ.0) THEN
PRES(IDN) = PIN - (PIN-PEx)*THET
IF (IPRINI .LT. 0.) PRES(IDN) = PEx
CEND
ELSE
PRES(IDN) = SQRT (PIN**2 - (PIN**2-PEX**2)*THET)
ENDIF
PRESO(IDN) = PRES(IDN)
CONTINUE
ENDIF
C
TIM = 0.
DTIM = 0.1
ISTOP = 0
MAX = 1
TOUT = 0
C
C--- UPDATE CURRENT VALUES OF PRESSURE ---
C
DO 710 K = 1, MAX
IF (ISTOP.EQ.1) GO TO 830
TIM = TIM + DTIM
IF (K.GT.1) THEN
DO 321 IDN = 1, NNOD
PRINT*, IDN, PRES0 (IDN)
PRES(IDN) = PRESO(IDN)
ENDIF
CONTINUE
C
C--- COMPUTE SPECIFIC SURFACE AREA ---
C
SUMA = 0.
SUMV = 0.
DO 312 ID = 1, NPAR
IF (IGEO .EQ. 2) THEN
SUMA = SUMA + PI *DIAM(ID)
SUMV = SUMV + PI *DIAM(ID)**2 /4.
ELSE
SUMA = SUMA + PI *DIAM(ID)**2
SUMV = SUMV + PI *DIAM(ID)**3 /6.
ENDIF
CONTINUE
C
C--- COMPUTE EFFECTIVE PORE RADII ---
C
FORSUM = 0.
DO 317 IDN = 1, NNOD
APORE(IDN) = 0.
DO 316 J = 1, NODAR(IDN)
IDT = NODAR (4+J, IDN)
ID = TUBNOD (3, IDT)
IDP = TUBNOD (4, IDT)
IF (ID.GT.0 .AND. IDP.GT.0 ) THEN
BASE2 = (XS(ID)-XS(IDP))**2 + (YS(ID)-YS(IDP))**2
BASE = SQRT (BASE2)
RLEG = BASE /2.
XMID = (XS(ID) + XS(IDP)) /2.
YMID = (YS(ID) + YS(IDP)) /2.
HEIGHT2 = (XNOD(IDN)-XMID)**2 + (YNOD(IDN)-YMID)**2
ENDIF
HIGHT = SQRT (HIGHT2)
DTHETA = ATAN (HIGHT / /RLEG) 
IF (RADI(ID)+RADI(IDP) .LE. BASE) THEN 
 APART = DTHETA /2. * (RADI(ID)**2 + RADI(IDP)**2)
ELSE 
 Rlap = (RADI(ID) + RADI(IDP) - BASE) /2.
 PHI1 = ACOS ((RADI(ID) - Rlap) / RADI(ID))
 Hlap = RADI(ID) * SIN (PHI1)
 PHI2 = ACOS ((RADI(IDP) - Rlap) / RADI(IDP))
 ATRI = (RADI(ID) - Rlap) * Hlap /2.
 APART = (DTHETA-PHI1) /2. * RADI(ID)**2 + ATRI
 APART = APART + (DTHETA-PHI2)/2. * RADI(IDP)**2 + ATRI
ENDIF 
DAREA = MAX (0., HIGHT * RLEG - APART)
ELSE 
IDSAV = ID
 ID = MAX (ID, IDP)
 IDP = MIN (IDSAV, IDP)
 HYPOT2 = (XS(ID)-XNOD(IDN))**2 + (YS(ID)-YNOD(IDN))**2
 HYPOT = MAX (1.E-10*XMAX, SQRT (HYPOT2))
 IF (IDP/2*2 .EQ. IDP) THEN 
 RLAP = RADI(ID) - HIGHT
 PHI = ACOS ((RADI(IDP)-RLEG) / RADI(ID))
 HLAP = RADI(ID) * SIN (PHI)
 ATRI = (RADI(ID)-RLEG) * HLAP /2.
 APART = (DTHETA-PHI) /2. * RADI(ID)**2 + ATRI
ENDIF 
DAREA = MAX (0., HIGHT * RLEG /2. - APART)
ENDIF 
APORE(IDN) = APORE(IDN) + DAREA 
C--- COMPARE MAX NODE DIAMETERS --- 
C
DO 318 IDN = 1, NNOD
 XTU(IDN) = XNOD(IDN)
 YTU(IDN) = YNOD(IDN)
 RTU(IDN) = XMAX /100.
 RMIN = XMAX
DO 319 J = 1, NODAR(1,IDN)
 IDT = NODAR (4+J, IDN)
 ID = TUBNOD (3, IDP)
 IF (ID.GT.0 .AND. IDP.GT.0 ) THEN 
 XAA(J,1) = -2. * (XS(ID) - XS(IDP)) 
 C316 CONTINUE 
 C317 CONTINUE 
 EPSVOID = FORSUM / (XMAX* YMAX) 
 C318 CONTINUE 
 C319 CONTINUE 
 C
XAA(J,2) = -2. * (YS(ID) - YS(IDP))
XAA(J,3) = -2. * (RADI(ID) - RADI(IDP))
RHS(J) = RADI(ID)**2 - RADI(IDP)**2
QHS(J) = RADI(ID)**2 + RADI(IDP)**2
CHS(J) = RADI(ID) - RADI(IDP)

DIS = SQRT ((XS(ID) - XS(IDP))**2 + (YS(ID) - YS(IDP))**2)

RTRY = DIS - RADI(ID) - RADI(IDP)
RMIN = MIN (RMIN, RTRY)

ELSE
XAA(J,1) = 0.
XAA(J,2) = 0.
XAA(J,3) = 0.
RHS(J) = 0.
QHS(J) = 0.
IDSAV = ID
ID = MAX (ID, IDP)
IDP = MIN (IDSAV, IDP)

IF (IDP .EQ. -2 .OR. IDP .EQ. -4) THEN
XAA(J,1) = 1.
RHS(J) = 0.
ENDIF

ELSE
XAA(J,3) = 1.
RHS(J) = 0.
ENDIF

CONTINUE

RHS(J) = 0.
IF (IDP .EQ. -3) RHS(J) = XMAX

RMIN = MIN (RMIN, RTRY)

BIOREMS

RHS(J) = 0.
IF (IDP .EQ. -3) RHS(J) = YMAX

ENDIF

DO 377 ITS = 1, 50

RTUF = (XS(ID) - XTU(IDN))**2 + (YS(ID) - YTU(IDN))**2
RXX = SQRT (RTUF) - RADI(ID)
RES(J) = SQRT(RTUF) - RADI(ID) - RTU(IDN)
ERR = MAX (ERR, ABS(RES(J)))

IF (ABS(RXX-RTU(IDN)) .GT. DELM) THEN
DEL = ABS (RXX-RTU(IDN))

RES = RXX

ENDIF

376 CONTINUE

IF (ERR .LT. 1.E-6*XMAX) GO TO 378

C

377 CONTINUE

C

RTU(IDN) = MAX (RTU(IDN), 1.E-30)

C

318 CONTINUE

C

C--- COMPUTE EFFECTIVE TUBE DIAMETERS FOR 3-D GEOMETRY

C

IF (IGEO .EQ. 3) THEN

BBAR = 0.

MSUM = 0.

ELBAR = 0.

DO 344 IDT = 1, NTUB

IN = TUBNOD(1,IDT)

IDNP = TUBNOD(2,IDT)

AR1 = PI *RTU(IDN)**2

AR2 = PI *RTU(IDNP)**2

AR1 = MAX (AR1, 1.E-20)

AR2 = MAX (AR2, 1.E-20)

TUL = TUBAR(1,IDT)

TUL1 = TUL /2.

TUL2 = TUL /2.

