10/5-1-97 JS ①

SANDIA REPORT

SAND97-8250 • UC-406
Unlimited Release
Printed April 1997

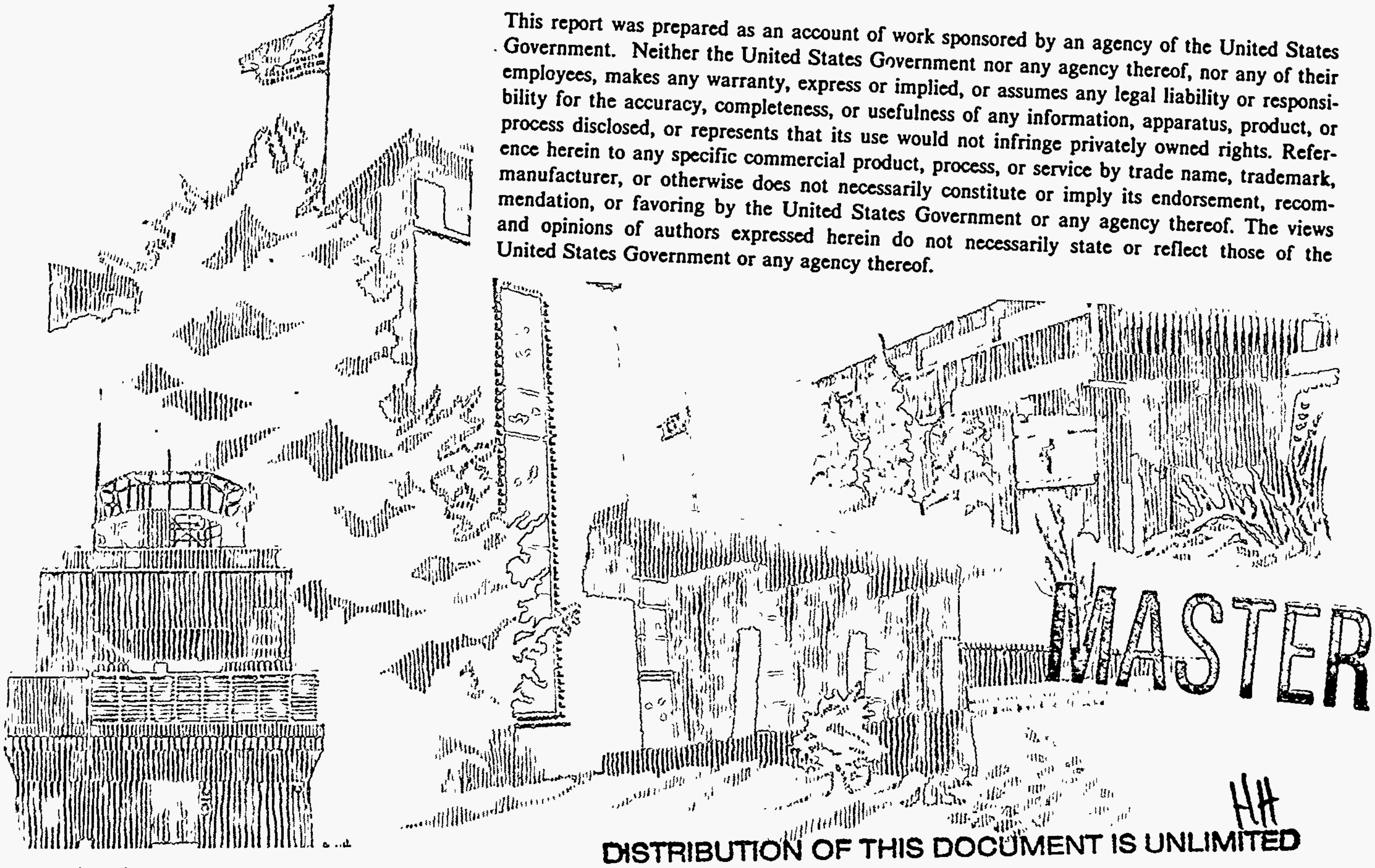
M97052749

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

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

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-94AL85000

Approved for public release, distribution is unlimited.



MASTER

HH

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

SF2900Q(8-81)

UC-406

SAND97-8250
Unlimited Release
Printed April 1997

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 metabolisis 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.

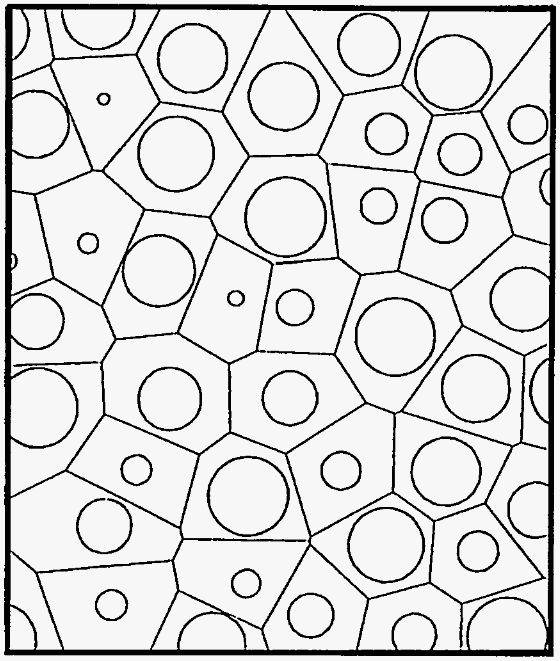
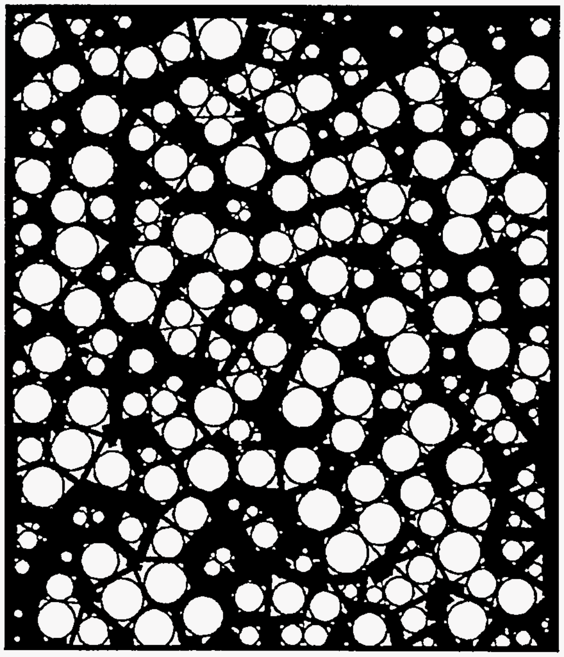


Figure 2. Schematic of particle array and channel network. Channel intersections define the nodes of the computational domain. Although these channels are displayed as having parallel walls, the governing transport equations take into account that channels conform to particle surfaces.



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

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

where V_k is the portion of tube volume associated with node k, ρ_k is the fluid density at node k, and $f_{i,k}$ 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 do 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.

$$q_{j,k} = -\rho \frac{\bar{\delta}^3_{j,k}}{24} \frac{\Delta p_{j,k}}{\ell_{j,k}} \quad \text{or} \quad q_{j,k} = -\rho\pi \frac{\bar{\delta}^4_{j,k}}{64} \frac{\Delta p_{j,k}}{\ell_{j,k}} \qquad \text{(2a,b)}$$

$\Delta p_{j,k}$ is the pressure drop across the tube joining the j and k nodes, ℓ is the tube length, and $\bar{\delta}_{j,k}$ 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, δ_V, defined by

$$V = \ell\,\delta_V \tag{3}$$

where ℓ is the tube length, and V is its volume. It is staightforward to show that in this case the equivalent aperture is given by

$$\delta_V = \frac{2}{\ell}\int_0^{\ell/2} \delta dz \tag{4}$$

where δ is the local tube aperture.

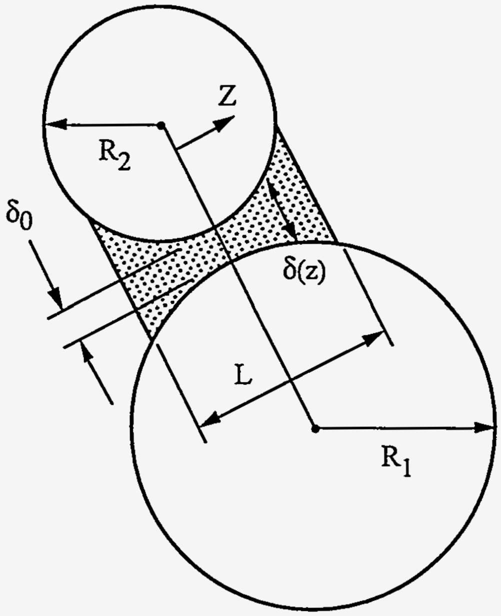
The second equivalent aperture is that of the permeability. In this case, the equivalent aperture for two-dimensional flow is defined by

$$q = -\frac{\delta^3}{12}\frac{dp}{dz} = -\frac{\delta_k}{12}\frac{\Delta p}{\ell} \tag{5}$$

where q is the constant volumetric flow rate through the tube, p is the local pressure and Δp 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 δ_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

$$\frac{1}{\delta_k^3} = \frac{2}{\ell}\int_0^{\ell/2} \frac{dz}{\delta^3} \tag{6}$$

Figure 3. Diagram of a particle pair and flow channel. Equivalent apertures for pore volume, flow, diffusion, and through-channel transit time are computed using the channel geometry.



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

$$f = \delta D \frac{dc}{dz} = \delta_D D \frac{\Delta c}{\ell} \tag{7}$$

where D is the coefficient of binary diffusion, c is the local concentration of the diffusing species and Δc 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

$$\frac{1}{\delta_D} = \frac{2}{\ell} \int_0^{\ell/2} \frac{dz}{\delta} \tag{8}$$

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

$$t = 2 \int_0^{\ell/2} \frac{dz}{u} = \frac{\ell}{\bar{u}} \tag{9}$$

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

$$u = -\frac{\delta^2}{12}\frac{dp}{dz} \qquad \text{and} \qquad \bar{u} = -\frac{\delta_u^2}{12}\frac{\Delta p}{\ell} \tag{10a,b}$$

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

$$\frac{1}{\delta_u^2} = \frac{2}{\ell}\int_0^{\ell/2} \frac{\delta}{\delta_k^3}\, dz = \frac{\delta_V}{\delta_k^3} \tag{11}$$

Thus the equivalent aperture for mean fluid speed and transit time can be written simply in terms of those for the permeability, δ_k, and the tube volume, δ_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

$$\delta = \delta_0 + R_1 + R_2 - \sqrt{R_1^2 - z^2} - \sqrt{R_2^2 - z^2} \tag{12}$$

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

$$\delta_0 = \sqrt{\left(x_1 - x_2\right)^2 + (y_1 - y_2)^2} - R_1 - R_2 \tag{13}$$

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

$$R_2 = R_1\,(1 + \epsilon) \qquad \text{and} \qquad z = R_1 \sin\theta \tag{14a,b}$$

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

$$\begin{aligned} \delta = \delta_0 + R_1 & \left[\!\!\left[\, 2 + \epsilon - \cos\theta - (1+\epsilon)\left[1 - \frac{\sin^2\theta}{(1+\epsilon)^2}\right]^{1/2} \right]\!\!\right] \\ & \approx \delta_0 + R_1\left[1 - 2\cos\theta + \epsilon(1 - \sec\theta)\right] \end{aligned} \tag{15a,b}$$

Note that the latter of these relations is based on a one-term expansion for small values of ϵ.

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

$$\begin{aligned} \frac{\delta_V}{R_1} = {} & 2\kappa + \epsilon - \frac{1}{2}\left[(1+\epsilon)^2 - \sin^2\theta\right]^{1/2} \\ & -\frac{R_1}{\ell}(1+\epsilon)^2 \sin^{-1}\left(\frac{\sin\theta}{1+\epsilon}\right) - \frac{1}{2}\left(1 - \sin^2\theta\right)^{1/2} - \frac{R_1}{\ell}\theta \end{aligned} \tag{16a,b}$$

for the equivalent volume aperture. The new parameter, κ, appearing here is given by

$$\kappa = 1 + \frac{\delta_0}{2R_1} \tag{17}$$

and in this case θ is evaluated at the angular limit of integration

$$\theta = \sin^{-1}\left(\frac{\ell}{2R_1}\right) \tag{18}$$

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

$$\frac{\delta_k^3}{\ell R_1^2} = 4\frac{\kappa^2 - 1}{f(\kappa,\theta)}\left[1 + \frac{3\epsilon I(\kappa,\theta)(\kappa^2-1)}{2f(\kappa,\theta)}\right] \tag{19}$$

where

$$I(\kappa,\theta) = \frac{1}{3(\kappa+1)}\left[\frac{\sin\theta}{(\kappa-\cos\theta)^3} - \frac{3h(\kappa,\theta)}{2(\kappa^2-1)} + \frac{2f(\kappa,\theta)}{2(\kappa^2-1)}\right] \tag{20}$$

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

$$\begin{aligned} h(\kappa,\theta) = {} & \frac{\sin\theta}{(\kappa-\cos\theta)^2} + \frac{3\kappa\sin\theta}{(\kappa^2-1)(\kappa-cos\theta)} \\ & + 2\frac{2\kappa^2+1}{(\kappa^2-1)^{3/2}}\tan^{-1}\left[\left(\frac{\kappa+1}{\kappa-1}\right)\tan\left(\frac{\theta}{2}\right)\right] \end{aligned} \tag{21}$$

and

$$\begin{aligned} f(\kappa,\theta) = {} & \frac{(2+\kappa^2)\sin\theta}{2(\kappa^2-1)(\kappa-\cos\theta)} + \frac{\kappa\sin\theta}{2(\kappa-\cos\theta)^2} \\ & + \frac{3\kappa}{(\kappa^2-1)^{3/2}}\tan^{-1}\left[\left(\frac{\kappa+1}{\kappa-1}\right)^{1/2}\tan\left(\frac{\theta}{2}\right)\right] \end{aligned} \tag{22}$$

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

$$\frac{\delta_D}{\ell} = \frac{1}{g(\kappa,\theta)}\left[1 + \frac{\epsilon\psi(\kappa,\theta)}{2(\kappa+1)g(\kappa,\theta)}\right] \tag{23}$$

where

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

and

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

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

$$\frac{\delta_V}{R} = 2\kappa - \left(1-\sin^2\theta\right)^{1/2} - 2\frac{R}{\ell}\theta \tag{26}$$

for the equivalent volume aperture. Similarly, Eq. (19) in this limit reduces to

$$\frac{\delta_k^3}{\ell R^2} = \frac{4\left(\kappa^2-1\right)}{f(\kappa,\theta)} \tag{27}$$

for the equivalent aperture for viscous fluid flow. Finally, the equivalent aperture for diffusion, given by Eq. (23) above, becomes

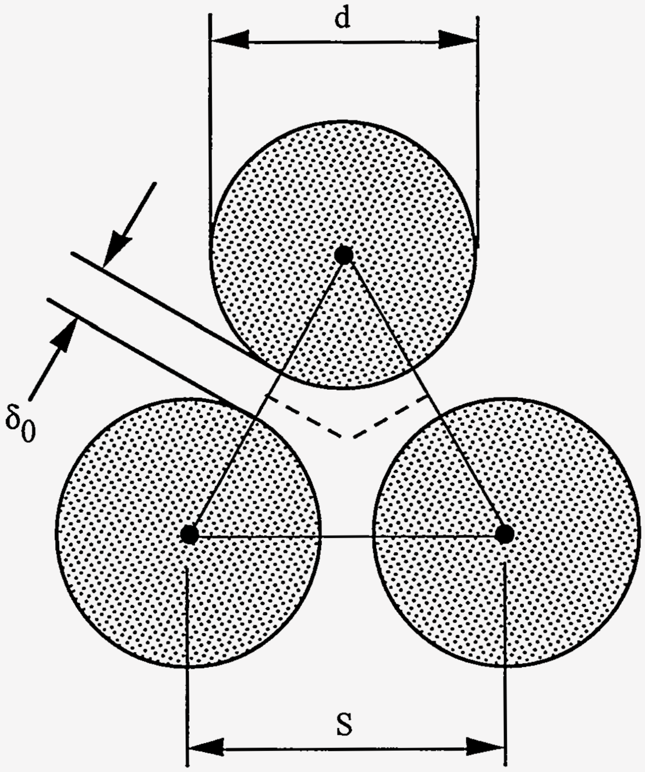
$$\frac{\delta_D}{\ell} = \frac{1}{g(\kappa,\theta)} \tag{28}$$

for this case of uniform particle size. As before, the angular limit of integration, θ, is taken as that corresponding to the tube length.

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

$$k = \frac{1}{12}\frac{\delta_k^3}{h\sqrt{\tau}} \tag{29}$$

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, δ_k is the equivalent aperture for viscous flow. Using Eq. (27), this result may be expressed as

$$k = \frac{1}{12}\frac{\left(\kappa^2 - 1\right)}{f(\kappa, \theta)}\frac{\ell d^2}{h\sqrt{\tau}} \tag{30}$$

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

$$\frac{\ell}{h} = \frac{2}{3} \quad \text{and} \quad \kappa = 1 + \frac{\delta_0}{d} = \sqrt{\frac{1-\phi_0}{1-\phi}} \tag{31a,b}$$

where ϕ 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

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

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

$$k_r = \frac{\pi d^2}{4(1-\phi)} \tag{33}$$

the normalized permeability may be expressed as

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

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.

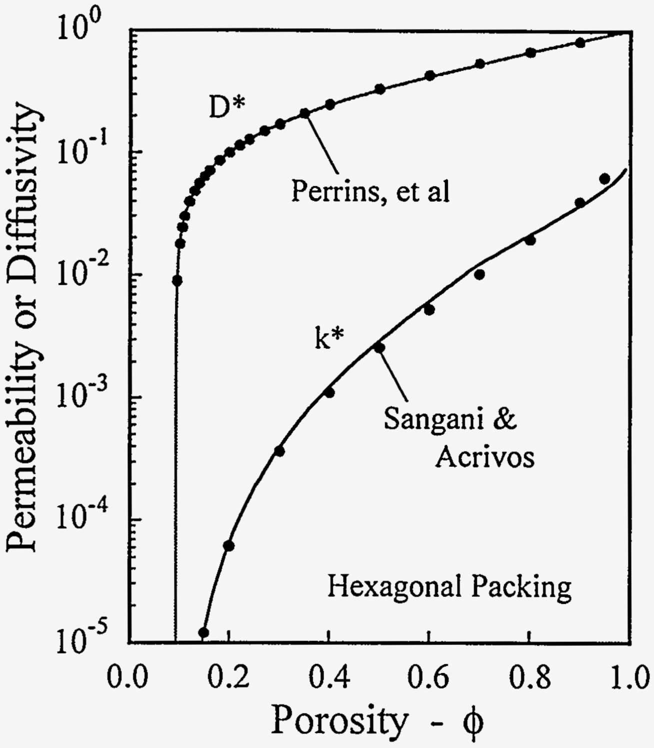
A comparison between these results and an exact analytical solution to the problem [32] is shown in Fig. 5. For this simplified geometry of a regular hexagonal array, Eq. (34) agrees to within 15% for all porosities above the percolation threshold and below $\phi = 0.9$. Based on this agreement, we conclude 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

$$D^* = \frac{D_e}{D} = \frac{\delta_D}{h\sqrt{\tau}} = \frac{\ell}{hg(\kappa, \theta)\sqrt{\tau}} \tag{35}$$

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 ℓ/h, κ, θ and τ 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

Figure 5. Comparison between network results and exact solutions from previous analyses of the normalized permeability and effective diffusivity for a regular hexagonal array of cylindrical particles.



