United States Department of Commerce
National Institute of Standards and Technology

NIST Technical Note 1269

Manual for the Cement Hydration Simulation Model

Leslie Struble, Steven Johnson, Mark Hartmann, Lawrence Kaetzel, and Hamlin Jennings

The National Institute of Standards and Technology[1] was established by an act of Congress on March 3, 1901. The Institute's overall goal is to strengthen and advance the Nation's science and technology and facilitate their effective application for public benefit. To this end, the Institute conducts research to assure international competitiveness and leadership of U.S. industry, science and technology. NIST work involves development and transfer of measurements, standards and related science and technology, in support of continually improving U.S. productivity, product quality and reliability, innovation and underlying science and engineering. The Institute's technical work is performed by the National Measurement Laboratory, the National Engineering Laboratory, the National Computer Systems Laboratory, and the Institute for Materials Science and Engineering.

The National Measurement Laboratory

Provides the national system of physical and chemical measurement; coordinates the system with measurement systems of other nations and furnishes essential services leading to accurate and uniform physical and chemical measurement throughout the Nation's scientific community, industry, and commerce; provides advisory and research services to other Government agencies; conducts physical and chemical research; develops, produces, and distributes Standard Reference Materials; provides calibration services; and manages the National Standard Reference Data System. The Laboratory consists of the following centers:

- Basic Standards[2]
- Radiation Research
- Chemical Physics
- Analytical Chemistry

The National Engineering Laboratory

Provides technology and technical services to the public and private sectors to address national needs and to solve national problems; conducts research in engineering and applied science in support of these efforts; builds and maintains competence in the necessary disciplines required to carry out this research and technical service; develops engineering data and measurement capabilities; provides engineering measurement traceability services; develops test methods and proposes engineering standards and code changes; develops and proposes new engineering practices; and develops and improves mechanisms to transfer results of its research to the ultimate user. The Laboratory consists of the following centers:

- Computing and Applied Mathematics
- Electronics and Electrical Engineering[2]
- Manufacturing Engineering
- Building Technology
- Fire Research
- Chemical Engineering[3]

The National Computer Systems Laboratory

Conducts research and provides scientific and technical services to aid Federal agencies in the selection, acquisition, application, and use of computer technology to improve effectiveness and economy in Government operations in accordance with Public Law 89-306 (40 U.S.C. 759), relevant Executive Orders, and other directives; carries out this mission by managing the Federal Information Processing Standards Program, developing Federal ADP standards guidelines, and managing Federal participation in ADP voluntary standardization activities; provides scientific and technological advisory services and assistance to Federal agencies; and provides the technical foundation for computer-related policies of the Federal Government. The Laboratory consists of the following divisions:

- Information Systems Engineering
- Systems and Software Technology
- Computer Security
- Systems and Network Architecture
- Advanced Systems

The Institute for Materials Science and Engineering

Conducts research and provides measurements, data, standards, reference materials, quantitative understanding and other technical information fundamental to the processing, structure, properties and performance of materials; addresses the scientific basis for new advanced materials technologies; plans research around cross-cutting scientific themes such as nondestructive evaluation and phase diagram development; oversees Institute-wide technical programs in nuclear reactor radiation research and nondestructive evaluation; and broadly disseminates generic technical information resulting from its programs. The Institute consists of the following divisions:

- Ceramics
- Fracture and Deformation[3]
- Polymers
- Metallurgy
- Reactor Radiation

[1] Headquarters and Laboratories at Gaithersburg, MD, unless otherwise noted; mailing address Gaithersburg, MD 20899.
[2] Some divisions within the center are located at Boulder, CO 80303.
[3] Located at Boulder, CO, with some elements at Gaithersburg, MD.

NIST Technical Note 1269

Manual for the Cement Hydration Simulation Model

Leslie Struble,* Steven Johnson, Mark Hartmann,**
Lawrence Kaetzel, and Hamlin Jennings†

Center for Building Technology
National Engineering Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899

*University of Illinois, Champaign-Urbana, IL
**North Carolina State University, Raleigh, NC
†Northwestern University, Chicago, IL

September 1989

NOTE: As of 23 August 1988, the National Bureau of Standards (NBS) became the National Institute of Standards and Technology (NIST) when President Reagan signed into law the Omnibus Trade and Competitiveness Act.

U.S. Department of Commerce
Robert A. Mosbacher, Secretary

National Institute of Standards and Technology
Raymond G. Kammer, Acting Director

National Institute of Standards
and Technology
Technical Note 1269
Natl. Inst. Stand. Technol.
Tech. Note 1269
125 pages (Sept. 1989)
CODEN: NTNOEF

U.S. Government Printing Office
Washington: 1989

For sale by the Superintendent
of Documents
U.S. Government Printing Office
Washington, DC 20402

PREFACE

This manual provides documentation and instructions for the Cement Hydration Simulation Model. A computer software system for this model was developed by Steven Johnson and Hamlin Jennings, and subsequently revised by Leslie Struble and Mark Hartmann, all of the Building Materials Division of the Center for Building Technology (CBT) of the National Institute of Standards and Technology. The software system is currently managed and maintained by Lawrence Kaetzel of CBT. In order to operate the program, he may be contacted at the following:

Lawrence Kaetzel
Building 226, Room B348
National Institute of Standards and Technology
Gaithersburg MD 20899

(301) 975-5912

ABSTRACT

This manual describes the Cement Hydration Simulation Model, a computer-based model developed in the Building Materials Division of the Center for Building Technology, National Institute of Standards and Technology (NIST). The model simulates microstructural changes during hydration of tricalcium silicate, the principal constituent of portland cement. Output of the model may be in the form of two- or three-dimensional images, showing the location and size of each constituent.

The model is written in FORTRAN 77. It is interactive and modular. The model is installed on a super-mini computer at NIST, which can be accessed by other, suitably configured computers. The manual provides documentation, instructions, and examples of input and output using the model.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

1. INTRODUCTION

This manual is intended to assist users of the software system that constitutes the Cement Hydration Simulation Model developed at the National Institute of Standards and Technology (NIST). The manual provides both an overview of the model and detailed instructions for its execution. Additional details and sample runs are included in the appendices.

The model was developed in the Building Materials Division of the Center for Building Technology at NIST. The software is installed on the Division's central computer, a 3C/3230[1,2], whose operation is described in Appendix H. The model is written in FORTRAN 77. The program utilizes TEMPLATE[3], a FORTRAN graphics library, and incorporates UNI[4], a pseudo-random number generator.

From its inception, the model was intended to be used by others, whether at NIST or other research institutions. To run or modify the software, it will be necessary to become familiar with this manual, to have access to the Division computer, and to obtain from L. Kaetzel an account and a password. The 3C/3230 computer is part of the NIST computer network, so the software may be run from other computers via this network or from off-site computers using a telephone and appropriate communications software (Appendix I). To facilitate ongoing development and revisions, the program is modular in structure. Specific guidelines and requirements for revising the program are provided in Appendix J.

This manual covers Version 8, which was modified from the model (Version 7) developed by H. Jennings and S. Johnson. Revisions from Version 7 are listed in Appendix K. Version 8 is the first for which documentation has been prepared. Further revisions and additions are anticipated, after which this manual will be updated as required.

---

[1]Concurrent Computer Corporation, Oceanport, NJ.

[2]This and other trade names and company products are identified to specify adequately the experimental procedure. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products are necessarily the best available for the purpose.

[3]Megatek Corporation, San Diego, CA.

[4]Developed at NIST by J. Blue, D. Kahaner, and G. Marsaglia.

2. CEMENT HYDRATION

The model simulates the microstructure that develops during the reaction with water of portland cement or, more precisely, of its principal constituent tricalcium silicate (C_3S[5]). Portland cement is a finely ground powder whose hydration produces the binding constituent of concrete. A brief discussion of cement chemistry will provide useful background for the simulation model. The description of cement composition and hydration reactions is quite general and simplified; references are available that provide a more complete understanding of cement chemistry (e.g., Lea [1][6]).

Approximately 95 percent of portland cement is a clinker produced by firing in a kiln a mixture of limestone and shale or clay. The clinker consists of four principal phases: tricalcium silicate (C_3S), dicalcium silicate (C_2S), tricalcium aluminate (C_3A), and tetracalcium aluminoferrite ($C_2(A,F)$). Typical proportions are ~65 percent C_3S, ~15 percent C_2S, ~10-percent C_3A, and ~5 percent $C_2(A,F)$. The clinker is interground with approximately 5 percent of a calcium sulfate (typically gypsum) to produce portland cement.

In developing the simulation model, it was decided to use hydration of C_3S as a model system for the hydration of portland cement. The principal reason for this decision is that the hydration reactions of C_3A, $C_2(A,F)$, and gypsum are not as important to the overall microstructural development as are those of C_3S and C_2S. Furthermore, hydration reactions of the calcium silicates are much simpler than those of the other phases. However, it is recognized that this simplification may limit how closely the simulated microstructures agree with actual microstructures produced by hydrating cement.

Both C_3S and C_2S react with water to form CH and C-S-H. The CH is a crystalline material whose composition varies little from the stoichiometric composition. On the other hand, C-S-H is a gel whose composition varies substantially depending on such parameters as overall composition and age. As reviewed recently by Taylor [2], gel formed from C_3S has a C/S molar ratio of approximately 1.8; Taylor [3] recently calculated that the same gel, when saturated, would have an H/S molar ratio of approximately 3.7. These ratios produce a gel composition of $C_{1.8}SH_{3.7}$ and a hydration reaction as follows:

$$C_3S + 4.9\ H \rightarrow C_{1.8}SH_{3.7} + 1.2\ CH \qquad (1)$$

Similar calculations by Young and Hansen [4] for water-saturated C-S-H lead to a slightly different reaction:

---

[5]The following abbreviations, common in describing cement chemistry, are used: C for CaO, S for SiO_2, A for Al_2O_3, F for Fe_2O_3, and H for H_2O. Using this notation, the compound Ca_3SiO_5 is written as C_3S.

[6]Figures in brackets indicate references listed in section 5.

$$C_3S + 5.3\ H \rightarrow C_{1.7}SH_4 + 1.3\ CH \quad (2)$$

The differences in reactions (1) and (2) illustrate the experimental difficulties encountered in determining C-S-H gel composition.

The other aspect of cement and C_3S hydration relevant to this model is microstructure. The term microstructure is intended to include various aspects of structure (i.e., size, shape, location, and interaction of each constituent) on a micron scale. The following description of microstructure has been simplified as much as possible, and additional details may be found in recent reviews of the microstructure of hydrated cement [5-7] and C_3S pastes [2,7]. The emphasis is on cement, and C_3S is discussed only for those aspects known to differ from cement.

The starting material, because it is ground in a mill, typically consists of irregular, angular fragments covering a broad size range. The particle size distribution for a typical cement [6] has a median (by mass) diameter of approximately 20 μm, and more than 90 percent (by mass) distributed between 5 μm and 100 μm. The grains are generally agglomerated unless a chemical dispersant is used.

The principal hydration products, C-S-H and CH, are quite different in their microstructure. The C-S-H is amorphous or poorly crystalline, with very high internal porosity and specific surface. It occurs both on the surface of the anhydrous grains, as a layer that thickens progressively as hydration proceeds, and in the original intergranular regions in a less dense and more porous form. The CH, on the other hand, is crystalline and often occurs as large crystals between cement grains. Such crystals may exhibit hexagonal habit. However, the size and morphology of CH crystals appear to depend on the space available for their growth.

Hydrated cement (or C_3S) is highly porous. Pores occur both within C-S-H gel and in the spaces originally between anhydrous grains. As summarized by Mindess and Young (ref. [8] p. 99), pores within the gel are very small, ~0.5 nm to 10 nm; while intergranular pores are larger, 10 nm to 50 nm for medium capillary pores, and 50 nm to 10 μm for large capillary pores. The capillary pores are important both in fracture and permeability, and their volume, size, and connectivity are considered to depend on the water-cement ratio.

Scrivener [6] recently discussed the microstructural changes which occur during cement hydration, and the following discussion is based on that summary. Hydration was divided into three periods. During the first hydration period, up to a few hours after the cement and water are mixed, a gelatinous layer forms on the cement surface. This layer is probably an amorphous colloidal product consisting of calcium, alumina, silica, and sulfate.

During the second period, which lasts until about 24 hours after the initial mixing, there is rapid hydration and growth of C-S-H and CH. The C-S-H often forms a shell that is separated from the surface of the hydrating

grain by as much as 1 μm. Such a shell may be hollow, in which case it has variously been referred to as a Hadley grain or a hollow-shell hydration grain. During this period, C-S-H gel grows outward from the surface of shells from adjacent grains, meets, and bonds, causing the paste to become rigid (i.e., to set). It is considered by Scrivener that C-S-H during this period forms by a through-solution mechanism, that ions diffuse through the shell to deposit C-S-H on the outer surface of the shell. The CH is deposited as massive crystals wherever there is sufficient space. These crystals are typical subhedral and show hexagonal morphology, and often engulf smaller cement grains or C-S-H product.

During the third period, which lasts from 24 hours onward, the hydrated shells become thicker and less permeable. The C-S-H tends to deposit on the inside of the shell, causing it to grow inward and decreasing the separation between shell and grain. For grains large enough so that anhydrous material remains by that time, the ensuing hydration may cause the region between the inside of the shell and the anhydrous surface to fill in. Once this region has filled in, Scrivener suggests from microstructural evidence that subsequent hydration, which is very slow, occurs by a topochemical mechanism. In general, the hydration of C_3S is similar to that of cement. However, the separated shells that commonly occur in hydration of cement are not found in studies of C_3S hydration. Furthermore, the gelatinous C-S-H layer formed during the first hydration period may take the form of an exfoliating film for C_3S.

In summary, C-S-H may be grouped into two types, outer and inner product (in reference to the boundary of the original grain). The outer product is the outermost thin layer (~1 μm thick) formed through solution during the second hydration period. Scrivener suggested that hydration of grains smaller than ~5 μm produces only this outer product. The inner product may be divided into two regions. One is a layer ~8 μm thick within the original grain boundary that apparently forms through solution during the second and third periods and constitutes most of the inner product. The other region of inner product is an innermost layer that may form by a topochemical mechanism during the final hydration period.

Although hydration mechanisms have been studied for a number of years, there is still no clear and detailed understanding of hydration kinetics. Cement (or C_3S) hydration is exothermic, and may conveniently be followed using isothermal calorimetry. The three hydration periods (above) are reflected in the three regions of the heat evolution curve [6]. Upon mixing with water, cement undergoes immediate dissolution and hydration reactions that release a considerable amount of heat. The rate of heat evolution then decreases, passing through a minimum at approximately 3 hours. The period of low heat evolution is often called the induction period, and at its end the rates of hydration and heat evolution increase to a second maximum at approximately 24 hours. Perhaps 30 percent of the cement hydrates during this second maximum. From 24 hours onward the rate of hydration and heat evolution declines.

The mechanisms responsible for the beginning and end of the induction period are also not clear. Taylor [2] divided the various hypotheses into two

groups, which may be characterized as physicochemical and chemical. The physicochemical group involves the formation and subsequent disruption of a barrier layer. The layer of hydration product during the first period is considered to form a barrier, causing the hydration rate to decrease to a low level. Subsequently this layer for some reason increases in permeability or ruptures, and the hydration rate increases. The chemical group involves effects associated with nucleation or growth of C-S-H or CH as the cause of the decrease in hydration rate during first period and the increase in rate at beginning of second period.

During the third and final hydration period, Taylor [9] proposed a topochemical mechanism, and suggested that diffusion of Si atoms through the layer of inner C-S-H is the rate controlling step. The hypothesized topochemical mechanism was supported by Scrivener [6] based on microstructural evidence.

3. SIMULATION MODEL

3.1 Background

The Cement Hydration Simulation Model (Version 7) was developed at NIST by Hamlin Jennings and Steven Johnson over a period of several years (~1983-1987) as part of a NIST Competence Initiative to model cement hydration. Several papers describe the model and its conceptual basis [10-13]. Subsequent development (Version 8), including this manual, was aimed at facilitating its use in other research efforts within and outside NIST.

By the term simulation model, it is intended to convey that the model represents the development of microstructure through the use of operations performed by the computer. Furthermore, these operations are carried out particle-by-particle. This is distinct from a model that describes an entire system on a statistical or probabilistic basis. The simulation was based on microstructural observations using electron-microscopy and other laboratory techniques. The model is largely empirical, and the simulation is controlled through geometric constraints involving amounts, densities, and spatial distributions of hydration products. The model thus relies little on kinetic or thermodynamic mechanisms involved in hydration. Hydration occurs in cycles that are not scaled with respect to time, although cycles may be made to represent equal time intervals using a "kinetic" algorithm (discussed below).

The model is three-dimensional. Although graphical output may be two-dimensional, the microstructure is simulated in three dimensions.

To facilitate subsequent development, the model is modular in structure. It consists of three subprograms, each utilizing subroutines whose selection is controlled by parameters set by the user. The first program is interactive and run directly from the terminal; this program establishes the parameters for the simulation. The second program is not interactive, and may be run directly from the terminal or submitted in batch mode. This program carries out the calculations to simulate the microstructure. The third program is interactive and run directly from the terminal, and generates graphical representations of the microstructure.

3.2 Overview

The conceptual basis of the model is straightforward: spheres are placed in a three-dimensional box, a specified portion of each sphere is allowed to react, and the reaction products are placed in the box according to certain rules. This procedure is repeated to produce a microstructure that at any stage in the reaction may be viewed graphically in two or three dimensions.

The size of the box, the number of spheres (particles), and the initial size of each sphere are set-up parameters that define the density of the system. Density (i.e., water-solid ratio) is not itself an input parameter, but is calculated during the simulation run. Both the number of C_3S particles and the number of CH crystals produced by the simulation are requested. The

size distribution of C_3S particles is selected by the user, either in the form of a Weibull distribution or a frequency table.

The algorithms used to place particles were developed so that particles do not overlap each other, though they may contact. While particles may extend outside the boundaries, the centers of all particles must be inside the box.

The C_3S particles are spheres whose initial radii are selected during the set up. They may be placed in contact (clustered) or at random (dispersed). For dispersed placement, each particle is placed (in decreasing order of size) in a randomly selected location. For clustered placement, the number of clusters (a cluster is composed of a seed particle and its subparticles) is a set-up parameter, and particles in decreasing order of size are designated as seeds for clusters. Smaller particles are then assigned to each seed as subparticles. The first seed may be placed in the center or randomly, depending on the set-up parameter, and its subparticles are then placed to form the cluster. Depending on the set-up parameter, subsequent seeds are placed either randomly, in contact with the previous seed, or in contact with already placed subparticles of the previous cluster. In this way either many clusters may be formed, or one large cluster may be formed, by placing seeds in contact with either the previous seed or the previous cluster.

After each particle is placed (for either clustered or dispersed placement), a check is made to ensure that its center is inside the box and that it does not overlap with any already-placed particle. If these criteria are not met, placement is repeated. With a high density (proportion of particles) and after a large number of particles have been placed, finding a location free of overlap may require a prohibitively large number of attempts. A preset number of attempts is used (now set to 100 attempts), after which placement is attempted using a less demanding criterion. If clustered placement is not successful, dispersed placement is attempted, and if dispersed placement is not successful, placement is attempted using a Voronoi heuristic. The Voronoi heuristic (described in greater detail in Appendix E) attempts to place the particle in the empty space between previously placed particles. If placement using the Voronoi heuristic is not successful, the CH particles are placed and the data are saved, but the program terminates without executing hydration.

The CH crystals are also spheres (though they are represented graphically as hexagons), with an initial radius of zero. They are dispersed randomly. Each crystal is placed, then similarly checked to be sure that its center is inside the box and that it does not overlap with any already-placed C_3S particle or CH crystal. The overlap check uses a projected radius of 0.10 μm for the CH crystals. Placement of CH crystals is repeated for a preset number of attempts (also set to 100 attempts). If dispersed placement is not successful, the data are saved and the program terminates without executing any hydration. A Voronoi algorithm has not been developed for placement of CH crystals, though one can be developed if necessary.

Once particles have been placed, the solid volume and total porosity are calculated. In calculating the total solid volume, the excluded volume (i.e., the portion of material extending outside the box) is not included (see

Appendix F for details of the excluded volume calculation). From the volume of total solids and pores, the water-solid ratio is calculated using a density of 3.2 g/cm^3 for the anhydrous material. The water-solid ratio is therefore calculated after particles have been placed in the box.

Particles are checked for potential overlap after each hydration cycle. To speed up the process of checking for overlap, those particles that may collide during hydration are determined prior to the initial hydration reaction. A table of potential collisions is generated based on the projected radius of each C_3S grain or CH crystal. The table lists for each particle those other particles it may collide with during hydration. During each hydration cycle, only these particles are checked for potential overlap.

The projected radius for the collision check is determined as follows. For C_3S particles, the projected radius is the initial radius multiplied by the projected radius factor, a set-up parameter. For CH crystals, the projected radius is the radius of each CH crystal that would be produced if all C_3S particles hydrated fully, multiplied by the same projected radius factor. Thus the projected radius of C_3S is different for each particle, but the projected radius of CH is the same for all crystals.

The reaction (or hydration) consists of removing a portion of anhydrous material from the surface of each C_3S particle and depositing a corresponding amount of hydration product either on the particle or elsewhere (shown in fig. 1). The C-S-H is deposited on the C_3S particle from which it formed. The CH from each hydration portion is distributed equally among the CH seed crystals, whose number is defined as a set-up parameter.

The C_3S particles hydrate one at-a-time in order of size from small to large. The volume of anhydrous material reacting is calculated for each particle according to the following equation:

$$V_a = (\frac{4\pi}{3})(R_a^3 - [R_a - HDX]^3) \qquad (3)$$

where V_a is the volume of anhydrous material reacting, R_a is the radius of the anhydrous particle before the hydration cycle, and HDX is the change in radius for the hydration cycle.

The user may select to use the kinetic algorithm described by Jennings and Johnson [12]. In this algorithm, the amount of material hydrated during each hydration cycle decreases as the thickness of the C-S-H layer increases according to the following equation:

$$HDX = HD \left[1 - \left(\frac{THK}{TC} \right)^2 \right] \qquad (4)$$

where HDX is the change in radius for the hydration cycle, HD is the change in radius for the initial cycle (a set-up parameter), THK is the thickness of the

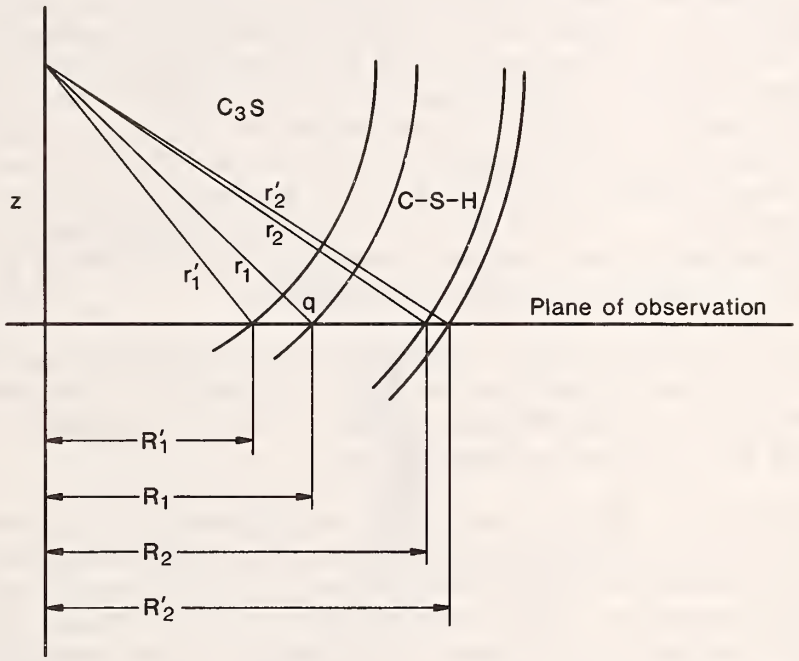


Figure 1. Schematic of the hydration reaction; r_1 is the C_3S radius before hydration, r_1' is the C_3S radius after hydration, q is the hydration increment, r_2 is the C-S-H radius before hydration, and r_2' is the C-S-H radius after hydration. (Figure 2 from Jennings and Johnson [12]).

C-S-H layer, and TC is the critical thickness (a set-up parameter). Thus as THK approaches TC, the size of the radius change approaches zero. Once THK reaches TC, the particle no longer hydrates. This equation was used to simulate diffusion control of hydration rate.

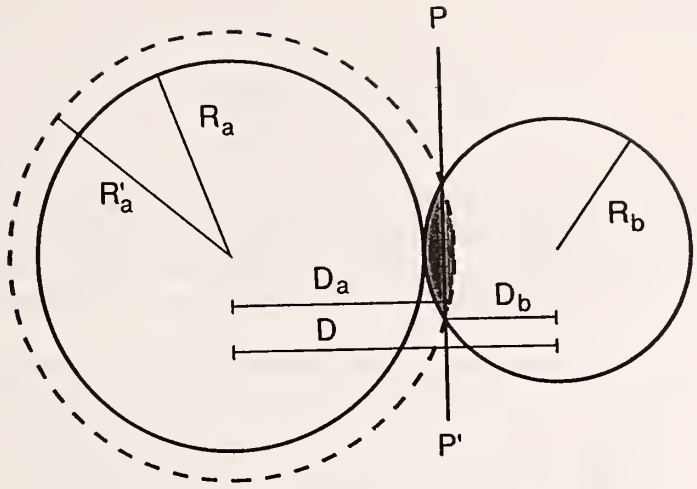
If the kinetic algorithm is not used, the change in radius of anhydrous material is the same for each particle regardless of its size. In that case, HDX remains equal to HD, the change in radius for the initial cycle (a set-up parameter).

The C-S-H may be placed entirely on the outer surface, or in part on the outer surface and in part on the inner surface of the hydrate shell. There are three set-up parameters that control this placement: the relative outer volume, the relative inner volume, and the backfill factor. The relative outer volume is the volume of C-S-H placed on the outer surface of the hydration shell, relative to the volume of C_3S that reacted to produce it. The relative inner volume is the volume of C-S-H placed on the inside surface of the hydration shell, relative to the volume of C_3S that reacted to produce it. The backfill factor is the proportion of reacting C_3S (by volume) that is used to produce inner product C-S-H (i.e., C-S-H placed within the original volume of reacted material). If the backfill factor is sufficiently low, separated-shell hydration grains are formed; whereas a high backfill factor prevents separated-shell hydration grains from forming.

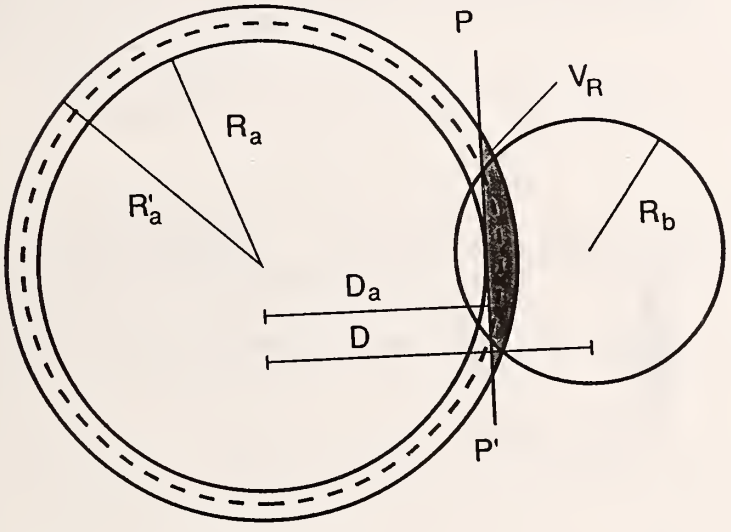
In each hydration cycle, after all C_3S particles have hydrated, the CH produced is added to the surface of the CH crystals. The number of CH crystals and the volume of CH produced relative to the volume of anhydrous material reacting are both set-up parameters.

For both C-S-H and CH, particles may contact but not overlap as they grow. Once a C-S-H or CH particle makes contact with another C-S-H or CH particle, the excess or overlapping volume is distributed over the remaining surface of the particle. This overlap correction is based on the algorithm described by Jennings and Johnson [12], and is illustrated in figures 2 and 3. The correction occurs in two steps (described below), first the calculation of overlapping volume due to hydration, second the distribution of the overlapping volume over the remaining (not excluded) surface of the hydrating particle.