AEFF = (TUL1 *AR1 + TUL2 *AR2) /TUL

RAD1 = SQRT (AR1 /PI)

RAD2 = SQRT (AR2 /PI)

RK1 = MAX (RAD1**2 /8., 1.E-20)

RK2 = MAX (RAD2**2 /8., 1.E-20)

IF (RK1 .LT. RK2) THEN

RKEFF = RK1 /BOT

ELSE

RKEFF = RK2 /BOT

ENDIF

RKEFF = MAX (RKEFF, 1.E-30)

DEFF = SQRT (32. *RKEFF)

DPER(IDT) = SQRT (12. *RKEFF)

DVOL(IDT) = AEFF

DDIF(IDT) = TUL / (TUL1 /AR1 + TUL2 /AR2)

TUBAR(14,IDT) = 2. *SQRT (AEFF /PI)

TUBAR(15,IDT) = DDIF(IDT)

IF (TUBAR(9,IDT) .LT. 0) THEN

DPER(IDT) = 0.

DVOL(IDT) = 0.

DDIF(IDT) = 0.

TUBAR(14,IDT) = 0.

TUBAR(15,IDT) = 0.

ENDIF

IF (TUBAR(9,IDT) .GT. 0) THEN

MSUM = MSUM + 1

BBAR = BBAR + TUBAR(14,IDT)

ELBAR = ELBAR + TUBAR(1,IDT)

ENDIF
02317     ENDIF
02318     CONTINUE
02319     BBAR = BBAR/MSUM
02320     ELBAR = ELBAR/MSUM
02321     ELSE
02322     C
02323     C--- COMPUTE CHARACTERISTIC 2-D PASSAGE DIMENSIONS ---
02324     C
02325     IPIK = 1
02326     BBAR = 0.
02327     MSUM = 0
02328     ELBAR = 0.
02329     DO 371 IDT = 1, NTUB
02330     ID = TUNOD(3,IDT)
02331     IDP = TUNOD(4,IDT)
02332     IF (ID.GT.0 .AND. IDP.GT.0) THEN
02333     R1 = RADI(ID)
02334     R2 = RADI(IDP)
02335     ELSE
02336     ID = MAX (ID, IDP)
02337     R1 = RADI(ID)
02338     ENDIF
02339
02340     DNEK = TUBAR(2,IDT)
02341     RLEN = MIN (MIN (TUBAR(1,IDT), 1.999*R1), 1.999*R2)
02342     CALL DELTAS (IPIK, R1, R2, DNEK, RLEN, DDIF(IDT),
02343             1)
02344     TUBAR(14,IDT) = DVOL(IDT)
02345     TUBAR(15,IDT) = DDIF(IDT)
02346     IF (TUBAR(9,IDT).GT.0) THEN
02347     MSUM = MSUM + 1
02348     BBAR = BBAR + TUBAR(1,IDT)
02349     ELSE
02350     ENDIF
02351     CONTINUE
02352     BBAR = BBAR/MSUM
02353     ELBAR = ELBAR/MSUM
02354     ENDIF
02355     C
02356     C--- CYCLE OVER RELAXATION STEPS ---
02357     C
02358     ICONV = 1
02359     DO 210 ICY = 1, NSOR
02360     IGO = IGO + 1
02361     ERR = 0.
02362     ERRMDT = 0.
02363     C
02364     C--- COMPUTE MASS FLOW RATES AND WEIGHTS FOR EACH NODE.
02365     C     SIGN CONVENTION FOR FLOW IS POSITIVE FOR POSITIVE FLOW FROM
02366     C     NODE N1 TO NODE N2; PRESSURE GRADIENT IS (P2-P1)/L.
02367     C
02368     DO 220 IDN = 1, NNOD
02369     VOL = 0.
02370     DOTM = 0.
02371     WDTSUM = 0.
02372     DDPDRHO = 1.
02373     IF (ICOMP.EQ.1) DDPDRHO = RGAS *TMP
02374     DO 230 L = 1, NODAR(1,IDN)
NUM = NODAR(L+1, IDN)
IDT = NODAR(L+4, IDN)
N1 = TUBMOD(1, IDT)
TUL = TUBAR(1, IDT)
TUD = TUBAR(2, IDT)
ALF = DPER(IDT)**2 /12.
VOL = VOL + DVOL(IDT) * TUL /2.
PBAR = (PRES(IDN)+PRES(NUM))/2.
RHO = 1.
IF (ICaMp.EQ.1) RHO = PBAR /RGAS /TMP
mu = 1.
DPDL = (PRES(IDN)-PRES(NUM)) /TUL
VEL = - ALF /Rhu *DPDL
DOT = DOT + VEL * RHO * DVOL(IDT)
WTSUM = WTSUM + ALF/Rhu/TUL * RHO * DVOL(IDT)
ISGN = 1
IF (NUM .NE. N1) ISGN = - 1
TUBAR(10, IDT) = ISGN * VEL
TUBAR(11, IDT) = ISGN * VEL * RHO * DVOL(IDT)
TUBAR(12, IDT) = DVOL(IDT) * TUL
BIORECONS

TUBAR(13, IDT) = ISGN * VEL * DVOL(IDT)

C--- COMPUTE RESIDUAL OF GOVERNING EQUATION FOR NODE IDN ---

IF (NODAR(8, IDN).EQ.-IDBIN .OR. NODAR(8, IDN).EQ.-IDBOU) THEN
DPRES(IDN) = 0.
ELSE
PDOT = (PRES(IDN)-PRESO(IDN)) /DTIM
RESP = 0.0000000 * PDOT - DOTM /VOL * DPDWO
WEIGHT = WST /VOL /DPDRHO
DPRES(IDN) = - WEIGHT * RESP
ENDIF

C--- UPDATE Pressures and NODE IDN ---
PRES(IDN) = PRES(IDN) + DPRES(IDN)
IF (NODAR(8, IDN).EQ.-IDBIN) PRES(IDN) = PIN
IF (NODAR(8, IDN).EQ.-IDBOU) PRES(IDN) = PEX

C--- COMPUTE LOCAL ERROR ESTIMATE AT NODE IDN ---
TOL = RELERR *ABS(PIN-PEX) * WOT
ERRIDN = ABS(DPRES(IDN)) /TOL
IF (ERRIDN .GT. ERR) THEN
ERR = ERRIDN
IINERR = IDN
ENDIF

C--- NOTE MAXIMUM NET FLOW RATE AT INTERIOR POINTS
IF ((NODAR(8, IDN).NE.-IDBIN) .AND. (NODAR(8, IDN).NE.-IDBOU)) THEN
IF (ABS(DOTM) .GT. ERRMDT) THEN
ERRMDT = ABS(DOTM)
IIMDT = IDN
ENDIF
ENDIF
CONTINUE IF (ERR.LT.1.0) GO TO 280

PRINT CURRENT MAXIMUM ERROR, NODE IDENTIFIERS AND PRESSURES ---

IF (MOD(IGO,100).EQ.0) THEN
  PRINT903, IGO, IDNERR, NODAR(8, IDNERR), ERRMDT,
  1 ERR, PRES(IDNERR), XNOD(IDNERR), YNOD(IDNERR)
ENDIF

CONTINUE

ICONV = 0

CONTINUE

C--- SUM FLOW RATES AND POROSITY

DOUTMIN = 0.

DOUT = 0.

BIOREMSMAIN

SUMDOT = 0.