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 posses 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, δt. 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

$$\delta \ell_D = \xi \sqrt{2D\delta t} \tag{36}$$

where ξ 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, σ^2 of the particle positions by

$$D^* = \frac{\phi}{D}\frac{\sigma^2}{2t} \tag{37}$$

where ϕ 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

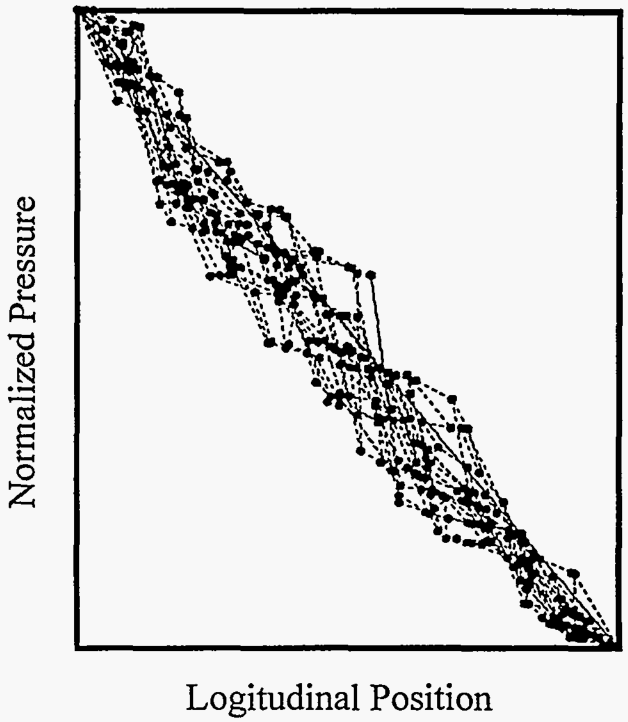
To demonstrate the unique capabilities of a network model, we now present the results of calculations for several sample problems. These sample problems include the pressure field for an incompressible flow, permeability reductions due to uniform film growth on particles of a granular medium, particle-scale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

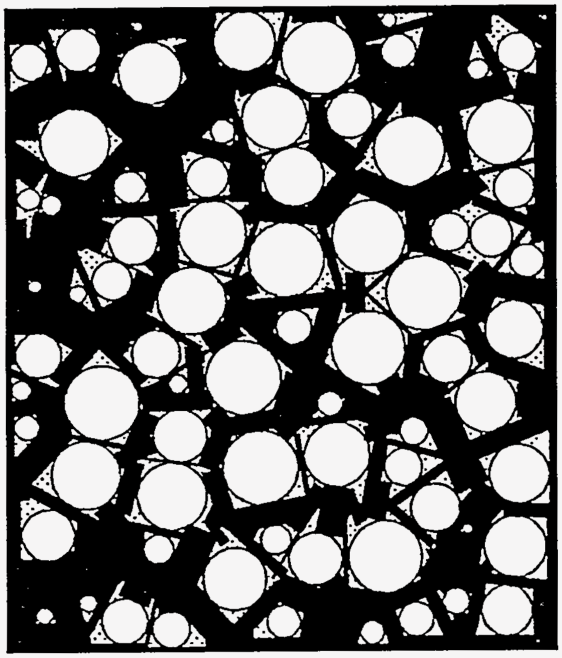
Three sample networks constructed in this way are shown if Fig. 7. Figure 7A represents the initial formation, while Figs. 7B and 7C represent the same collection of particles with 33% and 66% of the initial void filled. The initial network (7A) at a porosity of 0.6 consists of 223 tubes and has a normalized 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.

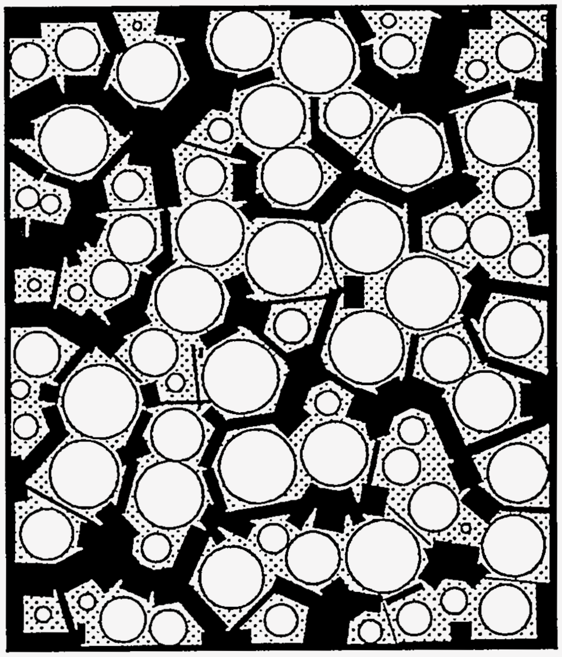
To illustrate the use of tracer particles in computing effective transport properties, we now consider the problems of longitudinal and lateral dispersion. Dispersion in a porous material is a complex process involving both flow and diffusion. Variations in local fluid velocities yield varying particle 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

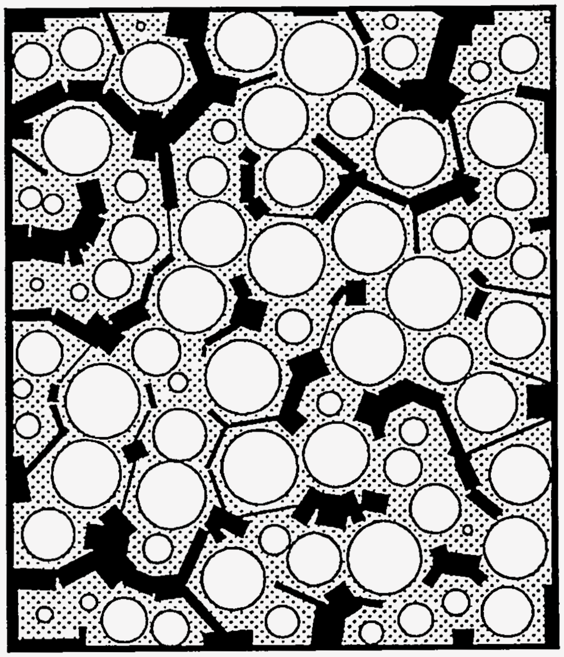


7A



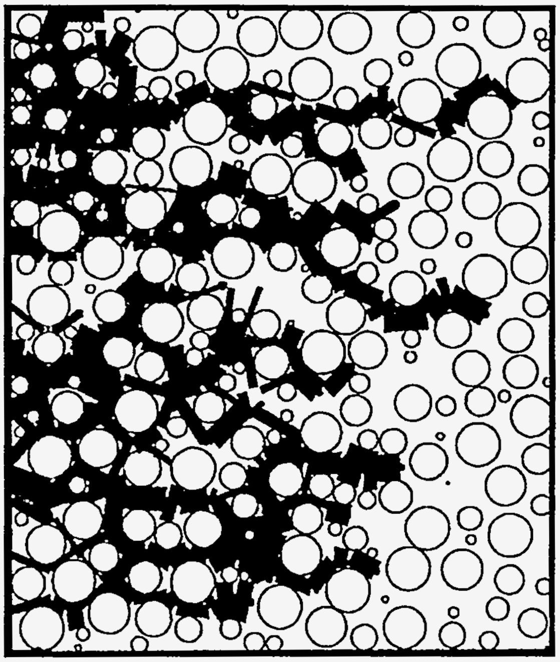
7B

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.



7C

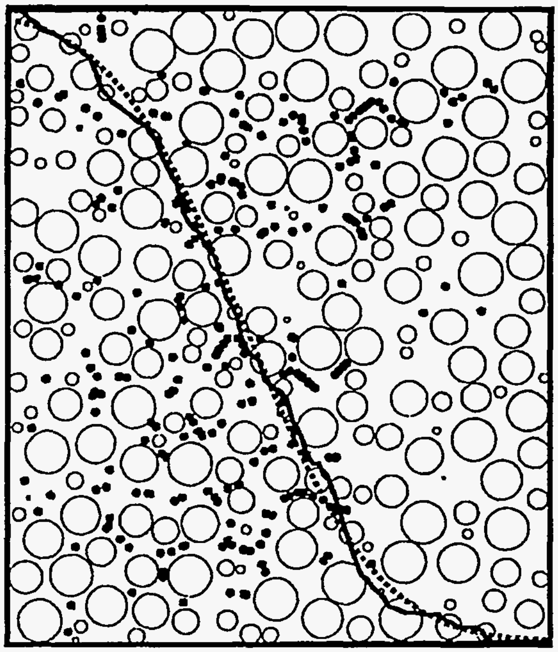
Figure 8. Particle-scale fingers in a pore network. Finger pattern results from local variations in permeability due to local variations in the size of interconnected pores. Unlike traditional viscous fingers, the pattern produced in a given pore network is largely deterministic in nature.



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 $D^* = \sigma^2/2t = \bar{u}\sigma^2/2\bar{x}$, where $\bar{u}$ is the mean fluid speed, σ^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.

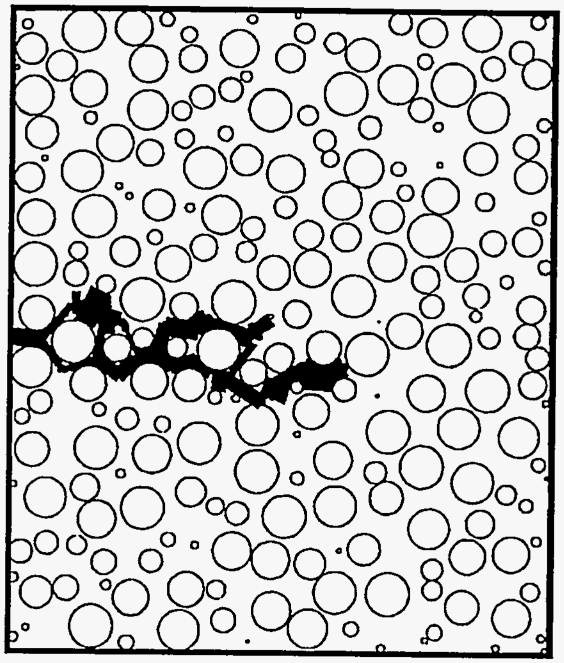
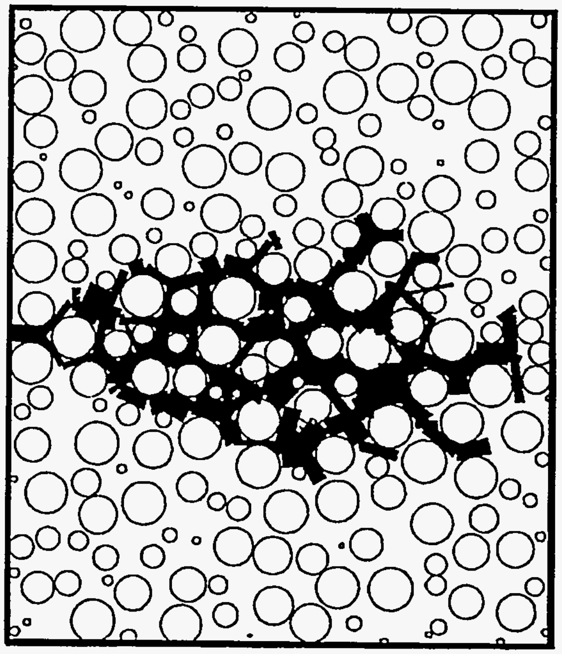


Figure 11. Lateral dispersion at low Peclet number. Large coefficient of diffusion allows tracer particles to cross streamlines and follow mean flow into an increasing lateral extent of the network. Results shown are for Pe = 1.



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 $\mathrm{Pe} = \rho\bar{u}d/D = 1$. 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 $\mathrm{Pe} = 1$, 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

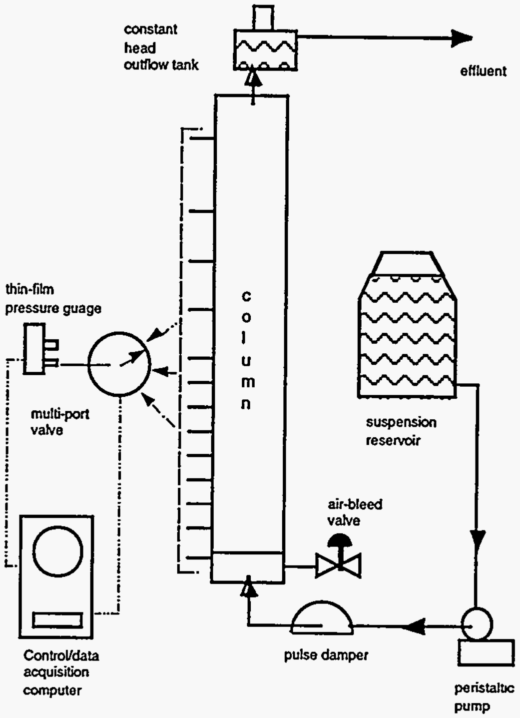


Figure 12. Diagram of experimental apparatus. Suspension is pumped through bottom of column against constant pressure of the reservoir at the top. Pressures are measured along the length of the column.

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.

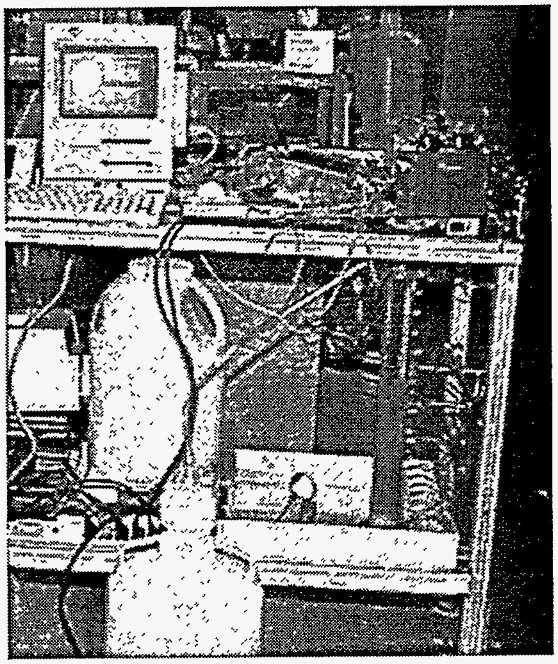
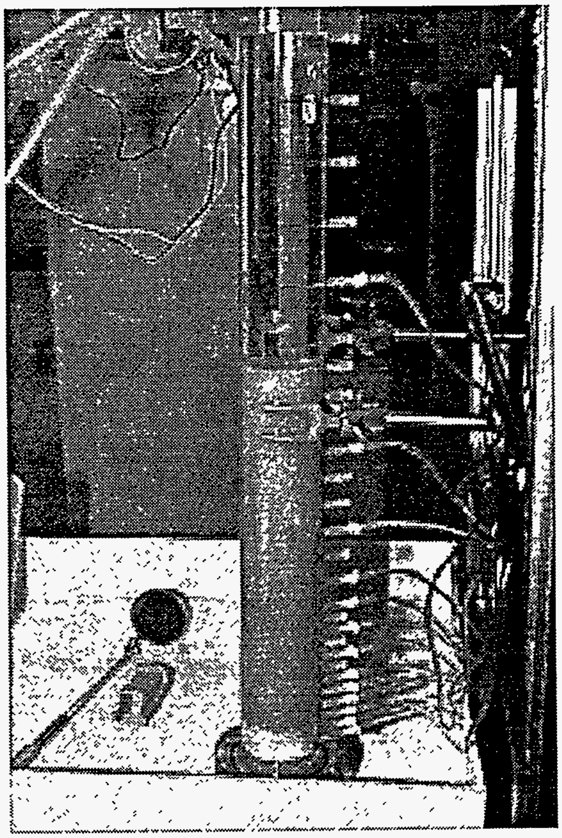


Figure 14. Photograph of a packed bed column as used for filtration experiments showing arrangement of the pressure monitoring ports. The white material at the bottom of the column is a deposit resulting from deposition of fines from a suspension.



Packing Material	Diameter (μm)	Mesh Size
P-170 Spheres*	300-450	40-50
A-055 Spheres*	500-600	30-35
Ottawa Sand**	600-850	20-30
A-150 Spheres*	1400-1700	12-14

* Potter's Industries, Inc., $\rho = 2.45 - 2.50$ gm/cc
** Fisher Scientific Co., $\rho = 2.62$ 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.

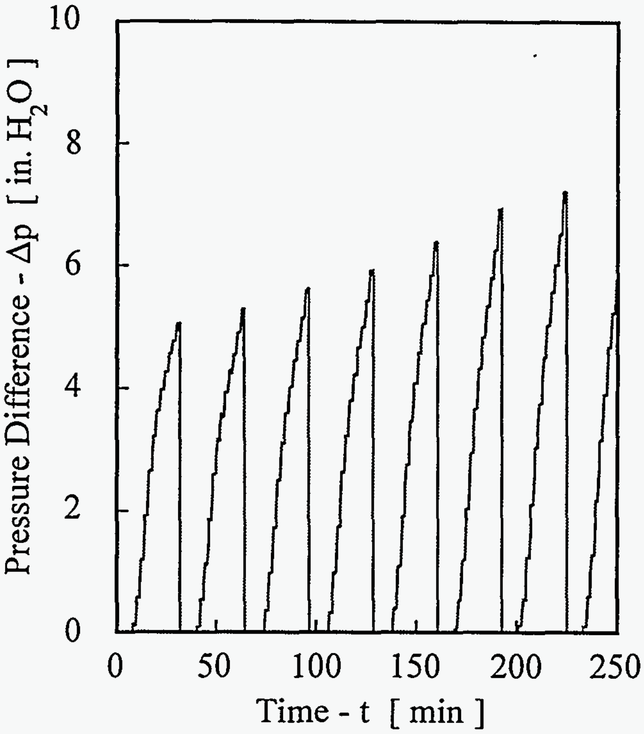
A peristaltic pump was used to supply a constant flow rate of suspension during these experiments. The suspension was pumped using a Masterflex Model 7523 peristaltic pump (Cole-Parmer Instrument Co., Vernon Hills, IL). Flow rates up to 270 milliliters per minute (ml/min) could be achieved. The flow 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

Figure 15. Representative data output showing the pressure distribution in a packed column during deposition from a suspension of fine clay particles.



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 multiposition 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 $H_2Al_2Si_2O_8$–X H_2O. Aluminum sulfate, $Al_2(SO4)_3$–X H_2O, 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 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 μm) 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 μm) 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 μm) and the data from this work for Ottawa sand (600-850 μm) also yields good agreement. Fig. 16 illustrates

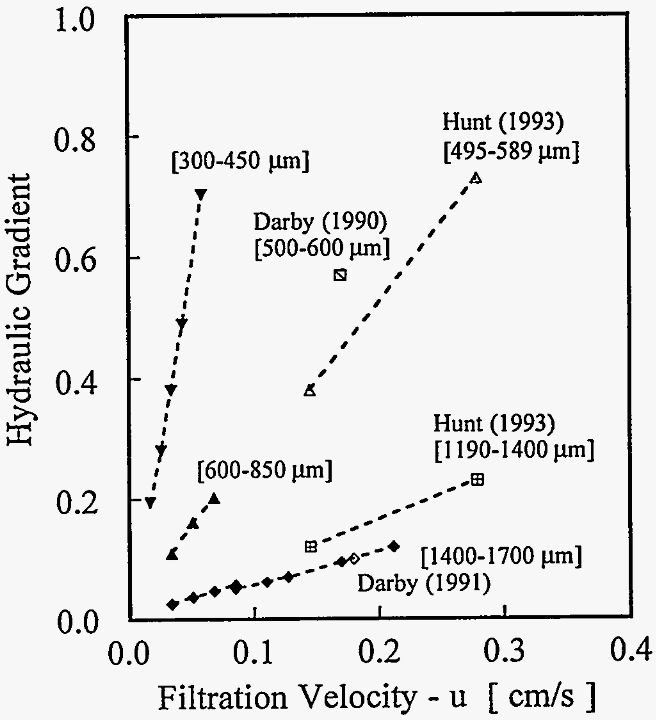
Data Source	Diameter (μm)	Velocity (cm/s)	Gradient
P-0170 Beads	300-450	0.017	0.195
"	300-450	0.026	0.281
"	300-450	0.034	0.381
"	300-450	0.043	0.490
Ottawa Sand	600-850	0.034	0.110
"	600-850	0.051	0.161
"	600-850	0.068	0.202
A-150 Beads	1400-1700	0.034	0.026
"	1400-1700	0.051	0.037
"	1400-1700	0.068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

Table 2. Hydraulic gradients measured for flow of water in clean packed columns.

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

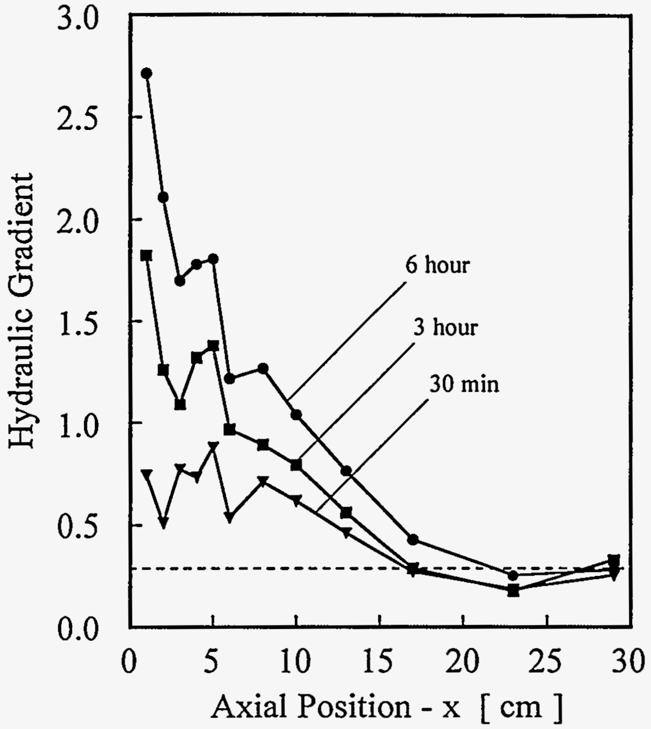
Figure 16. Hydraulic gradients measured in packed columns during flow of clean water.



fines 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

Figure 17. Observed increases in hydraulic gradient resulting from deposition of fines in a packed bed.



bed, under somewhat similar experimental conditions, rather than being distributed throughout the bed [45,46,47].

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

1. E. L. Madsen, "Determining In Situ Biodegradation: ES&T Critical Review," *Environ. Sci. Technol.*, **25** (10), 1663-1673-931, 1991.

2. S. Fiorenza, K. L. Duston, and C. H. Ward, "Decision Making: Is Bioremediation a Viable Option," *J. Haz. Mat.*, **28**, 171-183, 1991.