The calculation of overlapping volume depends on whether the particles are colliding for the first time (initial collision) or have already collided (subsequent collision). The geometry involved in the initial collision is illustrated schematically in figure 2(a). A plane PP' connects the intersections of the hydrated surface of the primary (hydrating) particle and the pre-existing surface of the secondary particle. The overlapping volume is the sum of the volume of the two domes formed by the spheres and this plane. This volume is shown in figure 2(a) using a hatch pattern.



(a)



(b)

Figure 2. Schematic of overlap correction illustrating the geometry of calculating overoapping volume for initial overlap (a) and subsequent overlap (b); R_a is the radius of particle a before the hydration cycle, R_a' is the radius of particle a after the hydration cycle, R_b is the radius of particle b, D_a is the distance from the center of particle a to the plane PP', D_b is the distance from the center of particle b to the plane PP', and D is the distance between the centers of particles a and b; the overlapping volume VR is hatched.

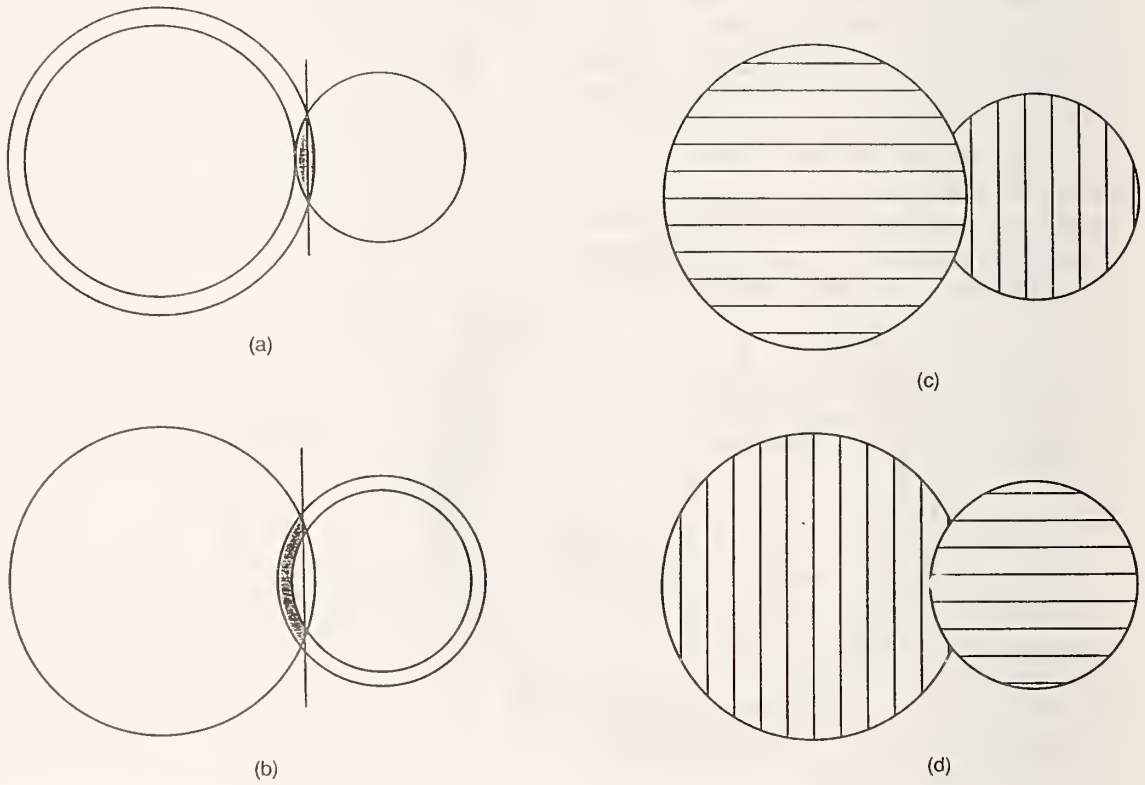


Figure 3. Schematic illustrating overlap correction for particles of radii 5 (left particle) and 3 (right particle), hydration decrement 0.5; left particle hydrates with overlapping volume shown (a), right particle hydrates with overlapping volume shown (b), resulting image if the right particle is drawn first (c), and resulting image if the left particle is drawn first (d).

The first step in the calculation is the following:

$$D_a = \frac{(R_a'^2 - R_b^2 + D^2)}{2D} \tag{5}$$

where D_a is the distance from the center of the primary particle to the plane PP', R_a' is the radius of the primary particle after hydration, R_b^2 is the existing radius of the secondary particle, and D is the distance between the center of the primary particle and the center of the secondary particle.

If the primary particle overlaps less than half of the secondary particle (or, more precisely, if $D_a < D$), then

$$D_b = D - D_a; \quad H_a = R_a' - D_a; \quad H_b = R_b - D_b; \tag{6}$$

or if the primary particle overlaps more than half of the secondary particle (or, more precisely, $D_a \geq D$), then

$$D_b = D_a - D; \quad H_a = R_a' - D_a; \quad H_b = R_b + D_b \tag{7}$$

where D_b is the distance from the center of the secondary particle to the plane PP', H_a is the height of the dome of the primary particle, and H_b is the height of the dome of the secondary particle.

In either case, the dome volumes are calculated as follows:

$$V_a = (\frac{\pi}{3})(H_a^2)(3R_a' - H_a) \tag{8}$$

and

$$V_b = (\frac{\pi}{3})(H_b^2)(3R_b - H_b) \tag{9}$$

where V_a is the volume of the primary particle dome and V_b is the volume of the secondary particle dome. The overlapping volume is the sum of these two volumes. The remaining surface of the primary particle, not excluded due to overlap is

$$S_a = 4\pi R_a'^2 - 2\pi(R_a')(H_a). \tag{10}$$

Unlike the calculation for initial collision, the calculation of overlapping volume for a subsequent collision, illustrated schematically in figure 2(b), is an approximation. In this case, the plane PP' connects the intersections of a surface midway between the pre-existing surface and the hydrated surface of the primary (hydrating) particle and the pre-existing surface of the secondary particle. The overlapping volume, shown in figure 2(b) using a hatch pattern, is the difference in volume of the two domes formed by spheres of the primary particle (one the surface before hydration, and the other the surface after hydration) and this plane. D_a is calculated according to eq 5. Then

$$H_a' = R_a' - D_a; \quad H_a = R_a - D_a \tag{11}$$

where H_a' is the dome height after the hydration cycle, and H_a is the dome height before the hydration cycle. Similar to eqs 8 and 9,

$$V_a' = (\frac{\pi}{3})(H_a'^2)(3R_a' - H_a') \tag{12}$$

and

$$V_a = (\frac{\pi}{3})(H_a^2)(3R_a - H_a) \tag{13}$$

where V_a' is the dome volume after the hydration cycle and V_a is the dome volume before the hydration cycle. The overlapping volume is the difference between these two dome volumes. The remaining surface of the primary particle, not excluded due to overlap is

$$S_a = 4\pi R_a'^2 - 2\pi(R_a')(H_a'). \tag{14}$$

It should be noted that the algorithm for calculating the volume to be redistributed due to overlap was developed only for the overlap of two particles. Whenever three or more particles overlap with a mutual overlapping volume, the amount of volume to be redistributed will be overestimated by the model.

The radius of the primary particle must finally be increased to correct for the overlapping volume. This correction is an estimate, which assumes that the hydration layer is very thin:

$$\Delta R_a = \frac{V}{S_a} \tag{15}$$

where ΔR_a is the increase in radius of the primary particle, V is the overlapping volume, and S_a is the unexcluded surface of the primary particle.

When a particle is overlapped to the extent that the unexcluded surface area of the primary particle approaches zero, then the increase in radius of the primary particle due to overlapping volume (eq 15) will become very large. This estimated correction is not valid, however, if the increase in radius is large. To prevent this situation, particles no longer hydrate when their unexcluded surface becomes small. More specifically, once the ratio of unexcluded surface to total surface is less than 0.1, that particle ceases to grow.

While the radius is corrected for overlapping volume, at least approximately, the shapes of overlapping particles are not represented correctly in the two-dimensional graphical display. The two-dimensional display is in the form of circles and hexagons. These shapes are not corrected for overlap. As illustrated in figures 3(c) and (d), the area (proportion) of overlapping particles is misrepresented. In some cases, this effect may cause a serious error in the apparent proportion of each phase.

After each hydration cycle the solid volume, pore volume and percent porosity are calculated, and the data file is saved. The hydration cycles are repeated until the number of cycles specified by the set-up parameter is reached, until no anhydrous material remains that is free to react, or until the layer of C-S-H on every particle has reached the critical thickness if the kinetic algorithm was selected.

3.3 Implementation

The simulation model is described in additional detail in Appendices A, B, C, and D, with sample output and examples in Appendix G. The model consists of three subprograms. The first, the set-up program, allows the user to select options and parameters that specify the particular situation to be simulated. The second subprogram performs the actual simulation using the parameters selected during the set-up. The results of each hydration cycle during a simulation run are stored in a data file. The last subprogram generates a graphical representation of the three-dimensional simulation data, allowing one to visualize the simulation results, in the form of two-dimensional or three-dimensional (perspective) images from selected hydration cycles.

The model is operated on a 3C/3230, 32 bit super-minicomputer equipped with a Tektronix[7] 4129 color graphics terminal. The code is written in FORTRAN 77 and utilizes TEMPLATE (a graphics library). Other graphics terminals may be supported by TEMPLATE if the program is linked to the appropriate device driver.

---

[7]Tektronix, Inc., Beaverton, OR.

Version 8 of the program contains only a few messages to trap errors. If a parameter value is out of range, the prompt requesting the parameter is generally repeated, providing no information concerning the error.

Accessing the 3C/3230 computer and the operating system are described in Appendix H. Accessing the computer from off-site is described in Appendix I.

Throughout this manual, output from the computer is indicated by **BOLD-FACE TYPE**, and input from the user is UNDERLINED. The <> brackets enclose keys by name (as distinct from keys by letter or number), not letters to be typed.

3.3.1 Set-up

To run the set-up program, M8A, the user enters the command:

RUNSET8<cr>

This command loads and initializes the set-up program. The user then responds to the prompts to select or review parameter values that control the simulation run.

The user may elect to set-up a new simulation or to rerun an existing simulation using new hydration parameters. For a rerun, the program skips to the hydration parameters after the names of the new and existing simulation files have been requested.

The first parameter requested is the filename for the simulation data. This name may have up to 8 characters but should have no extension, as the extension .CM8 is added automatically. This filename will be used to store the simulation results.

CAUTION: Attempting to use an existing simulation data filename in the set-up program will cause the subsequent simulation run to terminate.

The next set of parameters defines the size of the box and the number and size of C_3S particles. The maximum box size is 100 μm in each dimension, though it is often necessary to use a smaller dimension to obtain realistically low water-to-cement ratios. The maximum numbers of particles are 1000 for C_3S particles and 2000 for CH crystals; these serve to limit the amount of computer memory, and can be increased if necessary. The density or water-solid ratio is not selected by the user, but depends on the number of C_3S particles and the box dimensions, parameters that are selected by the user.

Particle size may be defined using a Weibull function or a user supplied distribution. The Weibull distribution is:

$$X = \exp\left[\frac{C - Y}{B}\right]^A \qquad (16)$$

where X is the relative proportion by number and Y is the particle radius. Values may be selected for the constants A, B, and C, or default values (A = 0.55, B = 0.40, and C = 0.60) may be used. The Weibull particle size distribution using the default values is shown in figure 4. This form of Weibull distribution produces a PSD somewhat different from that of a typical cement (fig. 5); in order to more closely approximate a cement size distribution, it will be necessary to generate the particles using the optional distribution table described below.

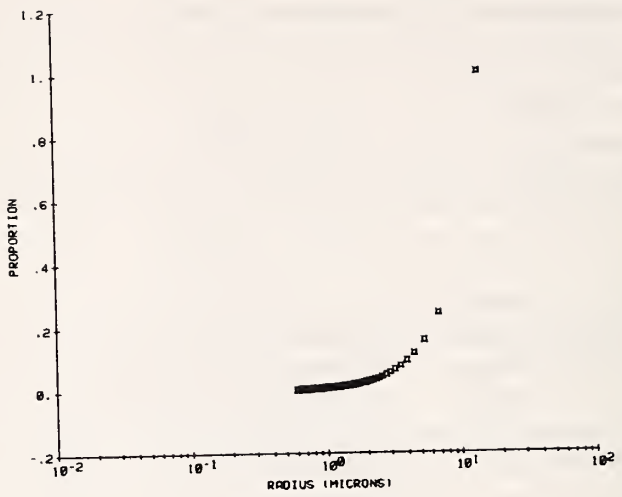
The user-supplied distribution, a frequency distribution showing the proportion of particles in each size range, is stored in the file M8TAB. This file must be written using the Edit program (Appendix H) prior to running the set-up program. The first line in this file is a label assigned to the data. The next line is a mode (0 or 1). For mode 0, each particle is randomly assigned a radius within its range; for mode 1, each particle is assigned the mean radius of its range. The mode is followed for each size range by the lower radius (in microns), the upper radius (in microns), and the relative proportion by number (0 to 1.0) of particles in that range. These data are listed when reviewing or printing the set-up parameters.

The particles are distributed either randomly or clustered. For clustered distribution, the number of clusters and the clustering method are defined by the user. It is possible to set the number of clusters so low that all particles cannot be placed. In the example in Appendix G (TEST5.PAR, table G-1), particles could be placed using 100 clusters, but they could not be placed using 10 clusters.

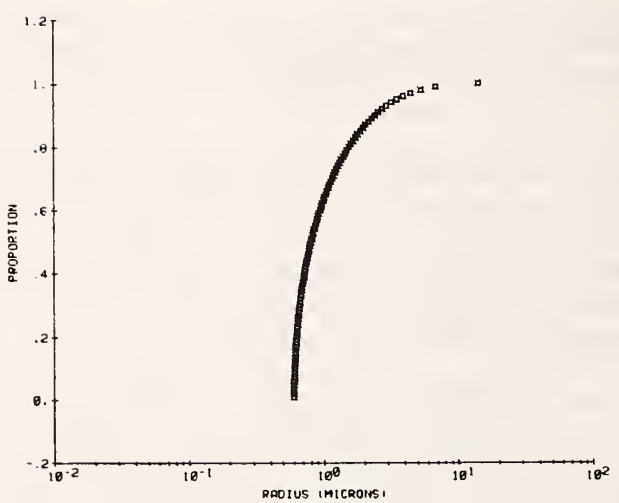
The next set-up parameter is a number to seed the random-number generator. Those functions in the model requiring a random number (e.g., particle size generation and particle placement) utilize UNI, a pseudo random-number generator. This routine generates quasi, uniform random numbers; the first time the routine is called, the seed is used to initialize the program and generate the first random number. Thereafter, a new random number is generated each time the routine is called. It should be noted that calling the routine repeatedly will produce the same sequence of random numbers if the same seed is used. Therefore, repeated random operations (i.e., particle size and placement) using different set-up parameters but the same random number seed may produce quite similar simulations.

Hydration is controlled using several additional set-up parameters. The number of hydration cycles may range between 0 and 1000, with a higher number involving more computing time. The initial change in radius may range between 0.01 μm and 1.0 μm. This parameter controls the extent of reaction during each hydration cycle, so a lower value provides greater sensitivity. If the user elects to use the kinetic algorithm, the critical thickness must be selected. This parameter ranges between 1.0 μm and 100.0 μm. For this algorithm to affect the rate of hydration, the value must be fairly low. In the example in Appendix G (TEST3.PAR), a critical thickness of 50 produced no observable effect on the microstructure after ~30 percent hydration.

Additional hydration parameters relate to volume and placement of C-S-H and CH. The volume of outer product C-S-H, inner product C-S-H, and CH are

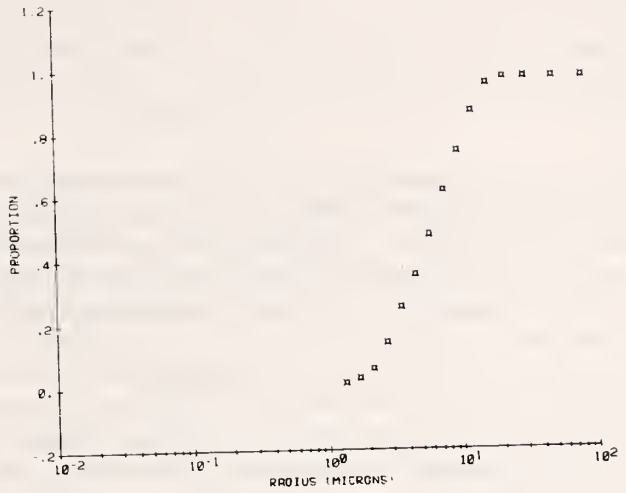


(a)

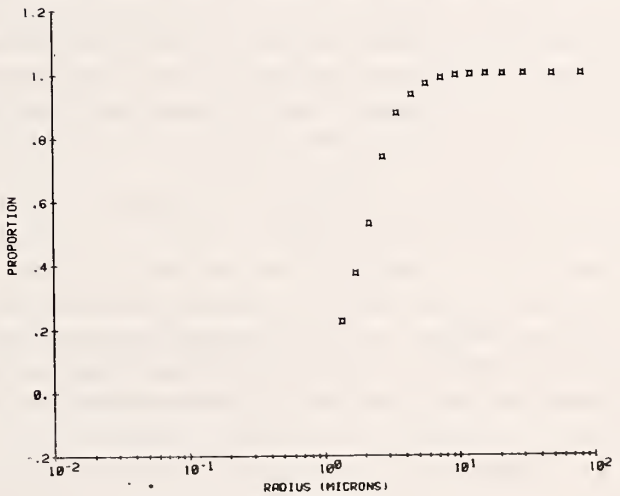


(b)

Figure 4. Weibull distribution with default parameters, showing cumulative proportion finer by weight (a) and by number (b).



(a)



(b)

Figure 5. Cement particle size distribution, converted from data reported by Scrivener [6], showing cumulative proportion finer by weight (a) and by number (b).

each set relative to the volume of C_3S that reacted to produce it. The suggested values are those estimated by Young and Hansen [4] based on eq 2. The suggested value for relative volume of inner and outer C-S-H is 1.75, and for relative volume of CH is 0.61. The backfill factor may be set between 0 and 1.0; lower values (below ~0.5) produce grains simulating separated-shell hydration grains. More specifically, if the product of the backfill factor and the relative volume of inner product is less than one, separated-shell grains will be produced.

The final set-up parameter is the projected radius factor. This is used to determine the the maximum radius of C_3S particles and CH crystals expected after hydration is complete. Only those particles whose center-to-center distance is less then the sum of their projected radii are checked for contact during each hydration cycle. If any hydrating particle or CH crystal exceeds its projected radius, the simulation run terminates. The projected radius factor may range from 0.01 to 5.0. A lower projected radius factor should be used in simulations with a low number of hydration cycles, and higher values in simulations with a large number of cycles. High values will substantially increase the time required to run the simulation, while low values may cause a simulation run to terminate prematurely. Too high a value may cause the collision table to exceed its configured size, in which case the simulation run will terminate. In the example in Appendix G (TEST3.PAR, Table G-1), the value 3.0 caused the simulation to terminate prematurely (during cycle 30); whereas a value of 4.0 caused the simulation to terminate before any hydration cycles because the collision table could not be generated.

At the end of the set-up program, the user may elect to review, change, or print the set-up parameters, or to save the parameters and exit the program. The file must be saved in order to use the parameters in the simulation program. The set-up parameters are stored in the parameter file, M8PAR. After exiting the set-up program the user can either proceed immediately with the simulation run(s) or submit the simulation at a future time.

CAUTION: If the set-up program is re-run, it will automatically overwrite the previous set-up information. Once a simulation run has been completed, the parameter file (M8PAR) can be written over without affecting the simulation data file(s). Therefore, if the simulation run has not been submitted and the user wants to retain the information in the parameters file without the risk of it being reused, the file should be renamed. When it is time to run the simulation, the new filename must be changed back to M8PAR.

3.3.2 Simulation

Program M8B executes the actual simulation. This program is not interactive. It can be run directly from the terminal or in a batch mode; because run times are often long, the batch mode may be preferred.

To run the program directly, enter:

RUNSIM8<cr>.

This procedure will load and start the program M8B, which automatically retrieves the set-up file M8PAR and runs the simulation according to the user's specifications. The user will be unable to use the terminal during this direct run.

For submitting a batch run, enter:

RUNBAT8<cr>.

The batch run is submitted to the batch queue for processing. The user will be able to use the terminal for other tasks, but should not attempt to run the set-up program or submit a second simulation run because the parameter file M8PAR remains attached to the simulation run until the run(s) have completed. To see if a batch run is still executing, enter the command:

INQ<cr>.

The results of each simulation run are stored in a data file (described in Appendix D). This file contains all parameters required to define the simulation, the position and size of each particle, and the size and status of each particle for each hydration cycle. This file can be renamed, but editing this file is discouraged because of the critical nature of the data format for any program that may read the data file for reruns or graphics. The data file can be printed for program testing or data analysis.

During the simulation run, the collision information is compiled into a collision table, and the index for this table is saved as part of the simulation data file. There are four types of collisions: collisions of primary (growing) C_3S particles with secondary C_3S particles and with secondary CH crystals, and collisions of primary CH crystals with secondary CH crystals and with secondary C_3S particles. For each type (CT1, CT2, CT3, and CT4), the collision data consist of the list of secondary particles. The collision table is indexed by line number: for each primary particle, the index contains the beginning line number and the number of collisions (or lines) for each type of collision.

The simulation program may terminate prior to completion for a number of reasons. For example, the simulation program will not write over an existing simulation data file, and attempting to name a data file with the name of an existing file will cause the program to terminate. If the system is so dense that all C_3S particles or CH crystals cannot be placed, the anhydrous data will be saved and the program will terminate. If the number of potential collisions exceeds the dimension set for the collision table, the program will terminate before any hydration cycles. Likewise, if any hydrating C3S particle or growing CH crystal exceeds its projected radius, whatever data have been generated will be saved and the program will terminate.

A log file, SIM8.LOG, is generated during each run (an example is shown in Appendix G). If the program terminates before hydration is complete, this log file may be checked to ascertain the reason for the termination.

3.3.3 Graphics

The simulation model uses a graphics program to provide two- or perspective three-dimensional images from the computed simulation data. The program is set up to run on a Tektronix 4115/4129 color graphics terminal or a terminal that emulates it. To run the program using another terminal, it will be necessary to compile and link the program with the required driver, and to add the driver to the configuration file, CONFIG.TEK.

To start the program, enter:

RUNGRF8 <cr>.

After the program loads and starts the user will be prompted for the simulation data filename.

The terminal/color menu is displayed. Three default sets are provided and the fourth option allows the user to specify feature colors directly through another menu. In a color mode, default colors allow differentiation of anhydrous C_3S, C-S-H associated with C_3S particles that have reacted fully, C-S-H associated with C_3S particles still hydrating, CH particles that have stopped growing, and CH particles that are still growing. In the black-and-white mode, hatch patterns allow differentiation of anhydrous C_3S, C-S-H, and CH.

The drawing modes consist of three types of two-dimensional graphics and one type of three-dimensional graphics. The first two-dimensional mode displays an X-Y plane cross-section at the current value of Z. The anhydrous and hydrated particles are shown using the selected color. The CH crystals are depicted using hexagon outline. The second two-dimensional mode draws a non-sliced set of particles which have been sorted background to foreground and then displayed as solid overlaid images, again color coded. Here the calcium hydroxide crystals are shown in outline form to reduce visual complexity. The third two-dimensional mode draws a non-sliced transparent view of all particles and crystals in the system box. The outlines are color coded as before. For any of these two-dimensional modes the Z-value may be changed.

The three-dimensional graphics displays hydrated C_3S particles, but not CH particles. Lines of latitude and longitude are used to depict the outer surface of each particle. In the three-dimensional graphics mode, a menu of options to control the graphics is first displayed. The user can change the number of "wires" used in the latitude and longitude directions, rotate the 3D box in increments of 90° or 180° (which require resorting the order background to foreground and redrawing the image in the new position), change the viewing angle, change the viewing window parameters, turn on and off the clipping option (for hither and yon clipping using values set in the three-dimensional window option[8]), reset all graphics parameters back to the default values, execute the three-dimensional image, or return to the main menu.

---

[8]The user may refer to the TEMPLATE manual for discussion of these terms.

Returning to the main menu, the user can select to change the size of the viewing window, change to a different hydration cycle, change the cutting plane's Z-value, get a new simulation data file, change the colors previously selected, or terminate the program.

The size of the viewing window may be changed for either two- or three-dimensional graphics. The initial size is based on the size of the box (the maximum and minimum values in each direction) selected in the set-up program. If the size of the viewing window is changed, the size and position of each particle will be rescaled to fill the viewing window. For two-dimensional graphics, this is a straightforward operation and should produce the desired effect. For the three-dimensional graphics, however, there are other size-related parameters that may need to be optimized using the three-dimensional graphics menu.

3.4 Discussion

The Simulation Model was modified extensively to produce Version 8. The modifications to Version 7, listed in more detail in Appendix K, fall into several general categories. The primary category of modifications provides more dense placement of particles, by incorporating a placement heuristic that searches for empty space into which particles may be placed. Without this modification, simulations could not be run using realistically dense water-to-cement ratios. Another modification provides optional control of hydration kinetics. Version 7 used the diffusion algorithm; in Version 8, this algorithm was made an option, with the other option being no kinetic control. Another modification is to calculate directly (rather than estimate) the excluded volume. Finally, various modifications served to correct errors observed in Version 7, to improve the logic of the program, to enhance the usefulness of the program, and to reduce the run time of the program.

The number of particles in a simulation limits how closely a single simulation may be considered to represent a larger system, in terms both of reproducibility and of validity of the model. The maximum number is currently 1000 for C_3S and 2000 for CH. With 1000 particles, the particle sizes generated from a particle size distribution table may not be reproducible. The maximum particle size, if set to be a random size within the largest possible size interval, can be as high as 50 μm, though more typically ~3 μm. Furthermore, a reasonable water-to-cement ratio (e.g., 0.60) and a particle size distribution typical for cement require a box size ~10.0 μm in each dimension. This is simulating a very small volume of material. The number of particles are also expected to limit the validity of simulation. For a simulation of this type, the maximum particle size should be kept small with respect to the box size, generally less than one-tenth of the box length. For a maximum particle size of 3 μm and a box length of 10 μm, this proportion is exceeded. The effect on validity in the Cement Hydration Simulation Model is made worse by the lack of periodic boundary conditions. For these reasons, work is currently underway to increase the maximum number of particles.

Another important development in the simulation model would be to incorporate hydration kinetics. At present, Version 8 simulates hydration either with no control of reaction kinetics, or using a kinetic algorithm

developed to simulate diffusion control of the hydration reaction. This kinetic algorithm represents an initial effort to develop kinetic control in the simulation model. In their description of the kinetic algorithm, Jennings and Johnson [12, p. 12] noted the need to use improved mathematical expressions as they become available. Thus developing improved kinetic algorithms is an important future effort. A related issue is that the model should track the liquid phase as well as the solid phase. Tracking the composition of the liquid phase is important in developing and validating kinetic algorithms. Tracking the proportion of the liquid phase is necessary to simulate effects such as chemical shrinkage and self desiccation, and is important in simulating flow behavior. These two areas may be pursued in the future, but no work is currently underway.

The Building Materials Division expects to continue development of this simulation model. Planned developments (Version 9) include adding a subprogram to estimate the size of the box or the number of particles for a target water-to-cement ratio, increasing the maximum number of particles, which requires that particle placement and generating the collision table require less computer time, adding periodic boundary conditions, and developing an algorithm for calculating overlapping volume when three or more particles share overlapping volume. In addition, a long-term effort is underway to add to the model a simulation of rheological behavior and to incorporate the model in a simulation of diffusion in hydrated cement.

A few studies have been carried out or are underway to assess validity of the model. The simulated microstructure has been shown to be qualitatively similar to the microstructure of hydrated C_3S as observed using scanning electron microscopy [12]. Likewise, the pore structures in a simulated microstructure and in a real microstructure of hydrated C_3S have been shown to be similar when analyzed using a 2-point correlation function and using a connectivity measure developed at NIST [14]. Additional validation experiments are underway.

4. ACKNOWLEDGMENTS

The authors are indebted to the many researchers who have discussed with us the various aspects of cement hydration and microstructure development, and thereby have influenced considerably the development of this model. We particularly wish to acknowledge the contributions of Dr. K. Scrivener, Prof. H.F.W. Taylor, and Dr. C. Witzgall. We also acknowledge the support provided by Dr. J. Clifton and Dr. G. Frohnsdorff throughout this effort. Finally, we are grateful to Dr. K. Scrivener for the particle size data used here.