DO 892 IDT = 1, NTUB

N1 = TUEINOD(1, IDT)

N2 = TUEINOD(2, IDT)

IF (NODAR(8, N1) .EQ. NODAR(8, N2)) GO TO 892

C--- GLOBAL MASS BALANCE FOR ALL INTERIOR NODES

C TUBE MUST HAVE ONE BOUNDARY ( < 0 ) AND ONE INTERIOR ( = 0 ) NODE

IF (NODAR(8, N1) .EQ. -IDBOU .OR. NODAR(8, N2) .EQ. -IDBOU) THEN
  ELSEIF (NODAR(8, N1) .EQ. -IDBIN .OR. NODAR(8, N2) .EQ. -IDBIN) THEN
    DOTMOUT = mur + ISNX * TUBAR(11, IDT)
    DOTMIN = DOTMIN + ISNX * TUBAR(11, IDT)
ENDIF

CONTINUE

PBAR = (PIN + PEX) / 2.

RHO = 1.

IF (ICOMP.EQ.1) RHO = PBAR / RGAS / TEMP

END
RMU = 1.
UBAR = (ABS(DOMMIN) + ABS(DOMOUT)) / 2. /YMAX /RHO
IF (IGEO.EQ.3) UBAR = UBAR /ELBAR
UBAR = UBAR /FORO
C--- OLD UREF WAS UBAR ---
UREF = UBAR
PERM = UBAR *RMU *XMAX /ABS(PIN-PEx)
SV0 = 4. /DREF
IF (IGEO.EQ.3) SV0 = 6. /DREF
PERM0 = EPSVOID**3 /((1.-EPSVOID)**2 /SV0**2
PERM0 = EPSMC**3 /((1.-EPSMC)**2 /SV0**2
ROZE = PERM0 /PERM
TAUO = DIAMP**2 *PI /2. /((1.-FORO)
EDSK = PERM /TAU0
PRINT900, UBAR, VBAR
PRINT900, (TUBAR(10,IT), IT=1,10)
PRINT900, (TUBAR(10,IT+NTUB/2), IT=1,10)
CONTINUE
VAVE = SMW /SUMV
RHOS = 1. - SUMV /YMAX /YMAX
PRINT900, VAVE, RHOS
PRINT900, SUMS, DOTMIN, DOTOUT, PERM, RKOZE
C--- SOLICIT INPUT FOR CONTINUATION ---
PRINT*, 'INPUT ICONT 1 = MORE CYCL 2 = NEW PRESS
READ*, ICONT
IF (ICONT.EQ.1) THEN
PRINT*, 'INPUT NSOR, W',NSOR,WGT
READ*, NSOR, WXP
GO TO 1630
GO TO 1620
GO TO 1600
ELSEIF (ICONT.EQ.2) THEN
ELSEIF (ICONT.EQ.4) THEN
ENDIF
C ADJUST TIME STEP
DO 832 IDT = 1, 0 *NTUB
N1 = TUBNOD(1,IDT)
N2 = TUBNOD(2,IDT)
TUL = TUBAR(1,IDT)
TUD = TuE3AR(2,IDT)
DOTM = TVBAR(11,IDT)
IF (NODAR(8,N1).EQ. NODAR(8,N2)) GO TO 832
IF (NODAR(8,N1).LT.0 .OR. NODAR(8,N2).LT.0) THEN
PRINT903, IDT, NODAR(8,N1), NODAR(8,N2), DOM, TUBAR(3,IDT),
TUBAR(4,IDT), TUBAR(5, IDT), TUBAR(6, IDT), PRES(N1), PRES(N2)
A44
ENDIF

C 832 CONTINUE
C 833 IDT = 1, 0 *NTUB
DO 833 IDT = 1, 0 *NTUB
N1 = TUBNOD(1, IDT)
N2 = TUBNOD(2, IDT)
TUL = TUBAR(1, IDT)
TUD = TUBAR(2, IDT)
DOM = TUBAR(11, IDT)
IF (NODAR(8, N1).NE. NODAR(8, N2)) GO TO 833
IF (NODAR(8, N1).LT.0 .OR. NODAR(8, N2).LT.0) THEN
PRINT903, IDT, NODAR(8, N1), NODAR(8, N2), DOM, TUBAR(3, IDT),
TUBAR(4, IDT), TUBAR(5, IDT), TUBAR(6, IDT), PRES(N1), PRES(N2)
ENDIF
C 832 CONTINUE
C PLOTT PRESSURE POINTS

BIOREM$MAIN

C IF (IPLTFR.EQ.1) THEN
CALL AREA2D(XAXIS, YAXIS)
CALL GRAP(XORIG, XSTP, XMAX, YORIG, YSTP, YMAX)
CALL THKFRM(.030)
CALL FRAME
PRINT*, ISEED0
PRINT913, NPR
PRINT913, NND
PRINT913, NTUB
PRINT913, NSOR
PRINT900, PIN
PRINT900, PEX
PRINT900, FORO
PRINT900, ABSSR
PRINT900, RELSSR
PRINT900, WGT
PRINT*, ' ' IF (ICONV.EQ.0) PRINT*, 'NO SOR CONV'
PRINT*, IGO
PRINT900, ERR
PRINT900, DMIN
PRINT900, DMIN
PRINT900, DMIN
PRINT900, DMIN
CALL MARKER(15)
DO 827 IDN = 1, NNOD
XB(1)=XNOD(IDN)
YB(1)=YNOD(IDN)
FAC = (PRES(IDN)-PEX)/(PIN-PEX)
FAC = FAC/SQR(1.*NNOD)*XAXIS/0.082
CALL SCLPIC(FAC)
CALL CURVE(XB,YB,1,1)
CONTINUE
CALL SCLPIC(1.)
CALL ENDFL(0)
ENDIF
C PLOTT X-P PRESSURE PROFILE
C XORIG = 0.
02607   YORIG = 0.
02608   YMAX2 = 1.
02609   CALL SCLPIC(1.)
02610   IF (IPLTPF.EQ.1) THEN
02611     IF (RATXY.GT.1.0) THEN
02612        XAXIS = 7.5
02613        YAXIS = 0.95775 *7.5
02614     ELSE
02615        XAXIS = 7.5 *RATXY
02616        YAXIS = 0.95775 *7.5
02617     ENDF
02618     XSTP=(YMAX-XORIG)/1.
02619     YSTP=(YMAX2-YORIG)/1.
02620     CALL AREAO2D(XAXIS, YAXIS)
02621     CALL Gراف(XORIG, XSTP, XMAX, YORIG, YSTP, YMAX2)
02622     CALL FRAME

BIOREMSMAIN

02623   CALL MARKER(15)
02624   PRINT*, ISEED0
02625   PRINT913,NPAR
02626   PRINT913,NNODE
02627   PRINT913,NTP
02628   PRINT913,NCR
02629   PRINT900,PIN
02630   PRINT900, PEX
02631   PRINT900, POR
02632   PRINT900, ABSPER
02633   PRINT900, RELPER
02634   PRINT900, WST
02635   PRINT*, ' '
02636   IF (ICONV.EQ.0) PRINT*, 'NO SOR CONV'
02637   PRINT901, IGO
02638   PRINT900, ERR
02639   PRINT900, DOTMIN
02640   PRINT900, DOTMAX
02641   PRINT900, SUMDET
02642   DO 826 IDN = 1, NNODE
02643       XB(1) = XNODE(IDN)
02644       YB(1) = (PRES(IDN) - PEX) / (PIN - PEX)
02645   CALL CURVE(XB, YB, 1, 1)
02646 826   CONTINUE
02647   IF (IDBIN .EQ. IDBEV) THEN
02648       JMX = 200
02649       DO 825 J = 1, JMX
02650          XB(J) = XMAX * (J-1.)/(JMX-1.)
02651          IF (IDBIN .EQ. 2) XB(J) = 1. - XB(J)
02652          IF (ICOMP.EQ.0) THEN
02653            YB(J) = 1. - XB(J) / XMAX
02654          ELSE
02655            YB(J) = SQRT (PIN**2 - (PIN**2-PIN**2)*XB(J)/XMAX)
02656          ENDIF
02657       CONTINUE
02658 825    CONTINUE
02659   CALL CURVE(XB, YB, JMX, 0)
02660   ENDF
02661   CALL DASH
02662   CALL CURVE(XB, YB, JMX, 0)
02663   ENDF
02664   C--- PLOT TUBE CONNECTIONS IN X-P SPACE ---
02665   C
02666   C