3. W. T. Frankenberger, K. D. Emerson, and D. W. Turner, "In Situ Bioremediation of an Underground Diesel Fuel Spill: A Case Study," *Environ. Manag.*, **13** (3), 325-332, 1989.

4. A. M. Thayer, "Bioremediation: Innovative Technology for Cleaning Up Hazardous Waste," *C&EN*, 23-37, August 26, 1991.

5. D. R. Cullimore, *Practical Manual of Groundwater Microbiology*, Lewis Publishers, Boca Raton, 1993.

6. M. E. Miller and M. Alexander, "Kinetics of Bacterial Degradation of Benzylamine in Montmorillonite Suspension," *Environ. Sci. Technol.*, **25**, 240-245, 1991.

7. J. M. Thomas and C. H. Ward, "In Situ Biorestoration of Organic Contaminants in the Subsurface," *Environ. Sci. Technol.*, **23** (7), 760-766, 1989.

8. H. H. M. Rijnaarts, A. Bachmann, J. C. Jumelet and A. J. B. Zehnder, "Effect of Desorption and Intraparticle Mass Transfer on the Aerobic Biomineralization of α-Hexachlorocyclohexane in a Contaminated Calcareous Soil," *Environ. Sci. Technol.*, **24** (9), 1349-1354, 1990.

9. P. N. Cheremisinoff ed., *Encyclopedia of Environmental Control Technology: Hazardous Waste Containment and Treatment*, Gulf Publishing Co., Houston, 1990.

10. J. R. Mihelcic and R. G. Luthy, "Sorption and Microbial Degradation of Naphthalene in Soil-Water Suspensions under Denitrification Conditions," *Environ. Sci. Technol.*, **25**, 169-177, 1991.

11. S. R. Hutchins, G. W. Sewell, D. A. Kovacs and G. A. Smith, "Biodegradation of Aromatic Hydrocarbons by Aquifer Microorganisms Under Denitrifying Conditions," *Environ. Sci. Technol.*, **25**, 68-76, 1991.

12. S. W. Taylor and P. R. Jaffe, "Biofilm Growth and the Related Changes in Physical Properties of a Porous Medium: 1. Experimental Investigation," *Water Resour. Res.,* **26** (9), 2153-2159, 1990.

13. S. W. Taylor and P. R. Jaffe, "Biofilm Growth and the Related Changes in Physical Properties of a Porous Medium: 2. Permeability," *Water Resour. Res.*, **26** (9), 2161-2169, 1990.

14. Y. M. Chen, L. M. Abriola, P. J. J. Alvarez, P. J. Anid and T. M. Vogel, "Modeling Transport and Biodegradation of Benzene and Toluene in Sandy Aquifer Material: Comparisons with Experimental Measurements," *Water Resour. Res.*, **28** (7), 1833-1847, 1992.

15. C. Y. Chiang, C. N. Dawson and M. F. Wheeler, "Modeling of In-Situ Biorestoration of Organic Compounds in Groundwater," *Transport in Porous Media*, **6**, 667-702, 1991.

16. D. C. Mangold and C. F. Tsang, "A Summary of Subsurface Hydrological and Hydrochemical Models," *Rev. Geophys.*, **29** (1), 51-79, 1991.

17. L. Semprini and P. L. McCarthy, "Comparison Between Model Simulations and Field Results for In Situ Biorestoration of Chlorinated Aliphatics," *Ground Water*, **29** (3), 365-373, 1991.

18. E. O. Frind, "Modeling of Multicomponent Transport with Microbial Transformation in Groundwater: The Fuhrberg Case," *Water Resour. Res.*, **26** (8), 1707-1719, 1990.

19. G. M. Hornberger, A. L. Mills and J. S. Herman, "Bacterial Transport in Porous Media: Evaluation of a Model Using Laboratory Observations," *Water Resour. Res.*, **28** (3), 915-938, 1992.

20. J. T. Angley, M. L. Brusseau, W. L. Miller, J. J. Delfino, "Nonequilibrium Sorption and Aerobic Biodegradation of Disolved Alkylbenzenes During Transport in Aquifer Material: Column Experiments," *Environ. Sci. Technol.*, **26** (7), 1404-1410, 1992.

21. M. Sahimi snd A. O. Imdakm, "Hydrodynamics of Particle Motion in Porous Media," *Phys. Rev. Lett.*, **66** (9), 1169-1172, 1991.

22. S. V. Sotirchos and V. N. Burganos, "Analysis of Multicomponent Diffusion on Pore Networks," *AIChE J.*, **34** (7), 1106-1118, 1988.

23. M. M. Tomadakis and S. V. Sotirchos, "Ordinary and Transition Regime Diffusion in Random Fiber Structures," *AIChE J.*, **39** (3), 397-412, 1993.

24. S. L. Bryant, D. W. Mellor and C. A. Cade, "Physically Representative Network Models of Transport in Porous Media," *AIChE J.*, **39**, 387-396, 1993.

25. S. L. Bryant, P. B. King and D. W. Mellor, "Network Model Evalutaion of Permeability and Spatial Correlation in a Real Random Sphere Packing," *Transport in Porous Media*, **11**, 53-70, 1993.

26. K. A. Akanni and J. W. Evans, "Effective Transport Coefficients in Heterogeneous Media," *Chem. Eng. Sci.*, **42** (8), 1945-1954, 1987.

27. L. M. Schwartz and J. R. Banavar, "Transport Properties of Disordered Continuum Systems," *Phys. Rev.* B, **39** (16), 11965-11970, 1989.

28. A. Okabe, B, Boots and K. Sugihara, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, John Wiley & Sons, New York, 1992.

29. S. Bryant and M. Blunt, "Prediction of Relative Permeability in Simple Porous Media," *Phys. Rev.* A, **46** (4), 2004-2011, 1992.

30. R. E. Larson and J. J. L. Higdon, "A Periodic Grain Consolidation Model of Porous Media," *Phys. Fluids* A, **1** (1), 38-46, 1988.

31. J. J. L. Higdon and G. D. Ford, "Permeability of Three Dimensional Models of Fibrous Porous Media," *J. Fluid Mech.*, **308**, 341-361, 1996.

32. A. S. Sangani and A. Acrivos, "Slow Flow Past Periodic Arrays of Cylinders With Application to Heat Transfer," *Int. J. Multiphase Flow*, **8** (3), 193-206, 1982.

33. W. T. Perrins, D. R. McKenzie and R. C. McPhedran, "Transport Properties of Regular Arrays of Cylinders," *Proc. Roy. Soc. Lond.* A, **369**, 207-225, 1979.

34. D. C. Hong, H. E. Stanley, and A. Coniglio,"Randon-Walk Approach to the Two-Component Random-Conductor Mixture," *Phys. Rev.* B, **33** (7), 4564-4573, 1986.

35. S. Reyes and K. F. Jensen,"Percolation Concepts in Modelling of Gas-Solid Reactions: 1. Application to Char Gasification in the Kinetic Regime," *Chem. Eng. Sci.*, **41** (2), 333-343, 1986.

36. S. Reyes, E. Iglesia, and K. F. Jensen,"Application of Percolation Theory Concepts to the Analysis of Gas-Solid Reactors," *Solid State Ionics*, **32**, 833-842, 1989.

37. L. M. McDowell-Boyer, "Chemical Mobilization of Micron-Sized Particles in Saturated Porous Media," *Environ. Sci. Technol.*, **26** (3), 586-593, 1992.

38. S. Torquato, "Trapping of Finite-Sized Brownian Particle in Porous Media," *J. Chem. Phys.*, **95** (4), 2838-2841, 1991.

39. R. E. Martin, E. J. Bouwer, and L. M. Hanna, "Application of Clean-Bed Filtration Theory to Bacterial Deposition in Porous Media," *Environ. Sci. Technol.*, **26** (5), 1053-1056, 1992.

40. P. G. Saffman, "A Theory of Dispersion in a Porous Medium," *J. Fluid Mech.*, **6**, 321-349, 1959.

41. D. L. Koch and J. F. Brady, "Dispersion in Fixed Beds" *J. Fluid Mech.* **154**, 399-427, 1985.

42. S. Hill, "Channeling in Packed Columns," *Chem. Eng. Sci.*, **1** (6), 247-253, 1952.

43. R. L. Chuoke and P. van Meurs, "The Instability of Slow Immiscible Viscous Liquid-Liquid Displacements in Permeable Media," Pet. Trans., **216**, 188-194, 1958.

44. P. G. Saffman and G. I. Taylor, "The Penetration of a Fluid into a Porous Medium or Hele-Shaw Cell Containing a More Viscous Liquid," *Proc. Roy. Soc. Lond.* A, **245**, 312-329, 1958.

45. J. B. Hunt, B-C Hwang and L. M. McDowell-Boyer, "Solids Accumulation during Deep Bed Filtration," *Environ. Sci. Technol.*, **27**, 1099-1107, 1993.

46. J. L. Darby and D. F. Lawler, "Ripening in Depth Filtration: Effect of Particle Size on Removal and Head Loss," *Environ. Sci. Technol.*, **24** (7), 1069-1079, 1990.

47. J. L. Darby, D. F. Lawler and T. P. Wilshusen, "Depth Filtration of Particle Size and Ripening," *J. Water Pollut. Control Fed.*, **63**, 228-238, 1991.

48. A. B. Cunningham, W. G. Characklis, F. Abedeen and D. Crawford, "Influence of Biofilm Accumulation on Porous Media Hydrodynamics," *Environ. Sci. Technol.*, **25**, 1305-1311, 1991.

APPENDIX A - SOURCE LISTING OF PROGRAM BIOREM

BIOREM$MAIN

```
C
      DIMENSION X0(500),Y0(500),SL(500)
      DIMENSION XX(131),YY(131),DDX(500),DDY(500)
      DIMENSION XS(500),YS(500),IPRM(12,500),LPRM(12,500)
      DIMENSION RADI(500),DIAM(500),HITE(500),DIAM0(500),HITE0(500)
      REAL LBON(12,500),LSID(12,500),AREA(500)
      DIMENSION XSID(12,500),YSID(12,500),DONE(500)
      DIMENSION PERI(500),APRX(500),APRY(500)
      REAL LOADR,LOADL,LOADT,LOADB,LBOUN(200),LOADX,LOADY
      INTEGER FSID(20),JB(200),NSID(500),NORM(4)
      DIMENSION WORK(4,500),IDBC(500),LBC(500)
C
      INTEGER IDNOD(12,500),NODAR(11,1500),IDTUB(12,500),TUBNOD(4,1500)
      INTEGER KBLOC(1500), NNBC(4), NBID(4,200), IPIKT(3), TBID(4,200)
      REAL XNOD(1500),YNOD(1500),TUBAR(15,1500), FLO(3), APORE(1500)
      REAL XBA1(2),YBA1(2),XBA2(2),YBA2(2),XB(500),YB(500)
      REAL FREQ(500), XYNPAR(4,1500)
      REAL DNEIGH(18,500), DBOX(500)
      REAL DPER(1500), DDIF(1500), DVOL(1500), ARRTIM(1500)
      REAL XAA(3,3), RHS(3), XTU(1500), YTU(1500), RTU(1500), QHS(3)
      REAL XHS(3)
      REAL DETA, DETX, DETY, DETR
      INTEGER INEIGH(18,500), JBLOK(1500)
C
      REAL*8 PRES(1500),PRESO(1500),DPRES(1500),DOTM,PSUM
      DIMENSION DX(1),DY(1)
      DIMENSION FX(1),FY(1),SMAX(1,1)
      DIMENSION TEM(1),DTEM(1),SSMX(1),FIDL(1),SIDL(1)
      DIMENSION XSOLD(1),YSOLD(1),DXOLD(1),DYOLD(1)
      DIMENSION UX(1),UY(1),UXOLD(1),UYOLD(1)
      DIMENSION AA(1,1),BB(1),RES(3),WX(1),WY(1)
      INTEGER IDS(1),IPOP(1,1),SED,SED0,JPOP(1,1),IWORK(1)
      DIMENSION VTRACE(1500)
C
      REAL MAS,HGT,KMOD,KBLK
      COMMON XXX
      COMMON /BLKQ/ ALOGMU,ALOGSIG,ZZZ
C
      DATA (NORM(I),I=1,4) /-1, 1, 1, -1/
C
C  NODAR (K,IDN)
C    1: NUMBER OF ADJACENT NODES AND/OR CONNECTED TUBES
C    2: NODE NUMBER OF ADJ NODE 1
C    3: NODE NUMBER OF ADJ NODE 2
C    4: NODE NUMBER OF ADJ NODE 3
C    5: TUBE NUMBER JOINING NODE 1
C    6: TUBE NUMBER JOINING NODE 2
C    7: TUBE NUMBER JOINING NODE 3
C    8: BOUNDARY NUMBER OF NOD IDN
C    9: PARTICLE NUMBER 1
C   10: PARTICLE NUMBER 2
C   11: PARTICLE NUMBER 3
C
C  TUBAR (K,IDT)
C    1: TUBE LENGTH
C    2: TUBE DIAMETER
C    3: X LOC OF TUBE END 1
```

```
C      4: Y LOC OF TUBE END 1
C      5: X LOC OF TUBE END 2
C      6: Y LOC OF TUBE END 2
C      7: X SHIFT IN TUBE POSITION
C      8: Y SHIFT IN TUBE POSITION
C      9: BOUNDARY NUMBER OF TUBE IDT
C     10: FLUID VELOCITY
C     11: MASS FLOW RATE
C     12: TUBE VOLUME
C
C    XNOD (IDN) AND YNOD (IDN)
C      1: X AND Y COORDINATES OF NODE IDN
C
C    IDNOD (L,ID)
C      1: NODE NUMBER ASST WITH SIDE L OF SEED ID
C
C    IDTUB (L,ID)
C      1: TUBE NUMBER OF SIDE L OF SEED ID
C
C    TUBNOD (K,IDT)
C      1: NODE NUMBER AT TUBE END 1
C      2: NODE NUMBER AT TUBE END 2
C      3: SEED NUMBER OF 1ST PARTICLE
C      4: SEED NUMBER OF 2ND PARTICLE
C
      PI=4.*ATAN(1.)
      TWOPI = 2. *PI
      RT3 = SQRT(3.)
      RT2 = SQRT(2.)
      ISED = 111311
      PRINT*,4.*ATAN(1.),4.*ATAN(-1.)
C
      CALL TK4014(960,1)
      CALL PAGE(11.,8.5)
      CALL BLOWUP(5./4.)
      CALL NOBRDR
      CALL YAXANG(0.)
      CALL HEIGHT(0.10)
      CALL DUPLX
      CALL XREVTK
      CALL YREVTK
      CALL GAPWID(.002)
      CALL THKFRM(.030)
      CALL GRACE(2.0)
      CALL NOCHEK
C
 1600 CONTINUE
      ICONT = 0
C
      DO 1480 ID=1,500
        DO 1480 L=1,12
 1480     IDNOD(L,ID)=0
C
      DO 1404 IDN=1,1500
 1404   NODAR(1,IDN)=0
C
      GMIN=0.
```

```
      GMAX=0.
      GAVE=0.
      CMIN=1.
      CMAX=0.
C
 1519 CONTINUE
      IPROB=1
      IF (IPROB.EQ.4) THEN
        RBORE=0.1
        PRINT*,'INPUT BORE RADIUS',RBORE
        READ*,RBORE
      ENDIF
      IMETH = 4
      IFILL = 0
      PRINT*,'INPUT IMETH,IFILL',IMETH,IFILL
      READ*,IMETH,IFILL
      XMAX=1.
      YMAX=1.
      PRINT*,'INPUT XMAX,YMAX',XMAX,YMAX
      READ*,XMAX,YMAX
      IDBIN = 4
      PRINT*,'INPUT INFLOW BOUNDARY ID',IDBIN
      READ*,IDBIN
      IDBEV = (IDBIN /2) *2
      IDBOU = MOD (IDBIN+1, 4) + 1
      IPLTS = 0
      IPLTB = 0
      IPLTC = 0
      PRINT*,'INPUT IPLTC,IPLTS,IPLTB',IPLTC,IPLTS,IPLTB
      READ*,IPLTC,IPLTS,IPLTB
      IPLTP = 0
      IPLTT = 0
      IPLTN = 0
      ISHAD = 0
      PRINT*,'INPUT IPLTP,IPLTT,IPLTN,ISHAD',IPLTP,IPLTT,IPLTN,ISHAD
      READ*,IPLTP,IPLTT,IPLTN,ISHAD
      IROTA = 0
      PRINT*,'ROTATE SEEDS PI/2 ?   0 = NO  1 = YES',IROTA
      READ*,IROTA
      DEMAX0=3.
      PRINT*,'INPUT DEMAX0',DEMAX0
      READ*,DEMAX0
      NBOX=4
C
C---  INITIALIZE SEED  ---
C
      IF (IMETH .LT. 0) THEN
C
        REWIND(-IMETH)
        READ(-IMETH,*) NSEED,ISEED0,XMAX,YMAX,PORO,SPAC,SPACM,TOL
        DO 413 I = 1, NSEED
  413     READ(-IMETH,*) XS(I),YS(I),HITE(I),DIAM(I),RADI(I)
        READ(-IMETH,*) JNOD
        DO 441 IDN = 1, JNOD
          READ(-IMETH,*) PRES(IDN)
  441     PRESO(IDN) = PRES(IDN)
        DIAMF = DIAM(1)
```

```
        PRINT*,'INPUT EPS0,DIAMF',EPS0,DIAMF
        READ*,EPS0,DIAMF
C
 417    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
      ELSEIF (IMETH .EQ. 99) THEN
        READ(99,*) NPAR,XMAX,YMAX,EPS0,DIAMF
        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
        ISEED0=ISED
        DSAT=1.
        PRINT*,'INPUT DSAT',DSAT
        READ*,DSAT
        DSMIN=ABS(DSAT)*0.80/SQRT(1.*NSEED)
        DREF = DSMIN
        POROIN=.4
        ALOGSIG = 1.E-8
        PRINT*,'INPUT PORO,SIG',POROIN,ALOGSIG
        READ*,POROIN,ALOGSIG
        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
        IF (DSAT.EQ.1) NSEED=50*NSEED
        DO 1010 I=1,NSEED
          IDBC(I)=0
          ITRY=0
          DONE(I)=0
          ALOGMU=ALOG(DSMIN/2.)
```

```
          ZZZ=RAN(ISED)
          CALL ERRINV(ALOGR)
          DIAM(I)=2.*EXP(ALOGR)
C
 1013     XS(I)=XMAX*RAN(ISED)
          YS(I)=YMAX*RAN(ISED)
          HITE(I)=DIAM(I)/2. *(1.-2.*RAN(ISED))
          HMINN = MAX (HITE(I), DSMIN/50.)
          IF (HITE(I) .LT. 0.) THEN
            HITE(I) = MIN (HITE(I), -HMINN)
          ELSE
            HITE(I) = MAX (HITE(I), HMINN)
          ENDIF
          RADI(I)=SQRT( (DIAM(I)/2.)**2 - HITE(I)**2 )
C
          IF (DSAT.GT.1.E-6) THEN
            RPART=RADI(I)
            DELBO=RPART-XS(I)
            DELBO=MAX(DELBO,XS(I)+RPART-XMAX)
            DELBO=MAX(DELBO,RPART-YS(I))
            DELBO=MAX(DELBO,YS(I)+RPART-YMAX)
            IF ( DELBO.GT.0.0) THEN
              ITRY=ITRY+1
              IF (ITRY.GT.MAX(100,200*KSEED)) GO TO 1014
              GO TO 1013
            ENDIF
            EPSR2 = (DSMIN /100.)**2
            DO 1012 J=1,I-1
              DELX2=(XS(I)-XS(J))**2
              DELY2=(YS(I)-YS(J))**2
              DELH2=(HITE(I)-HITE(J))**2
              DELR2=((DIAM(I)+DIAM(J))/2.)**2
              IF ( DELX2+DELY2+DELH2.LT.DELR2+EPS2 ) THEN
                ITRY=ITRY+1
                IF (ITRY.GT.MAX(100,200*KSEED)) GO TO 1014
                GO TO 1013
              ENDIF
 1012       CONTINUE
          ENDIF
C
          SUMP=SUMP+3.14*RADI(I)**2
          PORO=1.-SUMP/(XMAX*YMAX)
          IF (PORO.LT.POROIN) GO TO 1014

 1016     KSEED=KSEED+1
 1010   CONTINUE
 1014   NSEED=KSEED
C
      ELSEIF (IMETH.EQ.2) THEN
        NSEED = 150
        PRINT*,'INPUT NSEED,ISED',NSEED,ISED
        READ*,NSEED,ISED
        ISEED0 = ISED
        DSMIN = SQRT(XMAX*YMAX) *0.80/SQRT(1.*NSEED)
        DREF = DSMIN
        ALOGSIG = 1.E-8
        PRINT*,'INPUT SIG',ALOGSIG

BIOREM$MAIN

        READ*,ALOGSIG
```

```
      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 /(NDIVX+1.)
      PRINT*,NDIVX,NDIVY,EPS
      YTOP = 0.
      DO 2010 I = 1,NSEED
        IDBC(I) = 0
        DONE(I) = 0
C
        ALOGMU=ALOG(DSMIN/2.)
        ZZZ=RAN(ISED)
        CALL ERRINV(ALOGR)
        DIAM(I)=2.*EXP(ALOGR)
C
        HITE(I)=DIAM(I)/2. *(1.-2.*RAN(ISED))
        RADI(I)=SQRT( (DIAM(I)/2.)**2 - HITE(I)**2 )
        BOTF = 2.* RAN(ISED)
C
        YMIN = YMAX
        RPART = IFILL *RADI(I)
        DO 2022 II = 1,NDIVX
          XST = (XMAX *II) /(NDIVX+1.)
          DELBO = XST - RPART
          DELBO = MIN (DELBO,XMAX-XST-RPART)
          IF (DELBO .LT. EPS) GO TO 2022
          DO 2023 JJ = 1,NDIVY
            YST = YMAX *(1.- (1.*JJ)/(NDIVY+1.))
            IF (YST .GT. YTOP+DEMAX0*DSMIN) GO TO 2023
            DELBO = YST - RPART
            IF (IFILL .EQ. 0) DELBO = YST - RADI(I) *BOTF
            IF ( DELBO .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
            JJJJMX = MIN (I-1, 4 *NACROS)
            DO 2012 JJJJ = 1,JJJJMX

BIOREM$MAIN

              J = I-1 - JJJJ + 1
              DELX2 = (XST-XS(J))**2
```

```
                DELY2 = (YST-YS(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. EPS) THEN
                  IF (YST .LT. YMIN) THEN
                    YMIN = YST
                    XS(I) = XST + 1.E-2 *RADI(I)
                    YS(I) = YST + 1.E-2 *RADI(I)
                  ENDIF
                  GO TO 2022
                ENDIF
 2012         CONTINUE
 2023       CONTINUE
 2022     CONTINUE
          YTOP = YS(I)
C
          PRINT1901,I,XS(I),YS(I),RADI(I)
          DELBO = YMAX-YS(I)-RPART
          IF (DELBO .LT. EPS) GO TO 2014
          KSEED = KSEED + 1
          SUMP = SUMP + PI *RADI(I)**2
          PORO = 1.-SUMP/(XMAX*YMAX)
C
 2010   CONTINUE
 2014   NSEED = KSEED
        PRINT*,NSEED
        PRINT*,PORO
        READ*, DDXX
C
      ELSEIF (IMETH.EQ.5) THEN
        NSEED = 150
        PRINT*,'INPUT ISED',ISED
        READ*,ISED
        ISEED0 = ISED
        PRINT*,'INPUT FIBER DIAM AND POROISTY',DIAMF,EPS0
        READ*,DIAMF,EPS0
        IRAG = 1
        PRINT*,'DO YOU WANT A RAGGED BOTTOM:  0 = NO    1 = YES',IRAG
        READ*,IRAG
        DSMIN = DIAMF
        DREF = DSMIN
        DELDIA = 0.
        IREDU = 1
        PRINT*,'INPUT IREDU,DELDIA',IREDU,DELDIA
        READ*,IREDU,DELDIA
        SWELL = 0.
        PRINT*,'INPUT SWELLING FRACTION',SWELL
        READ*,SWELL
        IOK=0
        ACON = 1
        PRINT*,'INPUT RANDOM VARIATION IN SPACING ( 0 TO 1) ',ACON
        READ*,ACON
        PRINT*,'INPUT OK:  0 = NO  1 = YES',IOK
        READ*,IOK

BIOREM$MAIN

        IF (IOK.EQ.0) GO TO 1519
C
        KSEED=0
```

```
      SUMP=0.
      NBSUM=0
      NSEED = 2 * XMAX *YMAX / (PI/4. *DIAMF**2) *(1.-EPS0)
      NDIVX = 20 *XMAX /DIAMF
      NDIVY = 20 *YMAX /DIAMF
      NDIV = MAX (NDIVX, NDIVY)
      NACROS = XMAX /DSMIN
      TOL = 1.1 *XMAX /(NDIVX+1.)
      RHO = 1. - EPS0
      DSMIN = DIAMF /SQRT(2.*RT3/PI*RHO)
      PRINT*,NDIVX,NDIVY,TOL
      YTOP = 0.
      DO 5010 I = 1,NSEED
        IDBC(I) = 0
        DONE(I) = 0
C
        DIAM(I)=DIAMF
C
        HITE(I)=0.
        RADI(I)=SQRT( (DIAM(I)/2.)**2 - HITE(I)**2 )
        BOTF = 2.* RAN(ISED)
        SPAC = (1. + ACON*(1.-BOTF)) *(DSMIN - DIAMF)
        SPAC = MAX (0., SPAC)
        SPACM = SPAC + TOL
        RANX = (1. - BOTF) *SPACM /100.
        RANY = (1. - 2.*RAN(ISED)) *SPACM /100.
C
        YMIN = YMAX
        JMIN1 = 0
        RPART = IFILL *RADI(I)
        DO 5022 II = 1,NDIVX
          XST = (XMAX *II) /(NDIVX+1.)
          DELBO = XST - RPART
          DELBO = MIN (DELBO,XMAX-XST-RPART)
          IF (DELBO .LT. TOL) GO TO 5022
          DO 5023 JJ = 1,NDIVY
            YST1 = YMAX *(1.- (1.*JJ)/(NDIVY+1.))
            NUMY = (YTOP+DSMIN+SPACM) *(NDIVY+1.) /YMAX + 1
            YST2 = YMAX *(NUMY-JJ+1.) /(NDIVY+1.)
            YST = MIN (YST1, YST2)
            DELBO = YST - RPART
            IF (IFILL .EQ. 0) DELBO = YST - RADI(I) *IRAG *BOTF
            IF ( DELBO .LT. TOL) THEN
              JJY = JJ
              YMIN = YST
              XS(I) = XST + RANX
              YS(I) = YST + RANY
              GO TO 5022
            ENDIF
            JJJJMX = MIN (I-1, 2 *NACROS)
            DO 5012 JJJJ = 1,JJJJMX
              J = I-1 - JJJJ + 1
              DELX2 = (XST-XS(J))**2
              DELY2 = (YST-YS(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. SPACM) THEN
```

```
                  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
 5012         CONTINUE
 5023       CONTINUE
 5022     CONTINUE
C
C---  FIND SECOND CLOSEST PARTICLE BELOW ---
C
          SMIN = YMAX
          JMIN2 = 0
          DO 5044 J = 1, I-1
            IF (J .EQ. JMIN1) GO TO 5044
            IF (YS(I)-YS(J) .LT. 3.*TOL .AND. JMIN1.GT.0) GO TO 5044
            DELX2 = (XS(I)-XS(J))**2
            DELY2 = (YS(I)-YS(J))**2
            DELH2 = (HITE(I)-HITE(J))**2
            DELR2 = ((DIAM(I)+DIAM(J))/2.)**2
            DIST = SQRT (DELX2+DELY2+DELH2) - SQRT (DELR2)
            IF (DIST .GT. DIAM(I)/3.) GO TO 5044
            IF (DIST .LT. SMIN) THEN
              SMIN = DIST
              JMIN2 = J
            ENDIF
 5044     CONTINUE
          SPAC2 = SMIN
C
C---  MOVE FIBER TO EQUILIBRIUM POSITION
C
          BCON = TOL /10.
C
          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)/2. + 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))

BIOREM$MAIN

            BB2 = - (XS(I)-XS(JMIN2))
C
          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))
            BB2 = - (XS(I)-XS(JMIN1))
C
          ELSEIF (JMIN1.GT.0 .AND. JMIN2.GT.0) THEN
            CC1 = (DIAM(I)/2. + DIAM(JMIN1)/2. + SPAC + BCON)**2
     1            - (XS(I)-XS(JMIN1))**2 - (YS(I)-YS(JMIN1))**2
            CC2 = (DIAM(I)/2. + DIAM(JMIN2)/2. + SPAC + BCON)**2
     1            - (XS(I)-XS(JMIN2))**2 - (YS(I)-YS(JMIN2))**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
C
          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)
C
          YTOP = MAX (YS(I), YTOP)
C
          DELBO = YMAX-YS(I)-RPART
          IF (DELBO .LT. TOL) GO TO 5014
          IF (JJY .EQ. 1) GO TO 5014
          PRINT1903,I,JMIN1,JMIN2,XS(I),YS(I),RADI(I),SPAC,DLTX,DLTY
          KSEED = KSEED + 1
          SUMP = SUMP + PI *RADI(I)**2
          PORO = 1.-SUMP/(XMAX*YMAX)
C
 5010   CONTINUE
 5014   NSEED = KSEED
        PRINT*,NSEED
        PRINT*,PORO
        READ*, DDXX
C
      ELSEIF (IMETH.EQ.3) THEN
        NSEED = 150
        PRINT*,'INPUT NSEED,ISED',NSEED,ISED
        READ*,NSEED,ISED
        ISEED0 = ISED
```