5. REFERENCES

1. F.M. Lea (1970). The Chemistry of Cement and Concrete, 3rd Edition, 727 pages. New York: Chemical Publishing Co., Inc.

2. H.F.W. Taylor (1986). Chemistry of cement hydration. In: 8th International Congress on the Chemistry of Cement, I, 82-110. Rio: Secretaria Geral do 8^{0} CIQC.

3. H.F.W. Taylor (1987). Bound water in cement pastes and its significance for pore solution composition. In: Microstructural Development During Hydration of Cement, Materials Research Society Symposia Proceedings, 85, edited by L.J. Struble and P.W. Brown, 47-54. Pittsburgh: MRS.

4. J.F. Young and W. Hansen (1987). Volume relationships for C-S-H formation based on hydration stoichiometries. In: Microstructural Development During Hydration of Cement, Materials Research Society Symposia Proceedings, 85, edited by L.J. Struble and P.W. Brown, 313-322. Pittsburgh: MRS.

5. S. Diamond (1986). The microstructures of cement paste in concrete. In: 8th International Congress on the Chemistry of Cement, I, 122-147. Rio: Secretaria Geral do 8^{0} CIQC.

6. K.L. Scrivener (1988). The microstructure of concrete. In: Materials Science of Concrete, edited by J.P. Skalny. To be published by The American Ceramic Society.

7. H.M. Jennings (1983). The developing microstructure in portland cement. In: Advances in Cement Technology, edited by S.N. Ghosh, 349-396. New York: Pergamon Press.

8. S. Mindess and J.F. Young (1981). Concrete, 671 pages. Englewood Cliffs, NJ: Prentice-Hall, Inc.

9. H.F.W. Taylor (1984). The reactions of cement compounds with water. In: Proc. 10th Int. Symp. Reactivity of Solids, Dijon. Amsterdam: Elsevier Science Publishers.

10. H.M. Jennings, J. Clifton, G. Frohnsdorff, and S.K. Johnson (1986). On establishing functional relationships between reaction mechanisms and engineering properties of cement-based systems. In: Research on the Manufacture and Use of Cements, Proceedings of the Engineering Foundation Conference, edited by G. Frohnsdorff, 7-93. New York: Engineering Foundation.

11. S.K. Johnson and H.M. Jennings (1986). Computer simulated hydration of a cement model. In: CIB.86, Advancing Building Technology, Proceedings of the 10th Triennial Congress of the International Council for Building Research, Studies and Documentation, 6 2086-2095. Gaithersburg: NIST.

12. H.M. Jennings and S.K. Johnson (1986). Simulation of microstructure development during the hydration of a cement compound. Journal of the American Ceramic Society, 69(11) 790-795.

13. H.M. Jennings (1987). Towards computer-based microstructure models for cement-based systems. In: Microstructural Development During Hydration of Cement, Materials Research Society Symposia Proceedings, 85, edited by L.J. Struble and P.W. Brown, 291-300. Pittsburgh: MRS.

14. H.M. Jennings and D.P. Bentz (1987). Quantitative analysis of the pore structure of hydrating cement. Poster presented at the 89th Annual Meeting of the American Ceramic Society, April 28, 1987, Pittsburgh PA.

15. G. Voronoi (1908). Nouvelles applications des parametres continus a la theorie des formes quadratiques, Deuxieme Memoire, Recherches sur les parallelloedres primitifs. Journal fur die reine und angewandte Mathematik, 134 198-287.

16. B.J. Gellatly and J.L. Finney (1982). Characterizations of models of multicomponent amorphous metals: the radical alternative to the Voronoi polyhedron. Journal of Non-Crystalline Solids, 50 313-329.

17. J. Bernal (1989). On the expected complexity of the 3-dimensional Voronoi diagram. NISTIR 89-4100, 19 pages.

18. F.P. Preparata and D.E. Muller (1979). Finding the intersection of n half-spaces in time O(n log n). Theoretical Computer Science, 8 45-55.

19. H. Solomon and H. Weiner (1986). A review of the packing problem. Communications in Statistics - Theory and Methods 15 2571-2607.

20. A. Renyi (1958). On a one-dimensional problem concerning random space-filling. A Magyar Tudományos Akadémia Matematikai Kutató Intézetének Közlleményei 3(1/2) 109-127.

21. L. Friegold and J.T. Donnell (1979). Maximum density of random placing of membrane particles. Nature (London) 278 443-445.

22. M. Tanemura (1979). On random complete packing by discs. Annals of the Institute of Statistical Mathematics 31 351-365.

23. W.S. Jodrey and E.M. Tory (1979). Simulation of random packing of spheres. Simulation 32 1-12.

24. J. Feder (1980). Random sequential adsorption. Journal of Theoretical Biology 87 237-254.

25. H.W. Lotwick (1982). Simulation of some spatial hard core models, and the complete packing problem. Journal of Statistical Computation and Simulation 15 295-314.

26. E.M. Tory, W.S. Jodrey, and D.K. Pickard (1983). Simulation of random sequential adsorption: efficient methods and resolution of conflicting results. Journal of Theoretical Biology 102 439-445.

27. G.Y. Onoda and E.G. Liniger (1986). Experimental determination of the random-parking limit in two dimensions. Physical Review A: General Physics 33 715-176.

28. D.W. Cooper (1987). Parking problem (sequential packing) simulations in two and three dimensions. Journal of Colloid and Interface Sci. 119 442-450.

29. K. Gotoh, W.S. Jodrey, and E.M. Tory (1978). Average nearest-neighbour spacing in a random dispersion of equal spheres. Powder Technology 21 285-287.

30. K. Gotoh, M. Nakagawa, and H. Matsuoka (1985). Size distribution of clusters inherent in random dispersions of equal spheres with a coagulation radius. Particulate Science and Technology 3 27-36.

31. D.W. Cooper (1988). Random-sequential-packing simulations in three dimensions for spheres. Physical Review A: General Physics 38 522-524.

32. Y. Pomeau (1980). Some asymptotic estimates in the random parking problem. Journal of Physics A: Mathematical and General 13 L193-L196.

33. R.H. Swendsen (1981). Dynamics of random sequential adsorption. Physical Review A: General Physics 24 504-508.

34. P.T. Thach (1988). The design centering problem as a d.c. programming problem. Math Programming 41 229-248.

35. CRC Standard Mathematical Tables, 27th Edition, edited by W.H. Beyer. Boca Raton: CRC Press, Inc.

APPENDIX A. PROGRAM

Libraries and programs are listed in table A-1.

Table A-1. Simulation Model

Library	Main Program	Subprogram
-----	M8A	
-----	M8A	PARREV
M8SIM	M8B	
		M8G
		M8T
		M8P
		M8CT
		M8CTI
		M8HA
		M8SAV
		M8UNI
		M8SRT
		M8R
		M8HB
		M8V
		M8EV
M8GRF	M8GRF	
		M8G2SORT
		M8COLOR
		M8GR3D
		M8GTEXT
		M8G3SORT
		M8BOX
		M8SPHERE
		M8GR2D

A.1 M8A.FTN (Main Program)

FORTRAN Filename:	M8A.FTN
CALLED FROM MODULE(S):	NONE
CALLS MODULE(S):	PARREV (subroutine in M8A.FTN)
FILES USED:	M8PAR., Simulation Data Files for reruns M8TAB., Particle size distribution file
LOGICAL UNITS:	LU1 - Console (write) LU2 - Console (read) LU3 - M8PAR. LU8 - (reads existing Simulation Data) LU9 - Printer
PURPOSE OF MODULE:	User interface program for setting up simulation run parameters and storing selected parameters in M8PAR. Interactive only.

FORTRAN Filename:	M8A.FTN
MODULE INTERNAL NAME:	PARREV (subroutine in M8A.FTN)
CALLED FROM MODULE(S):	M8A.FTN
CALLS MODULE(S):	NONE
FILES USED:	M8TAB.
LOGICAL UNITS:	LU1 - Console (write) LU9 - Printer
PURPOSE OF MODULE:	Provides a formatted display of user selected parameters to the user's terminal or printer prior to saving the parameter data to M8PAR.

A.2 M8SIM (Library)

FORTRAN Filename:	M8B.FTN (MAIN PROGRAM)
CALLED FROM MODULE(S):	NONE
CALLS MODULE(S):	M8UNI.FTN, M8T.FTN, M8G.FTN, M8P.FTN, M8V.FTN, M8SAV.FTN, M8CTI.FTN, M8HA.FTN
FILES USED:	M8PAR., Simulation Data File (rerun), Simulation Data File (new).
LOGICAL UNITS:	LU1 - Console (write) LU2 - Console (read - if used) LU3 - M8PAR. (parameter file) LU4 - M8COLLSN.DAT (collision index) LU7 - Simulation Data File (rerun) LU8 - Simulation Data File (new) LU9 - Printer LU10 - M8TAB. (distribution table) (assignment made in M8T.FTN)
PURPOSE OF MODULE:	Main routine (root) for the simulation program. Initializes variables, opens and reads parameter files, controls repetitions of hydration cycles and checks for multiple simulation runs until all have completed. Calls particle generation, placement, hydration and data saving routines.

FORTRAN Filename:	M8G.FTN
CALLED FROM MODULE(S):	M8B.FTN
CALLS MODULE(S):	NONE
FILES USED:	NONE
LOGICAL UNITS:	NONE
PURPOSE OF MODULE:	Generates Weibull distribution for particle radii with adjustable size, scale and location parameters.

FORTRAN Filename:	M8T.FTN
CALLED FROM MODULE(S):	M8B.FTN
CALLS MODULE(S):	M8UNI.FTN
FILES USED:	M8TAB.
LOGICAL UNITS:	LU10 - M8TAB (distribution table)
PURPOSE OF MODULE:	Reads distribution table and generates particle radii within the specified range as random values within series of ranges or as the mean value within the series of ranges.

FORTRAN Filename:	M8P.FTN
CALLED FROM MODULE(S):	M8B.FTN
CALLS MODULE(S):	M8R.FTN, M8SRT.FTN, M8UNI.FTN
FILES USED:	NONE
LOGICAL UNITS:	NONE
PURPOSE OF MODULE:	Places the C3S particles and the CH crystals in the defined 3D system box according to the placement methods specified in the parameter file. Placements are either random or one of three levels of clustering. Seed assignments and random or sorted subparticle assignments are done for clustered runs. Overlap testing is performed for each particle to assure valid positioning. Random vectors are used for clustering.

FORTRAN Filename:	M8CT.FTN
CALLED FROM MODULE(S):	M8HA.FTN, M8HB.FTN
CALLS MODULE(S):	NONE
FILES USED:	NONE
LOGICAL UNITS:	NONE
PURPOSE OF MODULE:	Checks for collisions after CS or CH radii have been increased. The new radius is used to test for collisions or overlaps with the particles listed in the current particle's collision indexing. Uses CT1 & CT2. Surface areas involved in contacts with other particles are excluded from growth functions and the level of engulfing is tested and determines the method of computation used. Running totals are kept for system wide parameters such as occupied volumes and available surface areas.

FORTRAN Filename:	M8CTI.FTN
CALLED FROM MODULE(S):	M8B.FTN
CALLS MODULE(S):	NONE
FILES USED:	NONE
LOGICAL UNITS:	NONE
PURPOSE OF MODULE:	Initialize the four collision tables and stores an index and collision counter in the record of each particle. The data in the arrays CT1, CT2, CT3 & CT4 are the record numbers of the particles with which a collision is predicted based upon the "projected" radius of the particles. This radius is the initial radius multiplied by the PRF (projected radius factor). All record numbers are initially set to negative numbers to indicate that no collision has occurred as yet.

FORTRAN Filename:	M8HA.FTN
CALLED FROM MODULE(S):	M8B.FTN
CALLS MODULE(S):	M8CT.FTN, M8HB.FTN
FILES USED:	NONE
LOGICAL UNITS:	NONE
PURPOSE OF MODULE:	Calculates the new hydration radii for each particle for the current hydration cycle.

FORTRAN Filename:	M8SAV.FTN
CALLED FROM MODULE(S):	M8B.FTN
CALLS MODULE(S):	NONE
FILES USED:	M8COLLSN.DAT, Simulation Data File (new)
LOGICAL UNITS:	LU8 - Simulation Data File (new)
PURPOSE OF MODULE:	Saves the simulation data, consisting of system variables, C3S/CSH data, and CH data, to disk files after the particles are first placed and after each hydration cycle is completed.

FORTRAN Filename:	M8UNI.FTN
CALLED FROM MODULE(S):	M8B.FTN, M8T.FTN, M8P.FTN, M8R.FTN
CALLS MODULE(S):	NONE
FILES USED:	NONE
LOGICAL UNITS:	NONE
PURPOSE OF MODULE:	Function subroutine - generates random number sequence with a long cycle. This module must be adjusted to match the bit size of the computer (minimum is 16 bits). The routine is initialized by the main program M8B.FTN. This module was imported from UNI, developed in the Center for Applied Mathematics at NIST and the authors are listed in the routine's comments.

FORTRAN Filename:	M8SRT.FTN
CALLED FROM MODULE(S):	M8P.FTN
CALLS MODULE(S):	NONE
FILES USED:	NONE
LOGICAL UNITS:	NONE
PURPOSE OF MODULE:	Sort routine to place the particle radii in descending order of size prior to seed assignments and placement.

FORTRAN Filename:	M8R.FTN
CALLED FROM MODULE(S):	M8P.FTN
CALLS MODULE(S):	M8UNI.FTN
FILES USED:	NONE
LOGICAL UNITS:	NONE
PURPOSE OF MODULE:	Produces a randomized list of subparticles for "random" assignments to seeds from a presorted array. The Weibull generated particles are generated in sorted order and the distribution table generated particles are sorted because some operations require the particles to be in descending order of size.

FORTRAN Filename:	M8HB.FTN
CALLED FROM MODULE(S):	M8HA.FTN
CALLS MODULE(S):	M8CT.FTN
FILES USED:	NONE
LOGICAL UNITS:	NONE
PURPOSE OF MODULE:	Calculates the growth of CH based on the volume of C3S hydrated in M8HA.FTN.

FORTRAN Filename:	M8V.FTN
CALLED FROM MODULE(S):	M8B.FTN
CALLS MODULE(S):	M8EV.FTN
FILES USED:	NONE
LOGICAL UNITS:	NONE
PURPOSE OF MODULE:	Calculates volume of pores by subtracting corrected (non-overlapped and not extending out of box) particle volumes from the volume of the system box.

FORTRAN Filename:	M8EV.FTN
CALLED FROM MODULE(S):	M8V.FTN
CALLS MODULE(S):	NONE
FILES USED:	NONE
LOGICAL UNITS:	NONE
PURPOSE OF MODULE:	Calculates the volume of a particle extending beyond the system box. Three modes of calculating exclusion volumes are provided, planar, edge and corner. The exclusions seem simple at first, but varying positions create more complex calculations, especially in the corner calculations.

A.3 M8GRF (Library)

FORTRAN Filename:	M8GRF.FTN (MAIN PROGRAM)
CALLED FROM MODULE(S):	NONE
CALLS MODULE(S):	TEMPLATE, M8COLOR.FTN, M8G2SORT.FTN, M8GTEXT.FTN, and M8GR3D.FTN
FILES USED:	Simulation Data File (user selected)
LOGICAL UNITS:	LU1 - Console (write) LU2 - Console (read) LU4 - TEMPLATE Configuration File LU5 - NULL: LU6 - NULL: LU8 - Simulation Data File LU11 - TEMPLATE Font File LU15 - Console (TEMPLATE i/o) LU16 - Console (TEMPLATE i/o)
PURPOSE OF MODULE:	Main routine for the graphics program. Primary interactive user interface for data file selection, data set (hydration cycle) positioning.

FORTRAN Filename:	M8G2SORT.FTN
CALLED FROM MODULE(S):	M8GRF.FTN
CALLS MODULE(S):	NONE
FILES USED:	NONE
LOGICAL UNITS:	NONE
PURPOSE OF MODULE:	Produces and index of sorted particle locations for 2D non-sliced/solid filled graphics sequencing. TEMPLATE does not have hidden line removal, so particles are drawn in overlay fashion, from background (Z=0) to foreground.

FORTRAN Filename:	M8COLOR.FTN
CALLED FROM MODULE(S):	M8GRF.FTN
CALLS MODULE(S):	NONE
FILES USED:	NONE
LOGICAL UNITS:	NONE
PURPOSE OF MODULE:	User interface for setting color variables. Two methods are provided, default color sets or selection of individual colors for each supported feature. Three default sets are provided: Tek 4115 color, Tek 4115 gray-tones and Tek 4014 monochrome.

FORTRAN Filename:	M8GR3D.FTN
CALLED FROM MODULE(S):	M8GRF.FTN
CALLS MODULE(S):	M8G3SORT.FTN, M8GTEXT.FTN, M8BOX.FTN, and M8SPHERE.FTN
FILES USED:	NONE
LOGICAL UNITS:	NONE
PURPOSE OF MODULE:	Produces 3D wire-frame graphics of the maximum C3S/CSH radius for each particle. CH particles were excluded to minimize the complexity of the 3D image. It is possible to add the CH to the graphics, but with large numbers of particles the screen image becomes difficult to interpret and takes a great deal of time to generate. A menu allows the user to change the 3D window and view vector parameters interactively, rotate in 90 or 180 degree increments (requires redrawing), number of "wires" used to form each of the particles (longitude & latitude), set 3D clipping on or off, and reset to default 3D graphics values.

FORTRAN Filename:	M8GTEXT.FTN
CALLED FROM MODULE(S):	M8GRF, GR3D
CALLS MODULE(S):	NONE
FILES USED:	NONE
LOGICAL UNITS:	NONE
PURPOSE OF MODULE:	Produces the legend window to the left of the graphics window for 2D and 3D graphics displays. Shows data filename, hydration cycle number, number of particles, material densities, percent of C3S consumed, and current values of other variables of interest.

FORTRAN Filename:	M8G3SORT.FTN
CALLED FROM MODULE(S):	GR3D
CALLS MODULE(S):	NONE
FILES USED:	NONE
LOGICAL UNITS:	NONE
PURPOSE OF MODULE:	Produces an index of 3D sorted particles in descending order in reference to the current view point coordinates, so that the 3D graphics will produce an image of overlaid particles from background to foreground. This is required as TEMPLATE does not provide for hidden line removal.

FORTRAN Filename:	M8BOX.FTN
CALLED FROM MODULE(S):	GR3D
CALLS MODULE(S):	NONE
FILES USED:	NONE
LOGICAL UNITS:	NONE
PURPOSE OF MODULE:	Draws a 3D box, of the dimensions specified by the Simulation Data File, using the current 3D window and view vector values. X, Y & Z labels are displayed at the positive end of each axis.

FORTRAN Filename:	M8SPHERE.FTN
CALLED FROM MODULE(S):	GR3D
CALLS MODULE(S):	NONE
FILES USED:	NONE
LOGICAL UNITS:	NONE
PURPOSE OF MODULE:	Produces 3D wire-frame spheres with longitudinal and latitudinal lines, using current radius, number of wires, and line color values.

FORTRAN Filename:	M8GR2D.FTN
CALLED FROM MODULE(S):	M8GRF.FTN
CALLS MODULE(S):	M8GTEXT.FTN
FILES USED:	NONE
LOGICAL UNITS:	NONE
PURPOSE OF MODULE:	Produces 2D graphics in one of several modes.

A.4 Other Files

Task Files:

M8A.TSK
M8B.TSK
M8GRF.TSK

Library Files:

M8SIM.LIB
M8GRF.LIB

Data Files:

M8PAR.	-	set-up parameters
M8TAB.	-	particle size distribution table
fn.CM8	-	data file (Appendix D)

Command Files:

RUNSET8	-	starts set-up program M8A
RUNSIM8	-	starts simulation run in the interactive session
BATSIM8	-	submits simulation run in batch mode
RUNGRF	-	starts graphics program for Tektronix 4129

Batch File:

RUNM8.BAT	-	runs simulation in batch mode

Log Files:

SET8.LOG	-	log file for set-up run
SIM8.LOG	-	log file for simulation run

Configuration File:

CONFIG.TEK	-	configuration file for TEMPLATE

Test Files:

TEST1.CM8	-	data file for test 1 (Appendix G)
TEST2.CM8	-	data file for test 2 (Appendix G)
TEST3.CM8	-	data file for test 3 (Appendix G)
TEST4.CM8	-	data file for test 4 (Appendix G)
TEST5.CM8	-	data file for test 5 (Appendix G)
TEST6.CM8	-	data file for test 6 (Appendix G)
TEST1.TAB	-	particle size distribution file for test 1 (Appendix 6)
TEST2.TAB	-	particle size distribution file for test 2 (Appendix 6)

APPENDIX B. FLOW DIAGRAMS

Flow diagrams for the overall simulation model and for various subprograms in the model are shown in figures B-1 through B-11.

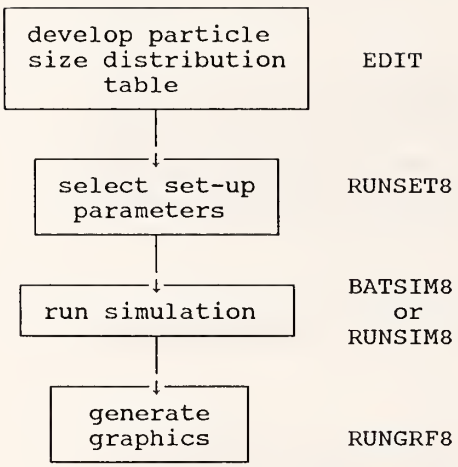


Figure B-1. Entire simulation model.

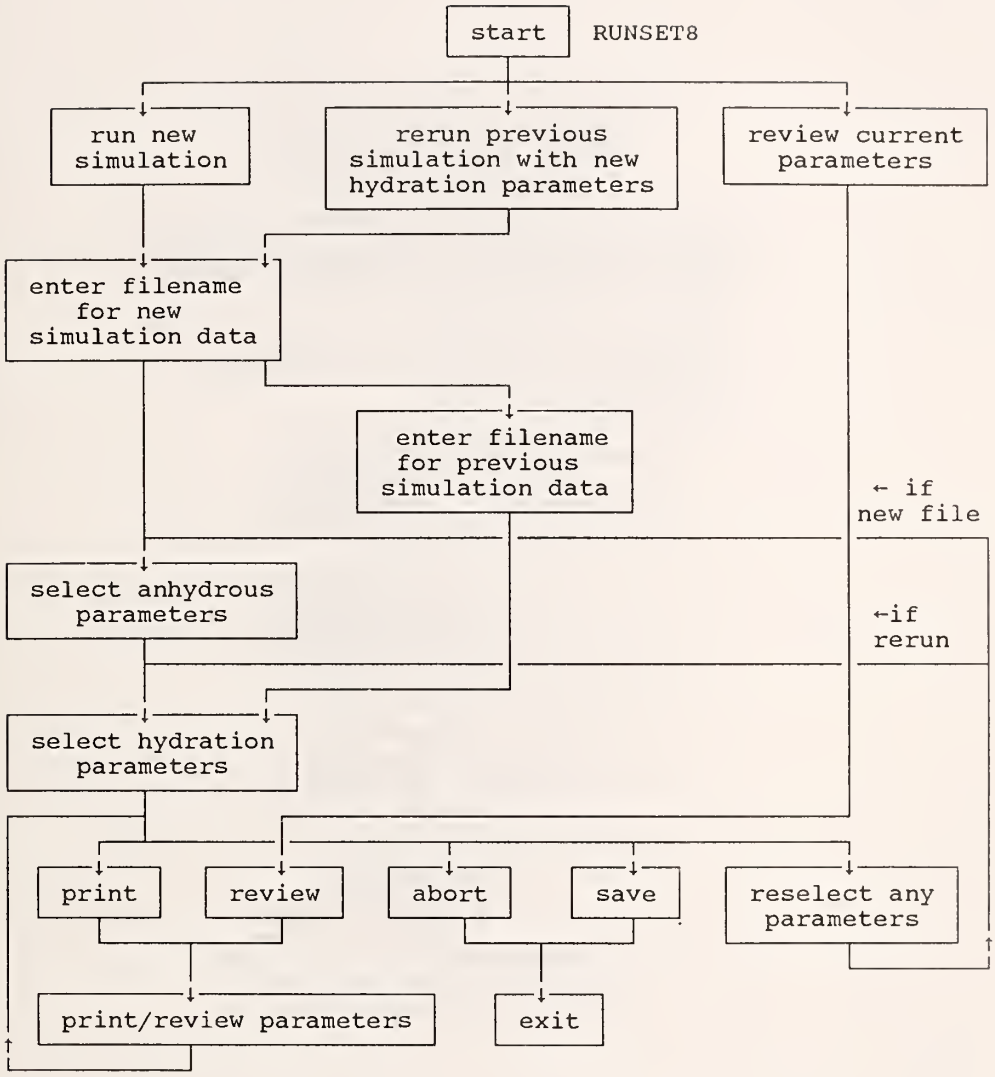


Figure B-2. Set-up program.

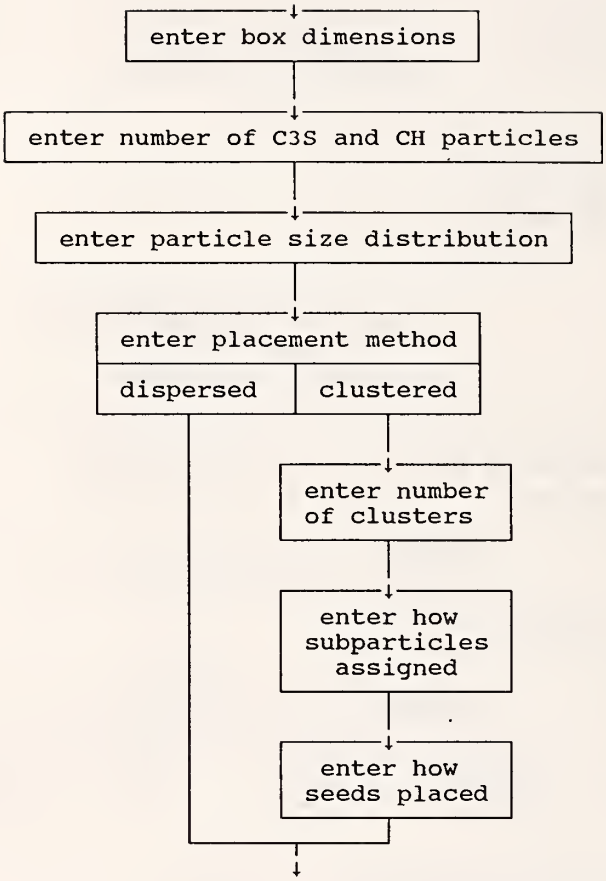


Figure B-3. Anhydrous set-up parameters.

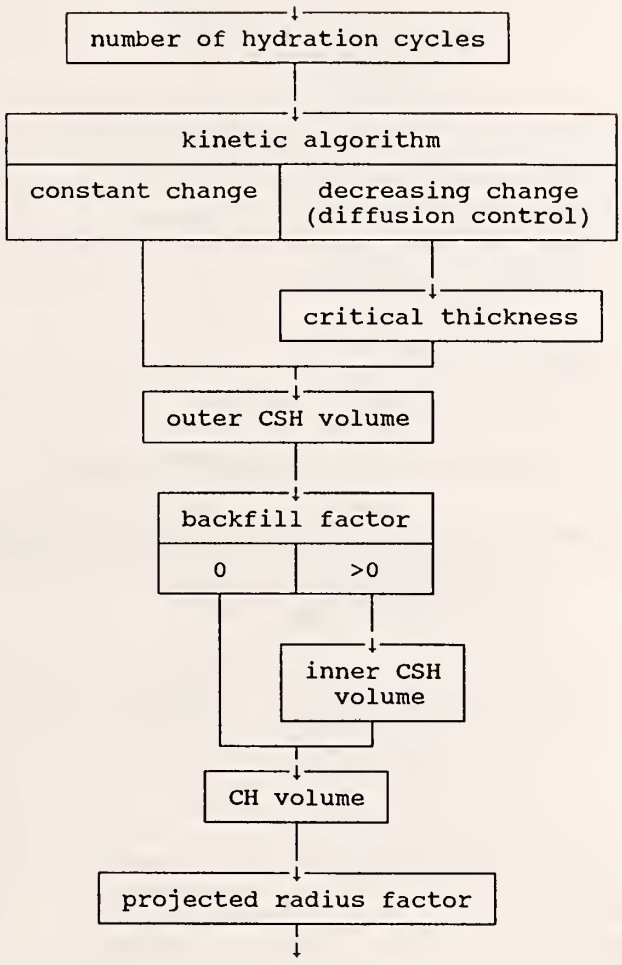


Figure B-4. Hydration set-up parameters.