A46
DO 828 IDT = 1, NTUB
   IF (KBLOC(IDT) .EQ. 1) THEN
      DO 829 J = 1, 2
         IDN = TUBNOD(J, IDT)
         XB(J) = XNOD(IDN)
         YB(J) = (PRES(IDN) - PEX) / (PIN - PEX)
         CALL CURVE (XB, YB, 2, 0)
      ENDIF
      829 CONTINUE
   ENDIF
828 CONTINUE
   CALL RFSET('DASH')
   CALL ENDPL(0)
   C
   C PLOTT Y-P PRESSURE PROFILE
   C
   XORIG = 0.
   YORIG = 0.
   YMAX2 = 1.
   IF (IPLTPFY .EQ. 1) THEN
      IF (RATXY .GT. 1.0) THEN
         XAXIS = 7.5
         YAXIS = 0.95775 * 7.5
      ELSE
         XAXIS = 7.5 * RATXY
         YAXIS = 0.95775 * 7.5
      ENDIF
   ELSE
      XAXIS = 7.5 * RATXY
      YAXIS = 0.95775 * 7.5
   ENDIF
   XSTP= (XMAX-XORIG) / 1.
   YSTP= (YMAX2-YORIG) / 1.
   CALL AREA2D(XAXIS, YAXIS)
   CALL GRAF(XORIG, XSTP, XMAX, YORIG, YSTP, YMAX2)
   CALL FRAME
   CALL MARKER(15)
   PRINT*, , , , , , ISEED0
   PRINT913, NNOD
   PRINT913, NINC
   PRINT913, NTUB
   PRINT913, NINC
   PRINT900, PIN
   PRINT900, PEX
   PRINT900, PEX
   PRINT900, PEX
   PRINT900, ABERR
   PRINT900, RELERR
   PRINT900, WST
   PRINT*, '
   IF (ICONV .EQ. 0) PRINT*, 'NO SOR CONV'
   PRINT901, IGO
   PRINT900, ERR
   PRINT900, DOMIN
   PRINT900, DOMOUT
   PRINT900, SUMCOT
   DO 856 IDN = 1, NNOD
      XB(1) = YNOD(IDN)
      YB(1) = (PRES(IDN) - PEX) / (PIN - PEX)
      CALL CURVE (XB, YB, 1, 1)
   CONTINUE
   856 CONTINUE
   IF (IDEIN .NE. IDEEV) THEN
      JMX = 200
   ENDIF
   DO 855 J = 1, JMX

A47
XB(J) = YMAX * (J-1.)/(JMX-1.)

IF (IDBN .EQ. 3) XB(J) = 1. - XB(J)/YMAX

IF (IOMP .EQ. 0) THEN
  YB(J) = 1. - XB(J)
ELSE
  YB(J) = SQRT ((PIN**2 - (PIN**2-PEX**2)*XB(J)/YMAX)
ENDIF

CONTINUE
CALL CURVE(XB,YB,JMX,0)
ENDIF

--- PLOT TUBE CONNECTIONS IN Y-P SPACE ---

CALL DASH
DO 858 I=1, NIUBE
  IF (KBLCC(I) .EQ. 1) THEN
    XB(I) = YNOD>IDN
    YB(I) = (PRES(I) -Pm) / (Pm-Pd)
    CALL CURVE(XB,YB,2,0)
  ENDIF
END

--- UPDATE VALUES ---
DO 740 I=1, IDMX
  PRES(I) = PRES(I)
CONTINUE

--- WRITE RESTART FILE ---
IF (IFILE .GT. 0) THEN
  WRITE(IFILE,922) NSEED, ISEED0, XMAX, YMAX, PORO, SPAC, SPACM, TOL,
  WRITE(IFILE,901) NNOD, XMAX, YMAX, PORO, SPAC, SPACM, TOL,
  WRITE(IFILE,*) =(I),YS(T), HITE(I), DIAM(I), RADI(I)
ENDIF

--- TRACER PARTICLE MOTION ---
IF (KPAR .GT. 0) THEN
  XORIG=0.
  YORIG=0.
  RATX=XM/XMAX
  IF (RATX.GT.1.0) THEN
    XAXIS = 7.5
  ENDIF
ENDIF
PROCEDURE MAIN

02781 YAXIS=XAXIS *0.95775 /RATXY
02782 ELSE
02783 XAXIS = 7.5 *RATXY
02784 YAXIS=XAXIS *0.95775 /RATXY
02785 ENDIF
02786 XSTP= (XMAX-XORIG) /1.
02787 YSTP= (YMAX-YORIG) /1.
02788 CALL AREA2D(XAXIS, YAXIS)
02789 CALL GRAP(XORIG, XSTP, XMAX, YORIG, YSTP, YMAX)
02790 CALL FRAME
02791 C
02792 DCOF = ABS (UREF *DIAMF/1.000 /PECL)
02793 PECS = PECL *SDm /DTBAR

BIOREQ$MAIN

02794 DTIM1 = ABS (XMAX /VBAR *DTSTR)
02795 DTIM2 = (XMAX *DTSTR)**2 /DCOF /2.
02796 DTIM = 1. /((1./DTIM1 + 1./DTIM2)
02797 DTIM = DTIM
02798 C
02799 PRINT981, 'METH', IMETH
02800 PRINT980, 'DIAM', DIAMF
02801 PRINT980, 'DREF', DREF
02802 PRINT980, 'UBAR', UBAR
02803 PRINT980, 'UREF', UREF
02804 PRINT980, 'VBAR', VBAR
02805 PRINT980, 'VAVE', VAVE
02806 PRINT980, 'PVAX', PVAX
02807 PRINT980, 'SV00', SV000
02808 PRINT980, 'RKOZ', RKOZ
02809 PRINT980, 'EDSK', EDSK
02810 PRINT980, 'EPS0', EPS0
02811 PRINT980, 'FVORO', FVORO
02812 PRINT980, 'EFVS', EPSVOID
02813 PRINT980, 'EPAX', EPSM
02814 PRINT980, 'RHOS', RHOSVOID
02815 PRINT980, 'BTIM', DTIM
02816 PRINT980, 'DTIM', DTIM
02817 PRINT980, 'DCOF', DCOF
02818 PRINT980, 'PECL', PECL
02819 PRINT980, 'PECS', PECS
02820 PRINT980, 'TBAR', DTBAR
02821 PRINT980, 'BBAR', BBAR
02822 PRINT980, 'TSIG', TSIG
02823 TSTAR = DCOF *DTIM /DTBAR**2
02824 PRINT980, 'TSIG', TSTAR
02825 IF (TSTAR.LT.1.0) PRINT980, 'TSTR TOO SMALL'
02826 C
02827 CALL MARKER(16)
02828 CALL BLREC(-0.2*XAXIS, -0.2*YAXIS, 0.2*XAXIS, 1.4*YAXIS, 0)
02829 CALL BLREC(-0.2*XAXIS, -0.2*YAXIS, 1.4*XAXIS, 0.2*YAXIS, 0)
02830 CALL BLREC( 1.0*XAXIS, -0.2*YAXIS, 0.2*XAXIS, 1.4*YAXIS, 0)
02831 CALL BLREC(-0.2*XAXIS, 1.0*XAXIS, 1.4*XAXIS, 0.2*YAXIS, 0)
02832 DO 1381 ID = 1, NSEED
02833 XB(ID)=XS(ID)
02834 YB(ID)=YS(ID)
02835 FAC= 2.* RADI(ID)/XMAX *YAXIS/0.082
02836 CALL SCLPIC(FAC)
02837 CALL CURVE(XB, YB, 1, 1)
02838 1381 CONTINUE
02839  CALL SCLPIC(1.)
02840  CALL RESET('BLSYM')
02841  CALL SCLPIC(1.0)
02842  C
02843  DO 1369 IFR = 1, 100
02844  1369  FREQ(IFR) = 0.
02845  C
02846  DO 1707 IDT = 1, NTUB
02847  1707  JBLK(IDT) = 0
02848  C
02849  ASUM = 0.
02850  DO 1374 I = 1, NNBC(IDBIN)