BIOREM$MAIN

```
        DSMIN = SQRT(XMAX*YMAX) *0.80/SQRT(1.*NSEED)
        DREF = DSMIN
        ALOGSIG = 1.E-8
        PRINT*,'INPUT SIG',ALOGSIG
        READ*,ALOGSIG
        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 = 10 *XMAX /DSMIN
      NDIVY = 10 *YMAX /DSMIN
      NDIVZ = 10
      NDIV = MAX (NDIVX, NDIVY)
      NACROS = XMAX /DSMIN
      EPS = 1.1 *XMAX /(NDIVX+1.)
      PRINT*,NDIVX,NDIVY,EPS
      YTOP = 0.
      DO 3010 I = 1,NSEED
        IDBC(I) = 0
        DONE(I) = 0
C
        ALOGMU=ALOG(DSMIN/2.)
        ZZZ=RAN(ISED)
        CALL ERRINV(ALOGR)
        DIAM(I)=2.*EXP(ALOGR)
        BOTF = 2.* RAN(ISED)
C
        YMIN = YMAX
        DO 3022 II = 1,NDIVX
          XST = (XMAX *II) /(NDIVX+1.)
          DELBO = XST - RPART
          DELBO = MIN (DELBO,XMAX-XST-RPART)
          IF (DELBO .LT. EPS) GO TO 3022
          DO 3021 KK = 1,NDIVZ
            HIT = DIAM(I)/2. *(1.- 2.*(1.*KK)/(NDIVZ+1.))
            IF (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+DEMAX0*DSMIN) GO TO 3023
              DELBO = YST - RPART
              IF (IFILL .EQ. 0) DELBO = YST - RAD *BOTF
              IF ( DELBO .LT. EPS) THEN
                YMIN = YST
```

```
                XS(I) = XST + 1.E-2 *RAD
                YS(I) = YST + 1.E-2 *RAD
                HITE(I) = HIT
                RADI(I) = RAD
                GO TO 3022
              ENDIF
              JJJJMX = MIN (I-1, 3 *NACROS)
```

```
                DO 3012 JJJJ = 1,JJJJMX
                  J = I-1 - JJJJ + 1
                  DELX2 = (XST-XS(J))**2
                  DELY2 = (YST-YS(J))**2
                  DELH2 = (HIT-HITE(J))**2
                  DELR2 = ((DIAM(I)+DIAM(J))/2.)**2
                  DIST = SQRT (DELX2+DELY2+DELH2) - SQRT (DELR2)
                  IF (DIST .LT. EPS) THEN
                    IF (YST .LT. YMIN) 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
                    GO TO 3021
                  ENDIF
 3012           CONTINUE
 3023         CONTINUE
 3021       CONTINUE
 3022     CONTINUE
          YTOP = YS(I)
C
          PRINT1901,I,XS(I),YS(I),2.*HITE(I)/DIAM(I),RADI(I)
          DELBO = YMAX-YS(I)-RPART
          IF (DELBO .LT. EPS) GO TO 3014
          KSEED = KSEED + 1
          SUMP = SUMP + PI *RADI(I)**2
          PORO = 1.-SUMP/(XMAX*YMAX)
C
 3010   CONTINUE
 3014   NSEED = KSEED
        PRINT*,NSEED
        PRINT*,PORO
        READ*,DDXX
C
      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 POROISTY',DIAMF,EPS0
        READ*,DIAMF,EPS0
        DREF = DIAMF
        DELDIA = 0.
        IREDU = 1
        PRINT*,'INPUT IREDU,DELDIA',IREDU,DELDIA
        READ*,IREDU,DELDIA
        ACON = 0
```

```
        PRINT*,'INPUT RANDOM VARIATION IN SPACING ( 0 TO 1) ',ACON
        READ*,ACON
        ISED = 12344321
        PRINT*,'INPUT SEED FOR RANDOM DISPLACEMENT',ISED
        READ*,ISED
C
        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.
        FAC = 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
        SPAC0 = 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.EQ.0) GO TO 1519
C
        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)
C
1015        CONTINUE
1025      CONTINUE
```

BIOREM$MAIN

```
        KSEED = ID
        NSEED = KSEED
        NPAR = NSEED
      ENDIF
C
C---  ROUGH POROSITY, SPAC, ETC.  ---
C
      NPAR = NSEED
      VSUM = 0.
```

```
      RBAR = 0.
      DBAR = 0.
      DO 887 ID = 1, NPAR
        RBAR = RBAR + RADI(ID)
        DBAR = DBAR + DIAM(ID)
        ARID = PI * RADI(ID)**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
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
 1784   NBORS = 6
        NDARR = 18
        CALL NERNEB (NDARR,NBORS,NSEED,XS,YS,X0,Y0,DNEIGH,INEIGH)
        DELS = SPAC /10.
        DELS0 = DELS
        KCYMX = DISD *DIAMF /DELS
        IF (KCYMX .GT. 0) THEN
          IF (IPLTWK .EQ. 1) 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
            XSTP = (XMAX - XORIG) /1.
            YSTP = (YMAX - YORIG) /1.
```

```
            CALL AREA2D(XAXIS,YAXIS)
            CALL GRAF(XORIG,XSTP,XMAX,YORIG,YSTP,YMAX)
            CALL FRAME
            CALL MARKER(15)
            CALL CURVE(XS,YS,NSEED,-1)
          ENDIF
C
          SLMAX = 0.
          ITRYMX = 100
          DO 1091 KCY = 1, KCYMX
```

```
      JMOV = 0
      DO 1092 I = 1, NSEED
        ITRY = 0
        DELS = DELS0
651     THET = 2. * PI *RAN(ISED)
        DELX = DELS *COS(THET)
        DELY = DELS *SIN(THET)
        XNEW = XS(I) + DELX
        YNEW = YS(I) + DELY
        TOL = DBOX(I)
        IF (XNEW .LT. TOL) XNEW = 2.*TOL - XNEW
        IF (XNEW .GT. XMAX-TOL) XNEW = 2.*(XMAX-TOL) - XNEW
        IF (YNEW .LT. TOL) YNEW = 2.*TOL - YNEW
        IF (YNEW .GT. YMAX-TOL) YNEW = 2.*(YMAX-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
1093    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)
1092  CONTINUE
C
      IF (MOD(KCY-1, MAX(1,KCYMX/500)) .EQ. 0) THEN
        CALL SPACEF (57,NSEED,XS,YS,XMAX,YMAX,SDEVX,SDEVY)
        WRITE(16,900) KCY*DELS/DIAMF,SDEVX,SDEVY
      ENDIF
C
      IF (SLMAX .GT. DIAMF/3.) THEN
        DISMX = 0.
        DISBR = 0
        DISX  = 0.
        DISY  = 0.
        DO 1855 ID = 1, NSEED
          DSID = (XS(ID)-WORK(1,ID))**2 + (YS(ID)-WORK(2,ID))**2
          DSID = SQRT (DSID)
          DISMX = MAX (DISMX, DSID)
          DISBR = DISBR + DSID
          DISX = MAX (DISX, ABS(XS(ID)-WORK(1,ID)))
          DISY = MAX (DISY, ABS(YS(ID)-WORK(2,ID)))
```

```
          IF (IPLTWK .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
1855    CONTINUE
        DISBR = DISBR /NSEED
        CALL SPACEF (87,NSEED,XS,YS,XMAX,YMAX,SDEVX,SDEVY)
        WRITE(17,900) KCY*DELS/DIAMF,DISMX/DIAMF,DISBR/DIAMF,
```

```
     1                    SDEVX,SDEVY,DISMX/SPAC,DISBR/SPAC
C
              SLMAX = 0.
              CALL NERNEB (NDARR,NBORS,NSEED,XS,YS,X0,Y0,DNEIGH,INEIGH)
C
              IF (IPLTWK .EQ.0) THEN
                CBAR = 0
                NSUM = 0
                DO 1101 ID = 1, NSEED
                  DBXX = MIN (XS(ID), XMAX-XS(ID)) /DIAMF
                  DBXY = MIN (YS(ID), YMAX-YS(ID)) /DIAMF
                  IF (DBXX.LT.2.0 .OR. DBXY.LT.2.0) GO TO 1101
                  DO 1101 J = 1, NBORS
                    NSUM = NSUM + 1
                    CBAR = CBAR + DNEIGH(J,ID)
 1101           CONTINUE
                CBAR = CBAR /NSUM
                SDEV = 0.
                DO 1102 ID = 1, NSEED
                  DBXX = MIN (XS(ID), XMAX-XS(ID)) /DIAMF
                  DBXY = MIN (YS(ID), YMAX-YS(ID)) /DIAMF
                  IF (DBXX.LT.2.0 .OR. DBXY.LT.2.0) GO TO 1102
                  DO 1102 J = 1, NBORS
                  SDEV = SDEV +(DNEIGH(J,ID)-CBAR)**2
 1102           CONTINUE
                SDEV = SQRT (SDEV /(NSUM-1.))
                PRINT901,JMOV,CBAR,SDEV,SDEVX,SDEVY,DISX,DISY
              ENDIF
            ENDIF
 1091     CONTINUE
C
          IF (IPLTWK .EQ. 1) CALL ENDPL(0)
          ICRAN = 1
          PRINT*,'MORE RANDOMIZING ?  0 = NO  1 = YES',ICRAN
          READ*,ICRAN
          IF (ICRAN .EQ. 1) THEN
            CALL SPACEF (87,NSEED,XS,YS,XMAX,YMAX,SDEVX,SDEVY)
            PRINT900,SDEX,SDEVY
            PRINT*,'INPUT FACTOR FOR RANDOMIZATION',DISD
            READ*,DISD
            PRINT*,'PLOT RANDOM MOTION ?  0 = NO  1 = YES',IPLTWK
            READ*,IPLTWK
            GO TO 1784
          ENDIF
        ENDIF
C
```

BIOREM$MAIN

```
C---  TAKE SUBSET OF PARTICLES
C
      ISUB = 0
      PRINT*,'TAKE SUBSET OF PARTICLES ?  0 = NO  1 = YES',ISUB
      READ*,ISUB
      IF (ISUB .EQ. 1) THEN
        X0NEW = 0.
        Y0NEW = 0.
        XMAXN = XMAX
        YMAXN = YMAX
        PRINT*,'INPUT XORIG, XMAX, YORIG, YMAX',
     1                X0NEW, XMAXN, Y0NEW, YMAXN
```

```
        READ*, X0NEW, XMAXN, Y0NEW, YMAXN
        XMAX = XMAXN - X0NEW
        YMAX = YMAXN - Y0NEW
        DO 519 ID = 1, NSEED
          XS(ID) = XS(ID) - X0NEW
          YS(ID) = YS(ID) - Y0NEW
 519    CONTINUE
      ENDIF
C
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))
        DBZZ = MIN (DBXX, DBXY)
        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)
          HITE0(NSUM) = HITE(ID)
C       ELSEIF (DBZZ .GT. EPS-RADI(ID)) THEN
        ELSEIF (DBZZ .GT. -0.5 *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, NSUM
        XS(ID) = X0(ID)
        YS(ID) = Y0(ID)
        DIAM(ID) = DIAM0(ID)
        HITE(ID) = HITE0(ID)
 1189 CONTINUE
      DO 1190 ID = 1, NOUT
        XS(ID+NSUM) = WORK(1,ID)
        YS(ID+NSUM) = WORK(2,ID)
        DIAM(ID+NSUM) = WORK(3,ID)
```

```
        HITE(ID+NSUM) = WORK(4,ID)
 1190 CONTINUE
      NSEED = NSUM + NOUT
      NPAR = NSEED
C
C---  ASSIGN RADII AND SWELL PARTICLES  ---
C
      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)
 1163 CONTINUE
      DIAMF = DIAMF /NPAR
C
C---  MONTE CARLO POROSITY  ---
C
  491 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
  493   CONTINUE
        NVOID = NVOID + 1
  492 CONTINUE
      EPSMC = (1. *NVOID) /NSHOT
C
C---  SUM POROSITY  ---
C
      SUMV = 0.
      DIAMF = 0.
      DO 677 ID = 1, NSEED
        DIAM0(ID) = DIAM(ID)
        DIAMF = DIAMF + DIAM(ID)
        DELBX = MIN (XS(ID), XMAX-XS(ID))
        DELBY = MIN (YS(ID), YMAX-YS(ID))
        DELB = MIN (DELBX, DELBY)
        ABSD = MIN (ABS(DELB), 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 (DELB .GT. 0.) THEN
            VID = VID - DELV
          ELSE
            VID = DELV
          ENDIF
```

