

Pebble-Bed Pebble Motion: Simulation and Applications

Joshua J. Cogliati
Abderrafi M. Ougouag

November 2011



The INL is a U.S. Department of Energy National Laboratory
operated by Battelle Energy Alliance

Pebble-Bed Pebble Motion: Simulation and Applications

**Joshua J. Cogliati
Abderrafi M. Ougouag**

November 2011

**Idaho National Laboratory
Idaho Falls, Idaho 83415**

<http://www.inl.gov>

**Prepared for the
U.S. Department of Energy
Office of Nuclear Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517**

Acknowledgments

Thanks are due to many people who have provided information, comments and insight. I apologize in advance for anyone that I have left out. Thanks go to Javier Ortensi for the encouragement and discussion as he figured out how use the earthquake data and I figured out how to generate it and for the assistance with the graphite dust portion. At INL the following people assisted: Rob Bratton and Will Windes with the graphite literature review, Brian Boer with discussion and German translation, Hongbin Zhang with encouragement and Chinese translation, Hans Gougar for the encouragement and ideas, and Suzette J. Payne for providing me with the earthquake motion data. The following PBMR (South Africa) employees provided valuable help locating graphite literature and data: Frederik Reitsma, Pieter Goede, and Alastair Ramlakan. The Juelich (Germany) people—Peter Pohl, Johannes Fachinger, and Werner von Lensa—provided valuable assistance with understanding AVR. Thanks to Professor Mary Dunzik-Gougar for introducing me to many of these people, as well as encouragement and feedback on this PhD and participating as co-chair on the dissertation committee. Thanks to the other members of my committee, Dr. Michael Lineberry, Dr. Steve C. Chiu and Dr. Steve Shropshire, for providing valuable feedback on the dissertation. Thanks to Gannon Johnson for pointing out that length needed to be tallied separately from the length times force tally for the wear calculation (this allowed the vibration issue to be found). Thanks to Professor Jan Leen Kloosterman of the Delft University of Technology for providing me the PEBDAN program used for calculating Dancoff factors. Thanks also to Akira Tokuhiko, Donald Carlson, and Luo Xiaowei for various suggestions. Thanks to Elizabeth Cogliati for proofreading a previous version of this report. The work was partially supported by the U.S. Department of Energy, Assistant Secretary for the office of Nuclear Energy, under DOE Idaho Operations Office Contract DEAC07-05ID14517. The financial support is gratefully acknowledged.

This report contains work that was first published in the following conferences: Mathematics and Computation, Supercomputing, Reactor Physics and Nuclear and Biological Applications, Palais des Papes, Avignon, France, September 12–15, 2005; HTR2006: 3rd International Topical Meeting on High Temperature Reactor Technology October 1–4, 2006, Johannesburg, South Africa; Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications (M&C + SNA 2007) Monterey, Califor-

nia, April 15–19, 2007; Proceedings of the 4th International Topical Meeting on High Temperature Reactor Technology, HTR2008, September 28–October 1, 2008, Washington, DC USA and PHYSOR 2010 - Advances in Reactor Physics to Power the Nuclear Renaissance Pittsburgh, Pennsylvania, USA, May 9–14, 2010. Most of this report was published as Joshua Cogliati’s dissertation for Idaho State University. Some of this was published in the paper “Survey of Dust Production in Pebble Bed Reactor Cores” in Nuclear Engineering and Design.

Contents

Acknowledgments	iii
Abstract	xi
1 Introduction	1
1.1 Pebble-Bed Reactors Introduction	1
1.2 Report Introduction	3
2 Motivation	5
3 Previous work	7
3.1 Static Friction Overview	9
3.2 Simulation of Mechanics of Granular Material	10
4 Mechanics Model	11
4.1 Overview of Model	11
4.2 Integration	14
4.3 Geometry Modeling	15
4.4 Packing Methods	17
4.5 Typical Parameters	17
5 A New Static Friction Model	19
5.1 Static Friction Formulation	19
5.1.1 Use of Parallel Velocity for Slip Updating	19
5.1.2 Adjustment of Oversize Slips	20
5.1.3 Rotation of Stuck-Slip	20
5.1.4 Differential Equation for Surface Slip Rotating	23
5.2 Testing of Static Friction Model With Pyramid Test	24
5.2.1 Derivation of Minimum Static Frictions	25
5.2.2 Use of Benchmark	28
5.3 Janssen's Formula Comparison	29
6 Unphysical Approximations	33

7	Code Speedup and Parallelization	37
7.1	General Information about Profiling	37
7.2	Overview of Parallel Architectures and Coding	38
7.3	Lock-less Parallel O(N) Collision Detection	39
7.4	MPI Speedup	41
7.5	OpenMP Speedup	43
7.6	Checking the Parallelization	44
7.7	Results	44
8	Applications	47
8.1	Applications in Support of Reactor Physics	47
8.1.1	Pebble Flow and Exit Chute	47
8.1.2	Modeling of Space-Dependent Dancoff Factors	48
8.1.3	Angle of Repose	49
8.1.4	Pebble Ordering with Recirculation	49
8.2	Application to Earthquake Modeling	50
8.2.1	Movement of Earthquakes	51
8.2.2	Method Of Simulation	52
8.2.3	Earthquake Results	52
8.2.4	Earthquake Equations	54
9	Construction of a Dust-Production Framework	59
9.1	HTR-10 Simulation Results	60
10	Future Work	67
11	Summary and Conclusions	69
	Bibliography	71
A	Calculation of Packing Fractions	77
B	Determination of dust-production coefficients	81
B.1	Calculation of Force in Reactor Bed	84
B.2	Prior data on dust production	87
B.3	Prior Prediction Work	89

List of Figures

3.1	Comparison between PEBBLES outputs and Benenati and Brosilow data.	9
4.1	Principle vectors in the interaction of two pebbles.	12
4.2	PRIME method illustration.	17
4.3	Virtual chute method.	18
5.1	Static friction vectors.	21
5.2	Projections to ds.	22
5.3	Static friction vectors for wall.	24
5.4	Sphere location diagram.	24
5.5	Pyramid diagram.	25
5.6	Force diagram.	26
5.7	Relevant forces on wall from pebble.	29
5.8	Comparison with 0.05 and 0.15 μ	31
5.9	Comparison with 0.25 and 0.9 μ	32
7.1	Sample cluster architecture.	39
7.2	Determining nearby pebbles from grid.	40
8.1	Flow field representation (arrow lengths are proportional to local average pebble velocity).	48
8.2	Dancoff factors for AVR.	49
8.3	Angle of repose.	50
8.4	Pebbles before recirculation.	50
8.5	Pebbles after recirculation.	51
8.6	Total earthquake displacement.	53
8.7	0.65 static friction packing over time.	54
8.8	0.35 static friction packing over time.	55
8.9	Different radial packing fractions.	56
8.10	Changes in packing fraction.	57
8.11	Neutronics and thermal effects from J. Ortensi.	58
9.1	Wall pressure.	61
9.2	Average wear force.	61

9.3	Velocity of pebbles at top.	62
9.4	Velocity of pebbles in chute.	62
9.5	Pebble-to-pebble wear length.	63
9.6	Pebble-to-wall wear length.	63
9.7	Pebble-to-pebble wear length by speed.	64
9.8	Pebble-to-pebble wear amount.	64
9.9	Pebble-to-wall wear amount.	65
9.10	Pebble-to-pebble wear amount by speed.	65
A.1	Area inside geometry.	78
A.2	Area outside geometry.	78
B.1	Wear compared to temperature (Luo et al., 2005; Stansfield, 1969).	84
B.2	AVR dimensions.	89

List of Tables

4.1	Typical constants used in simulation.	18
5.1	Sphere location table.	25
7.1	OpenMP speedup results.	45
7.2	MPI/OpenMP speedup results.	45
B.1	Different wear coefficients. Because these use different grades of graphite and experimental setups, they are not always directly comparable.	85
B.2	AVR data.	88
B.3	THTR data.	88
B.4	Helium impurities in AVR and THTR (Nieder, 1990).	88

Abstract

Pebble-bed reactors (PBRs) have moving graphite fuel pebbles. This unique feature provides advantages but also means that simulation of the reactor requires understanding the typical motion and location of the pebbles' granular flow.

This report presents a method for simulating the motion of PBR pebbles. A new mechanical motion simulator, PEBBLES, efficiently simulates the key elements of motion of PBR pebbles. This model simulates gravitational force and contact forces including kinetic and true static friction. It is used for a variety of tasks including simulation of the effect of earthquakes on a PBR, calculation of packing fractions, Dancoff factors, pebble wear, and the pebble force on the walls. The simulator includes a new differential static friction model for the varied geometries of PBRs. A new static friction benchmark was devised via analytically solving the mechanics equations to determine the minimum pebble-to-pebble friction and pebble-to-surface friction for a five-pebble pyramid. This pyramid check and a comparison to the Janssen formula were used to test the new static friction equations.

Because larger pebble-bed simulations involve hundreds of thousands of pebbles and long periods of time, the PEBBLES code has been parallelized. PEBBLES runs on shared memory architectures and distributed memory architectures. For the shared memory architecture, the code uses a new $O(n)$ lock-less parallel collision detection algorithm to determine which pebbles are likely to be in contact with other pebbles. The new collision detection algorithm improves on the traditional non-parallel $O(n \log(n))$ collision detection algorithm. These features combine to form a fast parallel pebble motion simulation.

The PEBBLES code provides new capabilities for understanding and optimizing PBRs, the pebble motion data required to calculate the motion of pebbles during a simulated earthquake, and the ability to determine the contact forces and the lengths of motion in contact. This information, combined with the proper wear coefficients, can be used to determine dust production from mechanical wear. These new capabilities enhance the understanding of PBRs, and the capabilities of the code will allow future improvements in understanding.

Chapter 1

Introduction

1.1 Pebble-Bed Reactors Introduction

Pebble-bed nuclear reactors are a unique reactor type that have been proposed and used experimentally. Pebble-bed reactors were initially developed in Germany in the 1960s when the Arbeitsgemeinschaft Versuchsreaktor (AVR) demonstration reactor was created. In China, the 10-megawatt HTR-10 reactor achieved first criticality in 2000, and more reactors are planned.

Pebble-bed nuclear reactors use graphite spheres (usually about 6 cm in diameter) for containing the fuel of the reactor. The graphite spheres encase smaller spheres of TRistructural-ISotropic (TRISO) particle fuel. Unlike most reactors, the fuel is not placed in an orderly static arrangement. Instead, the graphite spheres are dropped into the top of the reactor, travel randomly down through the reactor core, and are removed from the bottom. The pebbles are then possibly recirculated depending on the amount of burnup of the pebble and the reactor's method of operation.

The first pebble-bed reactor was conceived in 1950s in West Germany using helium-gas cooling and spherical graphite fuel elements. Construction on the AVR 15-MWe reactor was started in 1959 at the Kernforschungsanlage Jülich (KFA) Research Centre. It started operation in 1967 and continued for 21 years until 1988. The reactor operated with an outlet temperature of 950°C. The AVR demonstrated the potential for the pebble-bed reactor concept. Over the course of its operation, loss-of-coolant experiments were successfully performed.

The second pebble-bed reactor was the Thorium High Temperature Reactor (THTR). This reactor was built in West Germany for an electric utility. It was a 300-MWe plant that achieved full power in September 1986. In October 1988, when the reactor was shutdown for maintenance, 35 bolt heads were found in the hot gas ducts leading to the steam generators. The determination was made that the plant could be restarted, but funding difficulties prevented this, and the reactor was decommissioned (Goodjohn, 1991).

The third pebble-bed reactor to be constructed and the only one that is

currently operable is the 10-MWt High Temperature Reactor (HTR-10) at the Tsinghua University in China. Construction started in 1994 and reached first criticality in December 2000. This reactor is helium cooled and has an outlet temperature of 700°C (Wu et al., 2002; Xu and Zuo, 2002).

The use of high-temperature, helium-cooled, graphite-moderated reactors with TRISO fuel particles have a number of advantages. A TRISO particle consists of spherical fuel kernel (such as uranium oxide) surrounded by four concentric layers: (1) a porous carbon buffer layer to accommodate fission-product gases, which limits pressure on the outer layers, (2) an interior pyrolytic carbon layer, (3) a layer of silicon carbide, and (4) an outer layer of pyrolytic carbon. The pyrolytic layers shrink and creep with irradiation, partially offsetting the pressure from the fission products in the interior as well as helping contain the fission gases. The silicon carbide acts as a containment mechanism for the metallic fission products (Miller et al., 2002). These layers provide an in-core containment structure for the radioactive fuel and fission products.

The high-temperature gas reactors have some advantages over conventional light water reactors. First, the higher outlet temperatures allow higher Carnot efficiency to be obtained.¹ Second, the higher temperatures can be used for process heat, which can reduce the use of methane. Third, the high temperature under which TRISO particles can operate allows for the exploitation of the negative temperature coefficient to safely shut down the reactor without use of control rods.² Fourth, the higher temperature is above the annealing temperature for graphite, which safely removes Wigner energy.³ These are advantages of both prismatic and pebble-bed high-temperature reactors (Gougar et al., 2004; Wu et al., 2002).

Pebble-bed reactors, unlike most other reactors types, have moving fuel. This provides advantages but complicates modeling the reactors. A key advantage is that pebble-bed reactors can be refueled online—that is, reactor shutdown is not needed for refueling. As a consequence, the reactors have low excess reactivity, as new pebbles can be added or excess pebbles removed to maintain the reactor at critical. The low excess reactivity removes the need for burnable poisons. A final advantage is that the moving fuel allows the pebble bed to be run with optimal moderation, where both increases and decreases in the fuel-to-moderator ratio cause reduction in reactivity. Ougouag et al. (2004) discuss the advantages of optimal moderation, including improved fuel utilization. However, because the fuel is moving, many traditional methods of modeling nuclear reactors are inapplicable without a method for quantifying the motion. Hence, there is a need to develop usable methods for pebble-bed reactor modeling.