BIOREM5MAIN

02851  IDN = NBID(IDBIN,I)
02852  IDT = TBID(IDBIN,I)
02853  ASUM = ASUM + KBLOC(IDT) * TUBAR(15, IDT)
02854  1374  CONTINUE
02855  C
02856  VTBAR = 0.
02857  XBAR = 0.
02858  KPSUM = 0
02859  TORSUM = 0.
02860  ARTBAR = 0.
02861  DO 1310 J = 1, KPAR
02862  C
02863  C--- GENERATE NEW PARTICLE AND PLACE AT ENTRANCE NODE; PARTICLES ARE
02864  C  RANDOMLY DISTRIBUTED BETWEEN ENTRANCE NODES IN PROPORTION TO
02865  C  INLET TUBE FLOW RATES.
02866  C
02867  1315  CONTINUE
02868  DPART = -100 * SPACM / 10.
02869  XXX = RAN(ISED)
02870  XTRM = (4.*XXX-2.)/3.
02871  ZZZ = XTRM
02872  DO 1382 I = 1, 100
02873     ZZZ = ZZZ**3/3. + XTRM
02874     RESID = ZZZ**3 - 3.*ZZZ + 3.*XTRM
02875     IF (ABS(RESID) .LT. 1.E-4) GO TO 1383
02876  1382  CONTINUE
02877  WRITE(17,*) 'ZZZ DID NOT CONVERGE',XXX,ZZZ,RESID
02878  1383  CONTINUE
02879  C
02880  C--- OPTION FOR UNIFORM INJECTION ---
02881  C
02882  IF (IUNI .EQ. 1) THEN
02883     ZZZ = (2.*XXX - 1.)
02884  ENDIF
02885  C
02886  GGG = (ZZZ + 1.) / 2.
02887  C
02888  C--- SUM ENTRANCE FLOWS TO PICK INLET TUBE ---
02889  C
02890  XXX = RAN(ISED)
02891  XXX = EDGE + (1.-2.*EDGE) * XXX
02892  IF (XXX .LT. 0.) THEN
02893     XXX = -XXX
02894  ENDIF
02895  FLSUM = 0.
02896  DO 1364 I = 1, NNBC(IDBIN)
IDN = NBID(IDBIN, I)

IDT = TBID(IDBIN, I)

DFLUX = ABS(TUBAR(11, IDT)) /DOTMIN

DFLUX = DFLUX + 0.00000 *TUBAR(15, IDT) /ASUM /PECL

FLSUM = FLSUM + DFLUX /(1. + 0.00000 /PECL)

IF (FLSUM .GT. XXX) THEN

INOD = IDN
GO TO 1365

ENDIF

CONTINUE

WRITE(17,*)

'CANT FIND ENTRANCE NODE'

C
1365 TREM = DTIM

ID0N = 0

TORT = 0.

XYNPAR(4, J) = YNOD(INOD)

CALL THKCRV(.020)

C--- STEP THROUGH TUBE SET ALONG PARTICLE PATH ---

C

DO 1320 ISTP = 1, 10000

IF (IDON .EQ. 1) GO TO 1321

C

C--- IDENTIFY TUBES HAVING (POSITIVE) FLOW OUT OF NODE IDNOD

C

AND SUM TUBE CROSS SECTION AREAS ---

C

IF (ISTP .EQ. 1) GO TO 1363

IDIN = IDT

DIFSIM = 0.

FLOSUM = 0.

NCH = NODAR(1, INOD)

NFIN = 0

NFIN = 0

DO 1330 JNOD = 1, NCH

IDT = NODAR(4+JNOD, INOD)

ADIF = 1.00000 *TUBAR(15, IDT)

QFLO = TUBAR(13, IDT)

DIFSUM = DIFSUM + ADIF *DCOF /TUBAR(1, IDT)

ISN = 1

IF (TUBNOD(1, IDT) .NE. INOD) ISN = -1

FLOW = ISN *QFLO

IF (FLOW .GT. 0.0) THEN

NFOUT = NFOUT + 1

FLOSUM = FLOSUM + FLOW

FLO(JNOD) = FLOW

ELSE

NFIN = NFIN + 1

FLO(JNOD) = 0.

ENDIF

CONTINUE

C

QFLOIN = MIN (ABS (TUBAR(13, IDTIN)), FLOSUM) /FLOSUM

DO 1398 JNOD = 1, NCH

IF (NODAR(4+JNOD, INOD) .EQ. IDTIN) THEN

A51
JNODIN = JNOD
GO TO 1393
ENDIF
1398 CONTINUE
1393 CONTINUE
C
C---- MIX JUNCTION STREAMLINES ----
C
IF (IMIX .EQ. 1)
THEN
zzz = 2. *RAN(ISED) - 1.
ENDIF

C
C--- MIX JUNCTION STREAMLINES ---
C
BIORE$MAIN

ENDIF

C
C--- SELECT TUBE BASED ON RELATIVE FLUXES ---
C
GGG = (ZZZ + 1.) /2.
FLSUM = 0.
DO 1335 JCNT = 1, NCH
JNOD = MOD (JNODIN+JCNT-1, NCH) + 1
IDRJ = NODAR(4+JNOD, NOD)
DFLOW = FLO(JNOD) /FLSUM
FLSUM = FLSUM + DFLOW
IF (FLSUM .GE. GGG)
THEN
IDT = IDRJ
JGOUT = JCNT
QFLOUT = MIN (ABS (TUBAR(13,IDT)), FLSUM) /FLSUM
FLSUM = FLSUM - DFLOW
GO TO 1336
ENDIF
1335 CONTINUE
W1UTE(17,*)
'CAN'T FIND NEW TUBE'
WRITE(17,*)
XXX, FLSUM
1m=1
GO TO 1321
1336 CONTINUE
C
C--- COMPLETE NEW GGG AND ZZZ POSITION FOR NO MIXING ---
C
IF (IMIX .NE. 1)
THEN
IF (NFIN.EQ.1 .AND. NFOUT.EQ.1)
THEN
GGG = GGG
ELSEIF (NFIN.EQ.1 .AND. NFOUT.EQ.2)
THEN
GGG = (GGG - FLSUM) /QFLOUT
ELSEIF (NFIN.EQ.2 .AND. NFOUT.EQ.1)
THEN
IF (JOUT .EQ. 1)
THEN
GGG = QFLOIN *GGG
ELSEIF (JOUT .EQ. 2)
THEN
QFLOIN2 = 1. - QFLOIN
GGG = QFLOIN *GGG + QFLOIN2
ELSE
PRINT*, 'JGOUT IS NOT 1 OR 2', JGOUT
WRITE(17,*) 'JGOUT IS NOT 1 OR 2', JGOUT
ENDIF
ELSE
PRINT*, 'NUMBER OF TUBES IS WRONG'
WRITE(17,*) 'NUMBER OF TUBES IS WRONG'
ENDIF
ENDIF
ENDIF
ENDIF
C
A52
IF (GGG .GT. 1.0 .OR. GGG .LT. 0.) THEN
  PRINT*, 'GGG OUT OF RANGE', GGG
  WRITE(17,*) 'GGG OUT OF RANGE', GGG, NFIN, NFOUT, QFLOIN,
  QFLOOUT, FLSUM, JCOUNT
ENDIF

GGG = MIN (0.999, MAX (0.001, GGG))
ZZZ = 2. * GGG -1.