BIOREM$MAIN

```
        ENDIF
        SUMV = SUMV + VID
  677 CONTINUE
      PORO = 1. - SUMV /XMAX /YMAX
      DIAMF = DIAMF /NSEED
      PRINT980,'PORO', PORO
C
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
```

```
 1044     CONTINUE
            XTMP = YMAX
            YMAX = XMAX
            XMAX = XTMP
          ENDIF
C
C---      INTERMEDIATE DUMP  ---
C
          WRITE (99,*) NPAR,XMAX,YMAX,EPS0,DIAMF
          DO 692 ID = 1,NPAR
            WRITE(99,*) XS(ID),YS(ID),HITE(ID),DIAM(ID),RADI(ID)
  692     CONTINUE
C
C---      INITIALIZE PLOT  ---
C
          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
          XSTP=(XMAX-XORIG)/1.
          YSTP=(YMAX-YORIG)/1.
C
          CALL AREA2D(XAXIS,YAXIS)
          CALL GRAF(XORIG,XSTP,XMAX,YORIG,YSTP,YMAX)
          CALL FRAME
C
          PRINT*,' '
          PRINT*,' '
          PRINT*,' '
          PRINT*,' '
          PRINT981,'NPAR',NPAR
          PRINT980,'XMAX',XMAX
          PRINT980,'YMAX',YMAX
          PRINT980,'EPS0',EPS0
          PRINT980,'PORO',PORO
          PRINT980,'EPSM',EPSMC
          PRINT980,'SPAC',SPAC
```

```
          PRINT980,'DREF',DREF
C
C---      PLOTT SEEDS
C
          IF (IPLTS.EQ.1) THEN
            CALL SCLPIC(0.5)
            CALL MARKER(18)
            CALL CURVE(XS,YS,NSEED,-1)
            CALL RESET('MARKER')
            CALL SCLPIC(1.)
          ENDIF
C
          RJBAR=0.
          ABAR=0.
          NSUM=0
```

```
      NSUMOLD=0
      JLMAX=0
      JLMIN=200
      SUMLB=0.
      NBSUM=0
      DO 1621 K=1,20
1621    FSID(K)=0
C
      DO 1100 I=1,NSEED
C
C---  COMPUTE LINE CONSTANTS FOR Ith SEED  ---
C
        JL=0
        DEMX=DEMAX0*1.6/SQRT(1.*NSEED)
        DO 1030 J=1,NSEED
          IF (I.EQ.J) GO TO 1030
          DELS = SQRT((XS(I)-XS(J))**2 + (YS(I)-YS(J))**2)
          IF (DELS.GT.DEMX) GO TO 1030
          JL = JL + 1
          X0(JL) = (XS(I)+XS(J)) /2.
          Y0(JL) = (YS(I)+YS(J)) /2.
          DELX = XS(J) - XS(I)
          DELY = YS(J) - YS(I)
          DELY = SIGN(MAX(ABS(DELY),1.E-8),DELY)
          SL(JL) = -DELX /DELY
          JB(JL) = J
1030    CONTINUE
        NLIN = JL
C
C---  ADD BOUNDARY DOMAIN LINES  ---
C
        EPS = 0.
        RRR = 0.5 + EPS
        DO 1040 J = 1, NBOX
C
          THET = 2. *PI *(J-1)/NBOX - PI /2.
          X000 = (0.5 + RRR *COS(THET)) *XMAX
          Y000 = (0.5 + RRR *SIN(THET)) *YMAX
          SL00 = - 1. /SIGN (MAX (ABS(TAN(THET)),1.E-10), TAN(THET))
C
C---  REF: PURCEL PAGE 50
C
```

BIOREM$MAIN

```
          BBB = 1.
          AAA = - SL00
          CCC = SL00 *X000 - Y000
          DELS = ABS(AAA*XS(I)+BBB*YS(I)+CCC)/SQRT(AAA**2+BBB**2)
          IF (DELS.GT.DEMX) GO TO 1040
          JL = JL + 1
          JB(JL) = - J
          X0(JL) = X000
          Y0(JL) = Y000
          SL(JL) = SL00
1040    CONTINUE
        JLMAX = MAX (JL, JLMAX)
        JLMIN = MIN (JLMIN, JL)
C
C---  LOCATE NEAREST (PARTICLE) BOUNDING LINE  ---
C
```

```
        DMIN = 1.E8
        EPS = 1.E-4 *SMEAS
        DO 1050 J = 1, NLIN
C
C---  NEW STUFF  ---
C
          IF (X0(J).GT.XMAX-EPS .OR. X0(J).LT.EPS) GO TO 1050
          IF (Y0(J).GT.YMAX-EPS .OR. Y0(J).LT.EPS) GO TO 1050
          IF (JB(J) .GT. NSEED-NOUT) GO TO 1050
C
          DELS=SQRT((XS(I)-X0(J))**2+(YS(I)-Y0(J))**2)
          IF (DELS.LT.DMIN) THEN
            DMIN=DELS
            JMN=J
          ENDIF
1050      CONTINUE
C
        JC=-1
        JN=JMN
        XX(1)=X0(JN)
        YY(1)=Y0(JN)
C
C---  START WALK AROUND CELL BOUNDARY  ---
C
        KSAV=1
        NSYD=0
        MSID=0
        RAVE=0.
        RMIN=1.
        RMAX=0.
        ASUM=0.
        EPS = 1.E-2 *SMEAS
        DO 1110 K=1,20
C
          JO=JC
          JC=JN
C
          DMIN=SQRT(2.)
          XVJO=XX(K)
          YVJO=YY(K)
          DO 1120 J=1,JL
```

BIOREM$MAIN

```
C
            IF (J.EQ.JC) GO TO 1120
            IF (J.EQ.JO) GO TO 1120
C
            DSLO = SL(JC) - SL(J)
            DSLO = SIGN (MAX(ABS(DSLO),1.E-6), DSLO)
            XVJ=(  (Y0(J)-SL(J)*X0(J))
     1                -(Y0(JC)-SL(JC)*X0(JC)) )/DSLO
            YVJ=Y0(JC)+SL(JC)*(XVJ-X0(JC))
            IF (ABS(SL(J)).LT.ABS(SL(JC))) THEN
              YVJ=Y0(J)+SL(J)*(XVJ-X0(J))
            ENDIF
C
C---  NEW STUFF ---
C
            IF (XVJ.GT.XMAX+EPS .OR. XVJ.LT.-EPS) GO TO 1120
            IF (YVJ.GT.YMAX+EPS .OR. YVJ.LT.-EPS) GO TO 1120
```

```
C
C---  CHECK FOR COUNTER-CLOCKWISE PATH  ---
C     NOTE:  PUT THIS FIX IN BONZO
C
            IF (JC .LE. NLIN) THEN
              DIREC1 = (X0(JC)-XS(I)) *(YVJ-YVJO)
              DIREC2 = (YS(I)-Y0(JC)) *(XVJ-XVJO)
            ELSE
              DIREC1 = XVJ - XVJO
              DIREC2 = YVJ - YVJO
              IF (JB(JC).LT.-2) THEN
                DIREC1 = - DIREC1
                DIREC2 = - DIREC2
              ENDIF
            ENDIF
            DIREC = DIREC1
            IF (ABS(DIREC2) .GT. ABS(DIREC1)) DIREC = DIREC2
            IF (DIREC. LT. 0.0) GO TO 1120
C
            DVV=SQRT((XVJO-XVJ)**2+(YVJO-YVJ)**2)
            IF (DVV.LT.DMIN) THEN
              JN=J
              XX(K+1)=XVJ
              YY(K+1)=YVJ
              XSID(K,I)=XVJ
              YSID(K,I)=YVJ
              DMIN=DVV
              DSXMIN=DSX
              DVXMIN=DVX
              DSVMIN=DSV
              DSLOMIN=DSLO
            ENDIF
 1120     CONTINUE
          KSAV=KSAV+1
C
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)
            NBSUM=NBSUM+1
```

BIOREM$MAIN

```
            IF (JC.GT.NLIN) THEN
              LBON(K-1,I)=1.E10
              NBSUM=NBSUM-1
            ELSE
              L=JB(JC)
              LBON(K-1,I)=SQRT((XS(I)-XS(L))**2+(YS(I)-YS(L))**2)
              SUMLB=SUMLB+LBON(K-1,I)
            ENDIF
          ENDIF
C
C       COMPUTE AREA OF TRIANGULAR SEGMENT
C
          IF (K.GT.1) THEN
            DELH=SQRT((XS(I)-X0(JC))**2+(YS(I)-Y0(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)
```

```
              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)
            IF (IDBC(I).EQ.0) THEN
              IDBC(I)=IPRM(K,I)
              LBC(I)=K
            ELSEIF (IDBC(I).EQ.-1 .OR. IDBC(I).EQ.-3) THEN
              IDBC(I)=IPRM(K,I)
              LBC(I)=K
            ENDIF
C
          ELSE
            IDP=JB(JN)
            IPRM(K,I)=IDP
            IF (IDP.LT.I) THEN
              DO 1744 L=1,NSID(IDP)
                IF (IPRM(L,IDP).EQ.I) THEN
                  LPRM(K,I)=L
                  LPRM(L,IDP)=K
                ENDIF
1744          CONTINUE
            ENDIF
          ENDIF
C
          IF (JN.LE.NLIN) THEN
            XSJ=XS(JB(JN))
            YSJ=YS(JB(JN))
            DDSJ=SQRT((XS(I)-XSJ)**2+(YS(I)-YSJ)**2)
            RAVE=RAVE+DDSJ
            RMIN=MIN(RMIN,DDSJ)

BIOREM$MAIN

            RMAX=MAX(RMAX,DDSJ)
            CMIN=MIN(CMIN,DDSJ)
            CMAX=MAX(CMAX,DDSJ)
            MSID=MSID+1
          ENDIF
C
          NSYD=NSYD+1
C
C---  PLOTT BONDS  ---
C
          IF (IPLTB.EQ.1 .AND. JN.LE.NLIN) THEN
            XB(1)=XS(I)
            YB(1)=YS(I)
            XB(2)=XS(JB(JN))
            YB(2)=YS(JB(JN))
            CALL CURVE(XB,YB,2,1)
          ENDIF
C
1110    CONTINUE
```

```
1130   CONTINUE
       RAVE=RAVE/MAX(MSID,1)
C
C---  COMPUTE SIDE AND AREA DIAGNOSTICS  ---
C
       NSUM=NSUM+1
       RJBAR=RJBAR+NSYD
       FSID(NSYD)=FSID(NSYD)+1
       GAVE=GAVE+RAVE
       GMIN=GMIN+RMIN
       GMAX=GMAX+RMAX
       ABAR=ABAR+ASUM
C
C---  PLOTT CELL BOUNDARIES  ---
C
       IF (IPLTC.EQ.1) THEN
        IF (YS(I) .LT. 0.5) THEN
        CALL CURVE(XX,YY,KSAV,0)
        CALL RESET ('DASH')
        ENDIF
       ENDIF
C
       AREA(I)=ASUM
       NSID(I)=NSYD
C
1100  CONTINUE
C
C---  NUMBER NODES, ASSIGN X-Y COORDINATES, NODE CORRESPONDANCE AND
C     FLAG BOUNDARY NODES  ---
C
      IDN = 0
      DO 1400 ID=1,NSEED
       DO 1400 L=1,NSID(ID)
        IF (IDNOD(L,ID).EQ.0) THEN
C
         IDN = IDN + 1
         XNOD(IDN) = XSID(L,ID)
         YNOD(IDN) = YSID(L,ID)
```

BIOREM$MAIN

```
         IDNOD(L,ID) = IDN
         IDP = IPRM(L,ID)
         IF (IDP.GT.0) THEN
          LP = LPRM(L,ID)
          LPP1 = MOD(LP,NSID(IDP))+1
          IDNOD(LPP1,IDP) = IDN
         ENDIF
         LM1 = MOD(L+NSID(ID)-2,NSID(ID))+1
         IDP = IPRM(LM1,ID)
         IF (IDP.GT.0) THEN
          LP = LPRM(LM1,ID)
          IDNOD(LP,IDP) = IDN
         ENDIF
C
         NODAR(8,IDN) = 0
         IDP = IPRM(L,ID)
         IDPM1 = IPRM(LM1,ID)
         IF (IDP.LT.0) NODAR(8,IDN) = IDP
         IF (IDPM1.LT.0) NODAR(8,IDN) = IDPM1
         IF (IDP.EQ.-IDBIN .OR. IDPM1.EQ.-IDBIN )
```

```
     1            NODAR(8,IDN) = -IDBIN
            IF (IDP.EQ.-IDBOU .OR. IDPM1.EQ.-IDBOU )
     1            NODAR(8,IDN) = -IDBOU
C
          ENDIF
 1400 CONTINUE
      NNOD = IDN
C
C  NUMBER TUBES, IDENTIFY TUBE NODES, LENGTHS AND DIAMETERS
C  ASSIGN TUBE ENDPOINTS AND XY SHIFTS
C
 1610 IDT = 0
      DO 1402 ID=1,NSEED
        DO 1402 L=1,NSID(ID)
          IDP = IPRM(L,ID)
          IF (IDP.GT.ID .OR. IDP.LT.0) THEN
C
            IDT = IDT + 1
            TUBNOD(1,IDT) = IDNOD(L,ID)
            TUBNOD(3,IDT) = ID
            TUBNOD(4,IDT) = IDP
            LP1 = MOD(L,NSID(ID))+1
            TUBNOD(2,IDT) = IDNOD(LP1,ID)
            TUBAR(1,IDT) = MAX (LSID(L,ID),1.E-6*XMAX)
            TUBAR(9,IDT) = 1.* IDP
            IDTUB(L,ID) = IDT
            IF (IDP.GT.0) THEN
              LP = LPRM(L,ID)
              IDTUB(LP,IDP) = IDT
              DEL=SQRT( (XS(ID)-XS(IDP))**2 + (YS(ID)-YS(IDP))**2 )
              IF (IREDU.EQ.1) THEN
                TUBAR(2,IDT) = DEL - RADI(ID) - RADI(IDP) - DELDIA
              ELSEIF (IREDU.EQ.2) THEN
                TUBDI = DEL - RADI(ID) - RADI(IDP)
                TUBAR(2,IDT) = TUBDI *(1. - DELDIA)
              ELSEIF (IREDU.EQ.3) THEN
                TUBDI = DEL - RADI(ID) - RADI(IDP)
```

```
                TUBAR(2,IDT) = TUBDI /(1. + DELDIA*TUBDI)
              ENDIF
            ELSE
              IF (IDP .EQ. -1) DEL = YS(ID)
              IF (IDP .EQ. -2) DEL = XMAX - XS(ID)
              IF (IDP .EQ. -3) DEL = YMAX - YS(ID)
              IF (IDP .EQ. -4) DEL = XS(ID)
              IF (IREDU.EQ.1) THEN
                TUBAR(2,IDT) = DEL - RADI(ID) - DELDIA/2.
              ELSEIF (IREDU.EQ.2) THEN
                TUBDI = DEL - RADI(ID)
                TUBAR(2,IDT) = TUBDI *(1. - DELDIA)
              ELSEIF (IREDU.EQ.3) THEN
                TUBDI = DEL - RADI(ID)
                TUBAR(2,IDT) = TUBDI /(1. + TUBDI*DELDIA)
              ENDIF
            ENDIF
            KBLOC(IDT) = 1
            IF (TUBAR(2,IDT) .LT. 1.E-8*XMAX) THEN
              KBLOC(IDT) = 0
              TUBAR(2,IDT) = 1.E-8 *XMAX
```

```
            ENDIF
C
            IF (IDP.GT.0) THEN
              DEL=SQRT( (XS(ID)-XS(IDP))**2 + (YS(ID)-YS(IDP))**2 )
              DELTA = (RADI(ID)-RADI(IDP))/2.
              DXDL=(XS(IDP)-XS(ID))/DEL
              DYDL=(YS(IDP)-YS(ID))/DEL
              DELTAX = DELTA *DXDL
              DELTAY = DELTA *DYDL
            ELSE
              DELTAX = 0.
              DELTAY = 0.
              IF (IDP .EQ. -1) DELTAY = TUBAR(2,IDT) /2.
              IF (IDP .EQ. -2) DELTAX = - TUBAR(2,IDT) /2.
              IF (IDP .EQ. -3) DELTAY = - TUBAR(2,IDT) /2.
              IF (IDP .EQ. -4) DELTAX = TUBAR(2,IDT) /2.
            ENDIF
            TUBAR(3,IDT) = XSID(L,ID) + DELTAX
            TUBAR(4,IDT) = YSID(L,ID) + DELTAY
            TUBAR(5,IDT) = XSID(L+1,ID) + DELTAX
            TUBAR(6,IDT) = YSID(L+1,ID) + DELTAY
            TUBAR(7,IDT) = DELTAX
            TUBAR(8,IDT) = DELTAY
C
          ENDIF
1402  CONTINUE
      NTUB = IDT
C
C---  IDENTIFY NODE TRIOS AND ASSOCIATED TUBE NUMBERS  ---
C     SEQUENCE OF TRIOS IS IN ANTI-CLOCKWISE DIRECTION
C
      DO 1406 ID=1,NSEED
        DO 1406 L=1,NSID(ID)
          IDP = IPRM(L,ID)
          IF (IDP.GT.ID .OR. IDP.LT.0) THEN
            IDN = IDNOD(L,ID)
```