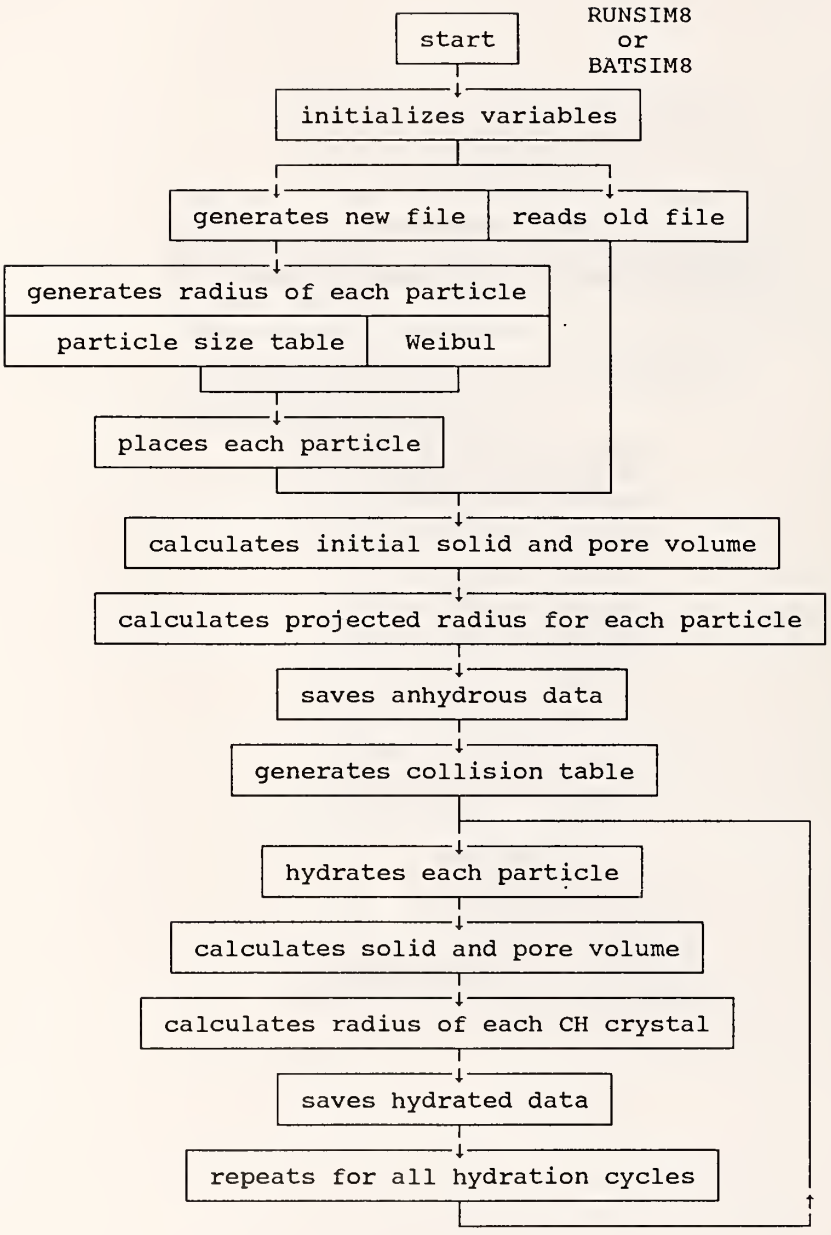


Figure B-5. Simulation program.

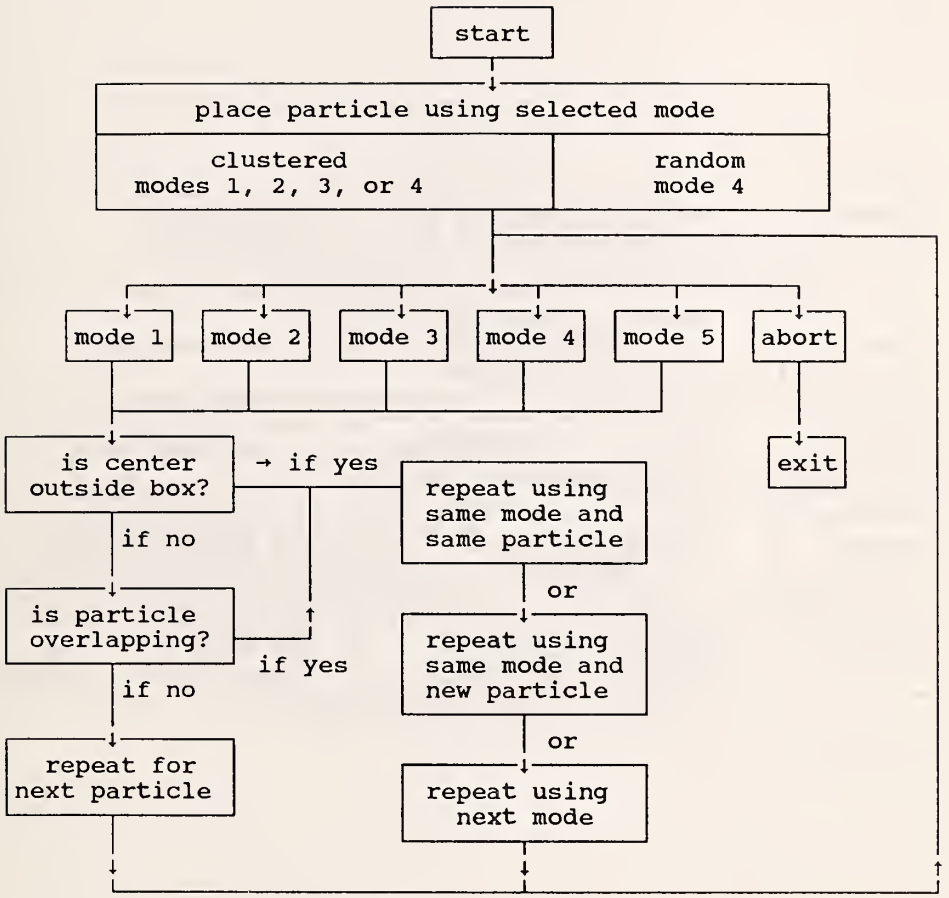


Figure B-6. C_3S particle placement.

mode 1	particle clustered on previous seed	C3S seed or subparticle	
mode 2	particle clustered on subparticle of previous seed	C3S seed only	
mode 3	particle clustered on any previous particle	C3S seed only	
mode 4	particle placed randomly	C3S seed or subparticle	CH nucleus
mode 5	particle placed in available empty space	C3S seed or subparticle	CH nucleus

Figure B-7. Modes for particle placement.

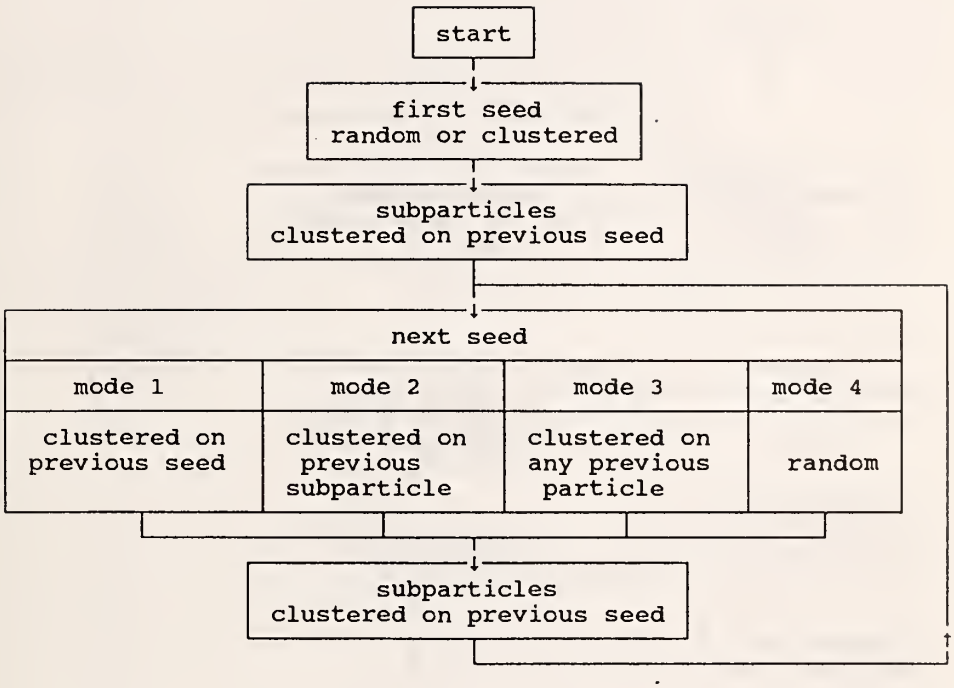


Figure B-8. Targeted placement methods in clustered placement of C_3S particles.

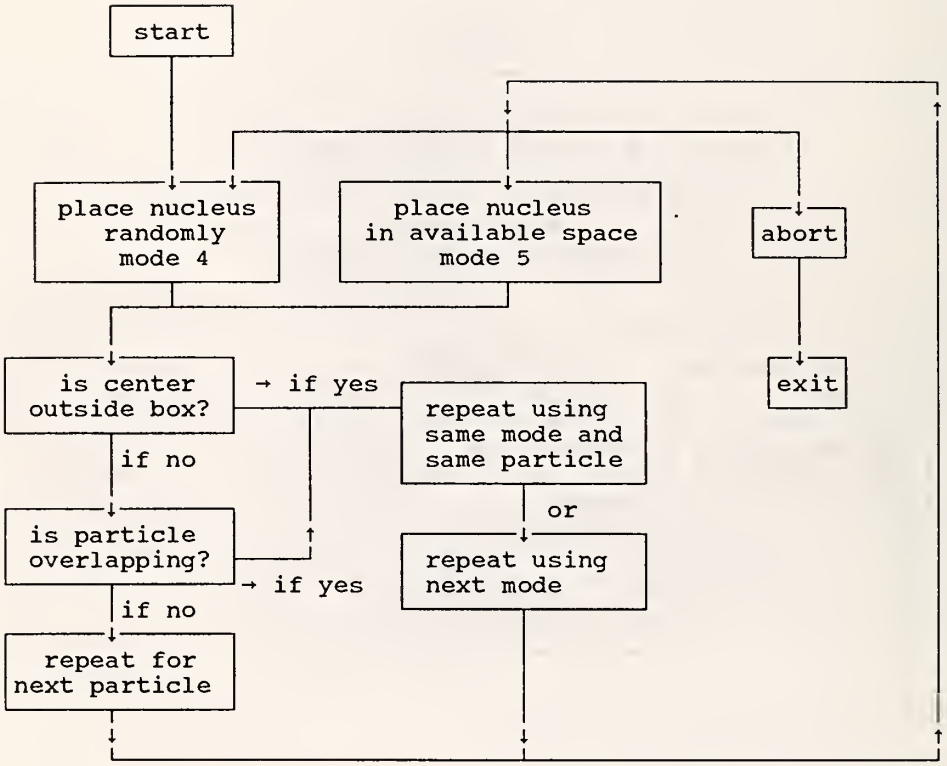


Figure B-9. CH nucleus placement.

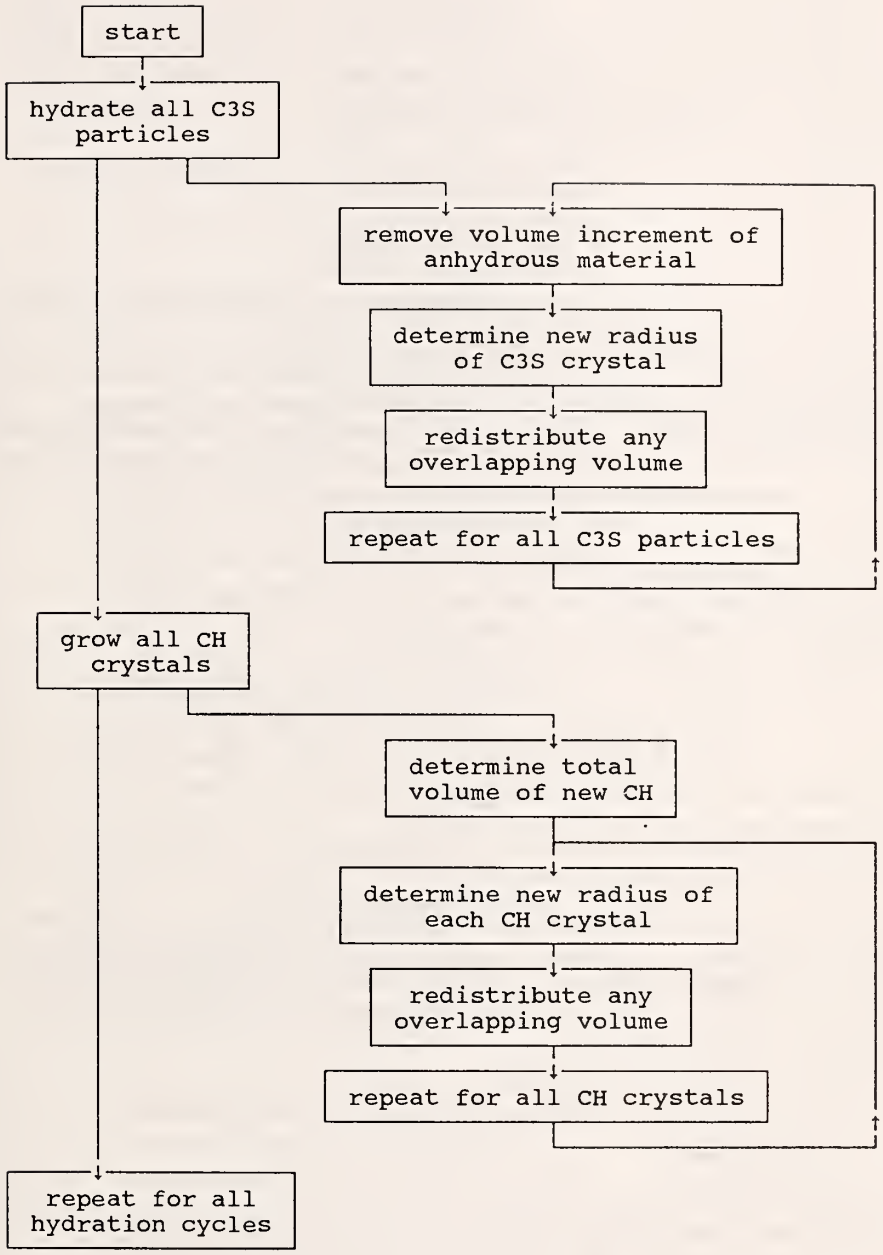


Figure B-10. Hydration.

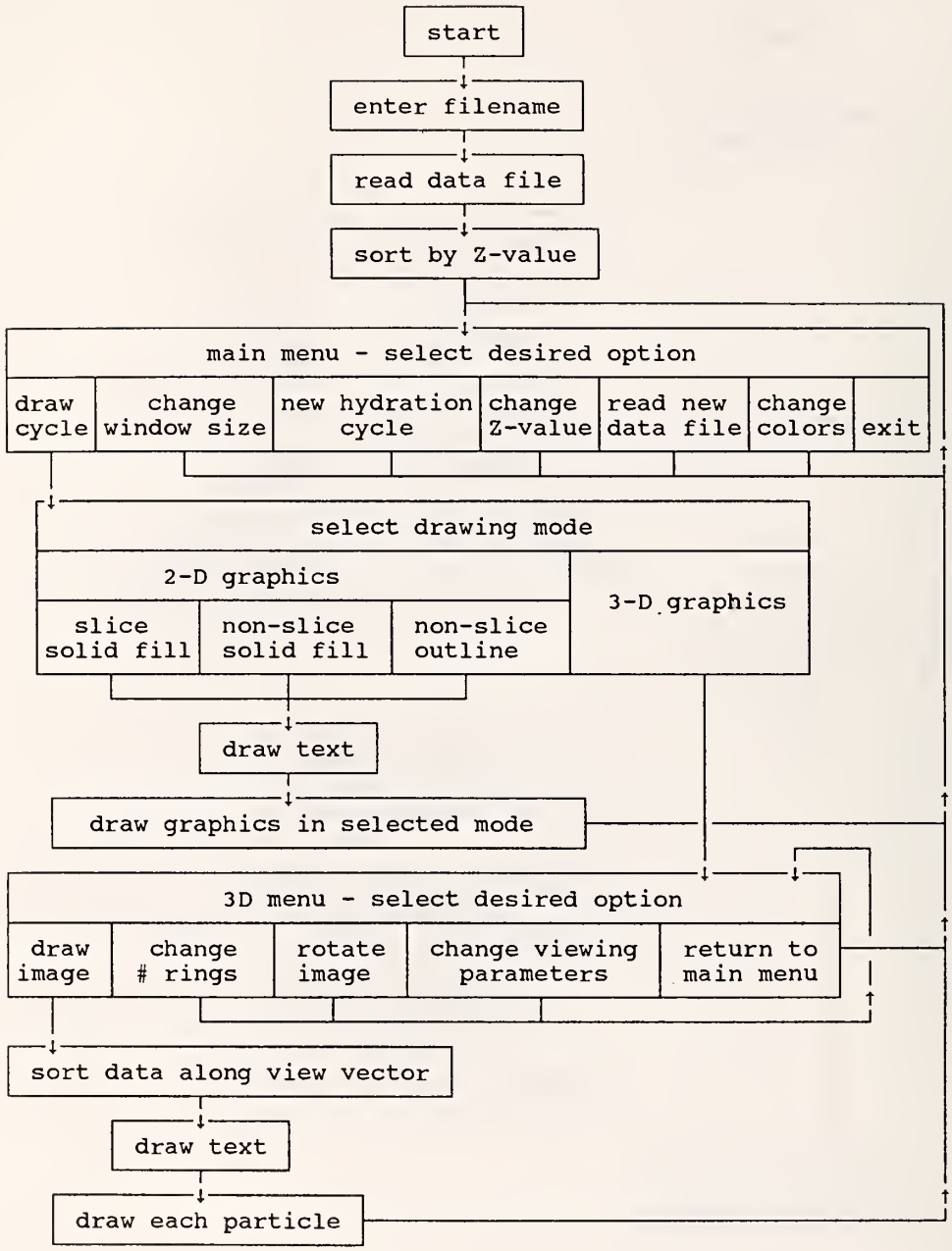


Figure B-11. Graphics display.

APPENDIX C. LIST OF VARIABLES

AN — user response for menu selection
AN1 — user response for terminal type
AN3 — user response for skip selection
AN4 — user response for program termination
AN5 — user response for 3D graphics menu selection

BF — backfill factor (0 to 1.0)

1.0 for no hollow-shell grain

CHI(N,J) — integer CH variables for particle J

CHI(1,J) — hydration code for hydrated CH (0 to 4)

0 hydrating actively with no collisions,
1 hydrating actively with collisions
2 inactive because no remaining surface area
3 inactive because engulfed
4 inactive because fully hydrated
5 inactive because projected radius exceeded

CHI(2,J) — CT3 line for anhydrous CH
CHI(3,J) — CT3 number for anhydrous CH
CHI(4,J) — CT4 line for anhydrous CH
CHI(5,J) — CT4 number for anhydrous CH
CHI(6,J) — placement code for anhydrous CH (1)

1 randomly placed particle

CHN — new CH effective radius
CHP — previous CH radius
CHPR — projected radius of CH crystals for placement
CHR(N,J) — real CH variables for particle J

CHR(1,J) — radius before cycle for hydrated CH
CHR(2,J) — x-coordinate for anhydrous CH
CHR(3,J) — y-coordinate for anhydrous CH
CHR(4,J) — z-coordinate for anhydrous CH
CHR(5,J) — projected radius for anhydrous CH
CHR(6,J) — radius after cycle for hydrated CH
CHR(7,J) — excluded volume for hydrated CH
CHR(8,J) — redistributed volume for hydrated CH

CI — collision index (1 to 4)

1 CSH-CSH
2 CSH-CH
3 CH-CH
4 CH-CSH

CLR color used in TEMPLATE subprogram
CM1 color for C_3S seed
CM2 color for C_3S subparticles
CM3 color for CSH from seed
CM4 color for CSH from subparticle
CM5 color for active CH crystals
CM6 color for inactive CH crystals
CM7 color for voids
COL1 x-coordinate for primary particle in collision
COL2 y-coordinate for primary particle in collision
COL3 z-coordinate for primary particle in collision
COL4 radius (before cycle) for primary particle in collision
COL5 x-coordinate for secondary particle in collision
COL6 y-coordinate for secondary particle in collision
COL7 z-coordinate for secondary particle in collision
COL8 radius (before cycle) for secondary particle in collision
CSA temporary anhydrous radius
CSI(N,J) integer C-S variables for particle J

CSI(1,J) hydration code for hydrated C3S (0 to 5)

0 hydrating actively with no collisions
1 hydrating actively with collisions
2 inactive because no remaining surface area
3 inactive because engulfed
4 inactive because fully hydrated
5 inactive because critical thickness reached
6 inactive because projected radius exceeded

CSI(2,J) seed number for anhydrous C3S
CSI(3,J) CT1 line for anhydrous C3S
CSI(4,J) CT1 number for anhydrous C3S
CSI(5,J) CT2 line for anhydrous C3S
CSI(6,J) CT2 number for anhydrous C3S
CSI(7,J) placement code for anhydrous C3S (0 to 5)

0 all clustered subparticles or for clustered seed placed in center of box
1 clustered seed clustered on previous seed
2 clustered seed clustered on subparticle of previous seed
3 clustered seed clustered on any previously placed particle
4 randomly placed particle (clustered or dispersed)
5 placement using Voronoi algorithm (clustered or dispersed)

CSN temporary hydrated radius for CS or CH
CSR CSH radius of second particle in collision
CSR(N,J) real C-S variables for particle J

CSR(1,J) anhydrous radius of anhydrous C3S
CSR(2,J) anhydrous radius after cycle for hydrated C3S
CSR(3,J) outer radius before cycle for hydrated C3S
CSR(4,J) inner radius for hydrated C3S

CSR(5,J)	x-coordinate for anhydrous C3S
CSR(6,J)	y-coordinate for anhydrous C3S
CSR(7,J)	z-coordinate for anhydrous C3S
CSR(8,J)	open
CSR(9,J)	excluded volume for hydrated C3S
CSR(10,J)	projected radius for anhydrous C3S
CSR(11,J)	outer radius after cycle for hydrated C3S
CSR(12,J)	redistributed volume for hydrated C3S
CSR(13,J)	group code for anhydrous C3S for sorting

CT1(J)	collision data, CSH/CSH; for particle I, -1*J where J is each potential collision particle
CT2(J)	collision data, CSH/CH; for particle I, -1*J where J is each potential collision particle
CT3(J)	collision data, CH/CH; for particle I, -1*J where J is each potential collision particle
CT4(J)	collision data, CH/CSH; for particle I, -1*J where J is each potential collision particle
CV	total volume of CH produced during each hydration cycle
CX	character variable to dump
D	distance between (x,y,z) coordinates of particles in collision
DAI	relative volume of inner product CSH
DAO	relative volume of outer product CSH
DB	relative volume of CH
DC	dash line color
DIB	distance from the center of primary particle to intercept plane
DIP	distance from the center of secondary particle to intercept plane
DQ	squared value of D
DX	delta x
DY	delta y
DZ	delta Z
EX	user response to exit query
FN	filename of user data file
FN2	filename of existing file for rerun
FNL	number characters for FN without extension
FNL1	position of "." in filename
FNL4	number of characters for FN with .CM8 extension
FS	percent porosity
HD	change in radius for initial hydration cycle (0.01 to 1.0)
HDA	height of dome of primary particle in collision
HDAI	height of dome of primary particle using radius before current hydration cycle
HDAO	height of dome of primary particle using radius after current hydration cycle
HDB	height of dome of secondary particle in collision
HDX	function value for change in radius of anhydrous material during hydration cycle
HN	hydration cycle number

HX height of excluded dome (x)
HY height of excluded dome (y)
HZ height of excluded dome (z)

I counter
IA index of C_3S particle number during hydration
IB index of CH crystal number during hydration
ICNT index (CT1 number) for primary particle in collision
ICOL1 counter (CTN(K)) for primary particle in collision
ICOL2 hydration code for primary particle in collision
ICOL3 hydration code for secondary particle in collision
INDX index (CT1 line) for primary particle in collision
IX integer variable to dump

J counter
JD random number generator seed (0 or SD)

K particle counter
KIN index for kinetic algorithm (0 or 1)

0 for no kinetic algorithm
1 for diffusion algorithm developed by Jennings and Johnson)

LN line number for text lines in graphics output

MODE in M8TAB, mode for intervals (0 for random and 1 for calculated mean value); in graphics program, drawing mode (slice, nonslice, etc.)

N counter
NA number of C3S particles simulated
NAG number of C_3S particles actively hydrating
NB number of CH particles simulated
NBG number of CH particles actively hydrating
NC1 number of collisions of type 1
NC2 number of collisions of type 2
NC3 number of collisions of type 3
NC4 number of collisions of type 4
NDX2 two-dimensional sorted particle index
NDX3 three-dimensional sorted particle index
NH number of hydration cycles requested
NHS number of hydrations selected for skip
NR number of repeats in placing particles
NRN randomly selected radius
NSD number of cluster seeds
NSP number of subparticles per seed
NSP1 NSP + 1
NSX maximum number for NSD
NXREC record number pointer for FN

P proportion in current particle size range in M8TAB.
P3 PI/3

P43 P3*4 or (4/3)*PI
PA percent of CSH growing
PB percent of CH growing
PCHR projected radius of CH particles for hydration
PD particle distribution method (1 to 3)

1 default Weibull
2 selected Weibull parameters
3 particle distribution table)

PFS placement method for first seed in box (1 or 2)

1 centered
2 random

PGC previous subparticle's group code in M8P.FTN
PI π (3.14159)
PM particle placement method (1 for random, 2 for clustered)
PR percent C_3S reacted
PRF projected radius factor
PRS placement method for remaining seeds (1 to 4)

1 previous seed
2 previous seed's subparticles
3 any seed or subparticle
4 random

PRV percent volume redistributed
PX particle volume exclusion dimension counter

R hydrated radius of current particle
RAB combined radii for collision test
RAV average of radii before and after present cycle (CS or CH)
RCH current (previous) radii
RN random number from M8UNI.FTN for generating particles
RO combined radii for collision test in M8P.FTN
ROD radius of domes (A and B) (not currently used)
RR rerun option (1 to 3)

1 new simulation
2 rerun of existing simulation
3 review current simulation parameters

RX real variable to dump

SAB user option at end of set-up program
SAF surface area free to hydrate (CS or CH)
SAX surface area excluded by overlaps
SCH sum of CH volume for each hydration cycle
SD random number generator seed

SPA		subparticle assignment method (1 or 2)
	1	random non-sorted
	2	round-robin sorted
SS		percent solids
SSD		system space dimension method (1 to 3)
	1	default
	2	0-100,0-100,0-2RMAX*ZR, where ZR is defined by user,
	3	min and max for x,y,z defined by user
SUR		total surface area before redistributions
SVP		for each particle (CS or CH), sum of particle volumes extending beyond system boundaries
SVR		for each particle (CS or CH), sum for all contacts of volumes to be redistributed
TBF		total volume of backfilled reaction product
TC		critical thickness parameter
TCH		sum of CH volume for all cycles
TCS		sum CSH volume in system
TERM		code to note that particle or crystal has exceeded projected radius and simulation is to be terminated
THK		CSH layer thickness value
TOT		total system volume
TRC		total sum of CH volumes to be redistributed
TRV		total sum of C-S-H volumes to be redistributed
TSA		sum of CS surface area free for reaction
TSS		total system solids
TVA		total volume of C3S not yet hydrated (initial = TVI)
TVI		total initial C3S volume
VA		C-S-H volume to be added to exterior during cycle
VAD		volume to add to each CH crystal still growing
VBF		actual volume added to the interior as backfill
VC		volume of anhydrous material consumed during hydration of current particle
VDA		volume of dome of primary particle in collision
VDAI		volume of dome of primary particle in collision using radius before current hydration cycle
VDAO		volume of dome of primary particle in collision using radius after current hydration cycle
VDB		volume of dome of secondary particle in collision
VH		previous total volume of CS particle (equivalent to a sphere of radius equal to the outer hydration radius)
VI		portion of volume consumed to be added to the interior
VNW		volume of CH after addition
VP		current particle volume extending beyond system boundaries
VPR		volume of previous CH crystals
VR		volume of primary particle to be redistributed due to collision

VT	new total volume of CS particle (equivalent to a sphere of radius equal to the outer hydration radius)
VX	volume of excluded dome (x)
VY	volume of excluded dome (y)
VZ	volume of excluded dome (z)
WCR	water-cement ratio (by weight) assuming density of 3.2 for cement
WPA	Weibull distribution parameter
WPB	Weibull distribution parameter
WPC	Weibull distribution parameter
WXL	minimum x-value in viewing window
WXM	maximum x-value in viewing window
WYL	minimum y-value in viewing window
WYM	maximum y-value in viewing window
WZL	minimum z-value in viewing window
WZM	maximum z-value in viewing window
X	x-coordinate for current particle
XMAX	maximum value for x-coordinate
XMIN	minimum value for x-coordinate
XMR	x-coordinate minus hydrated radius
XPR	x-coordinate plus hydrated radius
Y	y-coordinate for current particle
YMAX	maximum value for y-coordinate
YMIN	minimum value for y-coordinate
YMR	y-coordinate minus hydrated radius
YPR	y-coordinate plus hydrated radius
Z	z-coordinate for current particle
ZMAX	maximum value for z-coordinate
ZMIN	minimum value for z-coordinate
ZMR	z-coordinate minus hydrated radius
ZP	Z value of X-Y plane
ZPR	z-coordinate plus hydrated radius
ZR	multiplier of RMAX for alternate z-range

APPENDIX D. STRUCTURE OF DATA FILES

D.1 Parameter File

The parameter file (M8PAR.) contains the set-up parameters and is generated by the set-up program. The file consists of 6 lines with the following data:

Line 1
- filename (FN)
- filename of existing file for rerun (FN2)
- random number generator seed (SD)

Line 2
- index for kinetic algorithm (KIN)
- number of C_3S particles simulated (NA)
- number of CH particles simulated (NB)
- number of hydration cycles requested (NH)
- number of cluster seeds (NSD)
- number of subparticles per seed (NSP)

Line 3
- particle distribution method (PD)
- placement method for first seed in volume (PFS)
- particle placement method (PM)
- placement method for remaining seeds (PRS)
- rerun option (RR)
- subparticle assignment method (SPA)

Line 4
- system space dimension method (SSD)
- Weibull distribution parameter (WPA)
- Weibull distribution parameter (WPB)
- Weibull distribution parameter (WPC)
- relative volume of inner product C-S-H (DAI)
- backfill factor (BF)

Line 5
- relative volume of outer product C-S-H (DAO)
- relative volume of CH (DB)
- change in radius for initial hydration cycle (HD)
- projected radius factor (PRF)
- critical thickness parameter (TC)
- multiplier of RMAX for alternate z-range (ZR)

Line 6
- minimum value for x-coordinate (XMIN)
- maximum value for x-coordinate (XMAX)
- minimum value for y-coordinate (YMIN)
- maximum value for y-coordinate (YMAX)
- minimum value for z-coordinate (ZMIN)
- maximum value for z-coordinate (ZMAX)

D.2 Particle Size Distribution File

The particle size distribution file (M8TAB.) contains a size distribution in the form of cumulative proportion finer than the size increment. The

file consists of two lines, followed by columns containing the size distribution data.