CONTINUE

JBLOK(IDT) = 1
JNOD = TUBNOD(1, IDT)
IF (JNOD .EQ. INOD) JNOD = TUBNOD(2, IDT)
ISN = 1
IF (TUBNOD(1, IDT) .NE. INOD) ISN = -1
VEL = ISN * TUBAR(10, IDT)
VEL = SIGN (MAX (ABS (VEL), 1.E-30), VEL)

IF (DPART .GT. TUBAR(2, IDT)) THEN
  ISTIK = 1
  IDON = 1
  XB(1) = XNOD(INOD) + 1.000 * TUBAR(7, IDT)
  YB(1) = YNOD(INOD) + 1.000 * TUBAR(8, IDT)
ELSE
  IARIV = 0
  DELTIMF = TUBAR(1, IDT) / ABS (VEL) / 100.
  DIFFL = TUBAR(14, IDT)
  DELTIMD = (DIFFL / (20.*PI/3.))**2 / DCOF / 2.
  DELTIM = MIN (DELTIMF, DELTIMD)
  IF (PECL .GT. 9.E9) THEN
    DELTIM = TUBAR(1, IDT) / ABS (VEL) / 0.9
  ENDIF
  SSS = 0.
  ZZZIN = ZZZ
  TAKE INTERMEDIATE STEPS ALONG TUBE LENGTH
  TSUM = 0.
  BSUM = 0.
  DO 1340 KSTP = 1, 1000000
    IF (IARIV .NE. 0) GO TO 1341
    IF (IDON .EQ. 1) GO TO 1341
    IF (DELTIM .GT. TRED) THEN
      DELTIM = MAX (TRED, 1.E-30)
IDON = 1
ENDIF
CONTINUE
C
C--- COMPUTE LOCAL APERTURE ---
C
HHH = 0.
IF (RMETH .EQ. 2) THEN
BIOREMSMAIN
ID = TURNOD(3, IDT)
HHH = 0.
IF (ID .GE. 1) THEN
TRM = RADI(ID)**2 - (SSS - TUBAR(1, IDT)/2.)**2
TRM = MAX (TRM, 1.E-30)
HH1 = RADI(ID) + TUBAR(2, IDT)/2. - SQRT(TRM)
ENDIF
ID = TURNOD(4, IDT)
HH2 = 0.
IF (ID .GE. 1) THEN
TRM = RADI(ID)**2 - (SSS - TUBAR(1, IDT)/2.)**2
TRM = MAX (TRM, 1.E-30)
HH2 = RADI(ID) + TUBAR(2, IDT)/2. - SQRT(TRM)
ENDIF
HHH = MAX (HH1+HH2, 1.E-30)
ENDIF
C
C--- TAKE ADVECTIVE STEP ---
C
IF (IPARA .EQ. 1) THEN
VLOC = 1.5 * VEL * (1. - ZZZ**2)
ELSE
VLOC = VEL
ENDIF
IF (RAN(1SED) .LT. 0.5) IDIR = -1
DELSDIF = IDIR * SQRT (2. * DCOF * DELTIM)
DEL2 = DELSDIF / (TUBAR(14, IDI') / 2.)
IF (KMETHOD .EQ. 2) DELZ = DEL2 * TUBAR(14, IDT) / HHH
IF (ZZZ + DELZ .GT. 1.0)
DEL2 = 2.*(1.-ZZZ) - DEL2
ELSEIF (ZZZ + DELZ .LT. -1.0)
A54
DELZ = -2.* (1.+ZZZ) - DELZ
ENDIF
IF (PECL .GT. 9.E9) THEN
DE LZ = 0.
ENDIF
C--- UPDATE POSITION AND TIME ---

BIOREM$MAIN

C
SSS = SSS + DELS
ZZZ = ZZZ + DELZ
IF (IRANS .EQ. 1) ZZZ = (2. *RAN(ISED) - 1.)
TREM = TREM - DELTIM
C--- PWTT FINAL PARTICLE POSITION ---

XB(2) = XB(1) + DXDS * SSS
YB(2) = YB(1) + DYDS * SSS
IF (IRANS .EQ. 1) ZZZ = (2. * RAN(ISED) - 1.)
TREM = TREM - DELTIM
CONTINUE
C
CONTINUE
ENDIF
C
XB(2) = XB(1) + DXDS * SSS
YB(2) = YB(1) + DYDS * SSS
ELSEIF (IARIV .EQ. -1) THEN
INOD = INOD
ENDIF
INOD = JNOD
IF (IZZZ .EQ. 1) THEN
ENDIF
ELSEIF (IARIV .EQ. -1) THEN
INOD = INOD
ENDIF
INOD = JNOD
IF (IPLTITI .EQ. 1) THEN
CALL cuRvE(XB,YB,2,0)
ENDIF
C
BSUM = BSUM / TUBAR(1,IDT)
WRITE(17,900) TUBAR(2,IDT), TUBAR(14,IDT), BSUM, TSUM,
1 TUBAR(1,IDT)/VEL
C
C--- RESET STREAMLINE TO ENTRANCE VALUE ---
C
C
C
C
C
C--- PLOT TRACER PARTICLE TRAJECTORY ---
C
C
C
C--- PLOT FINAL PARTICLE POSITION ---
C
C
C
C
C
A55
ARTBAR = ARTBAR + ARRTIM(J)
CALL MARKER(13)
XB(2) = 0.99 *XMAX
CALL CURVE(XB(2), YB(2), 1, -1)
ELSEIF (EXIT .EQ. -1) THEN
CALL MARKER(13)

STORE PATICLE POSITION IN FREQUENCY DISTRIBUTION
AND SUM AVERAGE POSITION

XB(2) = 0.01 *XMAX
CALL CURVE(XB(2), YB(2), 1, -1)
ENDIF

C
TORT = (TORT /MAX (1.E-10, XB(2)))**2
TORSUM = TORSUM + TORT

C
C--- STORE PATICLE POSITION IN FREQUENCY DISTRIBUTION ---
C
C AND SUM AVERAGE POSITION
C
KPSUM = KPSUM + 1
IFR = NINT (XB(2) /XMAX *99) + 1
FREQ(IFR) = FREQ(IFR) + 1.
XBAR = XBAR + XB(2)
VTRACE(KPSUM) = XB(2) /DTIM
VBAR = VBAR + VTRACE(KPSUM)

C
1310 CONTINUE
CALL RESET('DASH')
CALL RESET('THKCRV')
ARTBAR = MAX (1.E-10, ARTBAR /KPAR)
XBAR = XBAR /KPAR
VBAR = VBAR /KPAR
TORT = TORSUM /KPAR

C
C--- COMPUTE STANDARD DEVIATIONS ---
C
SDEV = 0.
STIM = 0.
SIGV = 0.
SDTR = 0.