BIOREM$MAIN

```
            IF (NODAR(1,IDN).EQ.0) THEN
              LP1 = MOD(L,NSID(ID))+1
              LM1 = MOD(L+NSID(ID)-2,NSID(ID))+1
              NODAR(2,IDN) = IDNOD(LP1,ID)
              NODAR(3,IDN) = IDNOD(LM1,ID)
              NODAR(5,IDN) = IDTUB(L,ID)
              NODAR(6,IDN) = IDTUB(LM1,ID)
              NODAR(9,IDN) = ID
              NODAR(10,IDN) = IPRM(L,ID)
              NODAR(11,IDN) = IPRM(LM1,ID)
              NODAR(1,IDN) = 2
              IDP = IPRM(L,ID)
              IF (IDP.GT.0) THEN
                LP = LPRM(L,ID)
                LPP1 = MOD(LP,NSID(IDP))+1
                IF (IDNOD(L,ID).EQ.IDNOD(LPP1,IDP)) THEN
                  LPP2 = MOD(LPP1,NSID(IDP))+1
                  NODAR(4,IDN) = IDNOD(LPP2,IDP)
                  NODAR(7,IDN) = IDTUB(LPP1,IDP)
                  NODAR(1,IDN) = 3
                  GO TO 1406
                ENDIF
```

```
              ENDIF
              IDP = IPRM(LM1,ID)
              NODAR(10,IDN) = IDP
              IF (IDP.GT.0) THEN
                LP = LPRM(LM1,ID)
                IF (IDNOD(L,ID).EQ.IDNOD(LP,IDP)) THEN
                  LPM1 = MOD(LP+NSID(IDP)-2,NSID(IDP))+1
                  NODAR(4,IDN) = IDNOD(LPM1,IDP)
                  NODAR(7,IDN) = IDTUB(LPM1,IDP)
                  NODAR(1,IDN) = 3
                ENDIF
              ENDIF
            ENDIF
          ENDIF
 1406 CONTINUE
C
C---  TABULATE AND STORE BOUNDARY NODES AND CROSSING TUBES  ---
C     NNBC(J) IS THE NUMBER OF NODES/TUBES ON BOUNDARY J; NBID(I,J) IS
C     THE NODE NODE NUMBER OF THE JTH NODE ON BOUNDARY I; TBID(I,J) IS
C     THE TUBE NUMBER OF THE JTH TUBE CROSSING BOUNDARY I.
C
      NNBC(1) = 0
      NNBC(2) = 0
      NNBC(3) = 0
      NNBC(4) = 0
      DO 1447 IDN = 1, NNOD
        IDB = - NODAR(8,IDN)
        IF (IDB .GT. 0) THEN
          NNBC(IDB) = NNBC(IDB) + 1
          NBID(IDB,NNBC(IDB)) = IDN
          DO 1448 J = 1, NODAR(1,IDN)
            IDNJ = NODAR(1+J,IDN)
            IF (NODAR(8,IDN) .NE. NODAR(8,IDNJ)) THEN
              TBID(IDB,NNBC(IDB)) = NODAR(4+J,IDN)
              GO TO 1447

BIOREM$MAIN

            ENDIF
 1448     CONTINUE
        ENDIF
 1447 CONTINUE
C
C---  PLOTT PARTICLES
C
      IF (IPLTP.EQ.1) THEN
        DO 1811 ID = 1, NSEED
          XCEN = XS(ID) *XAXIS /XMAX
          YCEN = YS(ID) *YAXIS /YMAX
          RADIN = 0.95 *RADI(ID) *XAXIS /XMAX
          CALL BLCIR (XCEN, YCEN, RADIN, 0)
 1811   CONTINUE
      ENDIF
C
      IF (IPLTP.NE.0) THEN
        IF (IPLTP .EQ. 1) THEN
          CALL MARKER(16)
CXXX         CALL BLSYM
        ELSEIF (IPLTP .EQ. -1) THEN
          CALL MARKER(15)
        ENDIF
```

```
        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
          XB(1)=XS(ID)
          YB(1)=YS(ID)
          FAC= 2.* RADI(ID)/XMAX *XAXIS/0.082
          CALL SCLPIC(FAC)
          CALL CURVE(XB,YB,1,1)
 1801   CONTINUE
        CALL SCLPIC(1.)
        CALL RESET('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
C         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. *(XS(ID) - XS(IDP))
              XAA(J,2) = -2. *(YS(ID) - YS(IDP))
              XAA(J,3) = -2. *(RADI(ID) - RADI(IDP))
              RHS(J) = RADI(ID)**2 - RADI(IDP)**2
     1           - XS(ID)**2 + XS(IDP)**2 - YS(ID)**2 + YS(IDP)**2
              QHS(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.
                IF (IDP .EQ. -2) RHS(J) = XMAX
              ELSE
                XAA(J,2) = 1.
                RHS(J) = 0.
                IF (IDP .EQ. -3) RHS(J) = YMAX
              ENDIF
            ENDIF
 3019     CONTINUE
C
```

```
          WT = 1.
          WTR = 0.3
          RTU(IDN) = 1.1 *RMIN
          RNEW = RTU(IDN)
          ITSK = 0
          DO 3077 ITS = 1, 50
            ITSK = ITSK + 1
C
            RTU(IDN) = (1.-WTR) *RTU(IDN) + WTR *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)
            DETX = XHS(1)    *XAA(2,2) - XAA(1,2) *XHS(2)
            DETY = XAA(1,1) *XHS(2)  - XHS(1)    *XAA(2,1)
            XNEW = DETX /DETA
            YNEW = DETY /DETA
            XTU(IDN) = (1.-WT) *XTU(IDN) + WT *XNEW
            YTU(IDN) = (1.-WT) *YTU(IDN) + WT *YNEW
C
            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(RTUF) - 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

BIOREM$MAIN

            IF (ERR .LT. 1.E-6*XMAX) GO TO 3078
C
 3077     CONTINUE
 3078     CONTINUE
          RTU(IDN) = MAX (RTU(IDN), 1.E-30)
C
 3018   CONTINUE
C
C---  COMPUTE EFFECTIVE PORE RADII  ---
C
        PORSUM = 0.
        DO 3017 IDN = 1, NNOD
          APORE(IDN) = 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.
              XMID = (XS(ID) + XS(IDP)) /2.
              YMID = (YS(ID) + YS(IDP)) /2.
              HIGHT2 = (XNOD(IDN)-XMID)**2 + (YNOD(IDN)-YMID)**2
              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
                ATRI = (RADI(IDP)-RLAP) *HLAP /2.
                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
                RLEG = ABS (YS(ID) - YNOD(IDN))
              ELSE
                RLEG = ABS (XS(ID) - XNOD(IDN))
              ENDIF
              HIGHT = SQRT (HYPOT2 - RLEG**2)
              DTHETA = PI /2. - ACOS (RLEG /HYPOT)
              IF (RADI(ID) .LE. HIGHT) THEN
                APART = DTHETA /2. *RADI(ID)**2
              ELSE
                RLAP = RADI(ID) - HIGHT
                PHI = ACOS ((RADI(ID)-RLAP) /RADI(ID))
```

```
                HLAP = RADI(ID) *SIN (PHI)
                ATRI = (RADI(ID)-RLAP) *HLAP /2.
                APART = (DTHETA-PHI) /2. *RADI(ID)**2 + ATRI
              ENDIF
              DAREA = MAX (0., HIGHT *RLEG /2. - APART)
            ENDIF
            APORE(IDN) = APORE(IDN) + DAREA
 3016      CONTINUE
          PORSUM = PORSUM + APORE(IDN)
 3017    CONTINUE
        EPSVOID = PORSUM /(XMAX*YMAX)
C
C---  PLOTT NODES  ---
C
      IF (IPLTN.NE.0) THEN
        DO 1809 IDN = 1,NNOD
          XSHIF = 0.
          YSHIF = 0.
          FAC = 1.
          IF (NODAR(8,IDN).EQ.0) THEN
            SIZ = 0.
            DSHIF = 0.
            DO 1808 J=1,NODAR(1,IDN)
              IDT = NODAR(4+J,IDN)
              XSHIF = XSHIF + TUBAR(7,IDT)
              YSHIF = YSHIF + TUBAR(8,IDT)
```

```
              DSHIF = DSHIF + SQRT(TUBAR(7,IDT)**2 + TUBAR(8,IDT)**2)
              SIZ = SIZ + TUBAR(2,IDT)
 1808       CONTINUE
            XSHIF = XSHIF /NODAR(1,IDN)
            YSHIF = YSHIF /NODAR(1,IDN)
            SIZ = SIZ /NODAR(1,IDN)
            FAC = 0.8 *MIN(DSHIF,SIZ) /XMAX *XAXIS /0.082
          ENDIF
          FAC = MAX (FAC, .01)
          XB(1) = XNOD(IDN) + XSHIF
          YB(1) = YNOD(IDN) + YSHIF
          IF (IPLTN.NE.1) FAC = 1.
C
          CALL SCLPIC(FAC)
C
          XB(1) = XTU(IDN)
          YB(1) = YTU(IDN)
          FAC = 2. *RTU(IDN) /XMAX *XAXIS /0.082
          FAC = MAX (FAC, 0.01)
          FAC = MIN (FAC, 0.2 *XMAX /XMAX *XAXIS /0.082 )
          CALL SCLPIC(FAC)
          CALL MARKER(15)

C         CALL SCLPIC(0.5)
          CALL CURVE(XB,YB,1,1)
 1809   CONTINUE
        CALL SCLPIC(1.)
      ENDIF
C
C   PLOTT TUBES
C
```

BIOREM$MAIN

```
      IF (IPLTT.NE.0) THEN
      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 1803 IDT=1,NTUB
        XB(1) = TUBAR(3,IDT)
        YB(1) = TUBAR(4,IDT)
        XB(2) = TUBAR(5,IDT)
        YB(2) = TUBAR(6,IDT)
        WID = MIN (TUBAR(2,IDT), 3.*TUBAR(1,IDT))
        IF (TUBAR(2,IDT) .LT. 1.1E-2 *SPAC) WID = 0.
        WID = 0.98 *MAX (WID, 0.)
        WID = WID /XMAX *XAXIS
        IF (IPLTT.EQ.1) CALL THKCRV(WID)
        CALL CURVE(XB,YB,2,0)
 1803 CONTINUE
      CALL RESET('THKCRV')
      ENDIF
C
C---  PLOTT TUBE/NODE  TRIOS  ---
C
      IPLTTRI = 0
      IF (IPLTTRI.NE.0) THEN
      DO 1793 IDN = 1, NNOD
        XB(1) = XNOD(IDN)
        YB(1) = YNOD(IDN)
```

```
       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
C  SHADE BACKGROUND
C
      IF (ISHAD.NE.0) THEN
       IF (XBA1(2).LT.0.9) THEN
        XBA1(1) = 0.
        YBA1(1) = 0.
        XBA1(2) = XMAX
        YBA1(2) = 0.
        XBA2(1) = 0.
        YBA2(1) = YMAX
        XBA2(2) = XMAX
        YBA2(2) = YMAX
       ENDIF
       CALL SHDPAT (17)
       IF (ISHAD .LT. 0) CALL SHDPAT(16)
       CALL SHDCRV(XBA1,YBA1,2,XBA2,YBA2,2)
      ENDIF
C
      CALL ENDPL(0)
      PRINT*,ISEED0
```

BIOREM$MAIN

```
      PRINT913,NPAR
      PRINT913,NNOD
      PRINT913,NTUB
      PRINT900,PORO
      PRINT980,'EPSV',EPSVOID
      PRINT*,' '
C
C  CHECK CONSISTANCY
C
      IPAS = 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
         IPAS = 0
         PRINT*,'NO NODE CORRESPONDANCE'
         PRINT1904,IDN,NUM,L,IDT,XNOD(IDN),YNOD(IDN)
         DO 465 IKL = 1, NODAR(1,NUM)
          N2 = NODAR(IKL+1,NUM)
```

```
 465          PRINT1903,IKL,N2,NODAR(IKL+4,N2),XNOD(N2),YNOD(N2)
          ELSEIF (NOK .EQ. 1) THEN
            IDT2 = NODAR(L2OK+4,NUM)
            IF (IDT .NE. IDT2) THEN
              IPAS = 0
              PRINT*,'NO TUBE CORRESPONDANCE'
              PRINT1910,IDN,NODAR(1,IDN),L,IDT,
     1                  NUM,NODAR(1,NUM),L2OK,IDT2
            ENDIF
          ELSE
            IPAS = 0
            PRINT*,'MULTIPLE NODE CORRESPONDANCE'
          ENDIF
 503    CONTINUE
 502  CONTINUE
      IF (IPAS .EQ. 1) PRINT*,' CORRESPONDANCE IS OK'
      PRINT*,'INPUT ICONT   1 = CONT  2 = RESTART'
      READ*,ICONT
      IF (ICONT.EQ.2) GO TO 1600
C
C---  COMPUTE TUBE DIAGNOSTICS  ---
C
      SUMT = 0.
      NSUM = 0
      DMAX = 0.
      DO 1771 IDT = 1, NTUB
        IDBOUN = MIN (TUBNOD(3,IDT), TUBNOD(4,IDT))
        IF (IDBOUN .GT. 0) THEN
          NSUM = NSUM + 1
```

BIOREM$MAIN

```
          SUMT = SUMT + TUBAR(2,IDT)
          DMAX = MAX (DMAX, TUBAR(2,IDT))
        ENDIF
 1771 CONTINUE
      DTBAR = SUMT /NSUM
C
      SDEVT = 0.
      IFRE = NTUB /20
      DO 1773 IBIN = 1,IFRE
 1773   FREQ(IBIN) = 0.
C
      DTMIN = 0.
      DTMAX = 2.*DTBAR
      DO 1772 IDT = 1,NTUB
        IDBOUN = MIN (TUBNOD(3,IDT), TUBNOD(4,IDT))
        IF (IDBOUN .GT. 0) THEN
          SDEVT = SDEVT + (TUBAR(2,IDT)-DTBAR)**2
          IBIN = NINT((IFRE-1.)*(TUBAR(2,IDT)-DTMIN)/(DTMAX-DTMIN)) + 1
          FREQ(IBIN) = FREQ(IBIN) + 1.
        ENDIF
 1772 CONTINUE
      SDEVT = SQRT(SDEVT /NSUM)
      PRINT981,'NSUM',NSUM
      PRINT981,'NBOU',NTUB-NSUM
      PRINT980,'DBAR',DTBAR
      PRINT980,'SDEV',SDEVT
C
      PRINT900,DMAX
      WRITE(15,901) IFRE
```

```
      DO 1774 IBIN = 1, IFRE
        FSUM = 0.
        DO 1788 II = 1, IBIN
 1788     FSUM = FSUM + FREQ(II)
        FSUM = FSUM /NSUM
        FDIS = 1. *FREQ(IBIN) /NSUM /(1./IFRE)
        WRITE(15,900) (IBIN-0.5)/IFRE, FDIS, FSUM
 1774   PRINT901,IBIN, FDIS, FSUM
C
C---  COMPUTE BOUNDARY LENGTHS  ---
C
      DO 1776 J=1,NBOX
1776     LBOUN(J)=0.
      DO 1777 I=1,NSEED
        IBOU=-IDBC(I)
        IF (IBOU.GT.0) THEN
          LB=LBC(I)
          LBOUN(IBOU)=LBOUN(IBOU)+LSID(LB,I)
        ENDIF
1777  CONTINUE
C
C---  COMPUTE PERIMETER AND PROJECTED AREAS  ---
C
      DO 1778 I=1,NSEED
        PERI(I)=0.
        APRX(I)=0.
        APRY(I)=0.
        DO 1778 L=1,NSID(I)
```

BIOREM$MAIN

```
          PERI(I)=PERI(I)+LSID(L,I)
          IDP=IPRM(L,I)
          IF (IDP.LT.0) THEN
            THET=-PI/2.+(-IDP-1)*2.*PI/NBOX
            DXDL=COS(THET)
            DYDL=SIN(THET)
          ELSE
            DXXX=XS(IDP)-XS(I)
            DYYY=YS(IDP)-YS(I)
            DLLL=SQRT(DXXX**2+DYYY**2)
            DXDL=DXXX/DLLL
            DYDL=DYYY/DLLL
          ENDIF
          APRX(I)=APRX(I)+LSID(L,I)*ABS(DXDL)/2.
          APRY(I)=APRY(I)+LSID(L,I)*ABS(DYDL)/2.
1778  CONTINUE
C
      DO 1785 I=1,NSEED
1785        IF (IDBC(I).LT.-NBOX) PRINT902,I,IDBC(I),XS(I),YS(I)
C
      CKSUM=0.
      PRINT*,NSEED
      DO 1631 K=1,20
        CKSUM=CKSUM+(1.*K)*(1.*FSID(K))/NSUM
1631    PRINT902,K,FSID(K),(1.*FSID(K))/NSUM
      PRINT902,JLMIN,JLMAX
      GAVE=GAVE/NSUM*SQRT(1.*NSEED)
      GMIN=GMIN/NSUM*SQRT(1.*NSEED)
      GMAX=GMAX/NSUM*SQRT(1.*NSEED)
      CMIN=CMIN*SQRT(1.*NSEED)
```

```
      CMAX=CMAX*SQRT(1.*NSEED)
C
      PRINT900,CMIN
      PRINT900,GMIN
      PRINT900,GAVE
      PRINT900,GMAX
      PRINT900,CMAX
      READ*
2000  CONTINUE
C
C---  PRINT NODE/TUBE CORRESPONDANCE STUFF  ---
C
      DO 1733 IDN = 1, 0 *NNOD
        ICH = NODAR(1,IDN)
        DO 1733 J = 1, ICH
          IDT = NODAR(4+J,IDN)
          PRINT904,IDN,NODAR(1+J,IDN),TUBNOD(1,IDT),TUBNOD(2,IDT)
 1733 CONTINUE
C
C---  BEGIN CALCULATIONS OF PRESSURE FIELD  ---
C
1620  PIN = 1000.
      PEX = 1.
      TMP = 298.
      RGAS = 287.
      PRINT*,'INPUT PIN,PEX',PIN,PEX
      READ*,PIN,PEX
```

BIOREM$MAIN

```
      NSOR = 1000
      WGT = 0.5
      PRINT*,'INPUT NSOR, WGT',NSOR,WGT
      READ*,NSOR,WGT
      ABSERR = 1.E-20
      RELERR = 1.E-8
      PRINT*,'INPUT ABSERR,RELERR',ABSERR,RELERR
      READ*,ABSERR,RELERR
      IPLTPR = 0
      IPLTPF = 0
      IPLTPFY = 0
      IF (IDBIN .NE. IDBEV) THEN
        IPLTPF = 0
        IPLTPFY = 0
      ENDIF
      PRINT*,'INPUT IPLTPR,IPLTPF,IPLTFY',IPLTPR,IPLTPF,IPLTPFY
      READ*,IPLTPR,IPLTPF,IPLTPFY
      ICOMP = 0
      PRINT*,'INPUT ICOMP:  0 = INCOMP  1 = COMP',ICOMP
      READ*,ICOMP
      KPAR = 300
      DTSTR = 0.5
      PRINT*,'INPUT NUMBER OF TRACER PARTICLES AND TIME',KPAR,DTSTR
      READ*,KPAR,DTSTR
      PECL = 1.E10
      ISED = 12344321
      IF (KPAR .GT. 0) THEN
        PRINT*,'INPUT PECLET NUMBER',PECL
        READ*,PECL
        IMICF = 0
        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            IUNI
        READ*,IUNI
        IRANS = 0
        PRINT*,'RANDOMIZE STREAMLINE AT EVERY STEP ?,  0 = NO  1 = YES',
     1            IRANS
        READ*,IRANS
        IZZZ = 0
        PRINT*,'EXIT TUBE ON ENTRANCE STREAMLINE ?  0 = NO  1 = YES',
     1            IZZZ
        READ*,IZZZ
        IPLTTR = 0
        IF (IMICF .NE. 1) THEN
          PRINT*,'PARTICLE TRACES ?  0 = NO  1 = YES',IPLTTR
          READ*,IPLTTR
        ENDIF
        PRINT*,'INPUT SEED FOR TRACER INJECTION',ISED
        READ*,ISED
```

BIOREM$MAIN

```
      ENDIF
      IPRINI = 1
      IF (IMETH .LT. 0) IPRINI = 0
      IF (ICONT .NE. 0) IPRINI = 0
      PRINT*,'DO YOU WANT PRESSURES (RE)INITIALIZED',IPRINI
      READ*,IPRINI
      XXX0 = 0.5
      EDGE = 0.
      PRINT*,'INPUT XXX0, EDGE',XXX0,EDGE
      READ*,XXX0,EDGE
      KMETH = 1
      PRINT*,'AVERAGES OR LOCAL APERTURE:  1 = AVE  2 = LOC ?',KMETH
      READ*,KMETH
      IGEO = 2
      PRINT*,'INPUT IGEO',IGEO
      READ*,IGEO
      IOK = 0
      PRINT*,'INPUT OK',IOK
      READ*,IOK
      IF (IOK.EQ.0) GO TO 1620
C
C---  INITIALIZE PRESSURES  ---
C
      IF (IPRINI .NE. 0) THEN
        DO 440 IDN = 1,NNOD
          THET = XNOD(IDN) /XMAX
          IF (IDBIN .NE. IDBEV) THET = YNOD(IDN) /YMAX
          IF ((IDBIN.EQ.2) .OR. (IDBIN.EQ.3)) THET = 1. - THET
          PRES(IDN) = PIN
          IF (ICOMP.EQ.0) THEN
            PRES(IDN) = PIN - (PIN-PEX)*THET
            IF (IPRINI .LT. 0.) PRES(IDN) = PEX
```

```
          ELSE
            PRES(IDN) = SQRT (PIN**2 - (PIN**2-PEX**2)*THET)
            IF (IPRINI .LT. 0.) PRES(IDN) = PEX
          ENDIF
          PRESO(IDN) = PRES(IDN)
 440    CONTINUE
      ENDIF
C
      TIM = 0.
      DTIM = 0.1
      ISTOP = 0
      KMAX = 1
      TOUT = 0
C
C---  UPDATE CURRENT VALUES OF PRESSURE ---
C
        DO 710 K = 1, KMAX
        IF (ISTOP.EQ.1) GO TO 830
        TIM = TIM + DTIM
        IF (K.GT.1) THEN
          DO 321 IDN = 1, NNOD
            PRINT*,IDN,PRESO(IDN)
 321        PRES(IDN) = PRESO(IDN)
        ENDIF
C
```