¹The outlet temperatures for pebble-bed reactors have ranged from 700 °C to 950 °C, compared to typical outlet temperatures on the order of 300°C for light water reactors, so the intrinsic Carnot efficiency is higher.

²Control rods are needed for a cold shutdown, however.

³The accumulation of Wigner energy led to the Windscale fire in that lower temperature graphite reactor.

1.2 Report Introduction

This report describes a computer code, PEBBLES, that is designed to provide a method of simulating the motion of the pebbles in a pebble-bed reactor.

Chapter 4 provides the details of how the simulation works. Chapter 5 describes a new static friction model.

Several checks have been made of the PEBBLES code. Figure 3.1 in Chapter 3 compares the PEBBLES simulation to experimentally determined radial packing fractions. Section 5.2 describes a new analytical benchmark that was used to test the static friction model in PEBBLES. Section 5.3 uses the Janssen model to test the static friction in a cylindrical vat.

Motivating all the above are the new applications, including Dancoff factors (Section 8.1.2), calculating the angle of repose (Section 8.3), and modeling an earthquake in Section 8.2.

Chapter 2

Motivation

Most nuclear reactors have fixed fuel, including typical light water reactors. Some reactor designs, such as non-fixed fuel molten salt reactors, have fuel that is in fluid flow. Most designs for pebble-bed reactors, however, have moving granular fuel. Since this fuel is neither fixed nor easily treatable as a fluid, predicting the behavior of the reactor requires the ability to understand the characteristics of the positions and motion of the pebbles. For example, predicting the probability of a neutron leaving one TRISO's fueled region and entering another fueled region depends on the typical locations of the pebbles. A second example is predicting the effect of an earthquake on the reactivity of the pebble-bed reactor. This requires knowing how the positions of the pebbles in the reactor change from the forces of the earthquake. Accurate prediction of the typical features of the flow and arrangement of the pebbles in the pebble-bed reactor would be highly useful for their design and operation.

The challenge is to gain the ability to predict the pebble flow and pebble positions for start-up, steady-state, and transient pebble-bed reactor operation.

The research objective presented in this report is to provide this predicting ability. The approach used is to create a distinct element method computer simulation. The simulation determines the locations and velocities of all the pebbles in a pebble bed reactor and can calculate needed tallies from this data. Over the course of creating this simulation, various applications of the simulation were performed. These models allow the operation of the pebble-bed reactor to be better understood.

Chapter 3

Previous work

Because the purpose of this project is to produce a high-fidelity simulation that can provide predictions of the pattern and flow of pebbles, previous efforts to simulate granular methods and packing were examined. A variety of simulations of the motion of discrete elements have been created for different purposes. Lu et al. (2001) applied a discrete element method (DEM) to determine the characteristics of packed beds used as fusion reactor blankets. Jullien et al. (1992) used a DEM to determine packing fractions for spheres using different non-motion methods. Soppe (1990) used a rain method to determine pore structures in different-sized spheres. The rain method randomly chooses a horizontal position, and then lowers a sphere down until it reaches other existing spheres. This is then repeated to fill up the container. Freund et al. (2003) used a rain method for fluid flow in chemical processing.

The use of non-motion pebble packing methods provides an approximation of the positions of the pebble. Unfortunately, non-motion methods will tend to either under pack or over pack (sometimes both in the same model). For large pebble-bed reactors, the approximately 10-meter height of the reactor core will result in different forces at the bottom than at the top. This will change the packing fractions between the top and the bottom, so without key physics, including static friction and the transmittal of force, non-motion physics models will not even be able to get correct positional information. Non-physics-based modeling cannot be used for predicting the effect of changes in static friction or pebble-loading methods even if only the position data is required.

The initial PEBBLES code for calculation of pebble positions minimized the sum of the gravitational and Hooke's law potential energies by adjusting pebble positions. However, that simulation was insufficient for determining flow and motion parameters and simulation of earthquake packing.

Additional references addressing full particle motion simulation were evaluated. Kohring (1995) created a 3-D DEM simulation to study diffusional mixing and provided detailed information on calculating the kinetic forces for the simulation. The author describes a simple method of calculating static friction. Haile (1997) discusses both how to simulate hard spheres and soft spheres using only

potential energy. The soft sphere method in Haile proved useful for determining plausible pebble positions but is insufficient for modeling the motion. Hard spheres are simulated by calculating the collision results from conservation laws. Soft spheres are simulated by allowing small overlaps and then having a resulting force dependent on the overlap. Soft spheres are similar to what physically happens, in that the contact area distorts, allowing distant points to approach closer than would be possible if the spheres were truly infinitely hard and only touched at one infinitesimal point. Hard spheres are impractical for a pebble-bed due to the frequent and continuous contact between spheres, so soft spheres are used instead. The dissertation by Ristow (1998) describes multiple methods for simulating granular materials. On Ristow's list of methods was a model similar to that used as the kernel of the work supporting this project. Ristow's dissertation mentioned static friction and provided useful references that will be discussed in Section 3.2.

To determine particle flows, Wait (2001) developed a DEM that included only dynamic friction. Concurrently with this project's research, Rycroft et al. (2006b) used a DEM, created for other purposes, to simulate the flow of pebbles through a pebble-bed reactor.

Multiple other discrete-element codes have been created, and PEBBLES is similar to several of the full-motion models. For most of the applications discussed in this report, only a model that simulates the physics with high fidelity is useful. The PEBBLES dynamic friction model is similar to the model used by Wait or Rycroft, but the static friction model incorporates some new improvements that will be discussed later.

In addition to simulation by computer, other methods of determining the properties of granular fluids have been used. Bedenig et al. (1968) used a scale model to experimentally determine residence spectra (the amount of time that pebbles from a given group take to pass through a reactor) for different exit cone angles. Kadak and Bazant (2004) used scale models and small spheres to estimate the flow of pebbles through a full-scale pebble-bed reactor. These researchers also examined the mixing that occurred between different radial zones as the pebbles traveled downward. Bernal et al. (1960) carefully lowered steel spheres into cylinders and shook the cylinders to determine both loose and dense packing fractions. The packing fraction and boundary density fluctuations were experimentally measured by Benenati and Brosilow (1962). The Benenati and Brosilow data have been used to verify that the PEBBLES code was producing correct boundary density fluctuations (see Figure 3.1). Many experiments were performed in the designing and operating of the AVR reactor to determine relevant properties such as residence times and optimal chute parameters (Bäumer et al., 1990). These experiments provide data for testing the implementation of any computational model of pebble flow.

The PEBBLES simulation uses elements from a number of sources and uses standard classical mechanics for calculating the motion of the pebbles based on the forces calculated. The features in PEBBLES have been chosen to implement the necessary fidelity required while allowing run times small enough to accommodate hundreds of thousands of pebbles. The next sections will discuss

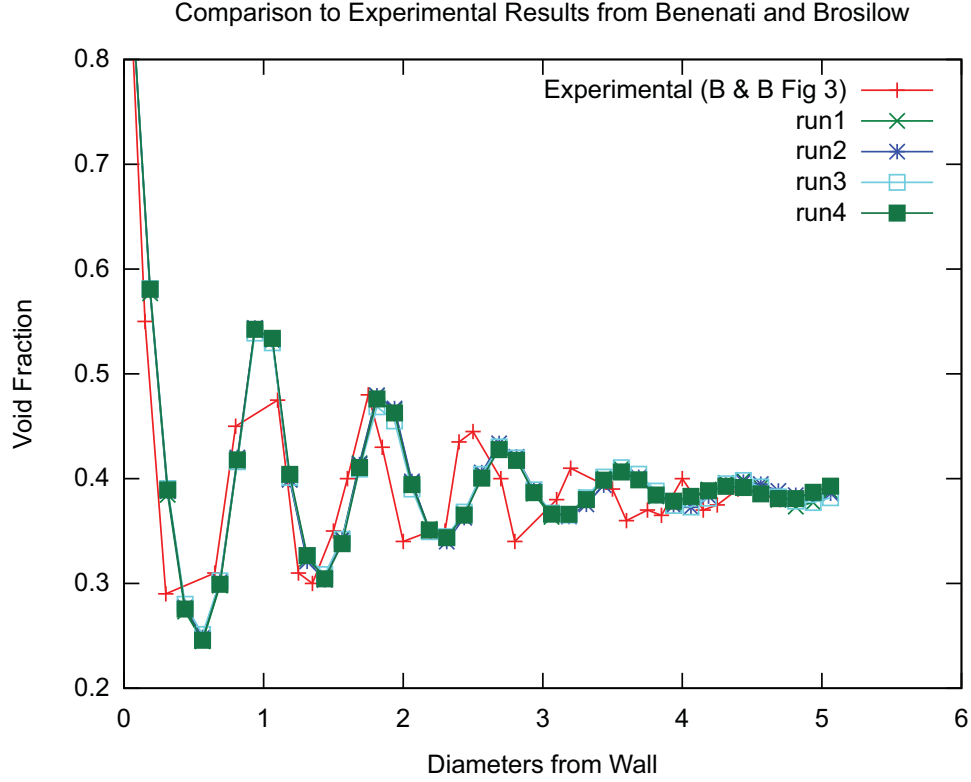


Figure 3.1. Comparison between PEBBLES outputs and Benenati and Brosilow data.

handling static friction.

3.1 Static Friction Overview

Static friction is an important effect in the movement of pebbles and their locations in pebble-bed reactors. This section briefly reviews static friction and its effects in pebble-bed reactors. Static friction is a force between two contacting bodies that counteracts relative motion between them when they are moving sufficiently slowly (Marion and Thornton, 2004). Macroscopically, the maximum magnitude of the force is proportional to the normal force with the following equation:

$$|\mathbf{F}_s| \leq \mu |\mathbf{F}_\perp| \quad (3.1)$$

where μ is the coefficient of static friction, \mathbf{F}_s is the static friction force and

\mathbf{F}_\perp is the normal (load) force.

Static friction results in several effects on granular materials. Without static friction, the angle of the slope of a pile of a material (angle of repose) would be zero (Duran, 1999). Static friction also allows “bridges” or arches to be formed near the outlet chute. If the outlet chute is too small, the bridging will be stable enough to clog the chute. Static friction will also transfer force from the pebbles to the walls. This will result in lower pressure on the walls than would occur without static friction (Sperl, 2006; Walker, 1966).

For an elastic sphere, static friction’s counteracting force is the result of elastic displacement of the contact point. Without static friction, the contact point would *slide* as a result of relative motion at the surface. With static friction, the spheres will experience local shear that distorts their shape so that the contact point remains constant. This change will be called *stuck-slip*, and continues until the counteracting force exceeds $\mu\mathbf{F}_\perp$. When the counteracting force exceeds that value, the contact point changes and slide occurs. The mechanics of this force with elastic spheres were investigated by Mindlin and Deresiewicz (1953). Their work created exact formulas for the force as a function of the past relative motion and force.

3.2 Simulation of Mechanics of Granular Material

Many simulations of granular materials incorporating static friction have been devised. Cundall and Strack (1979) developed an early distinct element simulation of granular materials that incorporated a computationally efficient static friction approximation. Their method involved integration of the relative velocity at the contact point and using the sum as a proxy for the current static friction force. Since their method was used for simulation of 2-D circles, adaptation was required for 3-D granular materials. One key aspect of adaptation is determining how the stuck-slip direction changes as a result of contacting objects’ changing orientation.

Vu-Quoc and Zhang (1999) and Vu-Quoc et al. (2000) developed a 3-D DEM for granular flows. This model was used for simulation of particle flow in chutes. They used a simplification of the Mindlin and Deresiewicz model for calculating the stuck-slip magnitude, and projected the stuck-slip onto the tangent plane each time-step to rotate the stuck-slip force direction. This correctly rotates the stuck-slip but requires that this rotation of the stuck-slip be done as a separate step since it is not written in a differential form.

Silbert et al. (2001) and Landry et al. (2003) describe a 3-D differential version of the Cundall and Strack method. The literature states that particle wall interactions are done identically. The amount of computation of the model is less than the Vu-Quoc, Zhang and Walton model. This model was used for modeling pebble-bed flow (Rycroft et al., 2006a,b). This model, however, does not specify how to apply their differential version to modeling curved walls.

Chapter 4

Mechanics Model

The PEBBLES simulation calculates the forces on each individual pebble. These forces are then used to calculate the subsequent motion and position of the pebbles.

4.1 Overview of Model

The PEBBLES simulation tracks each individual pebble's velocity, position, angular velocity, and static friction loadings. The following classical mechanics differential equations are used for calculating the time derivatives of those variables:

$$\frac{d\mathbf{v}_i}{dt} = \frac{m_i \mathbf{g} + \sum_{i \neq j} \mathbf{F}_{ij} + \mathbf{F}_{ci}}{m_i} \quad (4.1)$$

$$\frac{d\mathbf{p}_i}{dt} = \mathbf{v}_i \quad (4.2)$$

$$\frac{d\omega_i}{dt} = \frac{\sum_{i \neq j} \mathbf{F}_{\parallel ij} \times r_i \hat{\mathbf{n}}_{ij} + \mathbf{F}_{\parallel ci} \times r_i \hat{\mathbf{n}}_{ci}}{I_i} \quad (4.3)$$

$$\frac{d\mathbf{s}_{ij}}{dt} = \mathbf{S}(\mathbf{F}_{\perp ij}, \mathbf{v}_i, \mathbf{v}_j, \mathbf{p}_i, \mathbf{p}_j, \mathbf{s}_{ij}) \quad (4.4)$$

where

\mathbf{F}_{ij} = the force from pebble j on pebble i

\mathbf{F}_{ci} = the force of the container on pebble i

\mathbf{g} = the gravitational acceleration constant

m_i = the mass of pebble i

\mathbf{v}_i = the velocity of pebble i

\mathbf{p}_i = the position vector for pebble i

ω_i = the angular velocity of pebble i

$\mathbf{F}_{\parallel ij}$ = the tangential force between pebbles i and j

$\mathbf{F}_{\perp ij}$ = the perpendicular force between pebbles i and j

r_i = the radius of pebble i

I_i = the moment of inertia for pebble i

$\mathbf{F}_{\parallel ci}$ = the tangential force of the container on pebble i

$\hat{\mathbf{n}}_{ci}$ = the unit vector normal to the container wall on pebble i

$\hat{\mathbf{n}}_{ij}$ = the unit vector pointing from the position of pebble i to that of pebble j

\mathbf{s}_{ij} = the current static friction loading between pebbles i and j

\mathbf{S} = the function to compute the change in the static friction loading.