DO 1348 IPAR = 1, KPAR
SDEV = SDEV + (XBAR-XYNPAR(1, IPAR))**2
SDTR = SDTR + (XYNPAR(2, IPAR)-XYNPAR(4, IPAR))**2
STIM = STIM + (ARTBAR - ARRTIM(IPAR))**2
SIGV = SIGV + (VTRACE(IPAR) - VBAR)**2

1348 CONTINUE
SDEV = SQRT (SDEV /KPAR)
STIM = SQRT (STIM /KPAR)
SIGV = SQRT (SIGV /KPAR)

C
C--- COMPUTE CUMULATIVE DISTRIBUTION ---
C
DO 1391 IFR = 1, 100
XB(IFR) = (IFR-0.5) /100 *XMAX
YB(IFR) = 1.
DO 1392 JFR = 1, IFR
YB(IFR) = YB(IFR) - FREQ(JFR) /KPAR

1392 CONTINUE
IF (YB(IFR) .GE. 0.5) X50 = XB(IFR)
YB(IFR) = YB(IFR) *XMAX

A56
C

03245  1391  CONTINUE
03246           X50 = MAX (1.E-10, X50)
03247  C
03248           IF (IMICF .NE. 1) THEN
03249           CALL THKCRV(.050)

BIOREMS$MAIN

03250           CALL CURVE(XB,YB,100,0)
03251      ENDIF
03252  C
03253  C--- COMPUTE APPARENT DIFFUSIVITY AND PLOTT CORRESPONDING ERROR FUNC ---
03254  C
03265  C
03255           VPAR = XBAR /DTIM
03256           PEVP = PECL *VPAR /VBAR
03257           PEBB = PEVP *BBAR /DIAMP
03258           DIFF = SDEV**2 /2. /XBAR
03259           DIF2 = SDEV**2 /2. /DTIM
03260           DIFT = SDTR**2 /2. /DTIM
03261           DIP3 = (UREF /FORO *STIM)**2 /2. /ARTBAR
03262           DIF4 = SIGV**2 *DTIM /2.
03263           DIFF0 = MAX (1.E-20, SDEVT/XBAR*DIAMF)
03264           DCON = DIFF /DIFF0
03265           CN1 = FORO *DIF2 /DOCF /2. *PECL
03266           CONS = FORO *DIF2 /DOCF /2. *PECS
03267           DO 1367 IFR = 1, 101
03268               XB(IFR) = (IFR-1.) /100 *MAX
03269               ETA = (XBAR-XB(IFR)) / 2. /SQRT(XBAR*DIFF)
03270               YB(IFR) = 0.5 *(1. + ERF(ETA)) *XMAX

03271  1367  CONTINUE
03272  C
03273           IF (IMICF .NE. 1) THEN
03274           CALL DASH
03275           CALL CURVE(XB,YB,100,0)
03276           CALL RESET ('DASH')
03277      ENDIF
03278  C
03279  C--- PLOT USED TUBES FOR MICRO-FINGERS ---
03280  C
03281           IF (IMICF .EQ. 1) THEN
03282             DO 1703 IDT = 1, NTUB
03283               XB(1) = TUBAR(3,IDT)
03284               YB(1) = TUBAR(4,IDT)
03285               XB(2) = TUBAR(5,IDT)
03286               YB(2) = TUBAR(6,IDT)
03287               WID = MIN (TUBAR(2,IDT), 3.*TUBAR(1,IDT))
03288               WID = 0.98 *MAX (WID, 0.)
03289               WID = WID /XMAX *XAXIS
03290               WID = MAX (WID, .001)
03291               CALL THKCRV(WID)
03292          IF (JBLOK(IDT) .EQ. 1) CALL CURVE(XB,YB,2,0)
03293          1703  CONTINUE
03294  C
03295      ENDIF
03296  C
03297      CALL RESET ('THKCRV')
03298  C
03299      CALL SCLPIC(1.0)
03300  C
03301      CALL CURVE(XB,YB,1,1)
03302  C
03303      PRINT980, 'PECL', PECL

A57
03303 PRINT980, 'PEVP', PEVP
03304 PRINT980, 'PEBB', PEBB
03305 PRINT980, 'DIFL', DIF2/DCOF
03306 PRINT980, 'DIFT', DIFT/DCOF

BIOREMAN

03307 PRINT980, 'TORT', TORT
03308 PRINT980, '
03309 PRINT980, 'X50 ', X50
03310 PRINT980, 'VPAR', VPAR
03311 PRINT980, 'XBAR', XBAR
03312 PRINT980, 'ICON', ICON
03313 PRINT980, 'CONS', CONS
03314 PRINT980, 'CON1', CON1
03315 PRINT980, 'SDEV', SDEV
03316 PRINT980, 'DIFF', DIFF
03317 PRINT980, 'DIF2', DIF2
03318 PRINT980, 'DIF3', DIF3
03319 PRINT980, 'DRAT', DIF/DCOF
03320 PRINT980, 'DRA4', DIF4/DCOF
03321 PRINT980, 'SIGV', SIGV
03322 PRINT980, 'SVRA', SIGV/VTBAR
03323 PRINT980, '
03324 CALL ENDFPL(0)
03325 ENDF
03326 710 CONTINUE
03327 830 CONTINUE
03329 C
03330 PRINT*, 'INPUT ICONT  1 = NEW PRESS
03331   1 2 = NEW TUBES 3 = RESTART'
03332 READ*, ICONT
03333 IF (ICONT.EQ.1) THEN
03334   GO TO 1620
03335 ELSEIF (ICONT.EQ.2) THEN
03336   GO TO 1610
03337 ELSEIF (ICONT.EQ.3) THEN
03338   GO TO 1600
03339 ENDF
03340 C
03341 840 CALL DONEPL
03342 C
03343 988 FORMAT('1',A,2X,1PE11.3)
03344 980 FORMAT(1X,A,2X,1PE11.3)
03345 981 FORMAT(1X,A,2X,11,1PE11.3)
03346 900 FORMAT(11(1PE12.4))
03347 990 FORMAT(12(1PE11.3))
03348 901 FORMAT(110,10(1PE12.4))
03349 902 FORMAT(2(I10),8(1PE12.4))
03350 922 FORMAT(2(I10),10(1PE11.3))
03351 912 FORMAT(2(I5),8(1PE12.4))
03352 903 FORMAT(3(I10),8(1PE12.4))
03353 913 FORMAT(3(I6),8(1PE12.4))
03354 904 FORMAT(4(I10),8(1PE12.4))
03355 905 FORMAT(5(I10),8(1PE12.4))
03356 1900 FORMAT(10(1PE12.4))
03357 1901 FORMAT(I4,10(1PE12.4))
03358 1902 FORMAT(2(I4),8(1PE12.4))
03359 1912 FORMAT(2(I5),8(1PE12.4))
03360 1903 FORMAT(3(I4),8(1PE12.4))
SUBROUTINE SPACEF(NBIN, NSEED, XS, YS, XMAX, YMAX, SDEVX, SDEVY)

DO 761 J = 1, NBIN

FREQX(J) = 0.

761 FREQY(J) = 0.

DO 762 ID = 1, NSEED

IX = NINT(XS(ID)/XMAX *(NBIN-1)) + 1

IY = NINT(YS(ID)/YMAX *(NBIN-1)) + 1

FREQX(IX) = FREQX(IX) + 1.

762 FREQY(IY) = FREQY(IY) + 1.

SDEVX = 0.

SDEVY = 0.

BARN = (1.*NSEED) /NBIN

DO 763 J = 1, NBIN

SDEVX = SDEVX + (FREQX(J)-BARN)**2

763 SDEVY = SDEVY + (FREQY(J)-BARN)**2

SDEVX = SQRT(SDEVX / (NBIN-1.))

SDEVY = SQRT(SDEVY / (NBIN-1.))

RETURN

END

SUBROUTINE NERNEB(NDARR, NBORS, NSEED, XS, YS, XO, YO, DNEIGH, INEIGH)

INTEGER INEIGH(NDARR, 1)

C--- NERNEB LOCATES NBORS NEAREST NEIGHBORS. XS(I) AND YS(I) ARE SEED

C LOCATIONS, DNEIGH(I,J) IS THE DISTANCE TO NEIGHBOR I OF SEED J,

C AND INEIGH(I,J) IS THE SORTED SEED NUMBER OF THE ITH NEIGHBOR OF

C SEED J. NERBJZR USES A BUBBLE SORT TO RANK NEIGHBORS IN INCREASING

C DISTANCE.