Line 1 label (to identify source of data)

Line 2 mode (MODE) for radius within interval

0 for random value
1 for calculated mean value

The next set is particle size data, with the following columns:

lower size (radius, microns) of interval (R1)
upper size (radius, microns) in interval (R2)
cumulative proportion finer than interval size (P)

D.3 Simulation Data File

The data file (fn.CM8) contains parameters of the simulation run, both before hydration (Cycle 0) and after each hydration cycle. The format for each cycle starting with 0 consists of the following data sets: the set-up parameters, calculated parameters for that cycle, data for each C_3S particle, and data for each CH crystal.

First are the initial parameters for anhydrous data:

Line 1
filename (FN)
filename of existing file for rerun (FN2)
random number generator seed (SD)

Line 2
index for kinetic algorithm (KIN)
number of C_3S particles simulated (NA)
number of CH particles simulated (NB)
number of hydration cycles requested (NH)
number of cluster seeds (NSD)
number of subparticles per seed (NSP)

Line 3
particle distribution method (PD)
placement method for first seed in volume (PFS)
particle placement method (PM)
placement method for remaining seeds (PRS)
rerun option (RR)
subparticle assignment method (SPA)

Line 4
system space dimension method (SSD)
Weibull distribution parameter (WPA)
Weibull distribution parameter (WPB)
Weibull distribution parameter (WPC)
relative volume of inner product C-S-H (DAI)
backfill factor (BF)

Line 5 relative volume of outer product C-S-H (DAO)
relative volume of CH (DB)
change in radius for initial hydration cycle (HD)
projected radius factor (PRF)
critical thickness parameter (TC)

Line 6 minimum value for x-coordinate (XMIN)
maximum value for x-coordinate (XMAX)
minimum value for y-coordinate (YMIN)
maximum value for y-coordinate (YMAX)
minimum value for z-coordinate (ZMIN)
maximum value for z-coordinate (ZMAX)

Line 7 percent porosity (FS)
sum of C3S surface area (TSA)
sum of C3S volume (TVI)
water-cement ratio (WCR).

The next set is anhydrous C_3S data, with the following columns:

particle number (J)
x-coordinate (CSR(5,J))
y-coordinate (CSR(6,J))
z-coordinate (CSR(7,J))
radius (CSR(1,J))
projected radius (CSR(10,J))
excluded volume (CSR(9,J))
placement code (CSI(7,J))
seed number (CSI(2,J))
group code (CSR(13,J))
CT1 line (CSI(3,J))
CT1 number (CSI(4,J))
CT2 line (CSI(5,J))
CT2 number (CSI(6,J)).

The next set is anhydrous CH data with the following columns:

particle number (J)
x-coordinate (CHR(2,J))
y-coordinate (CHR(3,J))
z-coordinate (CHR(4,J))
projected radius (CHR(5,J))
placement code (CHI(6,J))
CT3 line (CHI(2,J))
CT3 number (CHI(3,J))
CT4 line (CHI(4,J))
CT4 number (CHI(5,J)).

For every hydration cycle after 0, the following initial parameters are listed:

Line 1 hydration cycle number (HN)
number of C_3S particles actively hydrating (NAG)
number of CH particles actively hydrating (NBG)
number of collisions of type 1 (NC1)
number of collisions of type 2 (NC2)
number of collisions of type 3 (NC3)
number of collisions of type 4 (NC4)

Line 2 percent porosity (FS)
total volume of backfilled reaction product (TBF)
total sum of CH produced (TCH)
sum of C3S + CSH volume (TCS)
sum of C3S surface area prior to hydration cycle (TPP)

Line 3 total sum of CH volumes to be redistributed (TRC)
total sum of C-S-H volumes to be redistributed (TRV)
sum of C-S-H surface area free for reaction (TSA)
total sum of C-S-H volume produced (TVC)
remaining C3S volume (TVA).

The next set is hydrated C_3S data with the following columns:

particle number (J)
anhydrous radius (CSR(2,J))
inner shell radius (CSR(4,J))
outer radius after cycle (CSR(11,J))
hydration code (CSI(1,J))
excluded volume (CSR(9,J))
redistributed volume (CSR(12,J)).

The final set is hydrated CH data with the following columns:

particle number (J)
radius after cycle (CHR(6,J))
hydration code (CHI(1,J))
excluded volume (CHR(7,J))
redistributed volume (CHR(8,J)).

APPENDIX E. PARTICLE PLACEMENT USING VORONOI HEURISTIC

The Voronoi particle placement heuristic has been implemented in the simulation model to place C_3S particles that cannot successfully be placed randomly. The heuristic uses the so-called radical method to partition the volume into a collection of polyhedra, each containing a particle which has already been placed. The motivation behind the heuristic is that a new particle should be able to be placed near one of the vertices of these polyhedra.

E.1 Voronoi Regions and Variants

Given a collection $S = \{(x_i,y_i,z_i),\ i = 1,\ldots,n\}$ of points in $\mathbb{R}^3$, the Voronoi [15] region of the point $(x_i,y_i,z_i) \in S$ is the set of all points in $\mathbb{R}^3$ which are closer to (x_i,y_i,z_i) than any other point in S. The Voronoi region is convex, and it is not difficult to see that it is the polyhedron which is the set of points which satisfy the inequalities

$$(x_j - x_i)(x - x_i) + (y_j - y_i)(y - y_i) + (z_j - z_i)(z - z_i)$$

$$\leq \frac{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2}{2} \qquad \text{(E1)}$$

for all $j \neq i$. These may equivalently be written as

$$(x_j - x_i)x + (y_j - y_i)y + (z_j - z_i)z$$

$$\leq \frac{(x_j^2 + y_j^2 + z_j^2)}{2} \qquad \text{(E2)}$$

for all $j \neq i$. Since we are concerned with particles and not just points, we instead use the radical[9] method [16] because it takes particle radius into account. The "radical region" of a particle is defined to be the set of points whose tangential distance (fig. E-1) to the particle is smaller than the its tangential distance to any other particle. If r_i is the radius of particle j for $j = 1,\ldots,n$, then the radical region of particle i is the

[9]The radical plane of two spheres is the locus of points from which the tangent lengths to the two spheres are equal. For a picture, see [16].

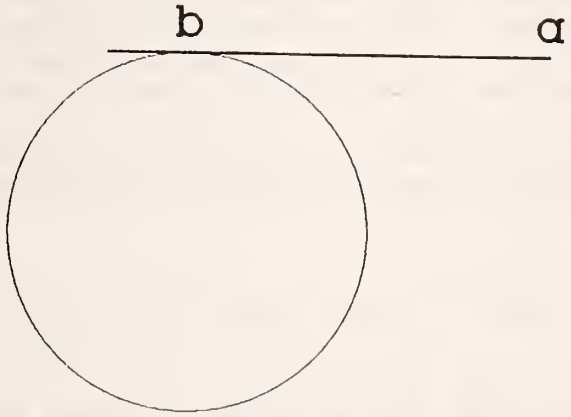


Figure E-1. Diagram showing the tangential distance: the distance along a tangent line from a point (a), which defines the radical region, to the point of tangency with the circle (b).

polyhedron that is the set of points which satisfy the inequalities

$$(x_j - x_i)x + (y_j - y_i)y + (z_j - z_i)z$$

$$\leq \frac{(x_j^2 + y_j^2 + z_j^2) - r_j^2 - (x_i^2 + y_i^2 + z_i^2) + r_i^2}{2} \qquad \text{(E3)}$$

for all $j \neq i$. These radical regions have the desirable property that they contain the particles with which they are associated. We obtain another generalization of the Voronoi regions if we instead say that the Voronoi region of a particle is the set of points which are closer to that particle than any other particle. These regions will no longer be polyhedra, and in fact need not be convex. The three methods are illustrated in figure E-2 (for a set of points in $\mathbb{R}^2$). We also note that if all particles have the same radius, then the methods all generate the same regions.

The complexity of determining the Voronoi regions of a set of n points is well known. If we are concerned with the intersection of each region with the box

$$\{(x,y,z) : x_{min} \leq x \leq x_{max},\ y_{min} \leq y \leq y_{max},\ z_{min} \leq z \leq z_{max}\}$$

then there is an algorithm with an expected running time O(n) [17] for n particles uniformly distributed over the box. The algorithm first partitions the box into a number of smaller boxes. For each point it then searches the neighboring boxes in a spiral until it determines that all neighbors have been found. It must then determine the vertices of the polyhedron determined by the inequalities associated with these m neighbors. This can be done using an algorithm whose worst case running time is O(mlogm) [18]. A similar approach may be used to determine the radical regions. The Voronoi regions for particles are somewhat more difficult to generate, let alone describe. The dual problem of finding the size of the largest sphere which may be placed in the box without intersecting any of the particles will be discussed in the next section.

E.2 Voronoi Particle Placement Heuristic

The Voronoi particle placement heuristic has been designed to place particles as randomly as possible when random placement itself is not feasible. The difficulty with random placement comes about when the volume fraction f_v of the box in which the center of the new particle that may be placed is small. In this case, it may take a long time before the "hit and miss" random placement is successful. To make this precise, the number of attempts required to place a particle will be a geometric random variable with mean $1/f_v$, which may be very large. The heuristic attempts to place the new

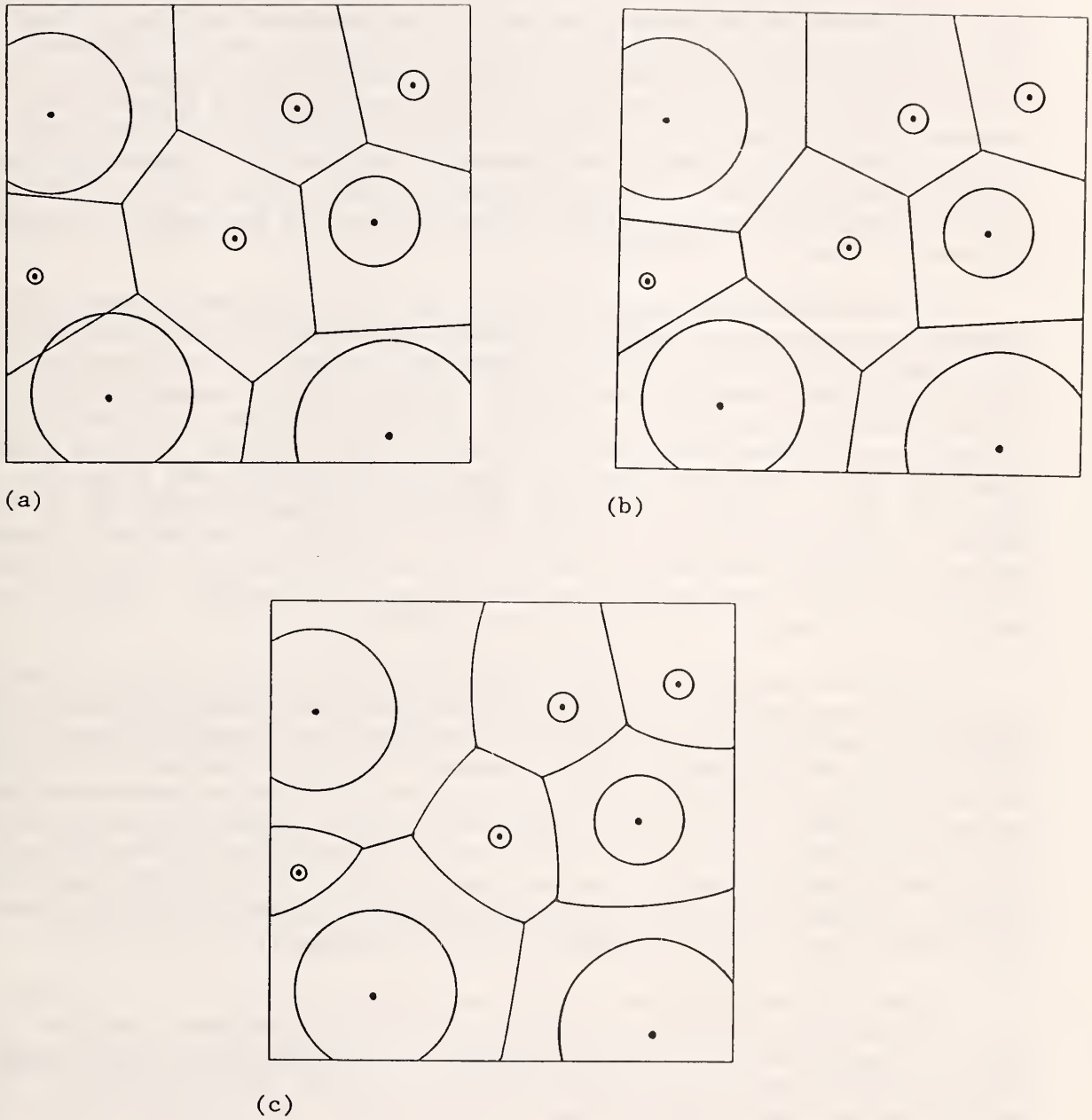


Figure E-2. Voronoi regions (points) (a), radical regions (b), and Voronoi regions (particles) (c).

particle in the empty space between previously placed particles, which should be a much smaller volume.

If all the particles are the same size, the problem of placing them in the box is known as the random sequential (or complete) packing problem [34]; Renyi [20] was able to solve this model analytically. He calculated 0.7476 as the mean density of particles (cars) which could be placed (parked). In two or more dimensions, the random sequential packing problem appears to be mathematically intractable. However simulations have been carried out in both two dimensions [21-28] and three dimensions [23, 29-31]. Pomeau [32] and Swendsen [33] were able to show that the approach to the final packing density is proportional to the reciprocal of the dth root of the number of attempted placements, where d is the dimension.

In two dimensions, several authors [22, 25, 26] have attempted to accelerate the approach to the final packing density by looking for the "holes" that comprise the area in which a sphere may be placed. As remarked by Lotwick [25] and Cooper [28 and 31] the method used by Tamemura [22] is only an approximation, and introduces bias. On the other hand, Lotwick [25] used an exact method based on a clever rejection scheme derived from the Voronoi tessellation. Our approach differs from that of Lotwick and is more in the spirit of [32], in that it uses the areas surrounding the vertices of the radical regions as possible placement sites. While the method only approximates the empty space between particles, the degree of approximation may be increased (at the expense of a greater number of attempts). Finally, our method steers clear of the inaccuracies of [22].

Our basic approach to determining this empty space is to take the areas surrounding the vertices of the radical regions. A drawback of this approach is that if one placed the largest particle possible in such a region, its center might not coincide with the vertex[10]. On the other hand, the radical method is easy to implement, fairly efficient, and can be updated when a new particle is added. So it was decided to use this method, and then associate with each vertex the location of the center of the largest sphere that could be placed in the vicinity. The net result is that the heuristic approximately finds the Voronoi regions of the particles while hopefully avoiding some of the pathological cases associated with that problem[11].

Since the particles are placed randomly, each vertex of a radical region will be common to exactly four regions with probability one. We first note that if a sphere does not touch each of the four particles, then a larger sphere may be found by perturbing the center of the sphere slightly. Thus the largest sphere that can be placed in the vicinity of a vertex[12] must touch all

---

[10]This is due to the way that the radical method takes particle radius into account.

[11]Some of these regions may not even have vertices.

[12]Assuming that the notion of "vicinity" is well defined.

four particles. If the sphere has center (x,y,z) and radius $r \geq 0$, then we must have

$$(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2 = (r_i + r)^2$$

$$(x_j - x)^2 + (y_j - y)^2 + (z_j - z)^2 = (r_j + r)^2$$

$$(x_k - x)^2 + (y_k - y)^2 + (z_k - z)^2 = (r_k + r)^2$$

$$(x_l - x)^2 + (y_l - y)^2 + (z_l - z)^2 = (r_l + r)^2.$$

These equations yield the system of three linear equations in four variables,

$$(x_j - x_i)x + (y_j - y_i)y + (z_j - z_i)z$$

$$= \frac{(x_j^2 + y_j^2 + z_j^2) - r_j^2 - (x_i^2 + x_i^2 + z_i^2) + r_i^2}{2} + (r_i - r_j)r \qquad \text{(E4)}$$

$$(x_k - x_i)x + (y_k - y_i)y + (z_k - z_i)z$$

$$= \frac{(x_k^2 + y_k^2 + z_k^2) - r_k^2 - (x_i^2 + x_i^2 + z_i^2) + r_i^2}{2} + (r_i - r_k)r \qquad \text{(E5)}$$

$$(x_l - x_i)x + (y_l - y_i)y + (z_l - z_i)z$$

$$= \frac{(x_l^2 + y_l^2 + z_l^2) - r_l^2 - (x_i^2 + x_i^2 + z_i^2) + r_i^2}{2} + (r_i - r_l)r \qquad \text{(E6)}$$

whose solutions are also required to satisfy

$$(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2 = (r_i + r)^2. \qquad \text{(E7)}$$

In some situations, the sphere so determined intersects some other particle, which makes it difficult to define the largest sphere that can be placed in the vicinity of a vertex. A possible way to overcome this difficulty will be discussed in the section on implementation.

Once we have associated with each vertex $V_1,\ldots,V_n$ the radius R_i and center C_i of the largest sphere which may be placed in its vicinity, the position of the new particle of radius r is determined by generating a point at random in the region which is the union for i = 1,...,N of the spheres with center C_i and radius $\min\{K(R_i-r),0\}$, where $K \geq 1$ is a scaling parameter. If K = 1 placement is guaranteed[13], whereas for larger values of K the approximation to random placement becomes increasingly closer. The number of attempts required to place a particle will be a geometric random variable whose mean is at most K^2 (which does not depend on f_v). If $r \geq R_i$ for i = 1,...,N, then the heuristic is unable to place the particle. If the particles have differing radii, then the largest particle which may be placed by the heuristic may be strictly smaller than the largest particle which may be placed in the box, but this should occur relatively infrequently[14].

E.3 Implementation

The current implementation of the Voronoi particle placement heuristic iteratively generates the radical regions of the particles. Initially, the region associated with the first particle is simply the box itself. The radical regions are next updated, particle by particle, when random placement fails for a particle[15]. When the Voronoi particle placement heuristic is required for a particle, the updated radical regions yield the collection of spheres whose union will be used as an approximation to the remaining empty space.

In order to update the radical regions after placing a new particle, we must do the following: (i) determine which of the vertices of the current regions will be vertices of one of the radical regions after adding the new particle, (ii) determine which of the particles will be neighbors of the new particle, and (iii) generate the vertices of the radical region of the new particle. For each vertex of a current radical region, we can easily check to see if it satisfies these inequalities which determine the radical region of the new particle; if it satisfies all of the inequalities, then it will not be a vertex of any of the radical regions after adding the new particle. A simple observation can save us from testing all of these inequalities. Suppose that the vertex V = (x,y,z) lies in the radical region of particle k, and that the new particle is particle i. If

---

[13]Assuming that the radii and centers of the spheres are found exactly.

[14]The size of the largest particle which may be placed in the box without intersecting any of the previously placed particles may be found in $O(n^4)$ time by solving the system of quadratic equations for each subset $\{(x_i,y_i,z_i), (x_j,y_j,z_j),(x_k,y_k,z_k),(x_l,y_l,z_l)\}$ of four particles, since the largest such particle must touch four particles (see also [34]). This may be implemented in Version 9.

[15]Improved efficiency would result if the regions were determined in a non-iterative fashion for all particles at the first point when random placement fails.

$$(x_k - x_i)x + (y_k - y_i)y + (z_k - z_i)z$$

$$> \frac{(x_k^2 + y_k^2 + z_k^2) - r_k^2 - (x_i^2 + x_i^2 + z_i^2) + r_i^2}{2} \qquad \text{(E8)}$$

then V lies outside the radical region of the new particle. If, on the other hand,

$$(x_k - x_i)x + (y_k - y_i)y + (z_k - z_i)z$$

$$\leq \frac{(x_k^2 + y_k^2 + z_k^2) - r_k^2 - (x_i^2 + x_i^2 + z_i^2) + r_i^2}{2} \qquad \text{(E9)}$$

then since V lies in the radical region of particle k, we have

$$(x_j - x_k)x + (y_j - y_k)y + (z_j - z_k)z$$

$$\leq \frac{(x_j^2 + y_j^2 + z_j^2) - r_j^2 - (x_k^2 + x_k^2 + z_k^2) + r_k^2}{2} \qquad \text{(E10)}$$

for all $j \neq i,k$. Adding inequality E9 to any one of the inequalities E10 yields

$$(x_j - x_i)x + (y_j - y_i)y + (z_j - z_i)z$$

$$\leq \frac{(x_j^2 + y_j^2 + z_j^2) - r_j^2 - (x_i^2 + x_i^2 + z_i^2) + r_i^2}{2} \qquad \text{(E11)}$$

and thus V lies inside the radical region of the new particle[16]. Next we note that a particle will be neighbor of the new particle if and only if one of the vertices of its current radical region lies inside the radical region of the new particle.

---

[16]This test is forgiving in the following sense. If for some reason not all neighbors of a particle are found, and the region is therefore larger than it should be, then in the future these vertices will be more likely to be assumed to be in a new region, and therefore will be eliminated.

To the list of neighbors of the new particle we also add translates of the particle itself, which has the effect of implicitly adding the constraints of the box to those which determine the new radical region. Neighbor j is then associated with a particle IND(J) (where IND(J) = 0 if the particle is a translate of the new particle). Since the radical regions are invariant under translation, before the vertices of the new region are found all positions are made relative to the new particle, which then lies at the origin. (These positions are stored in the array POS(,).) With each neighbor we associate the value

$$F(J) = \frac{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2 + r_i^2 - r_j^2}{2} \qquad \text{(E12)}$$

and its particle radius RJ, where j = IND(J) and i is the new particle. To find the vertices of the radical region, we then must solve the system of linear equations

$$\text{POS}(1,J)x + \text{POS}(2,J)y + \text{POS}(3,J)z = F(J) \qquad \text{(E13)}$$
$$\text{POS}(1,K)x + \text{POS}(2,K)y + \text{POS}(3,K)z = F(K) \qquad \text{(E14)}$$
$$\text{POS}(1,L)x + \text{POS}(2,L)y + \text{POS}(3,L)z = F(L) \qquad \text{(E15)}$$

for every J < K < L, and then check to see if the solution violates any of the other inequalities

$$\text{POS}(1,M)x + \text{POS}(2,M)y + \text{POS}(3,M)z \leq F(M)$$

which determine the new radical region[17]. For a fixed J and K the solutions to this system of equations all lie on some line $(X_1,Y_1,Z_1) + \alpha(X_2,Y_2,Z_2)$, where (X_2,Y_2,Z_2) is parallel to the planes determined by the first two equations, or in other words perpendicular to the vectors which give the relative positions of neighbors J and K. The cross product of these two vectors can play the role of (X_2,Y_2,Z_2), and (X_1,Y_1,Z_1) can be taken to be one of the intercepts of the line. Now for each L > K we need only determine α and then check to see if the point is a vertex[18]. If it is we record the position of the vertex (VPOS(,)) and the numbers of the particles whose radical regions it is a vertex of (VIND ( )).

In order to find the center (VPUT(,)) and radius (VRAD( )) of the largest sphere which can be placed in the vicinity of a vertex, we must solve the system of equations

---

[17]The method of [18] has not been implemented in Version 8. It should be available for Version 9 from J. Bernal, as it constitutes a part of this 3-dimensional Voronoi diagram algorithm [17] which is written in FORTRAN on the CYBER 205 at NIST.

[18]It is worth noting that if at some point we have found two vertices on a given line, then we may move on to the next values for J and K.

$$POS(1,J)x + POS(2,J)y + POS(3,J)z = F(J) + (RI-RJ)R \quad (E16)$$
$$POS(1,K)x + POS(2,L)y + POS(3,K)z = F(K) + (RI-RK)R \quad (E17)$$
$$POS(1,L)x + POS(2,L)y + POS(3,1)z = F(L) + (RI-RL)R \quad (E18)$$

subject to $x^2 + y^2 + z^2 = (RI + R)^2$. Suppose that (X_d,Y_d,Z_d) is the solution of the system of equations

$$POS(1,J)x + POS(2,J)y + POS(3,J)z = RI - RJ \quad (E19)$$
$$POS(1,K)x + POS(2,K)y + POS(3,K)z = RI - RK \quad (E20)$$
$$POS(1,L)x + POS(2,L)y + POS(3,L)z = RI - RL \quad (E21)$$

Since eqs E19-E21 differs from eqs E16-E18 only by their right hand side, we can find a point (X_p,Y_p,Z_p) so that $(X_d,Y_d,Z_d) = (X_p,Y_p,Z_p) + \alpha(X_2,Y_2,Z_2)$ for some α. Any solution to the previous system of equations involving R will then differ form the vertex[19] by (RX_d,RY_d,RZ_d). To determine the value of R for which $x^2 + y^2 + z^2 = (RI + R)^2$ now entails solving a quadratic equation in R, which will give the radius of the largest sphere that may be placed in the vicinity of the vertex. Because the radical regions are a "linear approximation" to the Voronoi regions of the particles, there are situations in which the value of R is so large that a sphere of radius R centered at (x,y,z) would intersect some other particle. In this case (x,y,z) would not be "in the vicinity" of the vertex. So if the value if R is too large, we instead take R to be the minimum distance from the vertex to the new particle and its three neighbors J,K and L and do not adjust the position of the vertex.