BIOREM$MAIN

```
 1630   IGO = 0
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
 312    CONTINUE
        SV000 = SUMA /SUMV
C
C---  COMPUTE EFFECTIVE PORE RADII  ---
C
        PORSUM = 0.
        DO 317 IDN = 1, NNOD
          APORE(IDN) = 0.
          DO 316 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.
              XMID = (XS(ID) + XS(IDP)) /2.
              YMID = (YS(ID) + YS(IDP)) /2.
              HIGHT2 = (XNOD(IDN)-XMID)**2 + (YNOD(IDN)-YMID)**2
```

```
              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
                ATRI = (RADI(IDP)-RLAP) *HLAP /2.
                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
                RLEG = ABS (YS(ID) - YNOD(IDN))
              ELSE
```

```
                RLEG = ABS (XS(ID) - XNOD(IDN))
              ENDIF
              HIGHT = SQRT (HYPOT2 - RLEG**2)
              DTHETA = PI /2. - ACOS (RLEG /HYPOT)
              IF (RADI(ID) .LE. HIGHT) THEN
                APART = DTHETA /2. *RADI(ID)**2
              ELSE
                RLAP = RADI(ID) - HIGHT
                PHI = ACOS ((RADI(ID)-RLAP) /RADI(ID))
                HLAP = RADI(ID) *SIN (PHI)
                ATRI = (RADI(ID)-RLAP) *HLAP /2.
                APART = (DTHET-PHI) /2. *RADI(ID)**2 + ATRI
              ENDIF
              DAREA = MAX (0., HIGHT *RLEG /2. - APART)
            ENDIF
            APORE(IDN) = APORE(IDN) + DAREA
 316      CONTINUE
          PORSUM = PORSUM + APORE(IDN)
 317    CONTINUE
        EPSVOID = PORSUM /(XMAX*YMAX)
C
C---  COMPUTE 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, IDT)
            IDP = TUBNOD (4, IDT)
            IF (ID.GT.0 .AND. IDP.GT.0 ) THEN
              XAA(J,1) = -2. *(XS(ID) - XS(IDP))
```

```
          XAA(J,2) = -2. *(YS(ID) - YS(IDP))
          XAA(J,3) = -2. *(RADI(ID) - RADI(IDP))
          RHS(J) = RADI(ID)**2 - RADI(IDP)**2
     1       - XS(ID)**2 + XS(IDP)**2 - YS(ID)**2 + YS(IDP)**2
          QHS(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.
            IF (IDP .EQ. -2) RHS(J) = XMAX
          ELSE
            XAA(J,2) = 1.
```

```
            RHS(J) = 0.
            IF (IDP .EQ. -3) RHS(J) = YMAX
          ENDIF
        ENDIF
  319 CONTINUE
C
      WT = 1.
      WTR = 0.3
      RTU(IDN) = 1.1 *RMIN
      RNEW = RTU(IDN)
      ITSK = 0
      DO 377 ITS = 1, 50
        ITSK = ITSK + 1
C
        RTU(IDN) = (1.-WTR) *RTU(IDN) + WTR *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)
        DETX = XHS(1)   *XAA(2,2) - XAA(1,2) *XHS(2)
        DETY = XAA(1,1) *XHS(2)   - XHS(1)   *XAA(2,1)
        XNEW = DETX /DETA
        YNEW = DETY /DETA
        XTU(IDN) = (1.-WT) *XTU(IDN) + WT *XNEW
        YTU(IDN) = (1.-WT) *YTU(IDN) + WT *YNEW
C
        DELM = 0.
        ERR = 0.
        DO 376 J = 1, NODAR(1,IDN)
          ID = NODAR(J+8,IDN)
          IF (ID .LT. 1) GO TO 376
          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
```

```
                DELM = ABS (RXX-RTU(IDN))
                RNEW = RXX
              ENDIF
 376        CONTINUE
            IF (ERR .LT. 1.E-6*XMAX) GO TO 378
C
 377      CONTINUE
 378      CONTINUE
          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
            IDN = 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
              BOT = AEFF /TUL *(TUL1 /AR1 + TUL2 *RK1 /RK2 /AR2)
              RKEFF = RK1 /BOT
            ELSE
              BOT = AEFF /TUL *(TUL1 *RK2 /RK1 /AR1 + TUL2 /AR2)
              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
  344       CONTINUE
            BBAR = BBAR /MSUM
            ELBAR = ELBAR /MSUM
          ELSE
C
C---  COMPUTE CHARACTERISTIC 2-D PASSAGE DIMENSIONS  ---
C
            IPIK = 1
            BBAR = 0.
            MSUM = 0
            ELBAR = 0.
            DO 371 IDT = 1, NTUB
              ID = TUBNOD(3,IDT)
              IDP = TUBNOD(4,IDT)
              IF (ID.GT.0 .AND. IDP.GT.0) THEN
                R1 = RADI(ID)
                R2 = RADI(IDP)
              ELSE
                ID = MAX (ID, IDP)
                R1 = RADI(ID)
```

```
                R2 = R1
              ENDIF
              DNEK = TUBAR(2,IDT)
              RLEN = MIN (MIN (TUBAR(1,IDT), 1.999*R1), 1.999*R2)
              CALL DELTAS (IPIK, R1, R2, DNEK, RLEN, DDIF(IDT),
     1                    DPER(IDT), DVOL(IDT))
              TUBAR(14,IDT) = DVOL(IDT)
              TUBAR(15,IDT) = DDIF(IDT)
              IF (TUBAR(9,IDT) .GT. 0) THEN
                MSUM = MSUM + 1
                BBAR = BBAR + DVOL(IDT)
                ELBAR = ELBAR + TUBAR(1,IDT)
              ENDIF
  371       CONTINUE
            BBAR = BBAR /MSUM
            ELBAR = ELBAR /MSUM
          ENDIF
C
C---  CYCLE OVER RELAXATION STEPS  ---
C
          ICONV = 1
          DO 210 ICY = 1,NSOR
            IGO = IGO + 1
            ERR = 0.
            ERRMDT = 0.
C
C---  COMPUTE MASS FLOW RATES AND WEIGHTS FOR EACH NODE.
C     SIGN CONVENTION FOR FLOW IS POSITIVE FOR POSITIVE FLOW FROM
C     NODE N1 TO NODE N2;  PRESSURE GRADIENT IS (P2-P1)/L.
C
            DO 220 IDN = 1,NNOD
              VOL = 0.
              DOTM = 0.
              WGTSUM = 0.
              DPDRHO = 1.
              IF (ICOMP.EQ.1) DPDRHO = RGAS *TMP
              DO 230 L = 1, NODAR(1,IDN)
```

```
            NUM = NODAR(L+1,IDN)
            IDT = NODAR(L+4,IDN)
            N1 = TUBNOD(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 (ICOMP.EQ.1) RHO = PBAR /RGAS /TMP
            RMU = 1.
            DPDL = (PRES(IDN)-PRES(NUM)) /TUL
            VEL = - ALF /RMU *DPDL
            DOTM = DOTM + VEL *RHO *DVOL(IDT)
            WGTSUM = WGTSUM + ALF/RMU/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
```

```
            TUBAR(13,IDT) = ISGN *VEL *DVOL(IDT)
 230      CONTINUE
C
C--- COMPUTE RESIDUAL OF GOVERNING EQUATION FOR NODE IDN  ---
C
          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 *DPDRHO
            WEIGHT = WGT *VOL /DPDRHO /WGTSUM
            DPRES(IDN) = - WEIGHT *RESP
          ENDIF
C
C--- UPDATE PRESSURES AND NODE IDN  ---
C
          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
C--- COMPUTE LOCAL ERROR ESTIMATE AT NODE IDN  ---
C
          TOL = RELERR *ABS(PIN-PEX) *WGT
          ERRIDN = ABS(DPRES(IDN)) /TOL
          IF (ERRIDN .GT. ERR) THEN
            ERR = ERRIDN
            IDNERR = IDN
          ENDIF
C
C--- NOTE MAXIMUM NET FLOW RATE AT INTERIOR POINTS
C
          IF ((NODAR(8,IDN).NE.-IDBIN) .AND.
     1        (NODAR(8,IDN).NE.-IDBOU)) THEN
            IF (ABS(DOTM) .GT. ERRMDT) THEN
              ERRMDT = ABS(DOTM)
              IDNMDT = IDN
            ENDIF
          ENDIF
```

```
C
  220      CONTINUE
           IF (ERR.LT.1.0) GO TO 280
C
C---  PRINT CURRENT MAXIMUM ERROR, NODE IDENTIFIERS AND PRESSURES  ---
C
           IF (MOD(IGO,100).EQ.0) THEN
             PRINT903,IGO,IDNERR,NODAR(8,IDNERR),ERRMDT,
     1           ERR,PRES(IDNERR),XNOD(IDNERR),YNOD(IDNERR)
           ENDIF
C
  210    CONTINUE
         ICONV = 0
  280    CONTINUE
C
C---  SUM FLOW RATES AND POROSITY
C
         DOTMIN = 0.
         DOTMOUT = 0.
```

```
         SUMDOT = 0.
         DO 892 IDT = 1, NTUB
C
           N1 = TUBNOD(1,IDT)
           N2 = TUBNOD(2,IDT)
           IF (NODAR(8,N1) .EQ. NODAR(8,N2)) GO TO 892
C
C---  GLOBAL MASS BALANCE FOR ALL INTERIOR NODES
C     TUBE MUST HAVE ONE BOUNDARY ( < 0 ) AND ONE INTERIOR ( = 0 ) NODE
C
           IF (NODAR(8,N1)*NODAR(8,N2) .EQ. 0) THEN
             NN = MIN (NODAR(8,N1), NODAR(8,N2))
             NEVEN = (NN /2) *2
             ISNN = 1
             IF ((XNOD(N1).GT.XNOD(N2)) .AND. (NN.EQ.NEVEN)) ISNN = - 1
             IF ((YNOD(N1).GT.YNOD(N2)) .AND. (NN.NE.NEVEN)) ISNN = - 1
             SUMDOT = SUMDOT + ISNN *NORM(-NN) *TUBAR(11,IDT)
           ENDIF
C
C---  SUM MASS FLOW RATES INTO AND OUT OF X = 0 AND X = L
C
           ISNX = 1
           IF (IDBIN .EQ. IDBEV) THEN
             IF (XNOD(N1) .GT. XNOD(N2)) ISNX = - 1
           ELSE
             IF (YNOD(N1) .GT. YNOD(N2)) ISNX = - 1
           ENDIF
C
           IF     (NODAR(8,N1).EQ.-IDBOU .OR. NODAR(8,N2).EQ.-IDBOU) THEN
             DOTMOUT = DOTMOUT + ISNX *TUBAR(11,IDT)
           ELSEIF (NODAR(8,N1).EQ.-IDBIN .OR. NODAR(8,N2).EQ.-IDBIN) THEN
             DOTMIN = DOTMIN + ISNX *TUBAR(11,IDT)
           ENDIF
C
  892    CONTINUE
C
         PBAR = (PIN + PEX) /2.
         RHO = 1.
         IF (ICOMP.EQ.1) RHO = PBAR /RGAS /TMP
```

```
      RMU = 1.
      UBAR = (ABS(DOTMIN)+ABS(DOTMOUT))/2. /YMAX /RHO
      IF (IGEO .EQ. 3) UBAR = UBAR /ELBAR
      VBAR = UBAR /PORO
C---  OLD UREF WAS UBAR  ---
      UREF = VBAR
      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
      RKOZE = PERM0 /PERM
      TAU0 = DIAMF**2 *PI /2. /(1.-PORO)
      EDSK = PERM /TAU0
      PRINT900,UBAR,VBAR
      PRINT900,(TUBAR(10,IT),IT=1,10)
      PRINT900,(TUBAR(10,IT+NTUB/2),IT=1,10)
C
```

BIOREM$MAIN

```
      SUMV = 0.
      SUMW = 0.
      DO 819 IDT = 1, NTUB
        VOLT = TUBAR(1,IDT) *DVOL(IDT)
        SUMV = SUMV + VOLT
        SUMW = SUMW + ABS(TUBAR(10,IDT)) *VOLT
 819  CONTINUE
      VAVE = SUMW /SUMV
      RHOS = 1. - SUMV /XMAX /YMAX
      PRINT900, VAVE, RHOS
C
      PRINT900,SUMDOT,DOTMIN,DOTMOUT,PERM,RKOZE
C
C---  SOLICIT INPUT FOR CONTINUATION  ---
C
      PRINT*,'INPUT ICONT 1 = MORE CYCL  2 = NEW PRESS
     1  3 = NEW TUBES  4 = RESTART'
      READ*,ICONT
      IF (ICONT.EQ.1) THEN
        PRINT*,'INPUT NSOR, WGT',NSOR,WGT
        READ*,NSOR,WGT
        GO TO 1630
      ELSEIF (ICONT.EQ.2) THEN
        GO TO 1620
      ELSEIF (ICONT.EQ.4) THEN
        GO TO 1600
      ENDIF
C
C     ADJUST TIME STEP
C
      DO 832 IDT = 1, 0 *NTUB
        N1 = TUBNOD(1,IDT)
        N2 = TUBNOD(2,IDT)
        TUL = TUBAR(1,IDT)
        TUD = TUBAR(2,IDT)
        DOTM = TUBAR(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),DOTM,TUBAR(3,IDT),
     1    TUBAR(4,IDT),TUBAR(5,IDT),TUBAR(6,IDT),PRES(N1),PRES(N2)
```

```
          ENDIF
C
 832    CONTINUE
        DO 833 IDT = 1, 0 *NTUB
          N1 = TUBNOD(1,IDT)
          N2 = TUBNOD(2,IDT)
          TUL = TUBAR(1,IDT)
          TUD = TUBAR(2,IDT)
          DOTM = 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),DOTM,TUBAR(3,IDT),
     1     TUBAR(4,IDT),TUBAR(5,IDT),TUBAR(6,IDT),PRES(N1),PRES(N2)
          ENDIF
 833    CONTINUE
C
C     PLOTT PRESSURE POINTS

BIOREM$MAIN

C
        IF (IPLTPR.EQ.1) THEN
          CALL AREA2D(XAXIS,YAXIS)
          CALL GRAF(XORIG,XSTP,XMAX,YORIG,YSTP,YMAX)
          CALL THKFRM(.030)
          CALL FRAME
          PRINT*,ISEED0
          PRINT913,NPAR
          PRINT913,NNOD
          PRINT913,NTUB
          PRINT913,NSOR
          PRINT900,PIN
          PRINT900,PEX
          PRINT900,PORO
          PRINT900,ABSERR
          PRINT900,RELERR
          PRINT900,WGT
          PRINT*,' '
          IF (ICONV.EQ.0) PRINT*,'NO SOR CONV'
          PRINT*,IGO
          PRINT900,ERR
          PRINT900,DOTMIN
          PRINT900,DOTMOUT
          PRINT900,DOTMUP
          PRINT900,DOTMDN
          CALL MARKER(15)
          DO 827 IDN = 1,NNOD
             XB(1)=XNOD(IDN)
             YB(1)=YNOD(IDN)
             FAC = (PRES(IDN)-PEX) /(PIN-PEX)
             FAC = FAC/SQRT(1.*NNOD) *XAXIS/0.082
             CALL SCLPIC(FAC)
             CALL CURVE(XB,YB,1,1)
 827      CONTINUE
          CALL SCLPIC(1.)
          CALL ENDPL(0)
        ENDIF
C
C     PLOTT X-P PRESSURE PROFILE
C
        XORIG = 0.
```

```
      YORIG = 0.
      YMAX2 = 1.
      CALL SCLPIC(1.)
      IF (IPLTPF.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
       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,NPAR
       PRINT913,NNOD
       PRINT913,NTUB
       PRINT913,NSOR
       PRINT900,PIN
       PRINT900,PEX
       PRINT900,PORO
       PRINT900,ABSERR
       PRINT900,RELERR
       PRINT900,WGT
       PRINT*,' '
       IF (ICONV.EQ.0) PRINT*,'NO SOR CONV'
       PRINT901,IGO
       PRINT900,ERR
       PRINT900,DOTMIN
       PRINT900,DOTMOUT
       PRINT900,SUMDOT
       DO 826 IDN = 1,NNOD
        XB(1) = XNOD(IDN)
        YB(1) = (PRES(IDN)-PEX) /(PIN-PEX)
        CALL CURVE(XB,YB,1,1)
826    CONTINUE
       IF (IDBIN .EQ. IDBEV) THEN
        JMX = 200
        DO 825 J = 1, JMX
         XB(J) = XMAX *(J-1.)/(JMX-1.)
         IF (IDBIN .EQ. 2) XB(J) = 1. - XB(J)
         IF (ICOMP.EQ.0) THEN
          YB(J) = 1. - XB(J)/XMAX
         ELSE
          YB(J) = SQRT (PIN**2 - (PIN**2-PEX**2)*XB(J)/XMAX)
          YB(J) = (YB(J)-PEX) /(PIN-PEX)
         ENDIF
 825    CONTINUE
        CALL CURVE(XB,YB,JMX,0)
       ENDIF
C
C---  PLOT TUBE CONNECTIONS IN X-P SPACE  ---
C
       CALL DASH
```

```
          DO 828 IDT = 1, NTUB
            IF (KBLOC(IDT) .EQ. 1) THEN
              DO 829 J = 1, 2
                IDN = TUBNOD(J,IDT)
                XB(J) = XNOD(IDN)
 829            YB(J) = (PRES(IDN)-PEX) /(PIN-PEX)
              CALL CURVE (XB,YB,2,0)
            ENDIF
 828      CONTINUE
          CALL RESET('DASH')
          CALL ENDPL(0)
        ENDIF
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
          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,NPAR
          PRINT913,NNOD
          PRINT913,NTUB
          PRINT913,NSOR
          PRINT900,PIN
          PRINT900,PEX
          PRINT900,PORO
          PRINT900,ABSERR
          PRINT900,RELERR
          PRINT900,WGT
          PRINT*,' '
          IF (ICONV.EQ.0) PRINT*,'NO SOR CONV'
          PRINT901,IGO
          PRINT900,ERR
          PRINT900,DOTMIN
          PRINT900,DOTMOUT
          PRINT900,SUMDOT
          DO 856 IDN = 1,NNOD
            XB(1) = YNOD(IDN)
            YB(1) = (PRES(IDN)-PEX) /(PIN-PEX)
            CALL CURVE(XB,YB,1,1)
 856      CONTINUE
          IF (IDBIN .NE. IDBEV) THEN
            JMX = 200
            DO 855 J = 1, JMX
```

```
              XB(J) = YMAX *(J-1.)/(JMX-1.)
              IF (IDBIN .EQ. 3) XB(J) = 1. - XB(J)/YMAX
              IF (ICOMP.EQ.0) THEN
                YB(J) = 1. - XB(J)
              ELSE
                YB(J) = SQRT (PIN**2 - (PIN**2-PEX**2)*XB(J)/YMAX)
                YB(J) = (YB(J)-PEX) /(PIN-PEX)
              ENDIF
 855        CONTINUE
            CALL CURVE(XB,YB,JMX,0)
          ENDIF
C
C---  PLOT TUBE CONNECTIONS IN Y-P SPACE  ---
C
```

BIOREM$MAIN

```
          CALL DASH
          DO 858 IDT = 1, NTUB
            IF (KBLOC(IDT) .EQ. 1) THEN
              DO 859 J = 1, 2
                IDN = TUBNOD(J,IDT)
                XB(J) = YNOD(IDN)
 859            YB(J) = (PRES(IDN)-PEX) /(PIN-PEX)
              CALL CURVE (XB,YB,2,0)
            ENDIF
 858      CONTINUE
          CALL RESET('DASH')
          CALL ENDPL(0)
        ENDIF
C
C     UPDATE VALUES
C
        DO 740 ID=1,IDMX
          PRESO(IDN) = PRES(IDN)
 740    CONTINUE
C
C--- WRITE RESTART FILE  ---
C
      IFILE = 99
      PRINT*,'INPUT FILE NUMBER FOR WRITING RESTART FILE',IFILE
      READ*,IFILE
      IF (IFILE .GT. 0) THEN
        REWIND (IFILE)
        WRITE(IFILE,922) NSEED,ISEED0,XMAX,YMAX,PORO,SPAC,SPACM,TOL,
     1                  EPS0,DIAMF,PIN,PEX
        DO 414 I = 1, NSEED
 414      WRITE(IFILE,*) XS(I),YS(I),HITE(I),DIAM(I),RADI(I)
        WRITE(IFILE,901) NNOD
        DO 415 I = 1, NNOD
 415      WRITE(IFILE,*) PRES(I)
      ENDIF
C
C---  TRACER PARTICLE MOTION  ---
C
      IF (KPAR .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
        XSTP=(XMAX-XORIG)/1.
        YSTP=(YMAX-YORIG)/1.
        CALL AREA2D(XAXIS,YAXIS)
        CALL GRAF(XORIG,XSTP,XMAX,YORIG,YSTP,YMAX)
        CALL FRAME
C
        DCOF = ABS (UREF *DIAMF/1.000 /PECL)
        PECS = PECL *SDEVT /DTBAR
```

BIOREM$MAIN

```
        DTIM1 = ABS (XMAX /VBAR *DTSTR)
        DTIM2 = (XMAX *DTSTR)**2 /DCOF /2.
        DTIM = 1. /(1./DTIM1 + 1./DTIM2)
        DTIM = DTIM1
C
        PRINT981,'METH',IMETH
        PRINT980,'DIAM',DIAMF
        PRINT980,'DREF',DREF
        PRINT980,'UBAR',UBAR
        PRINT980,'UREF',UREF
        PRINT980,'VBAR',VBAR
        PRINT980,'VAVE',VAVE
        PRINT980,'PERM',PERM
        PRINT980,'SV00',SV000
        PRINT980,'RKOZ',RKOZE
        PRINT980,'EDSK',EDSK
        PRINT980,'EPS0',EPS0
        PRINT980,'PORO',PORO
        PRINT980,'EPSV',EPSVOID
        PRINT980,'EPSM',EPSMC
        PRINT980,'RHOS',RHOS
        PRINT980,'DTIM',DTIM
        PRINT980,'DTST',DTSTR
        PRINT980,'DCOF',DCOF
        PRINT980,'PECL',PECL
        PRINT980,'PECS',PECS
        PRINT980,'TBAR',DTBAR
        PRINT980,'BBAR',BBAR
        PRINT980,'TSIG',SDEVT
        TSTAR = DCOF *DTIM /DTBAR**2
        PRINT980,'TSTR',TSTAR
        IF (TSTAR.LT.1.0) PRINT980,'TSTR TOO SMALL'
C
        CALL MARKER(16)
        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 1381 ID = 1, NSEED
          XB(1)=XS(ID)
          YB(1)=YS(ID)
          FAC= 2.* RADI(ID)/XMAX *XAXIS/0.082
          CALL SCLPIC(FAC)
          CALL CURVE(XB,YB,1,1)
 1381   CONTINUE
```

```
        CALL SCLPIC(1.)
        CALL RESET('BLSYM')
        CALL SCLPIC(1.0)
C
        DO 1369 IFR = 1, 100
 1369     FREQ(IFR) = 0.
C
        DO 1707 IDT = 1, NTUB
 1707     JBLOK(IDT) = 0
C
        ASUM = 0.
        DO 1374 I = 1, NNBC(IDBIN)
```

```
          IDN = NBID(IDBIN,I)
          IDT = TBID(IDBIN,I)
          ASUM = ASUM + KBLOC(IDT) *TUBAR(15,IDT)
 1374   CONTINUE
C
        VTBAR = 0.
        XBAR = 0.
        KPSUM = 0
        TORSUM = 0.
        ARTBAR = 0.
        DO 1310 J = 1, KPAR
C
C---  GENERATE NEW PARTICLE AND PLACE AT ENTRANCE NODE; PARTICLES ARE
C     RANDOMLY DISTRIBUTED BETWEEN ENTRANCE NODES IN PROPORTION TO
C     INLET TUBE FLOW RATES.
C
 1315     CONTINUE
          DPART = - 100 *SPACM /10.
          XXX = RAN(ISED)
          XTRM = (4.*XXX-2.)/3.
          ZZZ = XTRM
          DO 1382 I = 1, 100
            ZZZ = ZZZ**3 /3. + XTRM
            RESID = ZZZ**3 - 3.*ZZZ + 3.*XTRM
            IF (ABS(RESID) .LT. 1.E-4) GO TO 1383
 1382     CONTINUE
          WRITE(17,*) 'ZZZ DID NOT CONVERGE',XXX,ZZZ,RESID
 1383     CONTINUE
C
C--- OPTION FOR UNIFORM INJECTION  ---
C
          IF (IUNI .EQ. 1) THEN
            ZZZ = (2.*XXX - 1.)
          ENDIF
C
          GGG = (ZZZ + 1.) /2.
C
C---  SUM ENTRANCE FLOWS TO PICK INLET TUBE  ---
C
          XXX = RAN(ISED)
          XXX = EDGE + (1.-2.*EDGE) *XXX
          IF (XXX0 .LT. 0.) THEN
            XXX = -XXX0
          ENDIF
          FLSUM = 0.
          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
 1364     CONTINUE
          WRITE(17,*) 'CANT FIND ENTRANCE NODE'
```