The static friction model contributes to the $\mathbf{F}_{\parallel ij}$ term, which is also part of the \mathbf{F}_{ij} term. Figure 4.1 illustrates the principal vectors with pebble i going in the \mathbf{v}_i direction and rotating around the ω_i axis, and pebble j going in the \mathbf{v}_j direction and rotating around the ω_j axis.

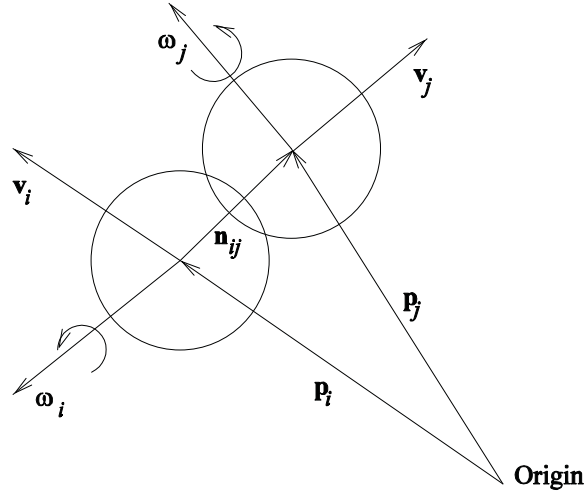


Figure 4.1. Principle vectors in the interaction of two pebbles.

The mass and moment of inertia are calculated assuming spherical symmetry with the equations:

$$m = \frac{4}{3}\pi [\rho_c r_c^3 + \rho_o (r_o^3 - r_c^3)] \quad (4.5)$$

$$I = \frac{8}{15}\pi [\rho_c r_c^5 + \rho_o (r_o^5 - r_c^5)] \quad (4.6)$$

where

r_c = the radius of inner (fueled) zone of the pebble

r_o = the radius of whole pebble

ρ_c = the average density of center-fueled region

ρ_o = the average density of outer non-fueled region.

The dynamic (or kinetic) friction model is based on the model described by Wait (2001). Wait's and the PEBBLES model calculate the dynamic friction using a combination of the relative velocities and pressure between the pebbles, as shown in *Equations (4.7) and (4.8)*:

$$\mathbf{F}_{\perp ij} = h l_{ij} \hat{\mathbf{n}}_{ij} - C_{\perp} \mathbf{v}_{\perp ij}, l_{ij} > 0 \quad (4.7)$$

$$\mathbf{F}_{d\parallel ij} = -\min(\mu |\mathbf{F}_{\perp ij}|, C_{\parallel} |\mathbf{v}_{\parallel ij}|) \hat{\mathbf{v}}_{\parallel ij}, l_{ij} > 0 \quad (4.8)$$

where

C_{\parallel} = the tangential dash-pot constant

C_{\perp} = the normal dash-pot constant

$\mathbf{F}_{\perp ij}$ = the normal force between pebbles i and j

$\mathbf{F}_{d\parallel ij}$ = the tangential dynamic friction force between pebbles i and j

h = the normal Hooke's law constant

l_{ij} = the overlap between pebbles i and j

$\mathbf{v}_{\parallel ij}$ = the component of the velocity between two pebbles perpendicular to the line joining their centers

$\mathbf{v}_{\perp ij}$ = the component of the velocity between two pebbles parallel to the line joining their centers

\mathbf{v}_{ij} = the relative velocity between pebbles i and j

μ = the kinetic friction coefficient.

Equations (4.9-4.12) relate supplemental variables to the primary variables:

$$\mathbf{F}_{ij} = \mathbf{F}_{\perp ij} + \mathbf{F}_{\parallel ij} \quad (4.9)$$

$$\mathbf{v}_{\perp ij} = (\mathbf{v}_{ij} \cdot \hat{\mathbf{n}}_{ij}) \hat{\mathbf{n}}_{ij} \quad (4.10)$$

$$\mathbf{v}_{\parallel ij} = \mathbf{v}_{ij} - \mathbf{v}_{\perp ij} \quad (4.11)$$

$$\mathbf{v}_{ij} = (\mathbf{v}_i + \boldsymbol{\omega}_i \times r_i \hat{\mathbf{n}}_{ij}) - (\mathbf{v}_j + \boldsymbol{\omega}_j \times r_j \hat{\mathbf{n}}_{ji}) \quad (4.12)$$

The friction force is then bounded by the friction coefficient and the normal force to prevent it from being too great:

$$\mathbf{F}_{f\parallel ij} = \mathbf{F}_{s\parallel ij} + \mathbf{F}_{d\parallel ij} \quad (4.13)$$

$$\mathbf{F}_{\parallel ij} = \min(\mu|\mathbf{F}_{\perp ij}|, |\mathbf{F}_{f\parallel ij}|) \hat{\mathbf{F}}_{f\parallel ij} \quad (4.14)$$

where

$\mathbf{F}_{s\parallel ij}$ = the static friction force between pebbles i and j

$\mathbf{F}_{d\parallel ij}$ = the kinetic friction force between pebbles i and j

h_s = the coefficient for force from slip

\mathbf{s}_{ij} = the slip distance perpendicular to the normal force between pebbles i and j

v_{\max} = the maximum velocity under which static friction is allowed to operate

μ = the static friction coefficient when the velocity is less than v_{\max} and the kinetic friction coefficient when the velocity is greater.

These two equations fully enforce the first requirement of a static friction method, $|\mathbf{F}_s| \leq \mu|\mathbf{F}_{\perp}|$.

4.2 Integration

When all the position, linear velocity, angular velocity, and slips are combined into a vector \mathbf{y} , the whole computation can be written as a differential formulation in the form:

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}) \quad (4.15)$$

$$\mathbf{y}(t_0) = \mathbf{y}_0 \quad (4.16)$$

This can be solved by a variety of methods, but the simplest is Euler's method:

$$\mathbf{y}_1 = \mathbf{y}_0 + \Delta t \mathbf{f}(t, \mathbf{y}_0) \quad (4.17)$$

In addition, both the Runge-Kutta method and the Adams-Moulton method can be used for solving this equation. These methods improve the accuracy of

the simulation. However, they do not improve the wall-clock time at the lowest stable simulation, since the additional time required for computation negates the advantage of being able to use somewhat longer time-steps. In addition, when running on a cluster, more data needs to be transferred since the methods allow non-contacting pebbles to affect each other in a single “time-step calculation.”

4.3 Geometry Modeling

For any geometry interaction, two things need to be calculated: the overlap distance l (or, technically, the mutual approach of distant points) and the normal to the surface \hat{n} . The input is the radius of the pebble, r and the position of the pebble, \mathbf{p} with components \mathbf{p}_x , \mathbf{p}_y , and \mathbf{p}_z .

For the floor contact, this is:

$$l = (\mathbf{p}_z - r) - \text{floor_location} \quad (4.18)$$

$$\hat{n} = \hat{z} \quad (4.19)$$

For cylinder contact on the inside of a cylinder, this is:

$$pr = \sqrt{\mathbf{p}_x^2 + \mathbf{p}_y^2} \quad (4.20)$$

$$l = (pr + r) - \text{cylinder_radius} \quad (4.21)$$

$$\hat{n} = \frac{-\mathbf{p}_x}{pr} \hat{x} + \frac{-\mathbf{p}_y}{pr} \hat{y} \quad (4.22)$$

For cylinder contact on the outside of a cylinder, this is:

$$pr = \sqrt{\mathbf{p}_x^2 + \mathbf{p}_y^2} \quad (4.23)$$

$$l = \text{cylinder_radius} + r - pr \quad (4.24)$$

$$\hat{n} = \frac{\mathbf{p}_x}{pr} \hat{x} + \frac{\mathbf{p}_y}{pr} \hat{y} \quad (4.25)$$

For contact on the inside of a cone defined by the $\text{radius} = mz + b$:

$$pr = \sqrt{\mathbf{p}_x^2 + \mathbf{p}_y^2} \quad (4.26)$$

$$z_c = \frac{m(pr - b) + z}{m^2 + 1} \quad (4.27)$$

$$r_c = mz_c + b \quad (4.28)$$

$$x_c = (r_c/pr)\mathbf{p}_y \quad (4.29)$$

$$y_c = (r_c/pr)\mathbf{p}_x \quad (4.30)$$

$$\mathbf{c} = x_c\hat{x} + y_c\hat{y} + z_c\hat{z} \quad (4.31)$$

$$d = \mathbf{p} - \mathbf{c} \quad (4.32)$$

$$l = |d| + r, r_c < pr \quad (4.33)$$

$$\hat{n} = -\hat{d}, r_c < pr \quad (4.34)$$

$$l = r - |d|, r_c \geq pr \quad (4.35)$$

$$\hat{n} = \hat{d}, r_c \geq pr \quad (4.36)$$

These equations are derived from minimizing the distance between the contact point \mathbf{c} and the pebble position \mathbf{p} .

For contact on a plane defined by $ax + by + cz + d = 0$ where the equation has been normalized so that $a^2 + b^2 + c^2 = 1$, the following is used:

$$dp = a\mathbf{p}_x + b\mathbf{p}_y + c\mathbf{p}_z + d \quad (4.37)$$

$$l = r - dp \quad (4.38)$$

$$\hat{n} = a\hat{x} + b\hat{y} + c\hat{z} \quad (4.39)$$

Combinatorial geometry operations can be done. Intersections and unions of multiple geometry types are done by calculating the overlaps and normals for all the geometry objects in the intersection or union. For an intersection, where there is overlap on all the geometry objects, then the smallest overlap and associated normal are kept, which may be no overlap. For a union, the largest overlap and its associated normal are kept.

For testing whether a geometry is correct, a simple check is to fill up the geometry with pebbles using one of the methods described in Section 4.4, and then make sure that linear and angular energy dissipate. Many geometry errors will show up by artificially creating extra linear momentum. For example, if a plane is only defined at the top, but it is possible for pebbles to leak deep into the bottom of the plane, they will go from having no overlap to a very high overlap, which will give the pebble a large force. This results in extra energy being added each time a pebble encounters the poorly defined plane, which will show up in energy tallies.

4.4 Packing Methods

The pebbles are packed using three main methods. The simplest creates a very loose packing with an approximately 0.15 packing fraction by randomly choosing locations, and removing the overlapping ones. These pebbles are then allowed to fall down to compact to a realistic packing fraction.

The second is the PRIME method developed by J. L. Kloosterman (2005). In this method large numbers of random positions (on the order of 100,000 more than will fit) are generated. The random positions are sorted by height, and starting at the bottom, the ones that fit are kept. Figure 4.2 illustrates this process. This generates packing fractions of approximately 0.55. Then they are allowed to fall to compact. This compaction takes less time than starting with a 0.15 packing fraction.

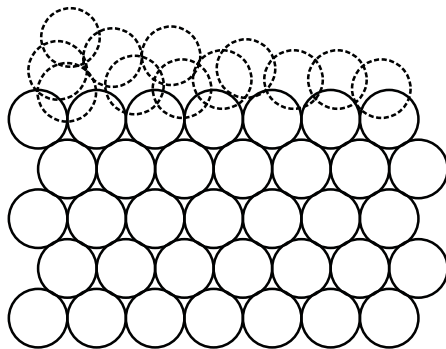


Figure 4.2. PRIME method illustration.

The last method is to automatically generate virtual chutes above the bed where the actual inlet chutes are and then load the pebbles into the chutes. Figure 4.3 shows this in progress. This allows locations that have piles where the inlet chutes are but can be done quicker than a recirculation. The other two methods generate flat surfaces at the top, which is unrealistic, since the surface of a recirculated bed will have cones under each inlet chute.

4.5 Typical Parameters

The typical parameters used with the PEBBLES code are described in Table 4.1. Alternative numbers will be described when used.

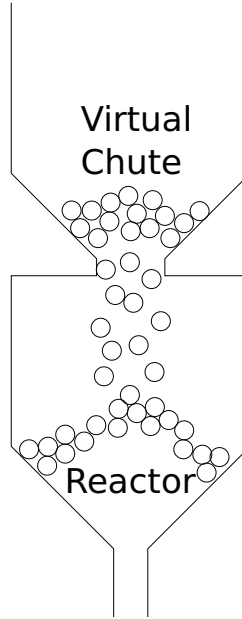


Figure 4.3. Virtual chute method.

Table 4.1. Typical constants used in simulation.

Constant	Value
Gravitational Acceleration g	9.8 m/s ²
radius of pebbles r	0.03 m
Mass of Pebble m	0.2071 kg
Moment of Inertia I	7.367e-05 kg m ²
Hooke's law constant h	1.0e6
Dash-pot constants C_{\parallel} and C_{\perp}	200.0
Kinetic Friction Coefficient μ	0.4 or sometimes 0.25
Static Friction Coefficient μ_s	0.65 or sometimes 0.35
Maximum static friction velocity v_{max}	0.1 m/s

Chapter 5

A New Static Friction Model

The static friction model in PEBBLES is used to calculate the force and magnitude of the static friction force. Other models have been created before to calculate static friction, but the PEBBLES model provides the combination of being a differential model (as opposed to one where the force is rotated as a separate step) and being able to handle the type of geometries that exist in pebble-bed reactors.