To place a particle of radius r, we first calculate

$$VSUM = \sum_{VRAD(J)\geq r} (VRAD(J) - r)^3 \quad (E22)$$

the volume of the region where particle placement is guaranteed. If VSUM = 0 we are unable to place the particle. If VSUM > 0 then a potential placement site K is randomly generated. If $VRAD(J) \geq r$, then site J has probability $(VRAD(J) - r)^3/VSUM$ of being chosen, and a point is selected at random from the sphere of radius SCALE(VRAD(J) - r) and center (VPUT(,J)). These spheres may overlap, so if the point lies in any other such sphere for J < K, then another potential placement site must be chosen.

---

[19]This is the solution to the system of equations which results from setting R = 0.

APPENDIX F. EXCLUDED VOLUME

The expressions described here are used to calculate that portion of the volume of a sphere which lies outside of the box. By summing over all spheres in the box, this gives an exact expression for the excluded volume.

The volume of the portion of a sphere of radius r centered at the point (a,b,c), which lies outside the positive orthant may be calculated as follows. First note that there is an obvious scale factor present, so that this volume is just r^3 times the volume of the portion of a sphere of radius 1 centered at the point (a/r, b/r, c/r), which lies outside the positive orthant. The functions are defined as

$$V(a,b,c) = \text{volume}\{(x,y,z) : x^2 + y^2 + z^2 \leq 1,\ x \geq a,\ y \geq b,\ z \geq c\} \quad \text{(F1)}$$

for $a,b,c \geq 0$ such that $a^2 + b^2 + c^2 < 1$;

$$V(a) = V(a,0,0) \quad \text{(F2)}$$

and

$$V(a,b) = V(a,b,0). \quad \text{(F3)}$$

It is easy to evaluate V(a) because 4V(a) is the volume of a "cap" or "zone of one base" (see [35], p. 128), and thus

$$V(a) = \frac{1}{4}\left[\frac{\pi}{3}(1 - a)^2(2 + a)\right] = \frac{\pi}{12}(1 - a)^2(2 + a). \quad \text{(F4)}$$

Evaluating V(a,b) for $a^2 + b^2 < 1$ is somewhat harder. First notice that the planes x = a and y = b cut the portion of the unit sphere that lies in the positive orthant into 4 different pieces, from which we obtain the identity

$$\frac{1}{8}\left(\frac{4\pi}{3}\right) = \int_0^b \int_0^a \sqrt{1 - x^2 - y^2}\, dxdy + V(a) + V(b) - V(a,b) \quad \text{(F5)}$$

or in other words,

$$V(a,b) = \int_0^b \int_0^a \sqrt{1 - x^2 - y^2}\, dxdy + V(a) + V(b) - \frac{\pi}{6}. \quad \text{(F6)}$$

So the next step will be to evaluate the following integral

$$I_1 = \int_0^b \int_0^a \sqrt{1 - x^2 - y^2}\, dx dy = \int_0^b \int_0^a \sqrt{(1 - y^2) - x^2}\, dx dy. \qquad \text{(F7)}$$

Equation F7 reduces as follows

$$I_1 = \int_0^b \left[\frac{x}{2} \sqrt{(1 - y^2) - x^2} + \frac{(1 - y^2)}{2} \sin^{-1} \frac{x}{\sqrt{1 - y^2}} \right] \Bigg|_0^a dy$$

$$= \frac{1}{2} \int_0^b \left[a \sqrt{1 - a^2 - y^2} + (1 - y^2) \sin^{-1} \frac{a}{\sqrt{1 - y^2}} \right] dy$$

$$= \frac{a}{2} \int_0^b \sqrt{(1 - a^2) - y^2}\, dy + \frac{1}{2} \int_0^b (1 - y^2) \sin^{-1} \frac{a}{\sqrt{1 - y^2}}\, dy$$

$$= I_2 + I_3. \qquad \text{(F8)}$$

The first integral in eq F8 is just

$$I_2 = \frac{a}{2} \left[\frac{y}{2} \sqrt{(1 - a^2) - y^2} + \frac{(1 - a^2)}{2} \sin^{-1} \frac{y}{\sqrt{1 - a^2}} \right] \Bigg|_0^b$$

$$= \frac{ab}{4} \sqrt{1 - a^2 - b^2} + \frac{a(1 - a^2)}{4} \sin^{-1} \frac{b}{\sqrt{1 - a^2}}.$$

The second may be integrated by parts, using

$$u = \sin^{-1} \frac{a}{\sqrt{1 - y^2}} \qquad dv = \frac{1 - y^2}{2}$$

$$du = \frac{aydy}{(1 - y^2) \sqrt{1 - a^2 - y^2}} \qquad v = \frac{3y - y^3}{6}.$$

So the value of the integral is

$$I_3 = uv\,\Big|_0^b - \int_0^b v\,du$$

$$= \frac{3y - y^3}{6}\ \sin^{-1}\frac{a}{\sqrt{1 - y^2}}\ \Big|_0^b - \frac{a}{6}\int_0^b \frac{(3y - y^3)y\,dy}{(1 - y^2)\sqrt{1 - a^2 - y^2}}$$

$$= \frac{3b - b^3}{6}\ \sin^{-1}\frac{a}{\sqrt{1 - b^2}} + \frac{a}{6}\int_0^b \frac{(y^4 - 3y^2)\,dy}{(1 - y^2)\sqrt{1 - a^2 - y^2}}$$

$$= \frac{3b - b^3}{6}\ \sin^{-1}\frac{a}{\sqrt{1 - b^2}} + I_4. \qquad \text{(F9)}$$

Now since $y^4 - 3y^2 = -y^2(1 - y^2) + 2(1 - y^2) - 2$, the integral I_4 in eq F9 can be written as

$$I_4 = \frac{a}{6}\int_0^b \frac{y^2\,dy}{\sqrt{1 - a^2 - y^2}} + \frac{a}{3}\int_0^b \frac{dy}{\sqrt{1 - a^2 - y^2}}$$

$$- \frac{a}{3}\int_0^b \frac{dy}{(1 - y^2)\sqrt{1 - a^2 - y^2}}$$

$$= -I_5 + I_6 - I_7. \qquad \text{(F10)}$$

Applying these integrals listed in table F-1, for the first integral in eq F10 we get

$$I_5 = \frac{a}{6}\int_0^b \frac{y^2\,dy}{\sqrt{(1 - a^2) - y^2}}$$

$$= \frac{a}{6}\left(-\frac{y}{2}\sqrt{(1 - a^2) - y^2} + \frac{1 - a^2}{2}\ \sin^{-1}\frac{y}{\sqrt{1 - a^2}}\right)\Big|_0^b$$

$$= \frac{ab}{12}\sqrt{1 - a^2 - b^2} - \frac{a(1 - a^2)}{12}\ \sin^{-1}\frac{b}{\sqrt{1 - a^2}}\,. \qquad \text{(F11)}$$

For the second integral in eq F10 we get

$$I_6 = \frac{a}{3} \sin^{-1} \frac{y}{\sqrt{1 - a^2}} \Bigg|_0^b = \frac{a}{3} \sin^{-1} \frac{b}{\sqrt{1 - a^2}}. \qquad \text{(F12)}$$

For the third integral in eq F10 we get

$$I_7 = \frac{a}{3} \left[\frac{1}{a} \tan^{-1} \frac{ay}{\sqrt{(1 - a^2) - y^2}} \right] \Bigg|_0^b$$

$$= \frac{1}{3} \tan^{-1} \frac{ab}{\sqrt{1 - a^2 - b^2}} . \qquad \text{(F13)}$$

Table F-1. Integrals[a]

200. $$\int \sqrt{a^2 - x^2}\, dx = \frac{x}{2} \sqrt{a^2 - x^2} + \frac{a^2}{2} \sin^{-1} \frac{x}{|a|}$$

201. $$\int \frac{dx}{\sqrt{a^2 - x^2}} = \sin^{-1} \frac{x}{|a|}$$

214. $$\int \frac{x^2 dx}{\sqrt{a^2 - x^2}} = -\frac{x}{2} \sqrt{a^2 - x^2} + \frac{a^2}{2} \sin^{-1} \frac{x}{|a|}$$

233. $$\int \frac{dx}{(b^2 - x^2)\sqrt{a^2 - x^2}} = \frac{1}{b\sqrt{b^2 - a^2}} \tan^{-1} \frac{x\sqrt{b^2 - a^2}}{b\sqrt{a^2 - x^2}}$$

$(b^2 > a^2)$

[a]From [35] pages 252-254.

Combining all terms, we get the following for the value of the integral in eq F10

$$I_4 = \frac{ab}{3}\sqrt{1 - a^2 - b^2} + \frac{3a - a^3}{6}\sin^{-1}\frac{b}{\sqrt{1 - a^2}}$$

$$+ \frac{3b - b^3}{6}\sin^{-1}\frac{a}{\sqrt{1 - b^2}} - \frac{1}{3}\tan^{-1}\frac{ab}{\sqrt{1 - a^2 - b^2}}. \qquad \text{(F14)}$$

This is symmetric in a and b (as it must be).

Fortunately, given V(a,b) it is relatively simple to compute V(a,b,c) for $a^2 + b^2 + c^2 < 1$. First notice that the planes x = a, y = b and z = c cut that portion of the unit sphere which lies in the positive orthant into eight different pieces, from which we obtain the identity

$$\frac{1}{8}\left(\frac{4\pi}{3}\right) = abc + V(a) + V(b) + V(c)$$

$$- V(a,b) - V(a,c) - V(b,c) + V(a,b,c)$$

or in other words,

$$V(a,b,c) = \frac{\pi}{6} - abc - V(a) - V(b) - V(c)$$

$$+ V(a,b) + V(a,c) + V(b,c). \qquad \text{(F15)}$$

Using the function V(a,b,c), it is easy to express the volumes of the intersections of one, two or three protruding domes in terms of the heights h_a, h_b and h_c of the protruding domes and the radius of the sphere: the volume of a dome is

$$4r^3 V\left(\frac{r - h_a}{r}\right).$$

If $(r - h_a)^2 + (r - h_b)^2 < r^2$, then the volume of the intersection of the three domes is

$$2r^3 V\left(\frac{r - h_a}{r}, \frac{r - h_b}{r}\right).$$

If $(r - h_a)^2 + (r - h_b)^2 + (r - h_c)^2 < r^2$, then the volume of the intersection of the three domes is

$$r^3 V \left(\frac{r - h_a}{r} , \frac{r - h_b}{r} , \frac{r - h_c}{r} \right) .$$

APPENDIX G. SAMPLE SESSIONS

Note: The sample sessions shown here consist of actual output from the computer and input from the user (UNDERLINED), taken from log files generated during the running of example TEST3.

G.1 Set-up

Note: When user input is requested during the set-up program, the current value of each parameter is written. These current values should not be confused with input from the user, which is underlined.

SETM8

```
   ************************************************
   **     COMPUTER SIMULATED MICROSTRUCTURE OF     **
   **     HYDRATING CEMENT - VERSION 8  (8/88)     **
   **     SET UP SIMULATION PARAMETERS             **
   ************************************************
   **     FOR EACH PARAMETER,                      **
   **     TYPE -1 TO KEEP PREVIOUS VALUE           **
   ************************************************


 SELECT:  (1) SET-UP A NEW SIMULATION
          (2) RERUN A PREVIOUS SIMULATION
              WITH NEW HYDRATION PARAMETERS
          (3) REVIEW CURRENT SIMULATION PARAMETERS
 SELECT 1, 2, OR 3:  0
>1

 ******  DATA FILES  ******

 ENTER A FILENAME FOR THE CURRENT SIMULATION DATA:
>TEST3

  SIMULATION DATA WILL BE FILED AS:  TEST3.CM8

 ******  SYSTEM SPACE DIMENSIONS  ******
         (1) DEFAULT DIMENSIONS FOR  X,Y,Z: {0 to 100}
         (2) X,Y = {0 to 100},  Z = {0 to 2RMAX * ?}
         (3) SPECIFY MIN & MAX VALUES FOR  X,Y,Z
 SELECT 1, 2, OR 3:  0
>3

******  SPECIFY 3D SYSTEM SPACE  ******

 ENTER MINIMUM X VALUE:  0.0
>0

 ENTER MAXIMUM X VALUE:  0.0
>67.75

 ENTER MINIMUM Y VALUE:  0.0
>0
```

```
 ENTER MAXIMUM Y VALUE:  0.0
>67.75

 ENTER MINIMUM Z VALUE:  0.0
>0

 ENTER MAXIMUM Z VALUE:  0.0
>67.75

******  PARTICLE GENERATION  ******

 ENTER # OF C3S PARTICLES TO BE SIMULATED
  (1-1000):  0
>1000

 ENTER # OF CH CRYSTALS TO BE SIMULATED
   (0-2000):  0
>1000

 ******  PARTICLE SIZE DISTRIBUTION   ******
          (1) DEFAULT WEIBULL VALUES:
                WPA=.55, WPB=.40, WPC=.60
                FOR  Y = (-LOG X)**(1/WPA)*WPB+WPC
          (2) SELECT WEIBULL PARAMETERS:
                FOR  Y = (-LOG X)**(1/WPA)*WPB+WPC
          (3) USE PARTICLE DISTRIBUTION TABLE
 SELECT 1, 2, OR 3:  0
>3

 SIZE DISTRIBUTION IN M8TAB IN EFFECT

 ******  PLACEMENT METHODS  ******
          (1) RANDOM PLACEMENT OF ALL PARTICLES
          (2) CLUSTERED PLACEMENT OF ALL PARTICLES
 SELECT 1 OR 2:  0
>1

******  RANDOM NUMBER GENERATOR (UNI)  ******

 ENTER A SEED NUMBER (0 TO 32767):  9999
>1989

**********  HYDRATION PARAMETERS  **********

 ENTER NUMBER OF HYDRATION CYCLES (0-1000):  0
>100
 ENTER CHANGE IN RADIUS DURING
  INITIAL HYDRATION CYCLE (.01-1):  0.0
>.02
```

```
 ENTER INDEX FOR KINETIC ALGORITHM
   (SETS METHOD FOR CALCULATING CHANGE IN RADIUS
   FOR ANHYDROUS PARTICLES
          (0) CONSTANT CHANGE EACH CYCLE
          (1) CHANGE ACCORDING TO DIFFUSION ALGORITHM
              OF JENNINGS AND JOHNSON
 SELECT 0 OR 1:  0
>1

 ENTER CRITICAL THICKNESS VALUE (1.0 - 100.0):  99.0
>50

 ENTER OUTER CSH VOLUME
   RELATIVE TO VOLUME OF C3S (0.01 - 5.0):  1.0
>1.75

 ENTER CSH BACKFILL FACTOR (0 - 1.0):  0.0
>.5

 ENTER INNER CSH VOLUME
   RELATIVE TO VOLUME OF C3S (0.01 - 5.0):  0.0
>1.75

 ENTER VOLUME OF CH
   RELATIVE TO VOLUME OF C3S (0.1 - 2.5):  1.0
>.61

 ENTER CS PROJECTED RADIUS FACTOR (1.5 - 5.0):  1.0
>3

 ******  END OF PARAMETER SELECTION  ******
          (1) REVIEW PARAMETERS SELECTED
          (2) PRINT COPY OF PARAMETERS SELECTED
          (3) SAVE CURRENT PARAMETERS AND EXIT
          (4) RESELECT ANY PARAMETERS
          (5) ABORT AND EXIT
 SELECT 1, 2, 3, 4, OR 5:
>3

*****  SAVING SIMULATION PARAMETERS (M8PAR)  *****

 PARAMETERS WRITTEN TO M8PAR.
 EXIT CEMENT MODEL SET-UP PROGRAM
STOP
TEK     -END OF TASK CODE=   0    PROCESSOR=0.363  TSK-ELAPSED=1:30
```

G.2 Simulation

RUNSIM8

```
05/04/89  10:40:21
 OPENING M8PAR
 M8PAR READ COMPLETED
 OPENING SIMULATION DATA FILE
 INITIALIZING VARIABLES
 USING DISTRIBUTION TABLE (M8TAB)
 PLACING PARTICLES IN BOX
 CALCULATING SYSTEM VOLUMES AND PORES
 INITIAL WATER/CEMENT RATIO = 0.300356
 **GENERATE COLLISION TABLE
  INITIAL INDEXING COMPLETED FOR PARTICLE HYDRATION
  NUMBER OF TYPE 1 COLLISIONS INDEXED = 49834
  NUMBER OF TYPE 2 COLLISIONS INDEXED = 55104
  NUMBER OF TYPE 3 COLLISIONS INDEXED = 61822
  NUMBER OF TYPE 4 COLLISIONS INDEXED = 55104
 **SAVE ANHYDROUS DATA IN USER SPECIFIED FILENAME
 ANHYDROUS DATA SAVED
 HYDRATION CYCLE 1
 HYDRATED DATA SAVED
 HYDRATION CYCLE 2
 HYDRATED DATA SAVED
 HYDRATION CYCLE 3
 HYDRATED DATA SAVED
 HYDRATION CYCLE 4
 HYDRATED DATA SAVED
 HYDRATION CYCLE 5
 HYDRATED DATA SAVED
 HYDRATION CYCLE 6
 HYDRATED DATA SAVED
 HYDRATION CYCLE 7
 HYDRATED DATA SAVED
 HYDRATION CYCLE 8
 HYDRATED DATA SAVED
 HYDRATION CYCLE 9
 HYDRATED DATA SAVED
 HYDRATION CYCLE 10
 HYDRATED DATA SAVED
 HYDRATION CYCLE 11
 HYDRATED DATA SAVED
 HYDRATION CYCLE 12
 HYDRATED DATA SAVED
 HYDRATION CYCLE 13
 HYDRATED DATA SAVED
 HYDRATION CYCLE 14
 HYDRATED DATA SAVED
 HYDRATION CYCLE 15
 HYDRATED DATA SAVED
 HYDRATION CYCLE 16
```

```
 HYDRATED DATA SAVED
 HYDRATION CYCLE 17
 HYDRATED DATA SAVED
 HYDRATION CYCLE 18
 HYDRATED DATA SAVED
 HYDRATION CYCLE 19
 HYDRATED DATA SAVED
 HYDRATION CYCLE 20
 HYDRATED DATA SAVED
 HYDRATION CYCLE 21
 HYDRATED DATA SAVED
 HYDRATION CYCLE 22
 HYDRATED DATA SAVED
 HYDRATION CYCLE 23
 HYDRATED DATA SAVED
 HYDRATION CYCLE 24
 HYDRATED DATA SAVED
 HYDRATION CYCLE 25
 HYDRATED DATA SAVED
 HYDRATION CYCLE 26
 HYDRATED DATA SAVED
 HYDRATION CYCLE 27
 HYDRATED DATA SAVED
 HYDRATION CYCLE 28
 HYDRATED DATA SAVED
 HYDRATION CYCLE 29
 HYDRATED DATA SAVED
 HYDRATION CYCLE 30
  HYDRATION TERMINATION:
  RADIUS OF CRYSTAL 260 EXCEEDS PROJECTED VALUE
  PROJECTED RADIUS EXCEEDED BY CRYSTAL 260
 HYDRATED DATA SAVED
  PARTICLES COULD NOT BE FULLY HYDRATED
 NOT ALL REQUESTED HYDRATION CYCLES WERE COMPLETED
 CEMENT MODEL SIMULATION RUN TERMINATING
STOP
TEK     -END OF TASK CODE=   0    PROCESSOR=1:52:09.894  TSK-ELAPSED=2:16:39
05/04/89  12:57:22
USER  CPU TIME    1:51:06.444    99.0 %
SVC   CPU TIME       1:03.450      .9 %
ROLL  CPU TIME           0.000     0.0 %
PROCESSOR TIME    1:52:09.894   100.0 %    82.0 %
WAIT      TIME      24:29.106              17.9 %
ROLL      TIME          0.000               0.0 %
ELAPSED   TIME    2:16:39                100.0 %
ROLLS           0
I/O             73939
```

G.3 Graphics

```
      RUNGRF8

05/04/89  08:17:07

 ENTER THE FILENAME (W/O .EXT) OF THE SIMULATION DATA:
>TEST3

 SELECT:  (1) DEFAULT 4115 COLORS
          (2) DEFAULT 4115 GREY TONES
          (3) DEFAULT 4014 B/W PATTERNS
          (4) SELECT 4115 COLORS
>3

  RETRIEVING INITIAL DATA, PLEASE STANDBY...

   0.00 %C3S REACTED FOR HYDRATION CYCLE     0

  SELECT OPTION:
  (1) DRAW CYCLE    0  (2) CHANGE WINDOW SIZE
  (3) NEW CYCLE        (4) CHANGE Z-SLICE ( 33.88)
  (5) NEW DATA FILE    (6) CHANGE COLORS    (7) EXIT
>1

  SELECT DRAWING MODE:  (1) SLICE / SOLID FILL
                        (2) NON-SLICED / SOLID FILL
                        (3) NON-SLICED / OUTLINED
>1

  SELECT OPTION:
  (1) DRAW CYCLE    9  (2) CHANGE WINDOW SIZE
  (3) NEW CYCLE        (4) CHANGE Z-SLICE ( 33.88)
  (5) NEW DATA FILE    (6) CHANGE COLORS    (7) EXIT
>7

  **  CM8 GRAPHICS SYSTEM - TERMINATION BY USER  **

STOP
TEK     -END OF TASK CODE=   0    PROCESSOR=5:55.577  TSK-ELAPSED=43:38
05/04/89  09:00:54
USER  CPU TIME       5:41.064    95.9 %
SVC   CPU TIME         14.513     4.0 %
ROLL  CPU TIME          0.000     0.0 %
PROCESSOR TIME       5:55.577   100.0 %    13.5 %
WAIT      TIME      37:42.423              86.4 %
ROLL      TIME          0.000               0.0 %
ELAPSED   TIME      43:38                 100.0 %
ROLLS           0
I/O             21011
```

G.4 Sample Output

A set of configurations (table G-1) have been developed to test the simulation model. Sample output is shown in figures G-1 through G-9. The parameter files and data files used for each configuration are available for demonstration purposes or for testing subsequent modifications.

Table G-1. Test Configurations

File	Configuration
TEST1.PAR	100 CS particles, default box size, monosized distribution (radius = 2 μm) (saved in TEST1.TAB), dispersed, no hydration cycles
TEST2.PAR	1000 CS particles, default box size, particle size distribution shown in figure 5 (saved in TEST2.TAB), dispersed, no hydration cycles
TEST3.PAR	1000 CS particles, 1000 CH crystals, box size 0-67.75 μm, same particle size distribution used in TEST2, dispersed, hydrated 100 cycles, kinetic control using 5.0 as critical thickness, initial hydration decrement 0.02. backfill factor 0.5
TEST4.PAR	1000 CS particles, 1000 CH crystals, same box size and particle size distribution used in TEST2, dispersed, hydrated 100 cycles, no kinetic control, initial hydration decrement 1.0. backfill factor 0.5
TEST5.PAR	1000 CS particles, 1000 CH crystals, same box size and particle size distribution used in TEST2, clustered, 100 clusters, method 1, hydrated 100 cycles, no kinetic control, initial hydration decrement 1.0, backfill factor 0.5
TEST6.PAR	1000 CS particles, 1000 CH crystals, default box size, Weibul distribution with default parameters, dispersed, hydrated 100 cycles, no kinetic control, initial hydration decrement 0.02, backfill factor 1.0

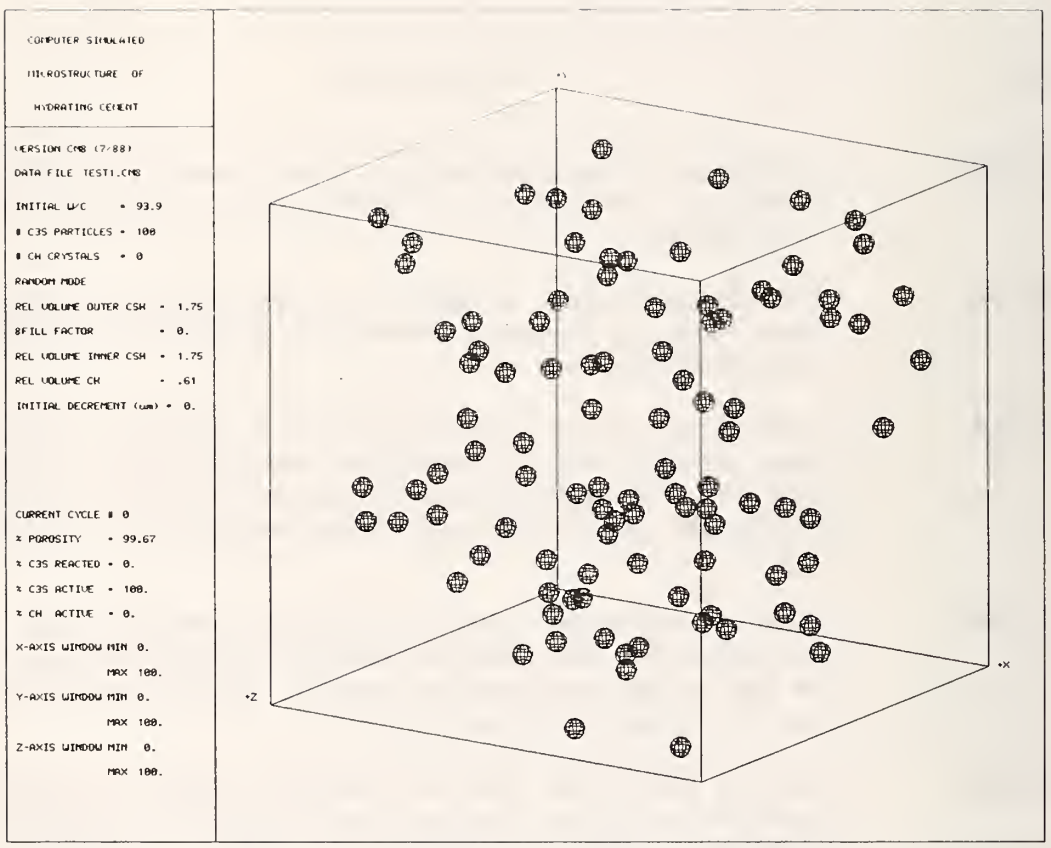


Figure G-1. 3D graphics of 100 unhydrated, monosized (2 μm), dispersed particles (TEST1.CM8).

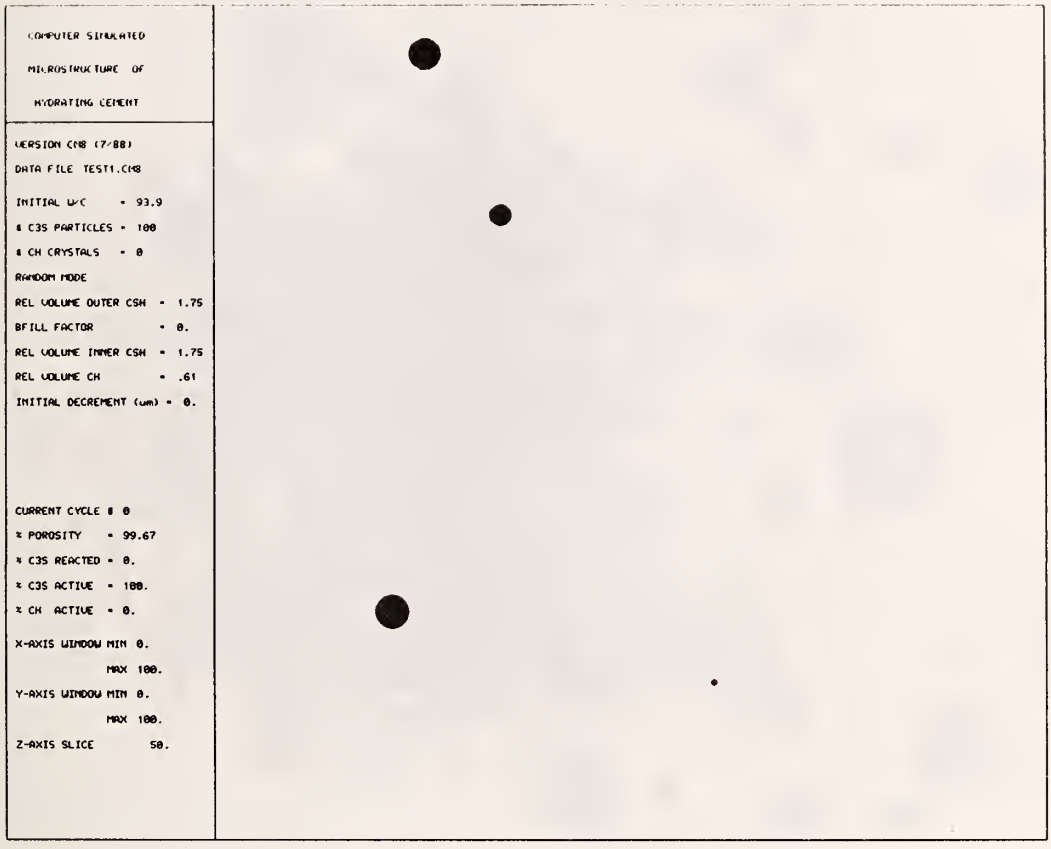


Figure G-2. 2D graphics of the same unhydrated particles in figure G-1 (TEST1.CM8).