BIOREM$MAIN

```
C
 1365     TREM = DTIM
          IDON = 0
          TORT = 0.
          XYNPAR(4,J) = YNOD(INOD)
          CALL THKCRV(.020)
C
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
            IDTIN = IDT
            DIFSUM = 0.
            FLOSUM = 0.
            NCH = NODAR(1,INOD)
            NFIN = 0
            NFOUT = 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
 1330       CONTINUE
            FLOSUM = MAX (FLOSUM, 1.E-30)
            DIFSUM = MAX (DIFSUM, 1.E-30)
C
C---  IDENDITFY TUBE/NODE SEQUENCE NUMBER OF INFLOW TUBE  ---
C
            QFLOIN = MIN (ABS (TUBAR(13,IDTIN)), FLOSUM) /FLOSUM
            DO 1398 JNOD = 1, NCH
              IF (NODAR(4+JNOD,INOD) .EQ. IDTIN) THEN
```

```
                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.
```

BIOREM$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
              IDTJ = NODAR(4+JNOD,INOD)
              DFLOW = FLO(JNOD) /FLOSUM
              FLSUM = FLSUM + DFLOW
              IF (FLSUM .GE. GGG) THEN
                IDT = IDTJ
                JCOUT = JCNT
                QFLOOUT = MIN (ABS (TUBAR(13,IDT)), FLOSUM) /FLOSUM
                FLSUM = FLSUM - DFLOW
                GO TO 1336
              ENDIF
 1335       CONTINUE
            WRITE(17,*) 'CANT FIND NEW TUBE'
            WRITE(17,*) XXX, FLSUM
            IDON = 1
            GO TO 1321
 1336       CONTINUE
C
C---  COMPUTE 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) /QFLOOUT
              ELSEIF (NFIN.EQ.2 .AND. NFOUT.EQ.1) THEN
                IF (JCOUT .EQ. 1) THEN
                  GGG = QFLOIN *GGG
                ELSEIF (JCOUT .EQ. 2) THEN
                  QFLOIN2 = 1. - QFLOIN
                  GGG = QFLOIN *GGG + QFLOIN2
                ELSE
                  PRINT*,'JCOUT IS NOT 1 OR 2',JCOUT
                  WRITE(17,*) 'JCOUT IS NOT 1 OR 2',JCOUT
                ENDIF
              ELSE
                PRINT*,'NUMBER OF TUBES IS WRONG'
                WRITE(17,*) 'NUMBER OF TUBES IS WRONG'
              ENDIF
            ENDIF
C
```

```
      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,
     1            QFLOOUT,FLSUM,JCOUT
      ENDIF
      GGG = MIN (0.999, MAX (0.001, GGG))
      ZZZ = 2. *GGG -1.
C
 1363 CONTINUE
```

BIOREM$MAIN

```
      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)
C
C---  START MOVE PARTICLE  ---
C
      ISTIK = 0
      IMOV = 0
      IEXIT = 0
      DYDS = (YNOD(JNOD)-YNOD(INOD)) /TUBAR(1,IDT)
      DXDS = (XNOD(JNOD)-XNOD(INOD)) /TUBAR(1,IDT)
      XB(1) = XNOD(INOD) + 1.000 *TUBAR(7,IDT)
      YB(1) = YNOD(INOD) + 1.000 *TUBAR(8,IDT)
      XB(2) = XB(1)
      YB(2) = YB(1)
C
C---  CHECK FOR TUBE BLOCKAGE  ---
C
      IF (DPART .GT. TUBAR(2,IDT)) THEN
        ISTIK = 1
        IDON = 1
        XB(2) = XNOD(INOD) + DXDS *TUBAR(1,IDT) /2.
        YB(2) = YNOD(INOD) + DYDS *TUBAR(1,IDT) /2.
      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
C
        SSS = 0.
        ZZZIN = ZZZ
C
C---  TAKE INTERMEDIATE STEPS ALONG TUBE LENGTH  ---
C
        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. TREM) THEN
            DELTIM = MAX (TREM, 1.E-30)
```

```
        IDON = 1
      ENDIF
 1343 CONTINUE
C
C---  COMPUTE LOCAL APERTURE  ---
C
      HHH = 0.
      IF (KMETH .EQ. 2) THEN
```

```
        ID = TUBNOD(3,IDT)
        HH1 = 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 = TUBNOD(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 (KMETH .EQ. 2) VLOC = VLOC *TUBAR(14,IDT) /HHH
      DELSADV = VLOC *DELTIM
      DELS = DELSADV
      IF (SSS+DELS .GT. TUBAR(1,IDT)) THEN
        IARIV = 1
        DELS = TUBAR(1,IDT) - SSS
        DELTIM = ABS (DELS /VLOC)
        IF (NODAR(8,JNOD) .EQ. -IDBOU) THEN
          IEXIT = 1
          IDON = 1
        ENDIF
      ENDIF
      BSUM = BSUM + HHH *DELS
      TSUM = TSUM + DELTIM

C
C---  TAKE DIFFUSIVE STEP  ---
C
      IDIR = 1
      IF (RAN(ISED) .LT. 0.5) IDIR = -1
      DELSDIF = IDIR *SQRT (2. *DCOF *DELTIM)
      DELZ = DELSDIF /(TUBAR(14,IDT) /2.)
      IF (KMETH .EQ. 2) DELZ = DELZ *TUBAR(14,IDT) /HHH
      IF (ZZZ+DELZ .GT. 1.0) THEN
        DELZ =  2.*(1.-ZZZ) - DELZ
      ELSEIF (ZZZ+DELZ .LT. -1.0) THEN
```

```
                  DELZ = -2.*(1.+ZZZ) - DELZ
                ENDIF
                IF (PECL .GT. 9.E9) THEN
                  DELZ = 0.
                ENDIF
C
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
 1340         CONTINUE
            ENDIF
 1341       XB(2) = XB(1) + DXDS *SSS
            YB(2) = YB(1) + DYDS *SSS
            IF (KMETH .EQ. 2) THEN
              DELZ = ZZZ *HHH /2.
              XB(2) = XB(2) - DELZ *DYDS
              YB(2) = YB(2) + DELZ *DXDS
            ENDIF
            TORT = TORT + SSS
            XYNPAR(1,J) = XB(2)
            XYNPAR(2,J) = YB(2)
            XYNPAR(3,J) = INOD
            IF (IARIV .EQ. 1) THEN
              INOD = JNOD
            ELSEIF (IARIV .EQ. -1) THEN
              INOD = INOD
            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
            IF (IZZZ .EQ. 1) THEN
              ZZZ = ZZZIN
            ENDIF
C
C---  PLOTT TRACER PARTICLE TRAJECTORY  ---
C
            IF (IPLTTR .EQ. 1) THEN
              CALL CURVE(XB,YB,2,0)
            ENDIF
C
 1320     CONTINUE
C
C---  PLOTT FINAL PARTICLE POSITION  ---
C
 1321     CONTINUE
          IF (IEXIT .EQ. 0) THEN
            CALL MARKER(15)
            CALL CURVE(XB(2),YB(2),1,-1)
          ELSEIF (IEXIT .EQ. 1) THEN
            ARRTIM(J) = DTIM - TREM
```

```
            ARTBAR = ARTBAR + ARRTIM(J)
            CALL MARKER(13)
            XB(2) = 0.99 *XMAX
            CALL CURVE(XB(2),YB(2),1,-1)
          ELSEIF (IEXIT .EQ.-1) THEN
            CALL MARKER(13)
```

BIOREM$MAIN

```
            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 FREQUNCY DISTRIBUTION  ---
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
          VTBAR = VTBAR + VTRACE(KPSUM)
C
 1310   CONTINUE
        CALL RESET('DASH')
        CALL RESET('THKCRV')
        ARTBAR = MAX (1.E-10, ARTBAR /KPAR)
        XBAR = XBAR /KPAR
        VTBAR = VTBAR /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) - VTBAR)**2
 1348   CONTINUE
        SDEV = SQRT (SDEV /KPAR)
        SDTR = SQRT (SDTR /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) *YMAX
```

```
 1391 CONTINUE
      X50 = MAX (1.E-10, X50)
C
      IF (IMICF .NE. 1) THEN
        CALL THKCRV(.050)
```

BIOREM$MAIN

```
        CALL CURVE(XB,YB,100,0)
      ENDIF
C
C---  COMPUTE APPARENT DIFFUSIVITY AND PLOTT CORRESPONDING ERROR FUNC  ---
C
      VPAR = XBAR /DTIM
      PEVP = PECL *VPAR /VBAR
      PEBB = PEVP *BBAR /DIAMF
      DIFF = SDEV**2 /2. /XBAR
      DIF2 = SDEV**2 /2. /DTIM
      DIFT = SDTR**2 /2. /DTIM
      DIF3 = (UREF /PORO *STIM)**2 /2. /ARTBAR
      DIF4 = SIGV**2 *DTIM /2.
      DIFF0 = MAX (1.E-20, SDEVT/DTBAR*DIAMF)
      DCON = DIFF /DIFF0
      CON1 = PORO *DIF2 /DCOF /(2.*PECL)
      CONS = PORO *DIF2 /DCOF /(2.*PECS)
      DO 1367 IFR = 1, 101
        XB(IFR) = (IFR-1.) /100 *XMAX
        ETA = (XBAR-XB(IFR)) / 2. /SQRT(XBAR*DIFF)
        YB(IFR) = 0.5 *(1. + ERF(ETA)) *YMAX
 1367 CONTINUE
      IF (IMICF .NE. 1) THEN
        CALL DASH
        CALL CURVE(XB,YB,100,0)
        CALL RESET ('DASH')
      ENDIF
C
C---  PLOTT USED TUBES FOR MICRO-FINGERS  ---
C
      IF (IMICF .EQ. 1) THEN
        DO 1703 IDT = 1, NTUB
          XB(1) = TUBAR(3,IDT)
          YB(1) = TUBAR(4,IDT)
          XB(2) = TUBAR(5,IDT)
          YB(2) = TUBAR(6,IDT)
          WID = MIN (TUBAR(2,IDT), 3.*TUBAR(1,IDT))
          WID = 0.98 *MAX (WID, 0.)
          WID = WID /XMAX *XAXIS
          WID = MAX (WID, .001)
          CALL THKCRV(WID)
          IF (JBLOK(IDT) .EQ. 1) CALL CURVE(XB,YB,2,0)
 1703   CONTINUE
      ENDIF
C
      CALL RESET('THKCRV')
      CALL SCLPIC(1.0)
      XB(1) = 0.
      YB(1) = 1.
      CALL SCLPIC(0.0001)
      CALL CURVE(XB,YB,1,1)
      CALL ENDGR(0)
      PRINT980,'PECL',PECL
```

```
        PRINT980,'PEVP',PEVP
        PRINT980,'PEBB',PEBB
        PRINT980,'DIFL',DIF2/DCOF
        PRINT980,'DIFT',DIFT/DCOF
```

BIOREM$MAIN

```
        PRINT980,'TORT',TORT
        PRINT980,' '
        PRINT980,'X50 ',X50
        PRINT980,'VPAR',VPAR
        PRINT980,'XBAR',XBAR
        PRINT980,'DCON',DCON
        PRINT980,'CONS',CONS
        PRINT980,'CON1',CON1
        PRINT980,'SDEV',SDEV
        PRINT980,'DIFF',DIFF
        PRINT980,'DIF2',DIF2
        PRINT980,'DIF3',DIF3
        PRINT980,'DRAT',DIFF/DCOF
        PRINT980,'DRA4',DIF4/DCOF
        PRINT980,'SIGV',SIGV
        PRINT980,'SVRA',SIGV/VTBAR
        PRINT980,'    '
        CALL ENDPL(0)
      ENDIF
C
 710  CONTINUE
 830  CONTINUE
C
      PRINT*,'INPUT ICONT   1 = NEW PRESS
     1    2 = NEW TUBES  3 = RESTART'
      READ*,ICONT
      IF (ICONT.EQ.1) THEN
        GO TO 1620
      ELSEIF (ICONT.EQ.2) THEN
        GO TO 1610
      ELSEIF (ICONT.EQ.3) THEN
        GO TO 1600
      ENDIF
C
 840  CALL DONEPL
C
 988  FORMAT('1',A,2X,1PE11.3)
 980  FORMAT(1X,A,2X,1PE11.3)
 981  FORMAT(1X,A,2X,I11,1PE11.3)
 900  FORMAT(11(1PE12.4))
 990  FORMAT(12(1PE11.3))
 901  FORMAT(I10,10(1PE12.4))
 902  FORMAT(2(I10),8(1PE12.4))
 922  FORMAT(2(I10),10(1PE11.3))
 912  FORMAT(2(I5),8(1PE12.4))
 903  FORMAT(3(I10),8(1PE12.4))
 913  FORMAT(3(I6),8(1PE12.4))
 904  FORMAT(4(I10),8(1PE12.4))
 905  FORMAT(5(I10),8(1PE12.4))
1900  FORMAT(10(1PE12.4))
1901  FORMAT(I4,10(1PE12.4))
1902  FORMAT(2(I4),8(1PE12.4))
1912  FORMAT(2(I5),8(1PE12.4))
1903  FORMAT(3(I4),8(1PE12.4))
```

```
1904  FORMAT(4(I4),8(1PE12.4))
1910  FORMAT(20(I4),8(1PE12.4))
      STOP


C
********************************************************************************
C
      SUBROUTINE SPACEF (NBIN,NSEED,XS,YS,XMAX,YMAX,SDEVX,SDEVY)
      REAL XS(1),YS(1),FREQX(100),FREQY(100)
C
      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


C
********************************************************************************
C
      SUBROUTINE NERNEB (NDARR, NBORS, NSEED, XS, YS, X0, Y0,
     1                   DNEIGH, INEIGH)
      REAL DNEIGH(NDARR,1), XS(1), YS(1), X0(1), Y0(1)
      INTEGER INEIGH(NDARR,1)
C
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.  NERBER USES A BUBBLE SORT TO RANK NEIGHBORS IN INCREASING
C     DISTANCE.
C
      DDMX = 1.E20
      DO 1107 J = 1, NBORS
        DO 1107 ID = 1, NSEED
 1107     DNEIGH(J,ID) = DDMX
C
      DO 1108 ID = 1, NSEED
        X0(ID) = XS(ID)
 1108   Y0(ID) = YS(ID)
C
      DO 1101 ID = 1, NSEED
        DO 1102 IDP = 1, NSEED
          IF (IDP .EQ. ID) GO TO 1102
          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
```

```
            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)
                DNEIGH(K+1,ID) = DTMP
                INEIGH(K+1,ID) = ITMP
              ELSE
                GO TO 1102
              ENDIF
 1103       CONTINUE
          ENDIF
 1102   CONTINUE
 1101 CONTINUE
      RETURN
      END
```

```
C
*******************************************************************************
C
      SUBROUTINE ERRINV(X)
      EXTERNAL FUNC
      DATA ERR,ERA /1.E-4,1.E-4/
      COMMON /BLKQ/ RMU,SIG,Z
      XL=RMU-20.*SIG
      XR=RMU+20.*SIG
      X=RMU
      CALL FZERO(FUNC,XL,XR,X,ERR,ERA,IFLAG)
      X=XL
      RETURN
      END
```

```
C
C
      FUNCTION FUNC(X)
      COMMON /BLKQ/ RMU,SIG,Z
      ETA=(X-RMU)/SIG/SQRT(2.)
      FUNC=(1.+ERF(ETA))/2.-Z
      RETURN
      END
```

```
C
C*******************************************************************************
C
      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 FZERO (GUNC, XL, XR, X, ERR, ERA, IFLAG)
      ERRETA = XL
      RETURN
      END
```

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

```
C
C********************************************************************************
C
      SUBROUTINE PLTBOX(XMAX,YMAX,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))*YMAX
 814  CONTINUE
      CALL THKCRV(.030)
      CALL CURVE(XX,YY,NBOX+1,0)
      CALL THKCRV(.010)
      RETURN
      END
```

```
C
C********************************************************************************
C
      SUBROUTINE PLTBNDS(IDMX,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 THKCRV(.010)
      DO 815 ID=1,IDMX
        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
```

```
C
C
      SUBROUTINE DELTAS (IFLAG, R1, R2, DELTA0, L, DELTAD, DELTAP,
     1                   DELTAV)
C
C   IFLAG = 1 => ANALYTICAL SOLUTION FOR R2 = R1 (ZEROTH-ORDER PERTURBATION
C                SOLUTION)
C   IFLAG = 2 => FIRST-ORDER PERTURBATION SOLUTION
C   IFLAG = 3 => EXACT SOLUTION (VIA NUMERICAL INTEGRATION IF NECESSARY)
```

```
C
      IMPLICIT DOUBLE PRECISION (A - H, O - Z)
      DOUBLE PRECISION K, I, L, INT3, INT1, LS4, KS, K1
      DIMENSION SUM(4), ZUM(4)
      DSTAR(TERM1, R3S, T, COST, SINT) = TERM1 - COST -
     1                                   SQRT(R3S - SINT*SINT)
      ARG = 0.5*L/R1
      RATIO2 = DELTA0/R1
      RATIO3 = R2/R1
      R1S = R1*R1
      LS4 = L*L/4.
      ROOT1 = SQRT(R1S - LS4)
      THETA = ASIN(ARG)
      IF (IFLAG .NE. 3) THEN
         K = RATIO2/2. + 1.
         K1 = K + 1.
         KS = K*K
         V1 = KS - 1.
         EPS = RATIO3 - 1.
         BASIC = ATAN(SQRT(K1/(K - 1.))*TAN(THETA/2.))/SQRT(V1)
         BASICK = 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)
         IF (IFLAG .EQ. 1) RETURN
         CD = (0.5*V2 - BASIC)/(K1*G)
         DELTAD = DELTAD*(1. + EPS*CD)
         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
```

```
            GRAND1 = GRAND2/(DELTA*DELTA)
            IFLIP = 6 - IFLIP
            SUM(IFLIP) = SUM(IFLIP) + GRAND1
            ZUM(IFLIP) = ZUM(IFLIP) + GRAND2
   11    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 - LS4)
         DELTAV = DELTA0 + R1 + R2 - 0.5*(ROOT1 + ROOT2) -
     1            (R2S/L)*ASIN(0.5*L/R2) - (R1S/L)*THETA
         DELTAP = SQRT(DECUBE/DELTAV)
      END IF
      RETURN
      END
```

```
C
C
      FUNCTION BOXMUL (ISED, RMU, SDEV, R2)
      DATA TWOPI /6.2831853072/
      XXX1 = RAN (ISED)
      XXX2 = RAN (ISED)
      BOXMUL = SQRT(-2. *ALOG(XXX1)) *COS (TWOPI *XXX2)
      BOXMUL = SDEV *BOXMUL + RMU
      R2 = SQRT(-2. *ALOG(XXX1)) *SIN (TWOPI *XXX2)
      R2 = SDEV *R2 + RMU
      RETURN
      END
```

UNLIMITED RELEASE
INITIAL DISTRIBUTION

1436	C. E. Meyers, 4523
9001	T. O. Hunter, 8000
	Attn: M. E. John, 8100
	A. West, 8200
	R. C. Wayne, 8400
	D. L. Crawford, 8900
9403	J. M. Hruby, 8230
9054	W. J. McLean, 8300
	Attn: W. Bauer, 8302
	L. A. Rahn, 8351
	F. P. Tully, 8353
	G. A. Fisk, 8355
	R. W. Carling, 8362
9042	G. H. Evans, 8345
9042	J. F. Grcar, 8345
9042	S. K. Griffiths, 8345 (5)
9042	C. M. Hartwig, 8345
9042	R. S. Larson, 8345
9042	R. H. Nilson, 8345 (5)
9056	P. H. Paul, 8351
9052	M. D. Allendorf, 8361
9052	D. R. Hardesty, 8361
9053	S. R. Vosen, 8362
9405	T. M. Dyer, 8700
	Attn: M. Perra, 8711
	M. I. Baskes, 8712
	G. J. Thomas, 8715
	K. L. Wilson, 8716
9904	W. D. Wilson, 8700
9403	J. C. F. Wang, 8713
9403	R. W. Bradshaw, 8713 (5)
9161	W. G. Wolfer, 8717
0834	M. J. Martinez, 9112
0826	M. R. Baer, 9112
9141	D. A. Nissen, 11600
9021	Technical Communications for OSTI, 8815 (10)
9021	Technical Communications, 8815
0899	Technical Library, 13414 (4)
9018	Central Technical Files, 8940 (3)