The static friction model has two key requirements. First, the force from stuck-slip must be updated based on relative motion of the pebbles. Second, the current direction of the force must be calculated since the pebbles can rotate in space.

5.1 Static Friction Formulation

5.1.1 Use of Parallel Velocity for Slip Updating

For elastic spheres, the true method of updating the stuck-slip force is to use the method of Mindlin and Deresiewicz (1953). This method requires computationally and memory-intensive calculations to track the forces. Instead, a simpler method is used to approximate the force. This method, described by Cundall and Strack (1979) uses the integration of the parallel relative velocity as the displacement. The essential idea is that the farther the pebbles have stuck-slipped at the contact point, the greater the counteracting static friction force needs to be. This is what happens under more accurate models such as Mindlin and Deresiewicz. There are two approximations imposed by this assumption. First, the amount the force changes is independent of the normal force. Second, the true hysteretic effects that depend on details of the loading history are ignored. For simulations where the exact dynamics of static friction are important, these could potentially be serious errors. However, since static friction only

occurs when the relative speed is low, the dynamics of the simulation usually are unimportant. Thus, for most circumstances, the following approximation can be used for the rate of change of the magnitude and non-rotational change of the stuck-slip:

$$\frac{ds_{ij}}{dt} = \mathbf{v}_{\parallel ij} \quad (5.1)$$

5.1.2 Adjustment of Oversize Slips

The slips can build up to unrealistically large amounts, that is, greater than $\mu|\mathbf{F}_{\perp}|$; *Equation (5.1)* places no limit on the maximum size of the slip. The excessive slip is solved at two different locations. First, when the frictions are added together to determine the total friction, they are limited by $\mu|\mathbf{F}_{\perp}|$ in *Equation (4.14)*. This by itself is insufficient, because the slip is storing potential energy that appears anytime the normal force increases. This manifests itself by causing vibration of the pebbles to continue for long periods of time. Two methods for fixing the hidden slip problem are available in PEBBLES. The simplest drops any slip that exceeds the static friction threshold (or an input parameter value somewhat above the static friction threshold so small vibrations do not cause the slip to disappear).

The second method used in PEBBLES is to decrease the slip that is over a threshold value. If the slip is too great, a derivative that is the opposite of the current slip is added as an additional term in the slip-time derivative. This occurs in the following additional term:

$$\frac{ds_{ij}}{dt} = -R(|s_{ij}| - s_{sd}\mu|\mathbf{F}_{\perp ij}|)s_{ij} \quad (5.2)$$

In this, $R(x)$ is the ramp function (which is x if $x > 0$ and 0 otherwise) and s_{sd} is a constant to select how much the slip is allowed to exceed the static friction threshold (usually 1.1 in PEBBLES). This derivative adder is used in most PEBBLES runs since it does allow vibrational energy to decrease, yet does not cause the pyramid benchmark to fail like complete removal of too great slips does.

When using non-Euler integration methods, the change in slip is calculated multiple times. Each time it is calculated, it might be set to be zeroed. In the PEBBLES code, if any of the added up slips for a given contact were set to be zeroed, the final slip is zeroed. This is not an ideal method, but it works well enough.

5.1.3 Rotation of Stuck-Slip

The static friction force must also be rotated so that it is in the plane of contact between the two pebbles. When there is a difference between the pebbles' center

velocities, which changes in the relative pebble center location, change in the direction in the stuck-slip occurs. That is:

$$\mathbf{p}_{in+1} - \mathbf{p}_{jn+1} \approx \mathbf{p}_{in} - \mathbf{p}_{jn} + (\mathbf{v}_{in} - \mathbf{v}_{jn})\Delta t \quad (5.3)$$

First, let $\mathbf{n}_{ijn} = \mathbf{p}_i - \mathbf{p}_j$ and $d\mathbf{n}_{ijn} = \mathbf{v}_i - \mathbf{v}_j$. The cross product $-d\mathbf{n}_{ijn} \times \mathbf{n}_{ijn}$ is perpendicular to both \mathbf{n} and $d\mathbf{n}$ and signed to create the axis around which \mathbf{s} is rotated in a right-handed direction. Then, using the cross product of the axis and \mathbf{s} , $-(d\mathbf{n}_{ij} \times \mathbf{n}_{ijn}) \times \mathbf{s}_{ijn}$ gives the correct direction that \mathbf{s} should be increased.

Next, determine the factors required to make the differential the proper length. By cross product laws,

$$|-(d\mathbf{n}_{ij} \times \mathbf{n}_{ijn}) \times \mathbf{s}_{ijn}| = |d\mathbf{n}_{ij}| |\mathbf{n}_{ijn}| |\mathbf{s}_{ijn}| \sin \theta \sin \phi \quad (5.4)$$

where θ is the angle between \mathbf{n}_{ijn} and $d\mathbf{n}_{ij}$ and ϕ is the angle between $d\mathbf{n}_{ij} \times \mathbf{n}_{ijn}$ and \mathbf{s}_{ijn} .

The relevant vectors are shown in Figure 5.1.

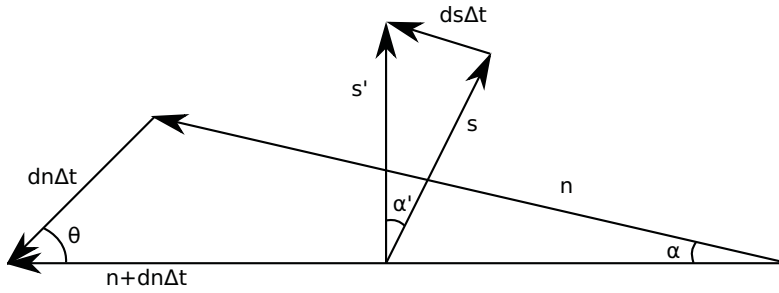


Figure 5.1. Static friction vectors.

The goal is to rotate \mathbf{s} by angle α' which is the “projection” into the proper plane of the angle α that \mathbf{n} rotates by. Since the direction has been determined, for simplicity the figure leaves the indexes off and concentrates on determining the lengths. In Figure 5.1, \mathbf{s} is the old slip vector, \mathbf{s}' is the new slip vector, \mathbf{n} is the vector pointing from one pebble to another. The vector $d\mathbf{n}\Delta t$ is added to \mathbf{n} to get the new \mathbf{n}' , $\mathbf{n} + d\mathbf{n}\Delta t$. The initial condition is that \mathbf{s} and \mathbf{n} are perpendicular. The final conditions are that \mathbf{s}' and \mathbf{n}' are perpendicular, that \mathbf{s} and \mathbf{s}' are the same length, and that \mathbf{s}' is the closest vector to \mathbf{s} as it can be while satisfying the other conditions. There is no requirement that \mathbf{s} or \mathbf{s}' are coplanar with $d\mathbf{n}\Delta t$ (otherwise α' would be equal to α). From the law of sines we have:

$$\frac{|d\mathbf{n}\Delta t|}{\sin \alpha} = \frac{|\mathbf{n}|}{\sin \theta} \quad (5.5)$$

so

$$\sin \alpha = \frac{|d\mathbf{n}\Delta t| \sin \theta}{|\mathbf{n}|} \quad (5.6)$$

In Figure 5.2 the projection to the correct plane occurs. First, by using ϕ , the length of \mathbf{s} is projected to the plane. Note that ϕ is the angle both to \mathbf{s} and to \mathbf{s}' . So, the length of the line on the $d\mathbf{n} \times \mathbf{n}$ plane is $|\mathbf{s}| \sin \phi$, and the length of the straight line at the end of the triangle is $|\mathbf{s}| \sin \phi \sin \alpha$ (note that the chord length is $|\mathbf{s}|(\sin \phi)\alpha$, but as Δt approaches 0, the other can be used). From these calculations, the length of the $d\mathbf{s}\Delta t$ can be calculated with the following formula:

$$d\mathbf{s}\Delta t = \frac{|s| \sin \phi |d\mathbf{n}\Delta t| \sin \theta}{|\mathbf{n}|} \quad (5.7)$$

Since $|-(d\mathbf{n}_{ij} \times \mathbf{n}_{ijn}) \times \mathbf{s}_{ijn}| = |d\mathbf{n}_{ij}| |\mathbf{n}_{ijn}| |\mathbf{s}_{ijn}| \sin \theta \sin \phi$ the formula for the rotation is:

$$\mathbf{s}_{ijn+1} = -\frac{(d\mathbf{n}_{ijn} \times \mathbf{n}_{ijn}) \times \mathbf{s}_{ijn}}{\mathbf{n}^2} \Delta t + \mathbf{s}_{ijn} \quad (5.8)$$

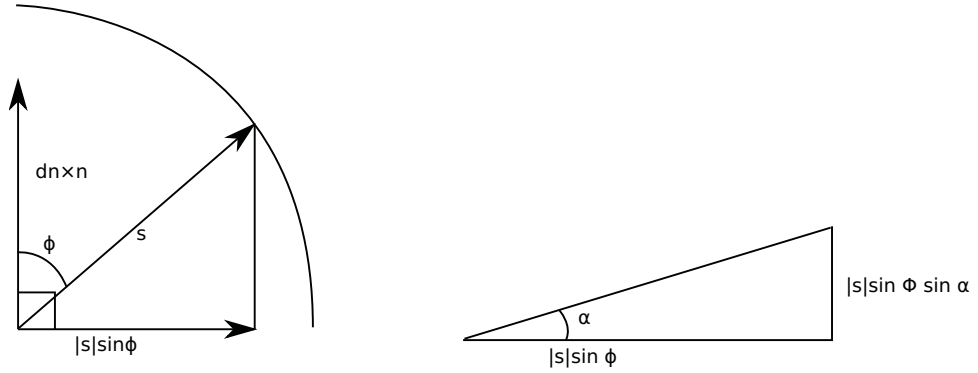


Figure 5.2. Projections to $d\mathbf{s}$.

As a differential equation, this is:

$$\frac{d\mathbf{s}_{ij}}{dt} = -\frac{[(\mathbf{v}_i - \mathbf{v}_j) \times (\mathbf{p}_i - \mathbf{p}_j)] \times \mathbf{s}_{ij}}{|\mathbf{p}_i - \mathbf{p}_j|^2} \quad (5.9)$$

By the vector property $\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = \mathbf{b}(\mathbf{a} \cdot \mathbf{c}) - \mathbf{c}(\mathbf{a} \cdot \mathbf{b})$ and since $(\mathbf{p}_i - \mathbf{p}_j) \cdot \mathbf{s}_{ij} = 0$, this can be rewritten as the version in Silbert et al. (2001):

$$\frac{d\mathbf{s}_{ij}}{dt} = -\frac{(\mathbf{p}_i - \mathbf{p}_j)(\mathbf{s}_{ij} \cdot (\mathbf{v}_i - \mathbf{v}_j))}{|\mathbf{p}_i - \mathbf{p}_j|^2} \quad (5.10)$$

5.1.4 Differential Equation for Surface Slip Rotating

It might seem that the wall interaction could be modeled the same way as the pebble-to-pebble interaction. For sufficiently simple wall geometries, this may be possible, but actual pebble-bed reactor geometries are more complicated and violate some of the assumptions that underpin the derivation. For a flat surface, there is no rotation, so the formula can be entirely dropped. For a spherical surface, it would be possible to measure the curvature at the pebble-to-surface contact point in the direction of relative velocity to the surface. This curvature could then be used as an effective radius in the pebble-to-pebble formulas.

The pebble reactor walls have additional features that violate assumptions made for the derivation. For surfaces such as a cone, the curvature is not, in general, constant, because the path can follow elliptical curves. As well, the curvature has discontinuities where different parts of the reactor join together (for example, the transition from the outlet cone to the outlet chute). At these transitions, the assumption that the slip stays parallel to the surface fails because the slip is parallel to the old surface, but the new surface has a different normal.

Because of the complications with using the pebble-to-pebble interaction, PEBBLES uses an approximation of the “rotation delta.” This is similar to the Vu-Quoc and Zhang (1999) method of adjusting the slip so that it is parallel to the surface each time. Each time when the slip is used, a temporary version of the slip that is properly aligned to the surface is computed and used for calculating the force. As well, a rotation to move the slip more parallel to the surface is also computed.

The rotation is computed as follows. Let the normal direction of the wall at the point of contact of the pebble be \mathbf{n} , and the old stuck-slip be \mathbf{s} . Let a be the angle between \mathbf{n} and \mathbf{s} . $\mathbf{n} \times \mathbf{s}$ is perpendicular to both \mathbf{n} and \mathbf{s} and so this cross product is the axis that needs to be rotated around. Then $(\mathbf{n} \times \mathbf{s}) \times \mathbf{s}$ is perpendicular to this vector, so it is either pointing directly towards \mathbf{n} if a is acute or directly away from \mathbf{n} if a is obtuse. To obtain the correct direction, this vector is multiplied by the scalar $\mathbf{s} \cdot \mathbf{n}$ which has the correct sign from $\cos a$. The magnitude of $(\mathbf{s} \cdot \mathbf{n})[(\mathbf{n} \times \mathbf{s}) \times \mathbf{s}]$ needs to be determined for reasonableness. We define the angle b , which is between $(\mathbf{n} \times \mathbf{s}) \times \mathbf{s}$ and \mathbf{s} . By these definitions the magnitude is $(|\mathbf{s}||\mathbf{n}| \cos a)[(|\mathbf{n}||\mathbf{s}| \sin a)|\mathbf{s}| \sin b]$. b is a right angle since $\mathbf{n} \times \mathbf{s}$ is perpendicular to \mathbf{s} , so $\sin b = 1$. Collecting terms gives the magnitude as $|\mathbf{s}|^3 |\mathbf{n}|^2 \cos a \sin a$, which is divided by $|\mathbf{n} \times \mathbf{s}| |\mathbf{n}| |\mathbf{s}|$ to give the full term the magnitude $|\mathbf{s}| \cos a$. This is the length of the vector that goes from \mathbf{s} to the plane perpendicular to \mathbf{n} . This produces *Equation (5.11)*, which can be used to ensure that the wall stuck-slip vector rotates towards the correct direction.

$$\frac{d\mathbf{s}}{dt} = (\mathbf{s} \cdot \mathbf{n}) \frac{[(\mathbf{n} \times \mathbf{s}) \times \mathbf{s}]}{|\mathbf{n} \times \mathbf{s}| |\mathbf{n}| |\mathbf{s}|} \quad (5.11)$$

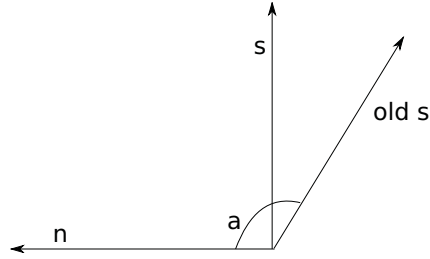


Figure 5.3. Static friction vectors for wall.