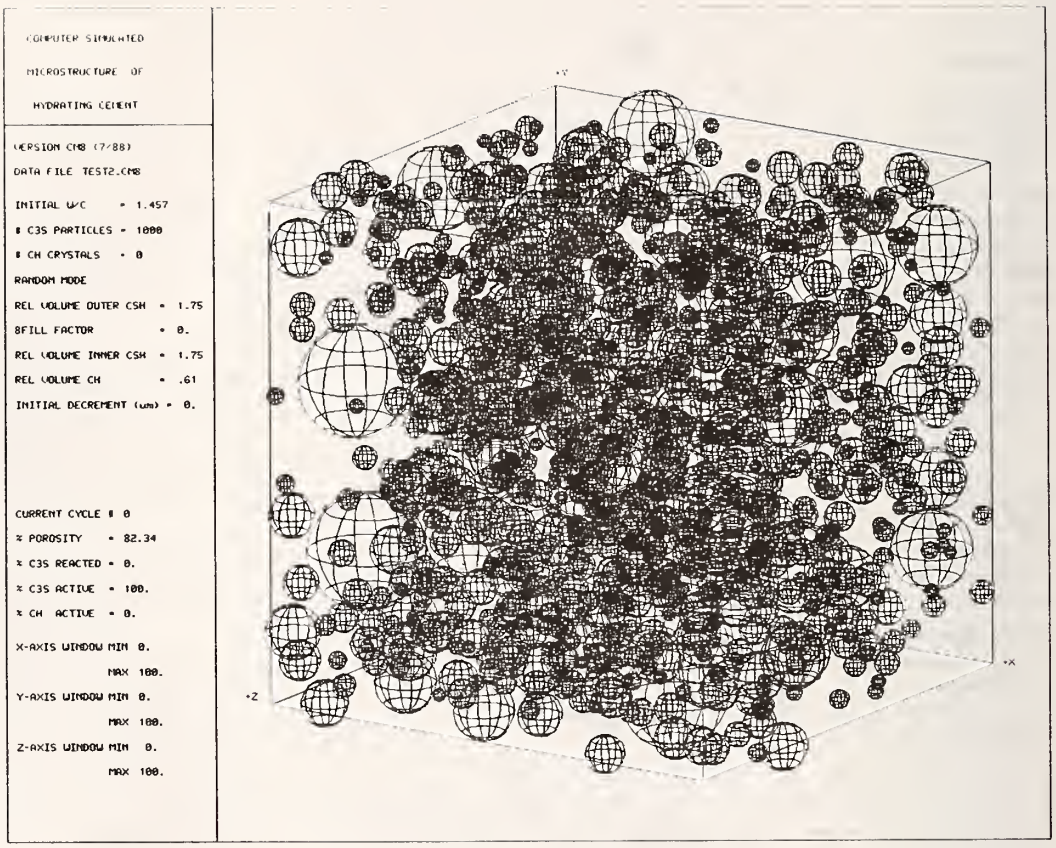


Figure G-3. 3D graphics of dispersed, unhydrated particles using the particle size distribution in figure 5 (TEST2.CM8).

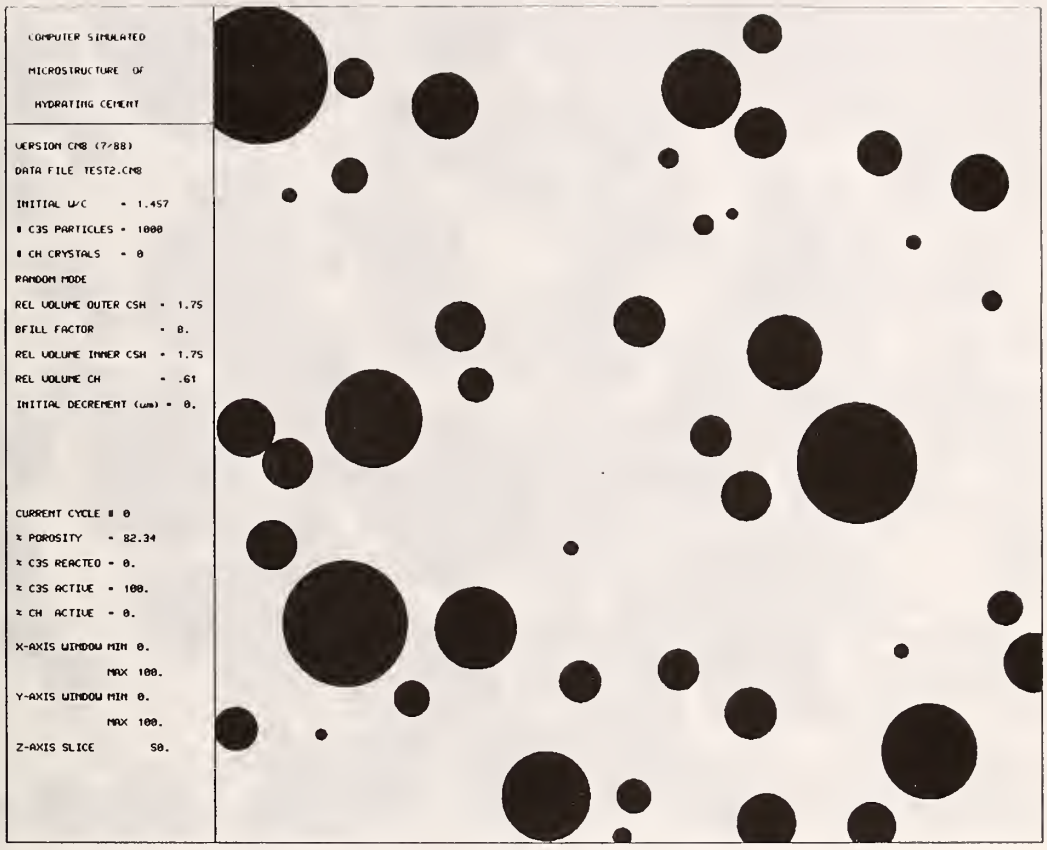


Figure G-4. 2D graphics of the same unhydrated particles in figure G-3 (TEST2.CM8).

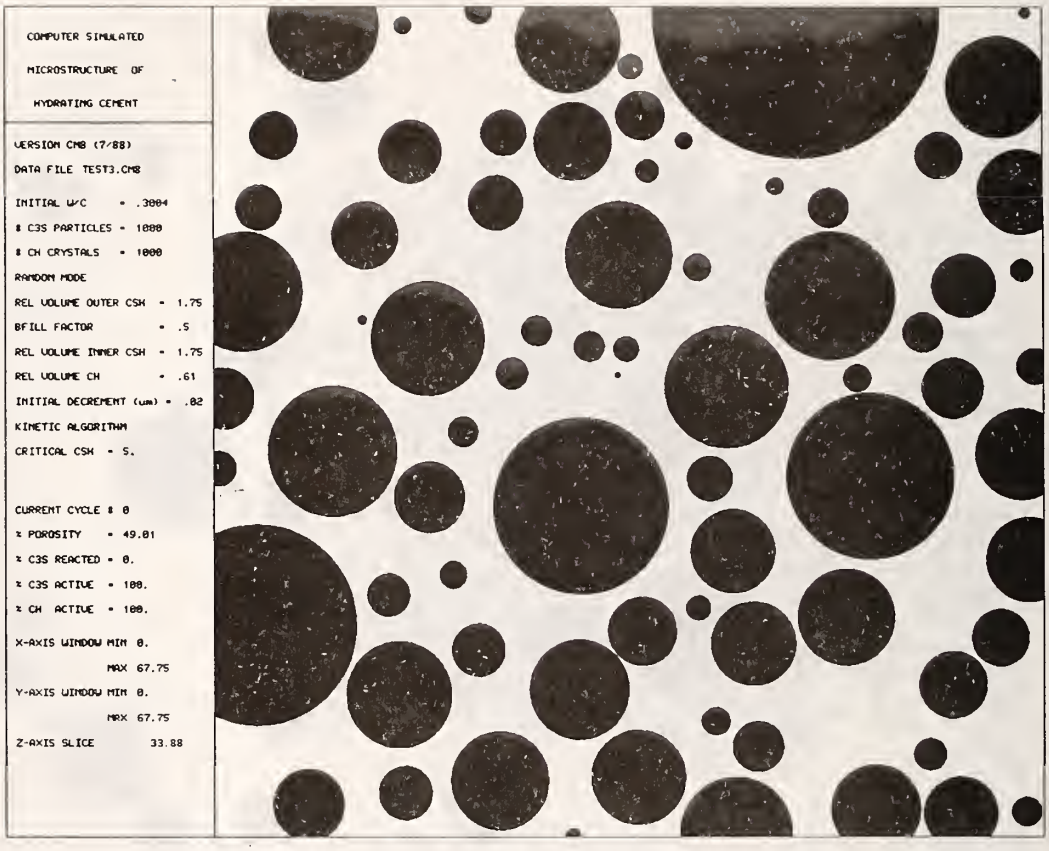


Figure G-5. 2D graphics of unhydrated particles, box size selected to provide water-to-cement ratio 0.30, otherwise similar to figures G-3 and G-4 (TEST3.CM8).

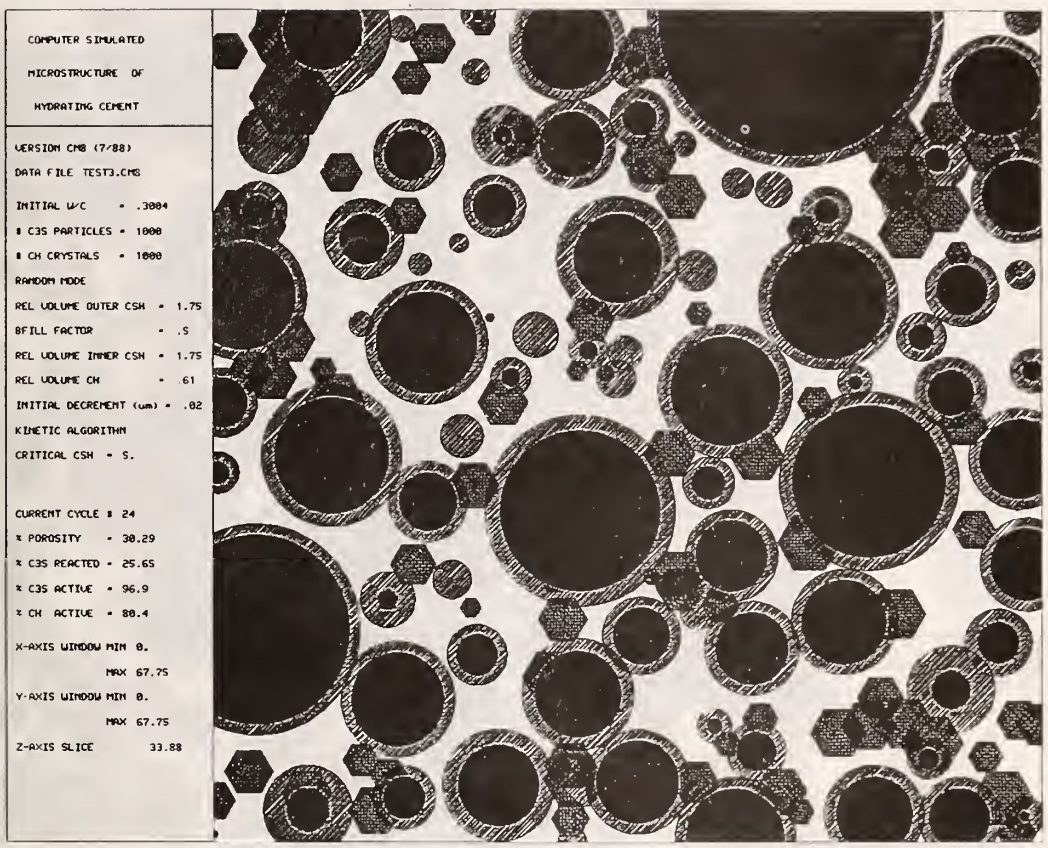


Figure G-6. 2D graphics of the same simulation used in figure G-5 after ~30% hydration with kinetic control and an intermediate backfill factor (TEST3.CM8).

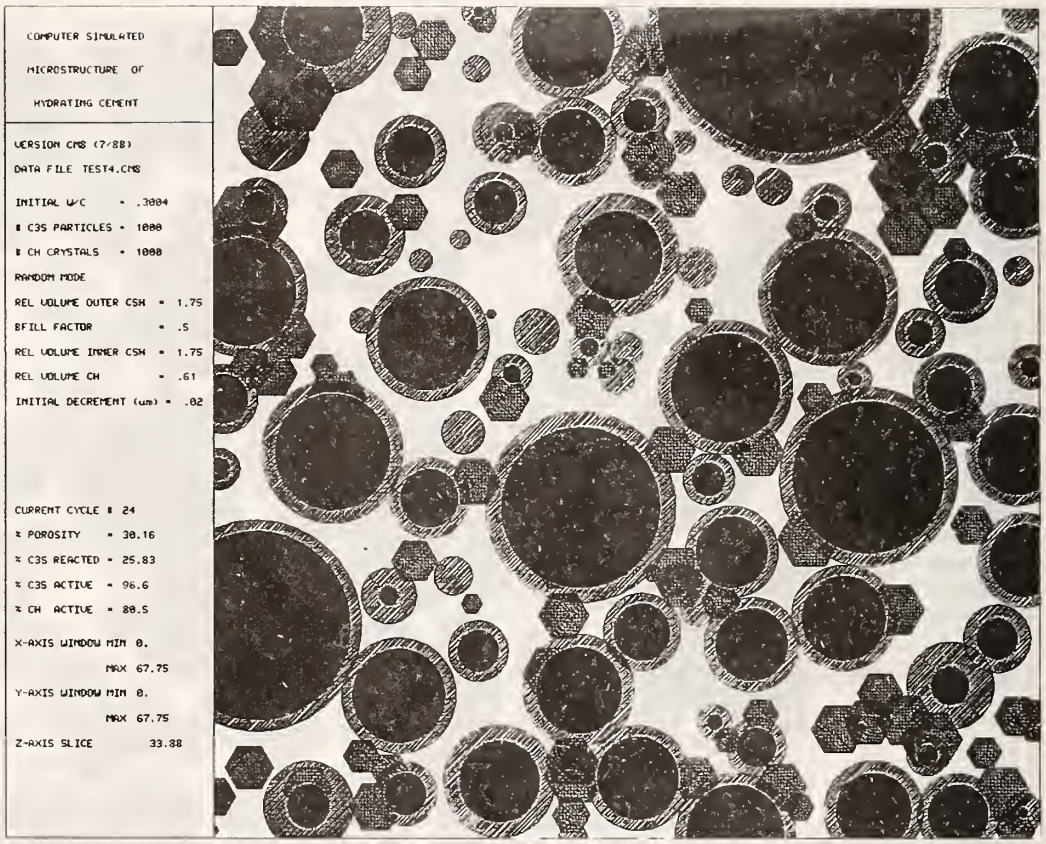


Figure G-7. 2D graphics of the same simulation used in figure G-5 after ~30% hydration with no kinetic control and an intermediate backfill factor (TEST4.CM8).

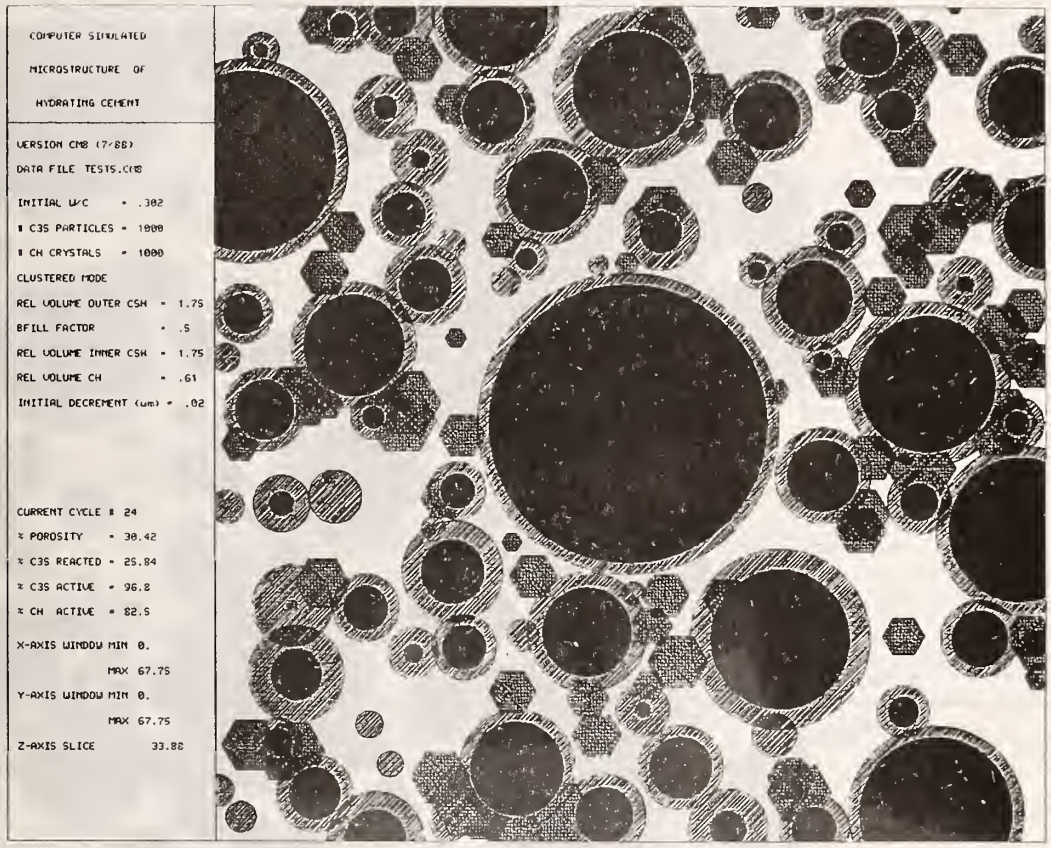


Figure G-8. 2D graphics of a similar simulation as figure G-5 but clustered particles, after ~50% hydration with kinetic control, an intermediate backfill factor, and a large initial hydration decrement (TEST5.CM8).

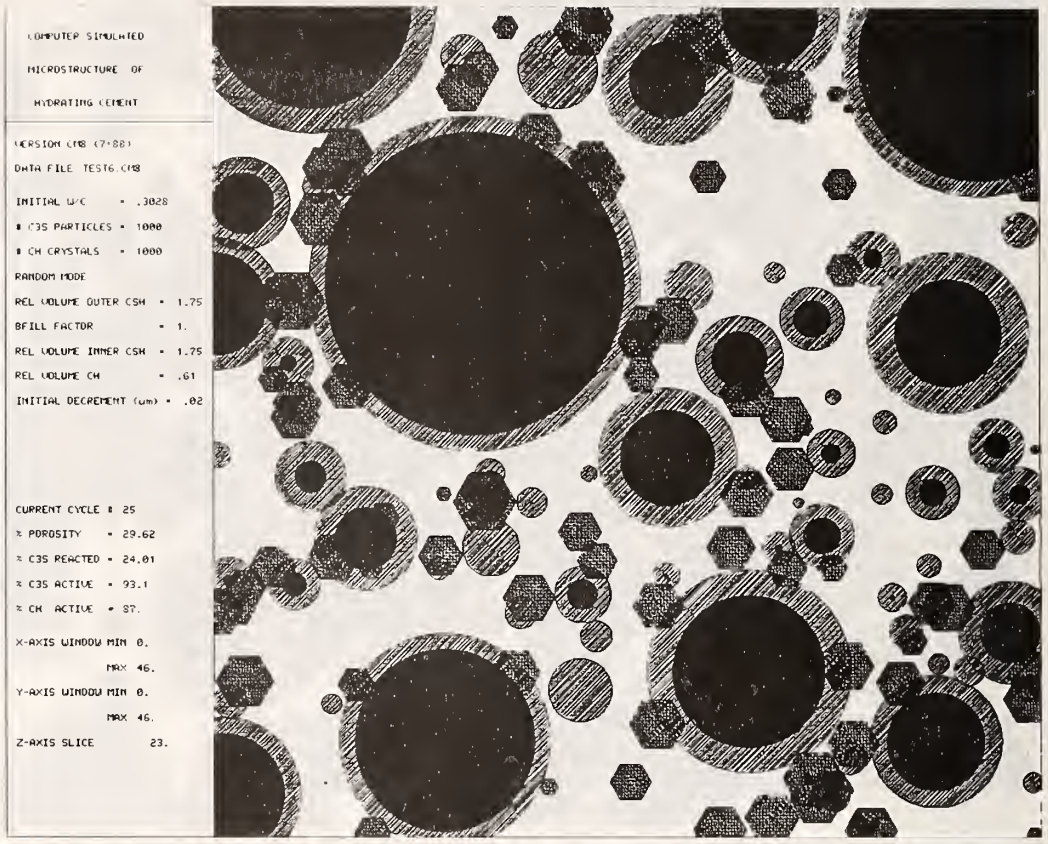


Figure G-9. 2D graphics of 1000 CS particles and 1000 CH crystals, using the default Weibul particle size distribution shown in figure 4, dispersed, after ~50% hydration with no kinetic control and a backfill factor of unity (TEST6.CM8).

APPENDIX H. 3C/3230 USER'S GUIDE

This appendix provides instructions to sign on the 3C/3230 computer system and perform some basic functions. More involved use will require the appropriate manual provided by the hardware or software source. Individual copies of the various manuals will be provided for reference upon specific request, within the limits of the software license or copyright law.

The 3C/3230 computer is currently accessible by two methods, by terminals connected directly to the computer or by terminals connected to the NIST computer network (NBSNET, including PCs with terminal emulation software, see Appendix I).

Signing on to the system can be as a prompted sign-on, shown in the following example:

* SIGNON<cr> or * S<cr>
USERID: abc<cr>
ACCOUNT NUMBER: ##<cr>
PASSWORD: password<cr>
ENVIRONMENT= <cr>

You may prefer the one-line sign-on as follows:

* SIGNON abc,##,password<cr> or * S abc,##,password<cr>

The "abc" is the user selected user-id, usually initials or a project mnemonic, and can be up to eight characters. All jobs and many of the printouts will be referenced by the user-id. If the user-id is consistent, it will aid the system operations and allow you to be notified by the console operator or other users if a system level problem arises. The account number and password must have been assigned by the system manager.

Once you are signed on, a system message is displayed. This often includes important notices concerning problems or scheduled down time.

Three user classes have been established on the Division computer to provide access to different user types (table H-1).

Each user account is restricted to one disk volume for file storage. The initialization file (USERINIT.CSS) in that volume should not be altered or deleted from your account.

The filename structure is an eight character name consisting of an alpha character for the first character and up to 7 additional alpha-numeric characters, a period ".", and up to three additional alpha-numeric characters as the filename "extension". The use of special characters is not permitted in filenames.

General purpose procedural files have been developed on the computer to assist the user in performing routine operations. These files are denoted with the file extension ".CSS". They contain the necessary commands to

Table H-1. Cement Hydration Simulation Model Account Classifications

Class	Type	Access Granted
1	Developer	- Modifications to model - Compilations, library building, linking - Execute model
2	Operator	- Execute model
3	Archive	- Maintained for storing current operating version. - Accessed by coordinator only.

perform operations such as starting the editor, compiling and linking programs and starting user written programs. Examples of these procedures are EDIT and F7D, which are described below. Procedural files also support user-specific parameters submitted at start time. A detailed description of the command substitution system can be found in the "Multi-Terminal Monitor (MTM) Reference Manual", Chapter 6.

A "Help" utility is available in MTM to assist users in the use of various procedures developed for this particular installation. Specialized "Help" systems are also available in the EDIT32 program. The MTM Help system is invoked by entering HELP<cr>.

FORTRAN programs are developed according to the procedures below (fig. H-1). Programs are first created using the editor, started by EDIT<cr>. You will need to reference the EDIT32 manual or help system for more information on the editing commands.

Programs are then compiled; if part of a library, the programs are linked to the library when they are compiled. To compile a program that is not part of a library (e.g., M8A), enter F7D filename,listdevice<cr>. The program is then ready to load and run. To create a library (e.g., M8SIM), each subprogram in the library (e.g., M8G in M8SIM) must be compiled using F7C filename,listdevice<cr>, and incorporated into the library using BULIB filename,libraryname<cr>. The main program in each library (e.g., M8B in M8SIM) must be compiled and linked to the library using F7D filename, listdevice,,,libraryname.LIB<cr>. Programs for the graphics library (M8GRF), which includes TEMPLATE, must be compiled using option H. Subprograms are compiled using F7C filename,listdevice,H<cr>. The main program may be compiled and linked for the Tektronix 4129 using a special program called MAKETK8; or it may be compiled and linked using F7TPL filename,device driver, libraryname,,H<cr>. For the latter procedure, the device driver is TK8 for the Tektronix 4129, and F7TPL.LST may be checked if a compilation error occurs.

In each case, the filename has the extension ".FTN", but the extension is not entered in these commands; the library name has the extension ".LIB", which is only used if shown. The list device can be a printer (PR2:) or a file created by entering "xal filename.LST,in,132" before the command is used. If no listing is desired, use, for example, F7D filename,,,, libraryname.LIB<cr>.

Programs are modified using the editor and compiled as described above. If a subprogram in a library, the modified program must be rebuilt into the library using BULIB filename,libraryname<cr>. The main program and library are then recompiled and linked using F7D filename,listdevice,,, libraryname.LIB<cr>.

Each compiled program is saved as an executable task with the name "filename.TSK". The task is loaded and started by entering LOAD filename<cr> and then assigning files or devices not "opened" within the FORTRAN program by entering the commands ASSIGN ##, filename.ext<cr>. More detail is provided in

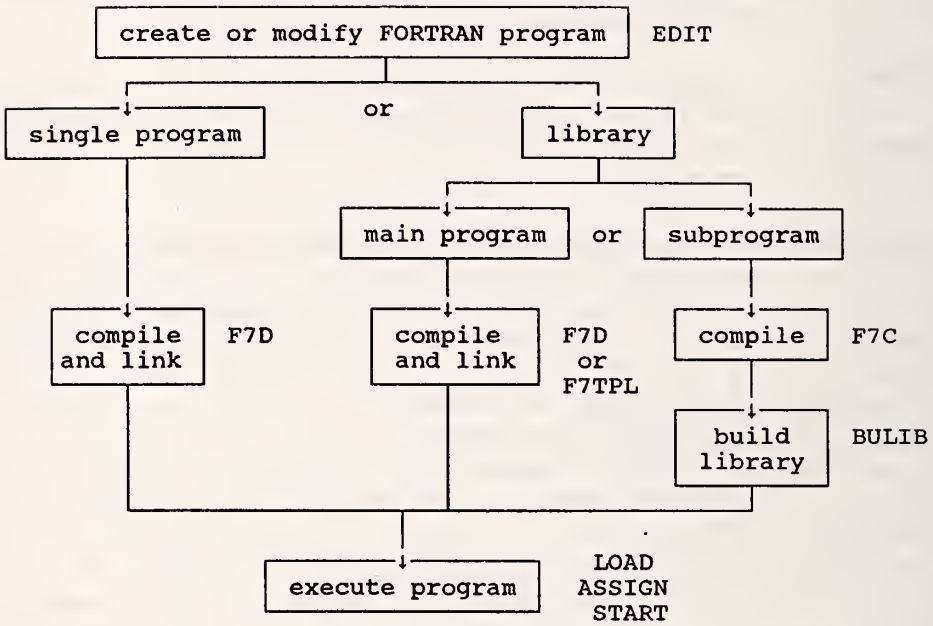


Figure H-1. Flow diagram illustrating steps in development or modification of FORTRAN programs.

the MTM help; refer to the FORTRAN VII manual for OS/32 for additional compiler information.

Printouts can be obtained on the high speed line printer, which has 132 column capability. Printing procedures are listed in the "Help" system. The procedure is to enter PR2 filename.ext<cr>.

System backups are performed daily to prevent potential data loss from hardware failure or power failures. This does not prevent users from deleting new files which have not been backed-up, so care is recommended when you delete old files. Removal of a file is done with the following MTM command: DELETE filename.ext<cr>. Additional techniques are available, but are recommended to be used with extreme care, so as to prevent deleting the wrong file. These techniques involve using the "GETDIR" procedure to create a batch file of deletions. See the "Help Getdir" for more information. Wildcard options are not available for the delete command, hence the batch method.