DDMX = 1.E20

DO 1107 J = 1, NBORS

1107 DNEIGH(J,ID) = DDMX

DO 1108 ID = 1, NSEED

1108 DNEIGH(ID,J) = DDMX

DO 1109 IDP = 1, NSEED

1109 IF (IDP .EQ. ID) GO TO 1102

IF (DIS .LT. DNEIGH(NBORS,ID)) THEN

DIS = SQRT((XS(ID)-XS(IDP))**2 + (YS(ID)-YS(IDP))**2)

IF (DIS .LT. DNEIGH(NBORS,ID)) THEN

DNEIGH(NBORS,ID) = DIS

INEIGH(NBORS,ID) = IDP

STOP
DO 1103 J = 1, NBORS-1

K = NBORS - J

IF (DNEIGH(K+1, ID) .LT. DNEIGH(K, ID)) THEN

DTMP = DNEIGH(K, ID)
ITMP = INEIGH(K, ID)

DNEIGH(K, ID) = DNEIGH(K+1, ID)
INEIGH(K, ID) = INEIGH(K+1, ID)

ELSE
GO TO 1102
ENDIF

ENDIF

CONTINUE

1101 CONTINUE

RFMTRN
END

FUNCTION FUNC(X)
COMMON /BLKQ/ RMU,SIG,Z

ETA = (X-RMU) /SIG/SQRT (2.)

FUNC = (1. +ERF(ETA)) /2. -Z

RETURN
END

FUNCTION ERRETA (Y)
EXTERNAL GUNC

DATA ERR, ERA /1.E-4,1.E-4/
COMMON /BLKZ/ VALU

VALU = Y
XL = -20
XR = 20
X = 0.

CALL PZERO (GUNC, XL, XR, X, ERR, ERA, IFLAG)

ERRETA = XL

RETURN
END
FUNCTION GUNC (X)
COMMON /BLKZ/ VALU
GUNC = ERF(X) - VALU
RETURN
END

SUBROUTINE PLTBOX(XMAX, YM0X, NBOX)
DIMENSION XX(20), YY(20)
PI=4. *ATAN(1.)
DDD=0.5*SQRT(1.+TAN(2.*PI/NBOX)**2)
DO 814 J=1,NBOX+1
THET=2.*PI*(J-1)/NBOX+2.*PI/NBOX/2.
XX(J)=(0.5+DDD*COS(THET))*XMAX
YY(J)=(0.5+DDD*SIN(THET))*YM0X
814 CONTINUE
CALL THKCV(.030)
CALL CURVE(XX, YY, NBOX+1, 0)
RETURN
END

SUBROUTINE PLTBNDS(ICMX, NSID, IPRM, XSID, YSID, IPOP)
DIMENSION NSID (1), IPRM(12,1), XSID (12,1), YSID(12,1), IPOP (12,1)
DIMENSION XB(2), YB(2)
CALL THKCV(.010)
DO 815 ID=1,ICMX
   DO 815 L=1,NSID(ID)
      IDP=IPRM(L, ID)
      IF (IDP.GT.ID) GO TO 815
      IF (IPOP(L, ID) .EQ.1) THEN
         XB(1)=XSID(L, ID)
         YB(1)=YSID(L, ID)
         XB(2)=XSID(L+1, ID)
         YB(2)=YSID(L+1, ID)
         CALL CURVE(XB, YB, 2, 0)
      ENDIF
   815 CONTINUE
RETURN
END

SUBROUTINE DELTAS (IFLAG, R1, R2, DELTA0, L, DELTAD, DELTAV)
C
C SOLU0ON)
IFLAG = 1 => ANALYTICAL SOLUTION FOR R2 = R1 (ZERO0-ORDER PERTURBATION
SOLUTION)
IFLAG = 2 => FIRST-ORDER PERTURBATION SOLUTION
IFLAG = 3 => EXACT SOLUTION (VIA NUMERICAL INTEGRATION IF NECESSARY)
IMPLICIT DOUBLE PRECISION (A - H, O - Z)
DIMENSION SUM(4), ZUM(4)
DSTAR(TERM1, R3S, T, COST, SINT) = TERM1 - COST -
               SQRT(R3S - SINT*SINT)

ARG = 0.5*L/R1
RATIO2 = DELTA0/R1
RATIO3 = R2/R1
R1S = R1*R1
ROOT1 = SQRT(R1S - LS4)
THETA = ASIN(ARG)
IF (IFLAG .NE. 3) THEN
   K = RATIO2/2. + 1.
   KS = K*K
   V1 = KS - 1.
   EPS = RATIO3 - 1.
   BASIC = ATAN(SQRT(K1/(K - 1.))*TAN(THETA/2.))/SQRT(V1)
   BASIC = BASIC*K
   G = -THETA + 2.*BASICK
   DELTAD = L/G
   DENOM = K - COS(THETA)
   V2 = SIN(THETA)/DENOM
   V3 = V2/DENOM
   F = (3.*BASICK + (1. + 0.5*KS)*V2)/V1 + 0.5*K*V3
   V1F = V1/F
   DELTAV = DELTA0 + R1*(2. - THETA/ARG) - ROOT1
   DELTAP = SQRT(4.*R1S*L*V1F/DELTAV) + 0.5*K*V3
   H = (2.*(2.*KS + 1.)*BASIC + 3.*K*V2)/V1 + V3
   CE = (V3*V1F/DENOM + 2. - 1.5*H/F)/(6.*K1)
   DV1 = R1*(1. - THETA/ARG)
   CP = 0.5*(3.*CE - DV1/DELTAV)
   DELTAV = DELTAV + EPS*Dv1
   DELTAP = DELTAP*(1. + EPS*CP)
   RETURN
ELSE
   TERM1 = RATIO2 + 1. + RATIO3
   R3S = RATIO3*RATIO3
   R2S = R2*R2
   DT = THETA/20.
   T = 0.
   DELTA = DSTAR(TERM1, R3S, T, 1.D0, 0.D0)
   ZUM(2) = 0.5/DELTA

DELTAS

ZUM(4) = 0.
SUM(2) = ZUM(2)/(DELTA*DELTA)
SUM(4) = 0.
IFLIP = 2
DO 11 J = 1, 20
   T = T + DT
   COST = COS(T)
   SINT = SIN(T)
   DELTA = DSTAR(TERM1, R3S, T, COST, SINT)
   GRAND2 = COST/DELTA
A62
GRAND1 = GRAND2/(DELTA*DELTA)

IFLIP = 6 - IFLIP

SUM(IFLIP) = SUM(IFLIP) + GRAND1

ZUM(IFLIP) = ZUM(IFLIP) + GRAND2

CONTINUE

SUM(2) = SUM(2) - 0.5*GRAND1

ZUM(2) = ZUM(2) - 0.5*GRAND2

FACTOR = 1.5*L/DT

DECUBE = FACTOR*R1S/(2.*SUM(2) + 4.*SUM(4))

DELTAD = FACTOR/(2.*ZUM(2) + 4.*ZUM(4))

ROOT2 = SQRT(R2S - L4)

DELTAV = DELTAV0 + R1 + R2 - 0.5*(ROOT1 + ROOT2) -

1

(R2S/L)*ASIN(0.5*L/R2) - (R1S/L)*THETA

DELTAP = SQRT(DECUBE/DELTAV)

RETURN

FUNCTION BOXMUL (ISED, RMU, SDEV, R2)

DATA TWOPI /6.2831853072/

XX1 = RAN (ISED)

XX2 = RAN (ISED)

BOXMUL = SQRT(-2. *ALOG(XX1)) *COS (TWOPI *XX2)

BOXMUL = SDEV *BOXMUL + RMU

R2 = SQRT(-2. *ALOG(XX1)) *SIN (TWOPI *XX2)

R2 = SDEV *R2 + RMU

RETURN

END