5.2 Testing of Static Friction Model With Pyramid Test

Static friction is an important physical feature in the implementation of mechanical models of pebbles' motion in a pebble-bed, and checking its correctness is important. A pyramid static friction test model was devised as a simple tool for verifying the implementation of a static friction model within the code. The main advantages of the pyramid test are that the model test is realistic and that it can be modeled analytically, providing an exact basis for the comparison. The test benchmark consists of a pyramid of five spheres on a flat surface. This configuration is used because the forces acting on each pebble can be calculated simply and the physical behavior of a model with only kinetic friction is fully predictable on physical and mathematical grounds: with only kinetic friction and no static friction, the pyramid will quickly flatten. Even insufficient static friction will result in the same outcome. The four bottom spheres are arranged as closely as possible in a square, and the fifth sphere is placed on top of them, as shown in Figure 5.4.

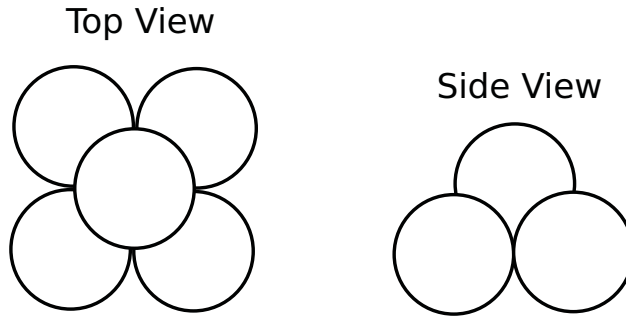


Figure 5.4. Sphere location diagram.

The lines connecting the centers of the spheres form a pyramid with sides $2R$, as shown in Figure 5.5, where R is the radius of the spheres. The length of a in the figure is $\frac{2R}{\sqrt{2}}$, and because b is part of a right triangle, $(2R)^2 - (\frac{2R}{\sqrt{2}})^2 =$

5.2. TESTING OF STATIC FRICTION MODEL WITH PYRAMID TEST25

$b^2 = 4R^2 - \frac{4R^2}{2} = 2R^2$, b has the same length as a , and thus the elevation angle for all vertexes of the pyramid are 45° from horizontal.

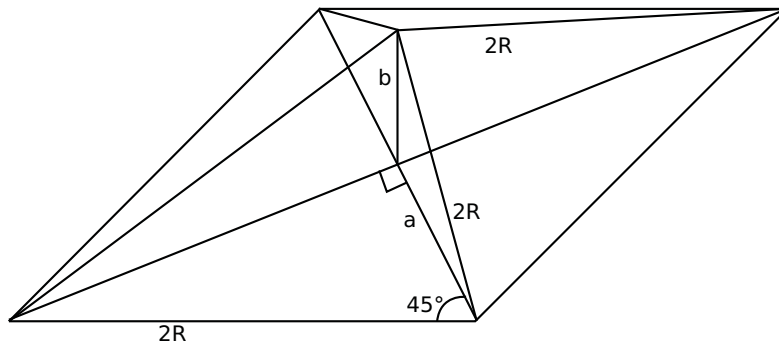


Figure 5.5. Pyramid diagram.

Using the projection of the pyramid summit onto the ground as the origin of the coordinates system, the locations (coordinates) of the sphere centers are given in Table 5.1.

Table 5.1. Sphere location table.

X	Y	Z
$-R$	$-R$	R
R	$-R$	R
$-R$	R	R
R	R	R
0	0	$R(1 + \sqrt{2})$

5.2.1 Derivation of Minimum Static Frictions

The initial forces on the base sphere are the force of gravity mg , and the normal forces \mathbf{Tn} and \mathbf{Fn} , as shown in Figure 5.6. This causes initial stuck-slip, which will cause \mathbf{Fs} to develop to counter the slip, and \mathbf{Ts} to counter the rotation of the base sphere relative to the top sphere. The top sphere will have no rotation because the forces from the four spheres will be symmetric and counteract each other.

The forces on the base sphere are:

- \mathbf{Tn} – Normal force from the top sphere
- \mathbf{Ts} – Static friction force from the top sphere
- mg – Force of gravity on the base sphere
- \mathbf{Fn} – Normal force from floor
- \mathbf{Fs} – Static friction force from the floor

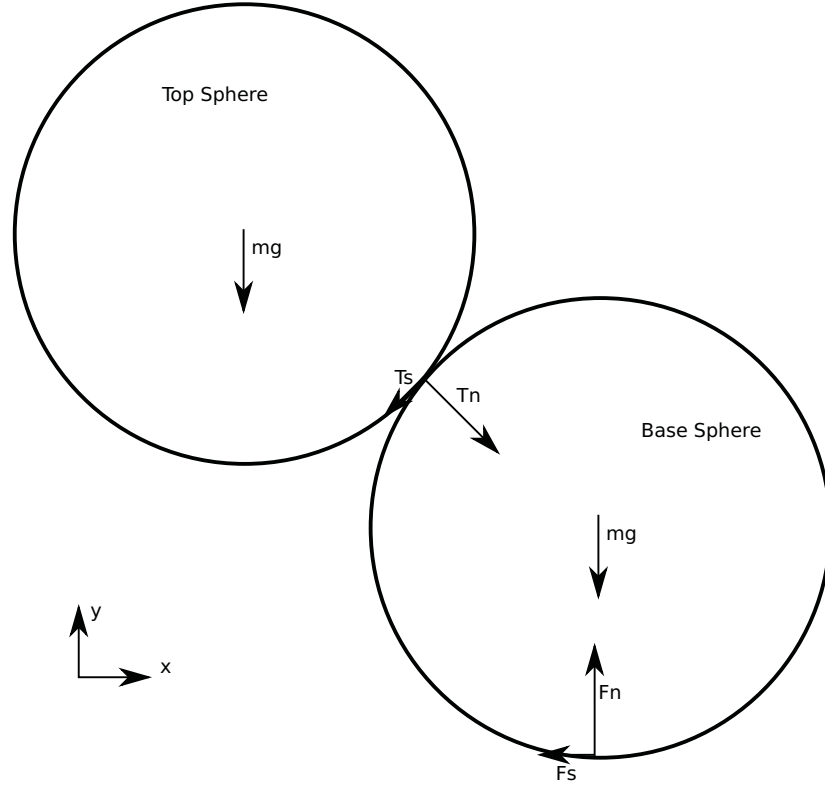


Figure 5.6. Force diagram.

Note that \mathbf{F}_n is larger than \mathbf{T}_n since \mathbf{T}_n is only a portion of the mg force since the top sphere transmits (and splits) its force onto all four base spheres.

There are three requirements for a base sphere to be non-accelerated.

If a base sphere is not rotating, then there is no torque, so:

$$|\mathbf{F}_s| = |\mathbf{T}_s| \quad (5.12)$$

The resultant of all forces must also be zero in the x and the y direction (vector notation dropped since they are in one dimension and therefore scalars) as follows:

$$-F_s - Tsx + Tnx = 0 \quad (5.13)$$

$$-mg - Tsy - Tny + Fn = 0 \quad (5.14)$$

Since the angle of contact between a base sphere and the top sphere is 45° , the following two equations hold (where T_s is the magnitude of \mathbf{T}_s and T_n is

5.2. TESTING OF STATIC FRICTION MODEL WITH PYRAMID TEST²⁷

the magnitude of \mathbf{Tn}):

$$Tsx = Tsy = \frac{Ts}{\sqrt{2}} \quad (5.15)$$

$$Tnx = Tny = \frac{Tn}{\sqrt{2}} \quad (5.16)$$

This changes *Equations (5.13) and (5.14)* into:

$$-Fs - \frac{Ts}{\sqrt{2}} + \frac{Tn}{\sqrt{2}} = 0 \quad (5.17)$$

$$-mg - \frac{Ts}{\sqrt{2}} - \frac{Tn}{\sqrt{2}} + Fn = 0 \quad (5.18)$$

Combining *Equations (5.12) and (5.17)* provides:

$$-Ts - \frac{Ts}{\sqrt{2}} + \frac{Tn}{\sqrt{2}} = 0 \quad (5.19)$$

Which gives the relation:

$$Tn = Ts(\sqrt{2} + 1) \quad (5.20)$$

By the static friction in *Equation (3.1)*,

$$Ts \leq \mu_{sphere} Tn \quad (5.21)$$

Combining *Equations (5.20) and (5.21)* and simplifying gives the requirement that

$$\sqrt{2} - 1 \leq \mu_{sphere} \quad (5.22)$$

For use with testing, the static friction program can be tested twice with a sphere-to-sphere friction coefficient slightly above 0.41421... and one slightly below 0.41421.... In the first case, the pyramid should be stable, and in the second case, the top ball should fall to the floor.

Since $\frac{1}{4}$ of the weight of the top pebble is on one of the base pebbles, the following holds:

$$Fn = \frac{5}{4}mg \quad (5.23)$$

Combining *Equations (5.18) and (5.23)* provides the following equation:

$$\frac{mg}{4} - \frac{Ts}{\sqrt{2}} - \frac{Tn}{\sqrt{2}} = 0 \quad (5.24)$$

Equations (5.17) and (5.24) can be added to produce

$$-Fs - \sqrt{2}Ts + \frac{mg}{4} = 0 \quad (5.25)$$

Using *Equations (5.12) and (5.24)* and solving for F_s gives the following value for F_s :

$$F_s = \frac{mg}{4(1 + \sqrt{2})} \quad (5.26)$$

By the static friction *Equation (3.1)*:

$$F_s \leq \mu_{surface} F_n. \quad (5.27)$$

Substituting the values for F_s and F_n gives:

$$\frac{mg}{4(1 + \sqrt{2})} \leq \mu_{surface} \frac{5}{4} mg \quad (5.28)$$

Simplifying provides the following relation for the surface-to-sphere static friction requirement:

$$\frac{1}{5(1 + \sqrt{2})} \leq \mu_{surface}. \quad (5.29)$$

This can be used similarly to the other static friction requirement by setting the value slightly above 0.08284... and slightly below 0.08284... and making sure that it is stable with the higher value and not stable with the lower value.

This test was inspired by an observation of lead cannon balls stacked into a pyramid. I tried to stack used glass marbles into a five-ball pyramid, and it was not stable. Since lead has a static friction coefficient around 0.9, and used glass has a much lower static friction, the physics of pyramid stability were further investigated, resulting in this benchmark test of static friction modeling.

5.2.2 Use of Benchmark

The benchmark test of two critical static friction coefficients is defined by the following equations. If both static friction coefficients are above the critical values, the spheres will form a stable pyramid. If either or both values are below the critical values, the pyramid will collapse.

$$\mu_{criticalsurface} = \frac{1}{5(1 + \sqrt{2})} \approx 0.08284 \quad (5.30)$$

$$\mu_{criticalsphere} = \sqrt{2} - 1 \approx 0.41421 \quad (5.31)$$

To set up the test cases, the sphere locations from Table 5.1 should be used as the initial locations of the sphere. Then, static friction coefficients for the sphere-to-sphere contact and the sphere-to-surface contact are chosen. The code is then run until either the center sphere falls to the surface or the pyramid obtains a stable state. There are three test cases that are run to test the model.

1. $\mu_{surface} = \mu_{criticalsurface} + \epsilon$ and $\mu_{sphere} = \mu_{criticalsphere} + \epsilon$ which should result in a stable pyramid.

2. $\mu_{surface} = \mu_{criticalsurface} - \epsilon$ and $\mu_{sphere} = \mu_{criticalsphere} + \epsilon$ which should result in a fall.
3. $\mu_{surface} = \mu_{criticalsurface} + \epsilon$ and $\mu_{sphere} = \mu_{criticalsphere} - \epsilon$ which should result in a fall.

For soft sphere models, there are fundamental limits to how close the model's results can be to the critical coefficient. Essentially, as the critical coefficients are approached, the assumptions become less valid. For example, with soft (elastic) spheres, the force from the center sphere will distort the contact angle, so the actual critical value will be different. For the PEBBLES code, an ϵ value of 0.001 is used.

5.3 Testing of the Static Friction Model by Comparison with Janssen's Formula

The pyramid static friction test is used as a simple test of the static friction model. Another test compares the static friction model against the Janssen formula's behavior (Sperl, 2006). This formula specifies the expected wall pressure as a function of depth. This formula only applies when the static friction is fully loaded, that is, when $\mathbf{F}_s = \mu|\mathbf{F}_\perp|$. This condition is generally not satisfied until some recirculation has occurred. Figure 5.7 shows the normal force and the static friction force from a pebble to the wall. With the PEBBLES code, this is only satisfied after recirculation with lower values of the static friction coefficient μ .