A file is copied with the command DUPE old filename, new filename<cr>.

Several Graphics and statistical packages and libraries are available for use on the 3C/3230 (Dataplot, Easygraph, TEMPLATE, Movie-BYU, Minitab, and the MATLAB, IMSL, CMLIB, and STARPAC FORTRAN libraries).

A task may be suspended by typing <ctr>C; then type CA<cr> to cancel the task or CO<cr> to continue.

Terminating a user session involves issuing a sign-off command from the user's terminal, by entering the command SIGNOFF<cr>.

APPENDIX I. REMOTE OPERATION

The following instructions have been prepared so that remote users of the Cement Hydration Simulation Model can access the computer, execute the model, run the graphics program, capture the graphics display to local disk, and playback the display on the screen of the local computer. These instructions assume that the user has the required equipment and software available and that the equipment is configured according to the specified parameters.

The recommended communications programs for accessing the model are Kermit[20] for serial communications access and ZAP[21] for Tektronix 4014 graphics terminal emulation. Users may capture graphical displays of the model to local disk for later playback using the ZAP "show" program.

Required equipment:

1. Either Tektronix 4014, 4115, 4129 graphics terminals with remote telecommunications capabilities,

2. or an IBM/Compatible Personal Computer System with minimum of:

   - CGA graphics adapter and display (or capable of emulation)
   - minimum 512 k bytes memory
   - Hayes compatible modem (1200 baud)
   - data communications connection
   - serial port interface port

3. ZAP Graphics Terminal Emulation program and manual, required for PC access only.

NOTE: A remote user access disk with a command file for ZAP and graphs for demonstration is available from L. Kaetzel.

Required configuration and parameters:

1. Communications Settings:

   - 1200 baud
   - even parity
   - 1 stop bit
   - 7 data bits
   - Hayes compatible

2. Graphics Adapter for PC

---

[20]Kermit Distribution, Columbia University Center for Computing Activities, New York, NY.

[21]Obtained from Solution Systems, South Weymouth, MA.

The local computer's graphics capabilities must be set to emulate CGA mode. Refer to the graphics adapter reference guide for the proper procedure.

Operating instructions for accessing the 3C/3230 computer are as follows:

1. Configure your communications equipment according to the above specifications.

2. Turn on your computer with the DOS (version 2.10 or higher) disk in drive A: or boot your computer from drive C:.

NOTE: You may have to perform step 2 before step 1.

NOTE: Omit Steps 3-5, 12-17, and 20 if a Tektronix graphics terminal is used.

3. Start the communications program: ZAP.

4. When the Kermit prompt is displayed: QUIT.

5. Depress the space bar to enter the ZAP communications mode.

6. Establish communications with the 3C/3230 computer: ATDT 1-301-975-2146 or, if your telephone is a pulse tone instead of a touch tone: ATDP 1-301-975-2146.

7. When you see the message, "Enter Number", type: NBSNET.

8. A message will be display when you have connected to the NBSNET; press <ctrl>T. You will see a message TIE: displayed.

9. Connect to the 3C/3230 host: CON 3108.

10. When an asterisk appears, type: SIGNON abc,account #,PASSWORD.

NOTE: The password must be all in capital letters.

11. Run the simulation model (RUNSET8, RUNSIM8, BATSIM8) or the graphs program (e.g., RUNGRF8 M8TK1).

12. You must now set the ZAP communications mode to display graphics: press <alt>C.

13. Put ZAP into graphics mode: MODE 6<cr>.

14. Select drawing mode: 1.

15. When the graphics display is complete, as noted by the beeping sound, type <cr>.

NOTE: If a capture of the graphical display is desired, press <alt>S.

16. Select next drawing mode: 1, or type 7 to exit.

17. Change ZAP mode to text: press <alt>C.

18. Put ZAP into text mode: type MODE 3<cr>.

19. Signoff the 3C/3230 host computer: type SIGNOFF<cr>.

20. Disconnect from nbsnet: type <ctrl>T.

21. Terminate the ZAP program: press <alt>E.

APPENDIX J. GUIDELINES FOR REVISIONS

The decision to incorporate revisions in the model lies with the coordinator, L. Kaetzel, and suggestions for enhancements and revisions should be made to him. These suggestions will be considered based on their contribution to the integrity of the model and their significance.

The coordinator will periodically release new versions of the model and update this manual. It is anticipated that a new version will be released whenever the data file format is altered or when sufficient modifications have been made that a new manual is needed.

The following guidelines have been established to maintain integrity of the software, to provide consistent operating policies, and to coordinate future modifications.

All new code and modifications to existing code must be compatible with the ANSI Standard FORTRAN 77 version X3.9 (1978). Exceptions to this policy will be allowed where the introduction of a modelling concept requires the extension of an auxiliary operation provided by the 3C/3230 computer's FORTRAN 77 library. An example would be to invoke a special purpose subroutine from the library that aids in the display of diagnostic messages to the user, such as a call to "EXIT", or a call to obtain system information such as "DATE" and "TIME".

The modular structure of the model will be maintained. Whenever a new function (operation) is to be introduced into the model, one should consider its applicability to other parts of the program. If the new function is likely to be called from other programs, then it should be separated from the main body of computer code or subroutine and be established as a library subprogram. An example of this would be computer code which operates on a series of vectors or arrays and solves a mathematical equation that has application in more than one calling program.

Developers should review the naming conventions currently given to subprograms, variable names and constants. It is important to maintain these names in order to retain the identification of data elements and program versions.

Data elements (variables, constants) shared by subprograms should be stored in "common" areas instead of being passed as arguments within subroutine calls.

Input/Output operations should include error trapping with English-like statements to diagnose the error. An example of this would be an OPEN statement for an existing file which is not currently catalogued. The program should trap the error code using the "IOSTAT=" convention of the OPEN statement and display a message on the user terminal device indicating the error message and a statement, such as: "Error code: 314 returned from file open . . . file does not exist". Failure to trap errors within the code will cause the computer program to become suspended and will require the user to take unnecessary steps to resolve the problem.

FORTRAN statements should be written using an indented style to improve readability. An example of this style is:

```
IF (VAR1.LT.0.OR.VAR2.GT.1000.) THEN
  IF(J.EQ.1) THEN
    WRITE(1,1000,IOSTAT=ISTAT) VAR1,VAR2
    IF (ISTAT.NE.0) CALL EXIT(255)
    ENDIF
  ELSE
    IF (VAR3.LT.0.OR.VAR3.GT.1000.) GO TO 100
    ENDIF
  ENDIF
ENDIF
```

Another factor to improve program readability is to include comments in the source code that identify program functions and operating parameters. An example of this would be a comment that heads a section of the computer program code generating particles:

```
C *****  CM8 PARTICLE GENERATION  *******
      IF (PD.EQ.3)  THEN
        WRITE(1,*) 'USING DISTRIBUTION TABLE (M8TAB)'
        CALL M8T
      ELSE
        WRITE(1,*) 'USE WEIBULL DISTRIBUTION'
        CALL M8G
      ENDIF
```

FORTRAN 77 requires a certain organization of program statements such as:

```
SUBROUTINE XXX
DATA INITIALIZATION
PROGRAM LOGIC
      .
      .
END
```

To improve readability of the program, this organization has been extended as follows:

```
SUBROUTINE XXX (AS NECESSARY)
DATA INITIALIZATION
IDENTIFICATION OF PROGRAM LOGICAL UNIT ASSIGNMENTS
PROGRAM LOGIC
      .
      .
      .

INPUT/OUTPUT FORMAT STATEMENTS
END OR RETURN
```

APPENDIX K. MODIFICATIONS FROM VERSION 7

Version 8 involved modifications to Version 7 including changes in both the parameters file and the simulation data file, so files generated using any earlier version cannot be used with Version 8.

Version 8 includes the following modifications to Version 7 (in no particular order):

1. used version-specific program and subprogram names throughout (e.g., M8A.FTN rather than MA.FTN)

2. removed the size multiplier factor (SM)

3. changed random number from a function to a variable (RN)

4. added a variable to select the kinetic algorithm

5. changed A, B, and C, to WPA, WPB, and WPC

6. changed data arrays to separate integers and real numbers

7. separated from M8H and M8H2 into a separate subprogram (M8CT) the calculations of redistributed volume

8. changed SD to an integer (as required in M8UNI)

9. added an option to to M8A to review existing parameters

10. changed relative densities to relative volumes

11. incorporated PARPRT subroutine in PARREV of M8A

12. corrected the flow of M8A to bypass parameters not in effect

13. corrected an error in M8A in printing ZMAX

14. added the option to use kinetic algorithms

15. changed M8TAB to use proportion, not number, of particles

16. corrected errors in M8P in the clustering algorithm

17. improved the logic of M8P

18. deleted the sliding algorithm in M8P

19. added a Voronoi algorithm in M8P

20. added a variable NR to M8P for the number of allowed placement attempts before moving to next placement method (initially set equal to 100)

21. changed M8P to terminate the program if unable to place particles by any method

22. changed the name M8P2 to M8R

23. deleted M8P3 as a separate subroutine and incorporated it in M8P

24. converted M8UNI to a subprogram rather than function, passing the variable RN

25. changed the diffusion algorithm to be optional

26. changed M8HA so THK equals the thickness of the shell (outer radius less inner radius of shell) rather than outer radius less anhydrous radius

27. skipped the CH calculation for hydration cycle 0

28. skipped generation of collision table if no hydration cycles requested

29. calculated excluded volume in M8V separately for C-S-H and CH

30. changed M8V2 to M8EV

31. corrected the excluded volume calculation

32. changed M8SAV to add the excluded volume and redistributed volume to data file

33. made the projected CH radius factor a parameter, and changed it from 1.0 to 0.1

34. changed M8B to correct errors noted in rerunning old data with new hydration parameters

35. replaced UNI with the up-to-date version

36. changed the window size in M8GRF to be initially set as the maximum and minimum values of x, y, and z

37. added an option to change window size

38. printing window size in graphics output

39. default colors changed in M8COLOR

40. two-dimensional graphics separated out from M8GRF to be a separate subprogram, M8GR2D

41. corrected flow of M8B when repeating a simulation

42. corrected error in CH placement code in M8P

43. corrected errors in M8HA and M8HB, hydration codes and termination parameter

44. deleted rerun option from M8A

45. corrected error in values of hydration parameters for hydration cycle zero printed in graphics program

46. changed graphics in black-and-white mode to be a hatch pattern instead of gray colors

47. if hydration cycles terminate without completing requested number, HN is set to the number actually run, so that graphics program gives correct range of hydration cycles available

48. in graphics program, added a printout of the proportion of C_3S reacted before menu for drawing graph

49. added calculation of SAX to M8CT for subsequent collisions

50. corrected calculation of VR in M8CT for subsequent collisions

51. established default values for input parameters to avoid division by zero when no hydration cycles requested

52. corrected calculation of hydrated radii with backfill factor 1.0 such that floating point error does not occur

The test configurations listed in table G-1 were run using Version 7 (figs. K-1 through K-5), as well as Version 8 (figs. G-1 through G-9). The results of the two versions are seen to differ in a number of respects.

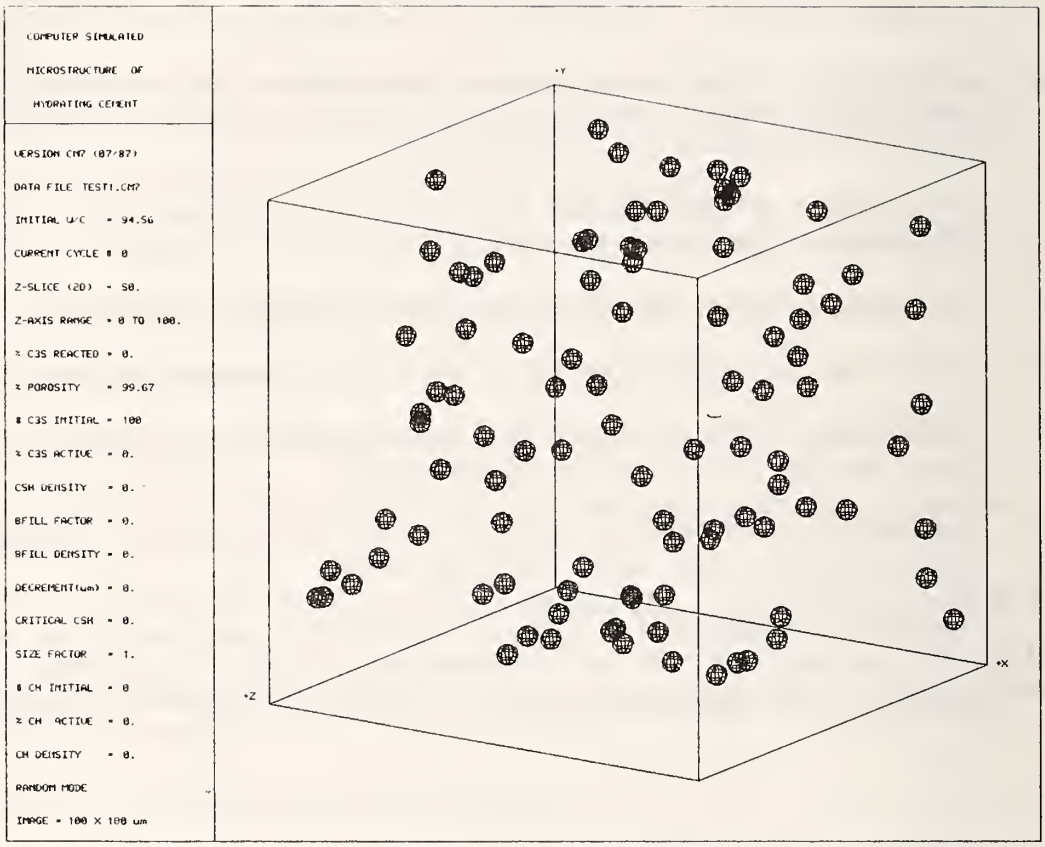


Figure K-1. 3D graphics of 100 unhydrated, monosized (2 μm), dispersed particles (TEST1.CM7) (same as fig. G-1).

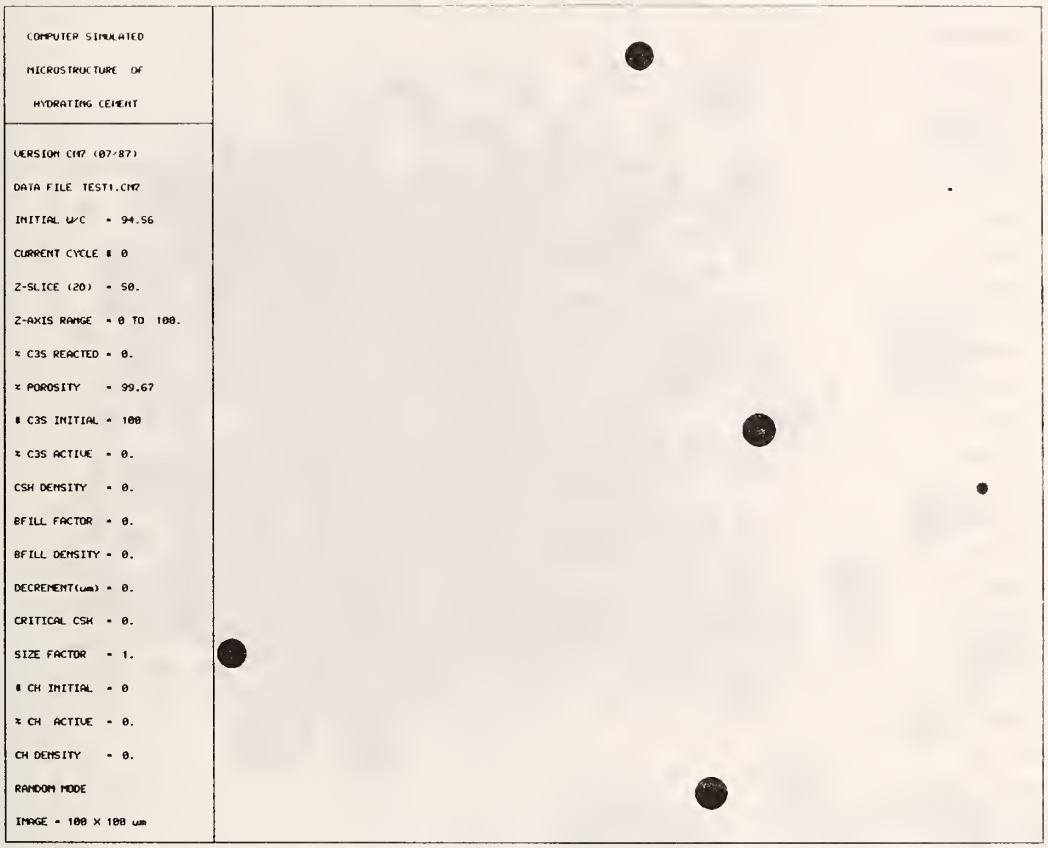


Figure K-2. 2D graphics of the same unhydrated particles in figure K-1 (TEST1.CM7) (same as fig. G-2).

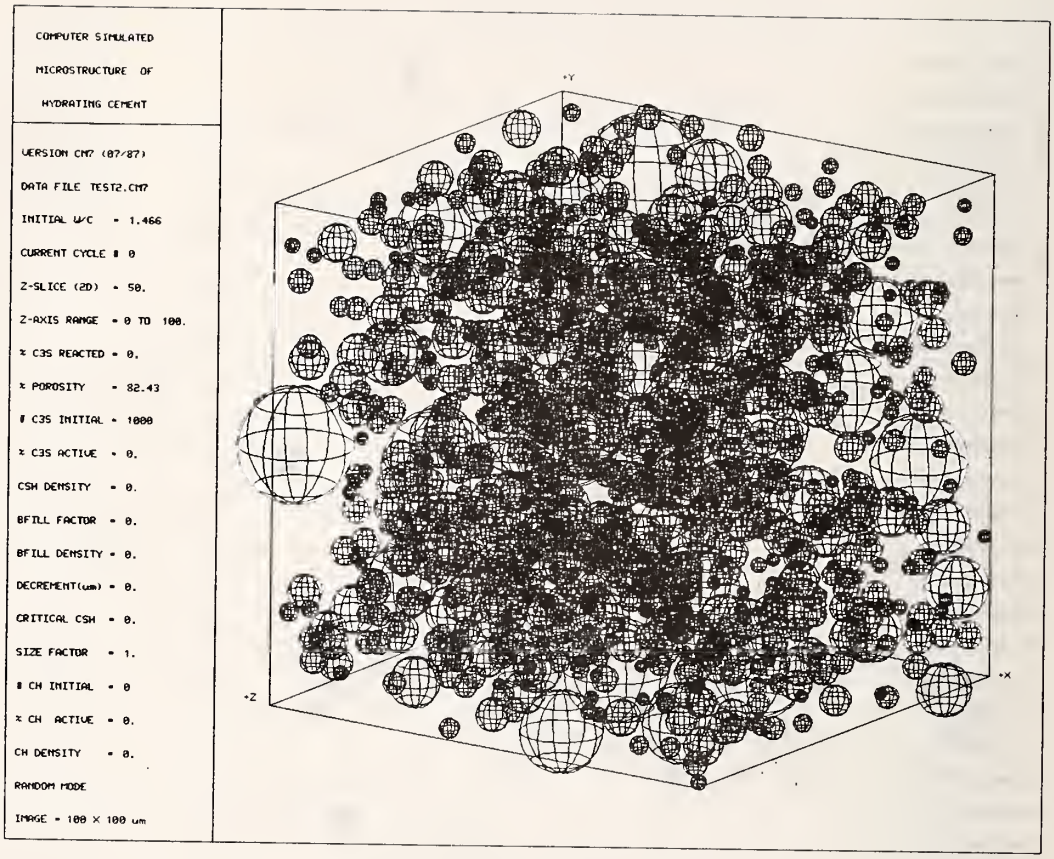


Figure K-3. 3D graphics of dispersed, unhydrated particles using the particle size distribution in figure 5, and using the default box size (TEST2.CM7) (same as fig. G-3).

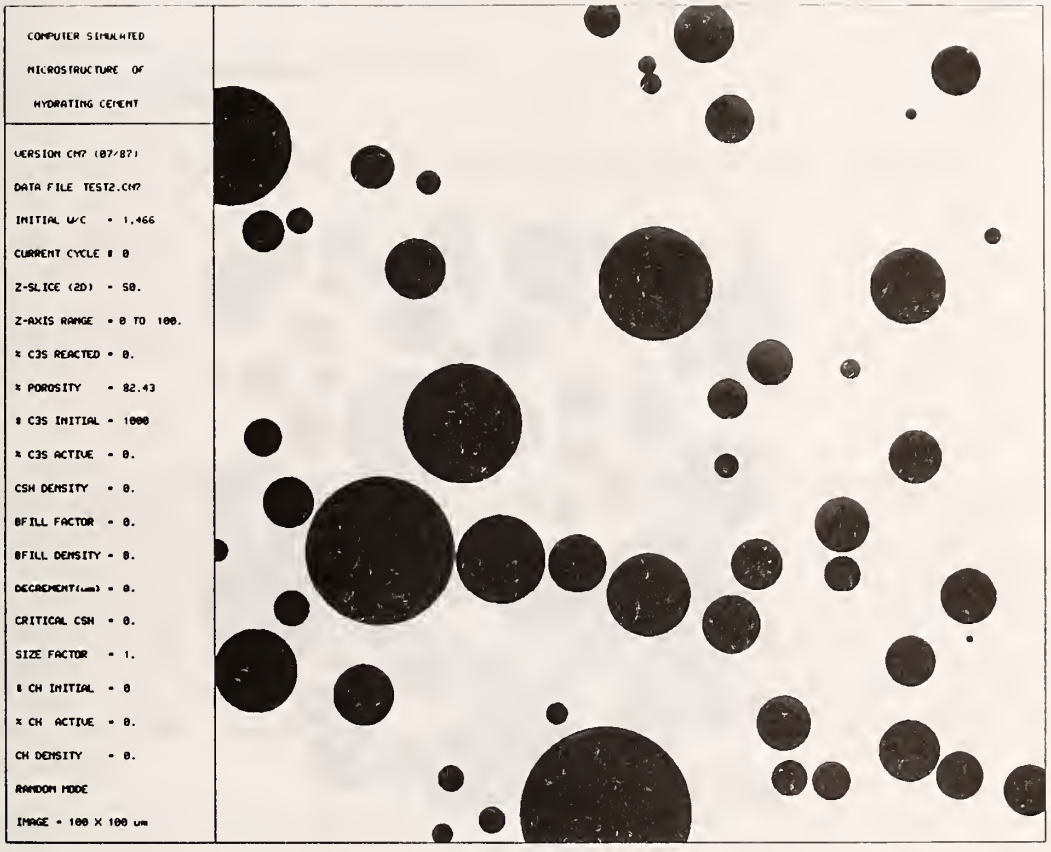


Figure K-4. 2D graphics of the same unhydrated particles in figure K-3 (TEST2.CM7) (same as fig. G-4).

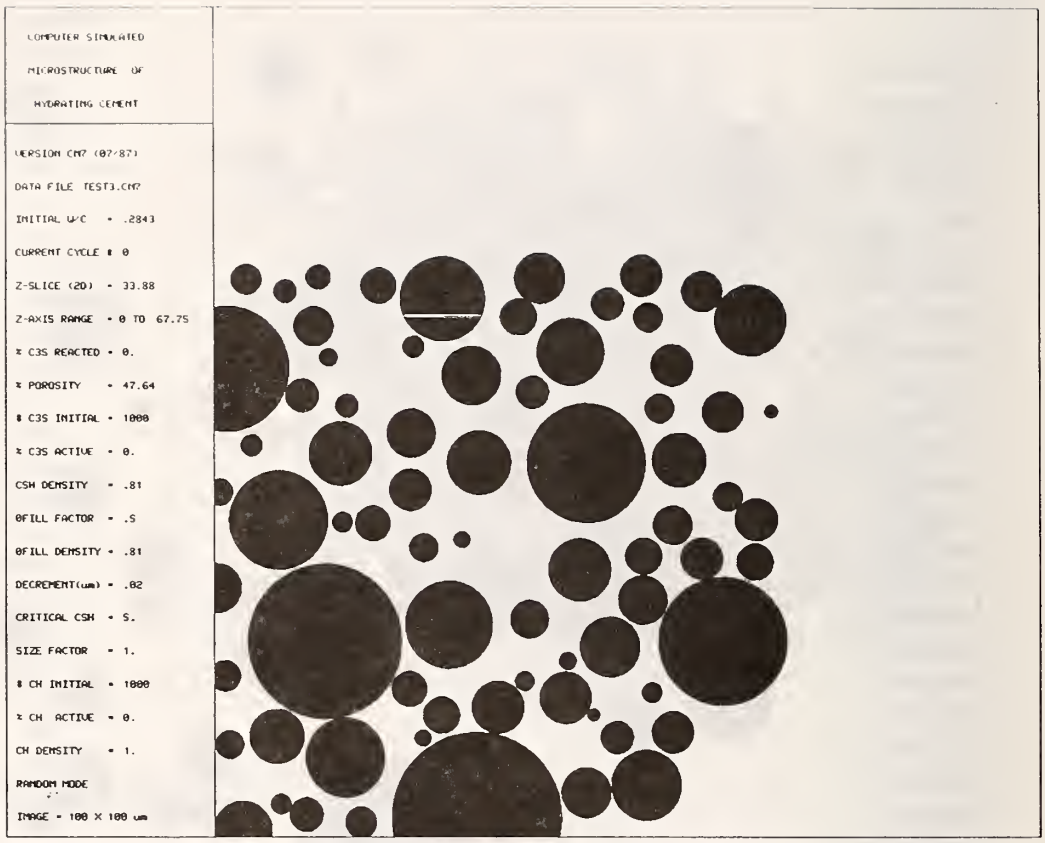


Figure K-5. 2D graphics of unhydrated particles, box size selected to provide water-to-cement ratio 0.30, otherwise similar to figures G-3 and G-4 (TEST3.CM7).

NBS-114A (REV. 2-8C)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NIST/TN-1269	2. Performing Organ. Report No.	3. Publication Date September 1989

4. TITLE AND SUBTITLE

Manual for the Cement Hydration Simulation Model

5. AUTHOR(S)

L. Struble, S. Johnson, M. Hartmann, L. Kaetzel, and H. Jennings

6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions)

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
(formerly NATIONAL BUREAU OF STANDARDS)
U.S. DEPARTMENT OF COMMERCE
GAITHERSBURG, MD 20899

7. Contract/Grant No.

8. Type of Report & Period Covered

Final

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)

Same as item #6

10. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)

This manual describes the Cement Hydration Simulation Model, a computer-based model developed in the Building Materials Division of the Center for Building Technology, National Institute of Standards and Technology (NIST). The model simulates microstructural changes during hydration of tricalcium silicate, the principal constituent of portland cement. Output of the model may be in the form of two- or three-dimensional images, showing the location and size of each constituent.

The model is written in FORTRAN 77. It is interactive and modular. The model is installed on a super-mini computer at NIST, which can be accessed by other, suitably configured computers. The manual provides documentation, instructions, and examples of input and output using the model.

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

cement; computer; documentation; hydration; manual; microstructure; model; simulation

13. AVAILABILITY

☒ Unlimited

☐ For Official Distribution. Do Not Release to NTIS

☒ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

☐ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

125

15. Price

☆ U.S. GOVERNMENT PRINTING OFFICE: 1989— 242 - 200 /05067

USCOMM-DC 6043-P80

NIST Technical Publications

Periodical

Journal of Research of the National Institute of Standards and Technology—Reports NIST research and development in those disciplines of the physical and engineering sciences in which the Institute is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Institute's technical and scientific programs. Issued six times a year.

Nonperiodicals

Monographs—Major contributions to the technical literature on various subjects related to the Institute's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NIST, NIST annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NIST under the authority of the National Standard Data Act (Public Law 90-396). NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published quarterly for NIST by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements are available from ACS, 1155 Sixteenth St., NW., Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Institute on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NIST under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NIST administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NIST research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order the **above** *NIST publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*

Order the **following** *NIST publications—FIPS and NISTIRs—from the National Technical Information Service, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NIST pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NIST Interagency Reports (NISTIR)—A special series of interim or final reports on work performed by NIST for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.

U.S. Department of Commerce
National Institute of Standards and Technology
(formerly National Bureau of Standards)
Gaithersburg, MD 20899