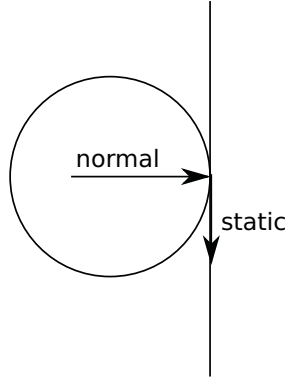


Figure 5.7. Relevant forces on wall from pebble.

The equation used to calculate the pressure on the region R from the normal force in PEBBLES is:

$$p = \frac{1}{R_h 2\pi r} \sum_{i \text{ in } R} |\mathbf{F}_{\perp ci}| \quad (5.32)$$

where

p = the pressure

R_h = the height of the region

r = the radius of the cylinder.

The equation for calculating the Janssen formula pressure is

$$K = 2\mu_{pp}^2 - 2\mu_{pp}\sqrt{\mu_{pp}^2 + 1} + 1 \quad (5.33)$$

$$b = f\rho g \quad (5.34)$$

$$p = \frac{b2r}{4\mu_{wall}} \left(1 - e^{-\frac{4\mu_{wall}Kz}{2r}} \right) \quad (5.35)$$

where

μ_{pp} = the pebble-to-pebble static friction coefficient

μ_{wall} = the pebble to wall

f = the packing fraction

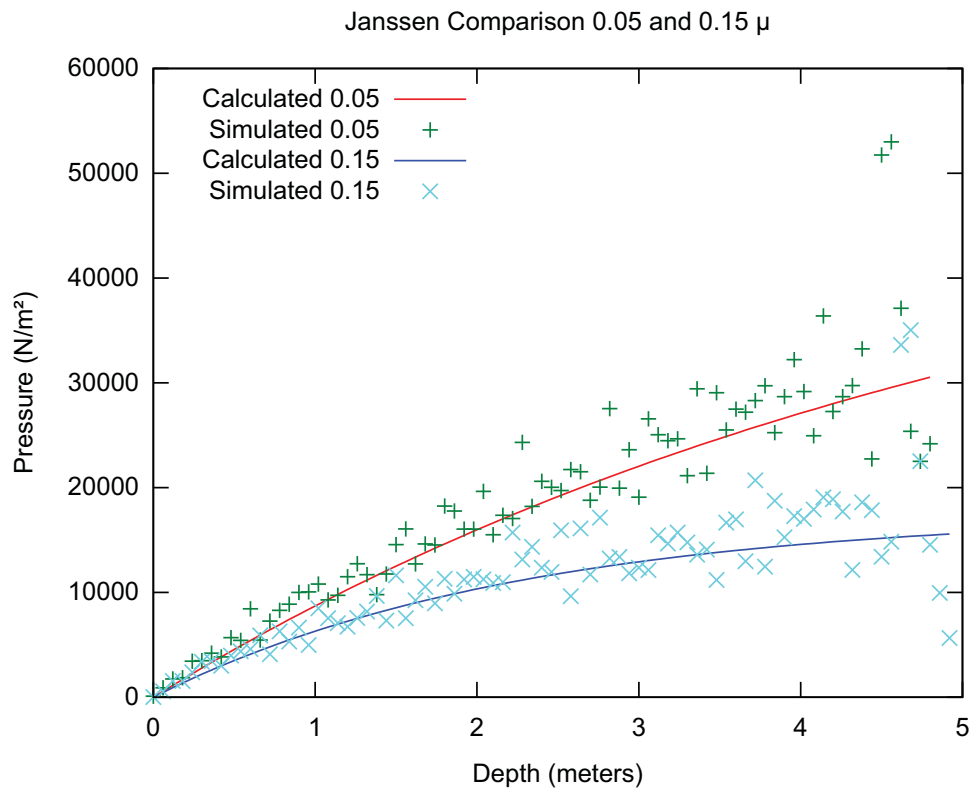
ρ = the density

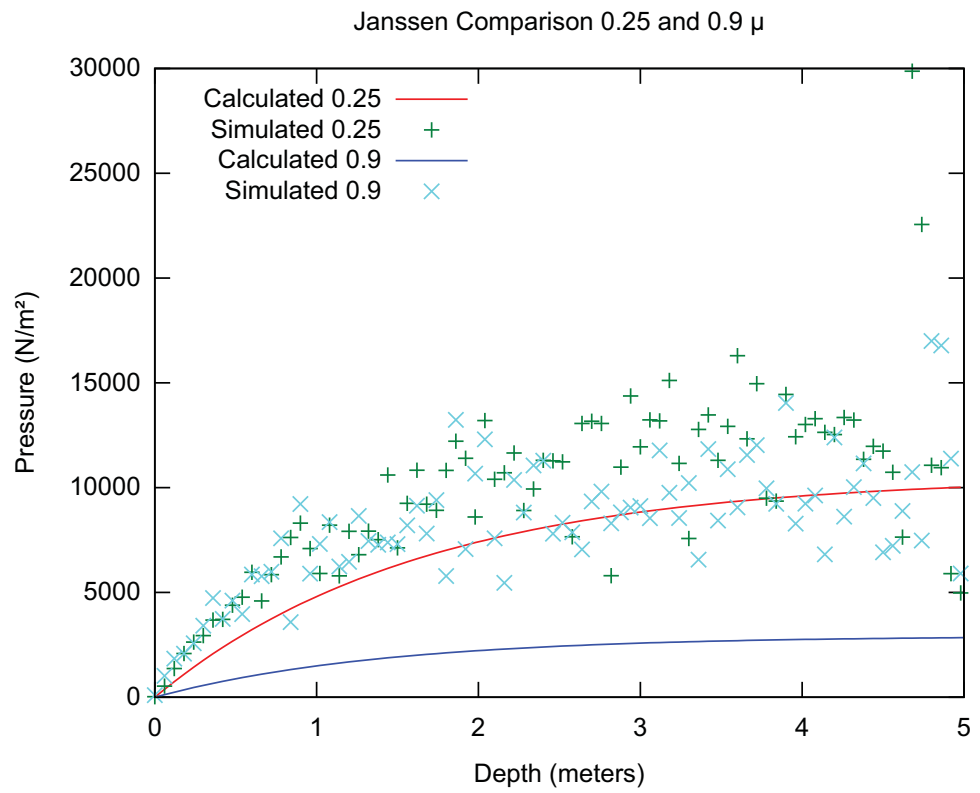
g = the gravitational acceleration

z = the depth that the pressure is being calculated.

For Figures 5.8 and 5.9, a packing fraction of 0.61 is used and a density of 1760 kg/m³ are used. There are 20,000 pebbles packed into a 0.5-meter radius cylinder, and 1,000 are recirculated before the pressure measurement is done.

Figure 5.8 compares the Janssen model with the PEBBLES simulation for static friction values of 0.05 and 0.15. For this case, the Janssen formula and the simulated pressures match closely. Figure 5.9 compares these again. In this case, the 0.25 μ values only approximately match, and the 0.9 static friction pressure values do not match at all. The static friction slip vectors were examined; they are not perfectly vertical, and they are not fully loaded. This results in the static friction force being less than the maximum possible, and thus the pressure is higher since less of the force is removed by the walls.

Figure 5.8. Comparison with 0.05 and 0.15 μ .

Figure 5.9. Comparison with 0.25 and 0.9 μ .

Chapter 6

Unphysical Approximations

Modeling the full physical effects that occur in a pebble-bed reactor mechanics is not computationally possible with current computer resources. In fact, even modeling all the intermolecular forces that occur between two pebbles at sufficient levels to reproduce all macroscopic behavior is probably computationally intractable at the present time. This is partially caused by the complexity of effects such as inter-grain boundaries and small quantities of impurities that affect the physics and different levels between the atomic effects and the macroscopic world. Instead, all attempts at modeling the behavior of pebble-bed reactor mechanics have relied on approximation to make the task computationally practical. The PEBBLES simulation has as high or higher fidelity than past efforts, but it does use multiple unphysical approximations. This chapter will discuss the approximations so that future simulation work can be improved, and an understanding of what limitations exist when applying PEBBLES to different problems.

In different regions of the reactor, the radioactivity and the fission will heat the pebbles differently, and the flow of the coolant helium will distribute this heat around the reactor. This will change the temperature of different parts of the reactor. Since the temperature will be different, the parameters driving the mechanics of the pebbles will be different as well. This includes parameters such as the static friction coefficients and the size of the pebbles, which will change through thermal expansion. As well, parameters such as static friction can also vary depending on the gas which they currently are in and in which they were, since some of the gas tends to remain in and on the carbon surface. Graphite dust produced by wear may also affect static friction in downstream portions of the reactor.

The pebbles in a pebble-bed reactor have helium gas flowing around and past them. PEBBLES and all other pebble-bed simulations ignore effects of this on pebble movement. However, the gas will cause both additional friction when the pebbles are dropping through the reactor, and the motion of gas will cause additional forces on pebbles.

Pebble-bed mechanics simulations use soft spheres. Physically, there will

be deflection of spheres under pressure (even the pressure of just one sphere on the floor), but the true compression is much smaller than what is actually modeled. In PEBBLES, the forces are chosen to keep the compression distance at a millimeter or below. Another effect related to the physics simulation is that force is transmitted via contact. This means the force from one end of the reactor is transmitted at a speed related to the time-step used for the simulation, instead of the speed of sound.

Since simulating billions of time-steps is time consuming, two approximations are made. First, instead of simulating the physical time that pebble-bed reactors have between pebble additions (on the order of 2–5 minutes), new pebbles are added at a rate between a quarter second and two seconds. This may result in somewhat unphysical simulations since some vibration that would have dampened out with a longer time between pebble additions still exists when the next pebble impacts the bed. Second, since full recirculation of all the pebbles is computationally costly, for some simulations, only a partial recirculation or no recirculation is done.

The physics models do not take into account several physical phenomena. The physics do not handle pure spin effects, such as when two pebbles are contacting and are spinning with an axis around the contact point. This should result in forces on the pebbles, but the physics model does not handle this effect since the contact velocity is calculated as zero. In addition, when the pebble is rolling so that the contact velocity is zero because the pebble's turning axis is parallel to the surface and at the same rate as the pebble is moving along the surface, there should be rolling friction, but this effect is not modeled either. As well, the equations used assume that the pebbles are spherically symmetric, but defects in manufacturing and slight asymmetries in the TRISO particle distribution mean that there will be small deviations from being truly spherically symmetric.

The physics model does not match classical Hertzian or Mindlin and Deresiewicz elastic sphere behavior. The static friction model is a simplification and does not capture all the hysteretic effects of true static friction. Effectively, this means that h_s , the coefficient used to calculate the force from slip, is not a constant. In order to fully discuss this, some features of these models will be discussed in the following paragraphs.

Since closed-form expressions exist for elastic contact between spheres, they will be used, instead of a more general case, which lacks closed-form expressions. Spheres are not a perfect representation of the effect of contact between shapes such as a cone and a sphere, but should give an approximation of the size of the effect of curvature.

The amount of contact area and displacement of distant points for two spheres or one sphere and one spherical hole (that is, negative curvature) for elastic spheres can be calculated via Hertzian theory (Johnson, 1985). For two spherical surfaces the following variables are defined:

$$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2} \quad (6.1)$$

and

$$\frac{1}{E^*} = \frac{1 - \nu_1^2}{E_1} + \frac{1 - \nu_2^2}{E_2} \quad (6.2)$$

with R_i the i th's sphere's radius, E_i the Young's modulus, ν_i the Poisson's ration of the material. For a concave sphere, the radius will be negative. Then, via Hertzian theory, the contact circle radius will be:

$$a = \left(\frac{3NR}{4E^*} \right)^{1/3} \quad (6.3)$$

where N is the normal force. The mutual approach of distant points is given by:

$$\delta = \frac{a^2}{R} = \left(\frac{9N^2}{16RE^{*2}} \right)^{1/3} \quad (6.4)$$

Notice that the above differs compared to the Hooke's Law formulation that PEBBLES uses. The maximum pressure will be:

$$p_0 = \frac{3N}{2\pi a^2} \quad (6.5)$$

So as a function of the radii R_1 and R_2 , the circle radius of the contact will be:

$$a = \left(\frac{3N}{4E^*} \left[\frac{1}{R_1} + \frac{1}{R_2} \right]^{-1} \right)^{1/3} \quad (6.6)$$

If m is used for the multiple of negative curvature sphere of the radius of the other, then the equation becomes:

$$a = \left(\frac{3N}{4E^*} \left[\frac{1}{R_1} - \frac{1}{mR_1} \right]^{-1} \right)^{1/3} \quad (6.7)$$

which can be rearranged to:

$$a = \left(\frac{3NR_1}{4E^*} \right)^{1/3} \left(1 - \frac{1}{m} \right)^{-1/3} \quad (6.8)$$

From this equation, as m increases, it has less effect on contact area, so if R_2 is much greater than R_1 , the contact area will tend to be dominated by R_1 rather than R_2 . For example, typical radii in PEBBLES might be an 18-cm outlet chute and a 3-cm pebble, which would put m at 6, so the effect on contact-area radius would be about 33% difference compared to pebble-to-pebble contact-area radius, or 6% compared to a flat surface.¹

¹ Sample values of $k = \left(1 - \frac{1}{m} \right)^{-1/3}$: $m = -1, k = 1.26$ for sphere to sphere, $m = 6, k = 0.94$ sphere to outlet chute and $m = \infty, k = 1$ sphere to flat plane.

To some extent, the actual contact area is irrelevant for calculating the maximum static friction force as long as some conditions are met. Both surfaces need to be of a uniform material. The basic macroscopic description $|F_S| \leq \mu|N|$ needs to hold, so changing the area changes the pressure $P = N/a$, but not the maximum static friction force. If the smaller area causes the pressure to increase enough to cause plastic rather than elastic contact, then through that mechanism, the contact area would cause actual differences in experimental values. PEBBLES also does not calculate plastic contact effects.

The contact area causes an effect through another mechanism. The tangential compliance in the case of constant normal and increasing tangential force, that is, the slope of the curve relating displacement to tangential force, is given in Mindlin and Deresiewicz as:

$$\frac{2 - \nu}{8\mu a} \tag{6.9}$$

Since the contact area radius, a , is a function of curvature, the slope of the tangential compliance will be as well, which is another effect that PEBBLES' constant h_s does not capture.

In summary, when the static friction uses a constant coefficient for h_s , causes two different approximations. First, using the same constants for wall contact when there are different curvatures is an approximation that will give somewhat inconsistent results. Since the equations for spherical contact are dominated by the smaller radius object, this effect is somewhat less but still exists. Second, using the same constant coefficient for different loading histories is an approximation. For a higher fidelity, these effects need to be taken into account.

Chapter 7

Code Speedup and Parallelization

Planned and existing pebble-bed reactors can have on the order of 100,000 pebbles. For some simulations, these pebbles need to be followed for long time periods, which can require computing billions of time-steps. Multiplying the time-steps required by the number of pebbles being computed over leads to the conclusion that large numbers of computations are required. These computations should be as fast as possible and as parallel as possible, to allow relevant calculations to be done in a reasonable amount of time. This chapter discusses the process of speeding up the code and parallelizing it.

The PEBBLES program has three major portions of calculation. The first is determining which pebbles are in contact with other pebbles. The second computational part is determining the time derivatives for all the vectors for all the pebbles. The third computational part is using the derivatives to update the values. Overall, for calculation of a single time-step, the algorithm's computation time is linearly proportional to the number of pebbles, that is, $O(n)$.¹

7.1 General Information about Profiling

There are four different generic parts of the complete calculation that need to be considered for determining the overall speed. The first consideration is the time to compute arithmetic operations. Modern processors can complete arithmetic operations in nanoseconds or fractions of nanoseconds. In the PEBBLES code, the amount of time spent on arithmetic is practically undetectable in wall clock changes. The second consideration is the time required for reading memory and writing memory. For main memory accesses, this takes hundreds of central

¹ $O(n)$: The algorithm scales linearly (n) with increasing input size, so if it runs with 100 pebbles, it takes roughly 10 times as long as when it runs with only 10 pebbles. Or, if it goes from 10 pebbles to 20 pebbles, it will take twice as long to run.

processing unit (CPU) clock cycles, so these times are on the order of fractions of microseconds (Drepper, 2007). Because of the time required to access main memory, all modern CPUs have on-chip caches that contain a copy of the recently used data. If the memory access is in the CPU's cache, the data can be retrieved and written in a small number of CPU cycles. Main memory writes are somewhat more expensive than main memory reads, since any copies of the memory that exist in other processor's caches need to be updated or invalidated. So, for a typical calculation like $a + b \rightarrow c$, the time spent doing the arithmetic is trivial compared to the time spent reading in a and b and writing out c .

The third consideration is the amount of time required for parallel programming constructs. Various parallel synchronization tools such as atomic operations, locks, and critical sections take time. These take an amount of time on the same order of magnitude as memory writes. However, they typically need a read and then a write without any other processor being able to access that chunk of memory in between which requires additional overhead, and a possible wait if the memory address is being used by another process. Atomic operations on x86_64 architectures are faster than using locks, and locks are generally faster than using critical sections. The fourth consideration is network time. Sending and receiving a value can easily take over a millisecond for the round trip time. These four time consuming operations need to be considered when choosing algorithms and methods of calculation.

There are a variety of methods for profiling the computer code. The simplest method is to use the FORTRAN 95 intrinsics `CPU_TIME` and `DATE_AND_TIME`. The `CPU_TIME` subroutine returns a real number of seconds of CPU time. The `DATE_AND_TIME` subroutine returns the current wall clock time in the `VALUES` argument. With `gfortran`, both these times are accurate to at least a millisecond. The difference between two different calls of these functions provides information on both the wall clock time and the CPU time between the calls. (For the `DATE_AND_TIME` subroutine, it is easiest if the days, hours, minutes, seconds, and milliseconds are converted to real seconds past some arbitrary time.) The time methods provide basic information and a good starting point for determining which parts of the program are consuming time. For more detailed profiling, the `oprofile` (opr, 2009) program can be used on Linux. This program can provide data at the assembly language level that makes it possible to determine which part of a complex function is consuming the time. Non-assembly language profilers are difficult to accurately use on optimized code, and profiling non-optimized code is misrepresentative.

7.2 Overview of Parallel Architectures and Coding

Parallel computers can be arranged in a variety of ways. Because of the expense of linking shared memory to all processors, a common architecture is a cluster of nodes with each node having multiple processors. Each node is linked to

other nodes via a fast network connection. The processors on a single node share memory. Figure 7.1 shows this arrangement. For this arrangement, the code can use both the OpenMP (Open Multi-Processing) (ope, 2008) and the MPI (Message Passing Interface) (mpi, 2009) libraries. MPI is a programming interface for transferring data across a network to other nodes. OpenMP is a shared memory programming interface. By using both programming interfaces, high-speed shared memory accesses can be used on memory shared on the node, and the code can be parallelized across multiple nodes.

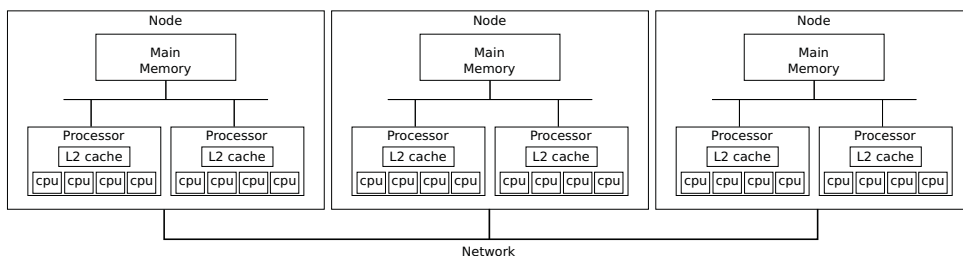


Figure 7.1. Sample cluster architecture.

7.3 Lock-less Parallel $O(N)$ Collision Detection

For any granular material simulation, which particles are in contact must be determined quickly and accurately for each time-step. This is called “collision detection,” though for pebble simulations it might be more accurately labeled “contact detection.” The simplest algorithm for collision detection is to iterate over all the other objects and compare each one to the current object for collision. To determine all the collisions using that method, $O(N^2)$ time is required.

An improved algorithm by Cohen et al. (1995) uses six sorted lists of the lower and upper bounds for each object. (There is one upper bound list and one lower bound list for each dimension.) With this algorithm, to determine the collisions for a given object, the bounds of the current objects are compared to bounds in the list—only objects that overlap the bounds in all three dimensions will potentially collide. This algorithm typically has approximately $O(N \log(N))$ time,² because of the sorting of the bounding lists (Cohen et al., 1995).

A third, faster method, grid collision detection, is available if the following requirements hold: (1) there is a maximum diameter of object, and no object exceeds this diameter, and (2) for a given volume, there is a reasonably small, finite, maximum number of objects that could ever be in that volume. These two constraints are easily satisfied by pebble-bed simulations, since the pebbles are effectively the same size (small changes in diameter occur due to wear and thermal effects). A three-dimensional parallelepiped grid is used over the entire

²In order from slowest to fastest (for sufficiently big N): $O(N^2)$, $O(N \log(N))$, $O(N)$, $O(1)$.

range in which the pebbles are simulated. The grid spacing gs is set at the maximum diameter of any object (twice the maximum radius for spheres).

Two key variables are initialized: $grid_count(x, y, z)$, the number of pebbles in grid locations x, y, z ; and $grid_ids(x, y, z, i)$, the pebble identification numbers (*ids*) for each x, y, z location. The *id* is a unique number assigned to each pebble in the simulation. The spacing between successive grid indexes is gs , so the index of a given x location can be determined by $\lfloor (x - x_{min})/gs \rfloor$, where x_{min} is the zero x index's floor; similar formulas are used for y and z .

The grid is initialized by setting $grid_count(:, :, :) = 0$, and then the x, y, z indexes are determined for each pebble. The $grid_count$ at that location is then atomically³ incremented by one and fetched. Because OpenMP 3.0 does not have a atomic add-and-fetch, the `lock xadd x86_64` assembly language instruction is put in a function. The $grid_count$ provides the fourth index into the $grid_ids$ array, so the pebble *id* can be stored into the *ids* array. The amount of time to zero the $grid_count$ array is a function of the volume of space, which is proportional to the number of pebbles. The initialization iteration over the pebbles can be done in parallel because of the use of an atomic add-and-fetch function. Updating the grid iterates over the entire list of pebbles so the full algorithm for updating the grid is $O(N)$ for the number of pebbles.

Once the grid is updated, the nearby pebbles can be quickly determined. Figure 7.2 illustrates the general process. First, index values are computed from the pebble and used to generate xc , yc , and zc . This finds the center grid location, which is shown as the bold box in the figure. Then, all possible pebble collisions must have grid locations (that is, their centers are in the grid locations) in the dashed box, which can be found by iterating over the grid locations from $xc - 1$ to $xc + 1$ and repeating for the other two dimensions. There are 3^3 grid locations to check, and the number of pebbles in them are bounded (maximum 8), so the time to do this is bounded. Since this search does not change any grid values, it can be done in parallel without any locks.

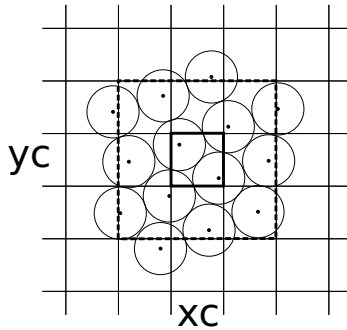


Figure 7.2. Determining nearby pebbles from grid.

³In this chapter, atomic means uncuttable, that is, the entire operation is done in one action without interference from other processors.

Therefore, because of the unique features of pebble-bed pebbles simulation, a parallel lock-less $O(N)$ algorithm for determining the pebbles in contact can be created.

7.4 MPI Speedup

The PEBBLES code uses MPI to distribute the computational work across different nodes. The MPI/OpenMP hybrid parallelization splits the calculation of the derivatives and the new variables geometrically and passes the data at the geometry boundaries between nodes using messages. Each pebble has a primary node and may also have various boundary nodes. The pebble-primary-node is responsible for updating the pebble position, velocity, angular velocity, and slips. The pebble-primary-node also sends data about the pebble to any nodes that are the pebble boundary nodes and will transfer the pebble to a different node if the pebble crosses the geometric boundary of the node. Boundary pebbles are those close enough to a boundary that their data needs to be present in multiple nodes so that the node's primary pebbles can be properly updated. Node 0 is the master node and does processing that is simplest to do on one node, such as writing restart data to disk and initializing the pebble data. The following steps are used for initializing the nodes and then transferring data between them:

1. Node 0 calculates or loads initial positions of pebbles
2. Node 0 creates the initial domain to node mapping
3. Node 0 sends domain to node mapping to other nodes
4. Node 0 sends other nodes their needed pebble data.

Order of calculation and data transfers in main loop:

1. Calculate derivatives for node primary and boundary pebbles
2. Apply derivatives to node primary pebble data
3. For every primary pebble, check with the domain module to determine the current primary node and any boundary nodes
 - (a) If the pebble now has a different primary node, add the pebble id to the transfer list to send to the new primary node
 - (b) If the pebble has any boundary nodes, add the pebble id to the boundary send list to send it to the node for which it is a boundary
4. If this is a time step where Node 0 needs all the pebble data (such as when restart data is being written), add all the primary pebbles to the Node 0 boundary send list
5. Send the number of transfers and the number of boundary sends that this node has to all the other nodes using buffered sends

6. Initialize three Boolean lists of other nodes that this node has:
 - (a) data-to-send-to with “true” if the number of transfers or boundary sends is nonzero, and “false” otherwise
 - (b) received-data-from to “false”
 - (c) received-the-number-of-transfers and the number-of-boundary-sends with “false”
7. While this node has data to send to other nodes and other nodes have data to send to this node loop:
 - (a) Probe to see if any nodes that this node needs data from have data available
 - i. If yes, then receive the data and update pebble data and the Boolean lists as appropriate
 - (b) If there are any nodes that this node has data to send to, and this node has received the number of transfers and boundary sends from, then send the data to those nodes and update the Boolean data send list for those nodes
8. Flush the network buffers so any remaining data gets received
9. Node 0 calculates needed tallies
10. If this is a time to rebalance the execution load:
 - (a) Send wall clock time spent computing since last rebalancing to node 0
 - (b) Node 0 uses information to adjust geometric boundaries to move work towards nodes with low computation time and away from nodes with high computation time
 - (c) Node 0 sends new boundary information to other nodes and needed data to other nodes
11. Continue to next time step and repeat this process until all time-steps have been run.

All the information and subroutines needed to calculate the primary and boundary nodes that a pebble belongs to are calculated and stored in a FORTRAN 95 module named `network_domain_module`. The module uses two derived types: `network_domain_type` and `network_domain_location_type`. Both types have no public components so the implementation of the domain calculation and the location information can be changed without changing anything but the module, and the internals of the module can be changed without changing the rest of the PEBBLES code. The location type stores the primary node and the boundary nodes of a pebble. The module contains subroutines for determining the location type of a pebble based on its position, primary and boundary

nodes for a location type, and subroutines for initialization, load balancing, and transferring of domain information over the network.

The current method of dividing the nodes into geometric domains uses a list of boundaries between the z (axial) locations. This list is searched via binary search to find the nodes nearest to the pebble position, as well as those within the boundary layer distance above and below the zone interface in order to identify all the boundary nodes that participate in data transfers. The location type resulting from this is cached on a fine grid, and the cached value is returned when the location type data is needed. The module contains a subroutine that takes a work parameter (typically, the computation time of each of the nodes) and can redistribute the z boundaries up or down to shift work towards nodes that are taking less time computing their share of information. If needed in the future, the z -only method of dividing the geometry could be replaced with a full 3-D version by modifying the network domain module.

7.5 OpenMP Speedup

The PEBBLES code uses OpenMP to distribute the calculation over multiple processes on a single node. OpenMP allows directives to be given to the compiler that direct how portions of code are to be parallelized. This allows a single piece of code to be used for both the single processor version and the OpenMP version. The PEBBLES parallelization typically uses OpenMP directives to cause loops that iterate over all the pebbles to be run in parallel.

Some details need to be taken into consideration for the parallelization of the calculation of acceleration and torque. The physical accelerations imposed by the wall are treated in parallel, and there is no problem with writing over the data because each processor is assigned a portion of the total zone inventory of pebbles. For calculating the pebble-to-pebble forces, each processor is assigned a fraction of the pebbles, but there is a possibility of the force addition computation overwriting another calculation because the forces on a pair of pebbles are calculated and then the calculated force is added to the force on each pebble. In this case, it is possible for one processor to read the current force from memory and add the new force from the pebble pair while another processor is reading the current force from memory and adding its new force to that value; they could both then write back the values they have computed. This would be incorrect because each calculation has only added one of the new pebble pair forces. Instead, PEBBLES uses an OpenMP ATOMIC directive to force the addition to be performed atomically, thereby guaranteeing that the addition uses the latest value of the force sum and saves it before a different processor has a chance to read it.

For calculating the sum of the derivatives using Euler's method, updating concurrently poses no problem because each individual pebble has derivatives calculated. The data structure for storing the pebble-to-pebble slips (sums of forces used to calculate static friction) is similar to the data structure used for the collision detection grid. A 2-D array exists where one index is the from-

pebble and the other index is for storing *ids* of the pebbles that have slip with the first pebble. A second array exists that contains the number of *ids* stored, and that number is always added and fetched atomically, which allows the slip data to be updated by multiple processors at once. These combine to allow the program to run efficiently on shared memory architectures.

7.6 Checking the Parallelization

The parallelization of the algorithm is checked by running the test case with a short number of time steps (10 to 100). Various summary data are checked to make sure that they match the values computed with the single processor version and between different numbers of nodes and processors. For example, with the NGNP-600 model used in the results section, the average overlap of pebbles at the start of the run is 9.665281e-5 meters. The single processor average overlap at the end of the 100 time-step run is 9.693057e-5 meters, the 2-node average overlap is 9.693043e-5 meters, and the 12-node average overlap is 9.693029e-5 meters. The lower order numbers change from run to run. The start-of-run values match each other exactly, and the end-of-run values match the start-of-run values to two significant figures. However, the three different end-of-run values match to five significant digits. In short, the end values match each other more than they match the start values. The overlap is very sensitive to small changes in the calculation because it is a function of the difference between two positions. During coding, multiple defects were found and corrected by checking that the overlaps matched closely enough between the single processor calculation and the multiple processor calculations. The total energy or the linear energy or other computations can be used similarly since the lower significant digits also change frequently and are computed over all the pebbles.

7.7 Results

The data in Tables 7.1 and 7.2 provide information on the time used with the current version of PEBBLES for running 80 simulation time steps on two models. The NGNP-600 model has 480,000 pebbles. The AVR model contains 100,000 pebbles. All times are reported in units of wall-clock seconds. The single processor NGNP-600 model took 251 seconds, and the AVR single processor model took 48 seconds when running the current version. The timing runs were carried out on a cluster with two Intel Xeon X5355 2.66 GHz processors per node with a DDR 4X InfiniBand interconnect network. The nodes had 8 processors per node. The gfortran 4.3 compiler was used.

Significant speedups have resulted with both the OpenMP and MPI/OpenMP versions. A basic time step for the NGNP-600 model went from 3.138 seconds to 146 milliseconds when running on 64 processors. Since a full recirculation would take on the order of 1.6e9 time steps, the wall clock time for running a full recirculation simulation has gone from about 160 years to a little

Table 7.1. OpenMP speedup results.

Processes	AVR	Speedup	Efficiency	NGNP-600	Speedup	Efficiency
1	47.884	1	100.00%	251.054	1	100.00%
1	53.422	0.89633	89.63%	276.035	0.90950	90.95%
2	29.527	1.6217	81.09%	152.479	1.6465	82.32%
3	21.312	2.2468	74.89%	104.119	2.4112	80.37%
4	16.660	2.8742	71.85%	80.375	3.1235	78.09%
5	13.884	3.4489	68.98%	68.609	3.6592	73.18%
6	12.012	3.98635	66.44%	61.168	4.1043	68.41%
7	10.698	4.4760	63.94%	54.011	4.6482	66.40%
8	9.530	5.0246	62.81%	49.171	5.1057	63.82%

Table 7.2. MPI/OpenMP speedup results.

Nodes	Procs	AVR	Speedup	Efficiency	NGNP-600	Speedup	Efficiency
1	1	47.884	1	100.00%	251.054	1	100.00%
1	8	10.696	4.4768	55.96%	55.723	4.5054	56.32%
2	16	6.202	7.7207	48.25%	30.642	8.1931	51.21%
3	24	4.874	9.8244	40.93%	23.362	10.746	44.78%
4	32	3.935	12.169	38.03%	17.841	14.072	43.97%
5	40	3.746	12.783	31.96%	16.653	15.076	37.69%
6	48	3.534	13.550	28.23%	15.928	15.762	32.84%
7	56	3.285	14.577	26.03%	15.430	16.271	29.05%
8	64	2.743	17.457	27.28%	11.688	21.480	33.56%
9	72	2.669	17.941	24.92%	11.570	21.699	30.14%
10	80	2.657	18.022	22.53%	11.322	22.174	27.72%
11	88	2.597	18.438	20.95%	11.029	22.763	25.87%
12	96	2.660	18.002	18.75%	11.537	21.761	22.67%

over 7 years. For smaller simulation tasks, such as simulating the motion of the pebbles in a pebble-bed reactor during an earthquake, the times are more reasonable, taking about $5e5$ time steps. Thus, for the NNGP-600 model, a full earthquake can be simulated in about 20 hours when using 64 processors. For the smaller AVR model, the basic time step takes about 34 milliseconds when using 64 processors. Since there are fewer pebbles to recirculate, a full recirculation would take on the order of $2.5e8$ time steps, or about 98 days of wall clock time.

Chapter 8

Applications

The knowledge of the packing fraction, flow patterns, and, to a lesser extent, the position of pebbles in the pebble-bed reactor is an essential prerequisite for many in-core fuel cycle design activities as well as for safety assessment studies. Three applications have been done with the PEBBLES code. The major application has been the computation of pebble positions during a simulated earthquake. Two other applications that have been done are calculation of space-dependent Dancoff factors and calculation of the angle of repose for a HTR-10 simulation.

8.1 Applications in Support of Reactor Physics

8.1.1 Pebble Flow Choking and Optimization of Exit Chute Diameter

When the PEBBLES code was first used, it was noticed that certain runs resulted in a blockage of the exit chute at the bottom of the reactor vat. The blockage was initially mistaken for a code bug. However, upon thorough examination of the code and experimentation with various chute diameters, it was determined that the behavior was in fact a realistic representation of an expected occurrence, as pebbles above the chute occasionally form a physical bridge that stops the downward flow of pebbles. A systematic study of the exit chute diameter was undertaken. It was found that a chute diameter of 30 cm results in smooth flow with no arching and hence no pebble flow blockage. This result is close to the value determined experimentally for chute designs (Reitsma, 2005).

Figure 8.1 shows the average velocity as the pebbles pass through a simulated chute. In this chute, the static and dynamic friction are constant. In an actual PBR core, the effect of temperature variation across the core results in conditions with different (locally) He viscosity and graphite-on-graphite friction factors, and hence that in an actually operating PBR with temperature accounted for, the flow pattern would be different, with the possibility of a faster

moving inner core region. However, this cannot be fully determined without a complete simulation, as the effect of temperature on the friction factors and on the He viscosity would produce competing effects on the pebble flow rates.

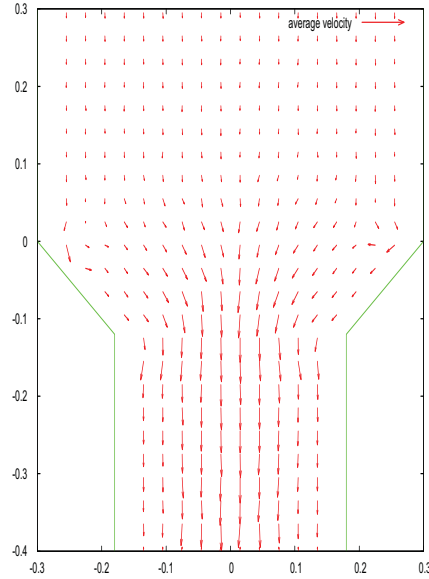


Figure 8.1. Flow field representation (arrow lengths are proportional to local average pebble velocity).

8.1.2 Modeling of Space-Dependent Dancoff Factors

The calculation of Dancoff factors is an example application that needs accurate pebble position data. The Dancoff factor is used for adjusting the resonance escape probability for neutrons. There are two Dancoff factors that use pebble position data. The first is the inter-pebble Dancoff factor that is the probability that a neutron escaping from the fuel zone of a pebble crosses a fuel particle in another pebble. The second is the pebble-pebble Dancoff factor, which is the probability that a neutron escaping one fuel zone will enter another fuel zone without interacting with a moderator nuclide. J. L. Kloosterman (2005) used pebble location information to calculate the probability by ray tracing from fuel lumps until another is hit or the ray escapes the reactor. The PEBBLES code has been used for providing position information to J. L. Kloosterman and A. M. Ougouag's PEBDAN program. This program calculates these factors, as shown in Figure 8.2, which calculates them for the AVR reactor model.

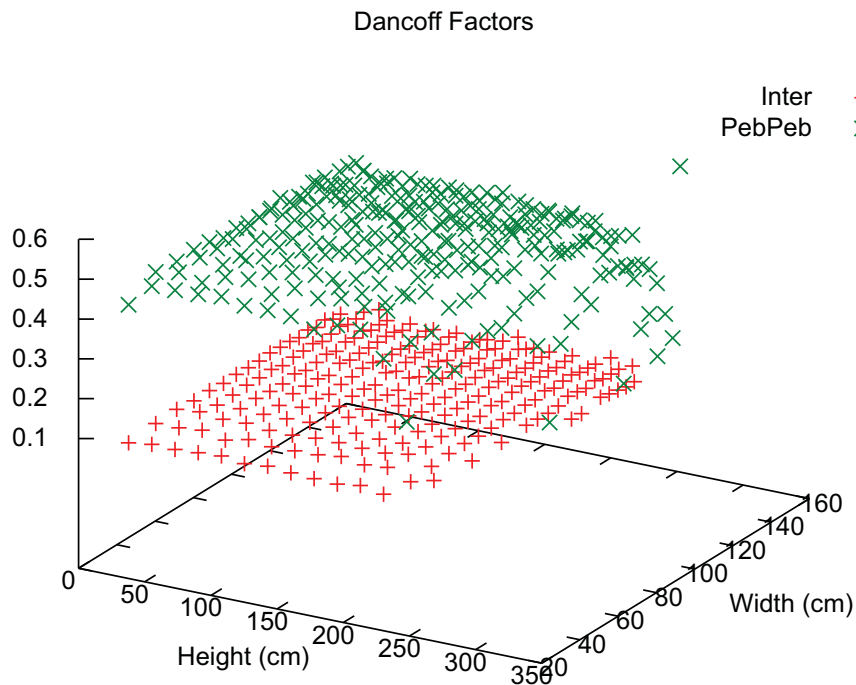


Figure 8.2. Dancoff factors for AVR.

8.1.3 Angle of Repose

When granular matter is poured onto a surface, friction causes the matter not to spread evenly and hence causes the formation of a cone. The cone may have implications for the neutronic behavior. The angle the cone makes with the vertical is dubbed the angle of repose. The PEBBLES code was used for calculating the angle of repose for an analysis of the HTR-10 first criticality (Terry et al., 2006). The pebble-bed code recirculated pebbles to determine the angle that the pebbles would stack at the top of the reactor, as show in Figure 8.3, since this information was not provided, but was needed for the simulation of the reactor.

8.1.4 Pebble Ordering with Recirculation

During experimental work before the construction of the AVR, it was discovered that when the pebbles were recirculated, the ordering in the pebbles increased. Figures 8.4 and 8.5 show that this effect occurs in the PEBBLES simulation as

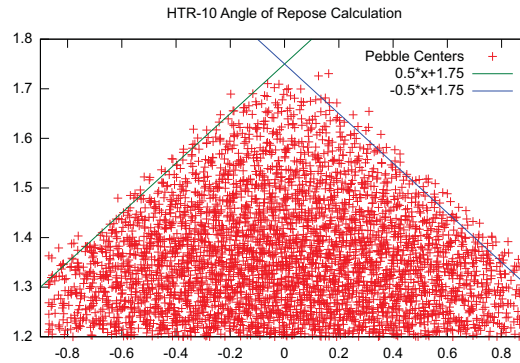


Figure 8.3. Angle of repose.

well. The final AVR design incorporated indentations in the wall to prevent this from occurring.

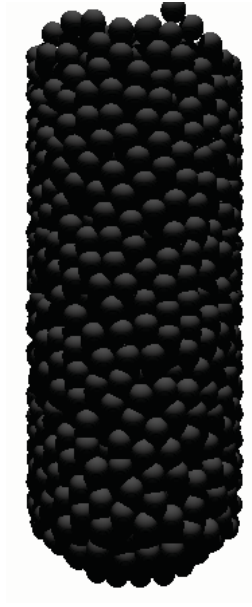


Figure 8.4. Pebbles before recirculation.

8.2 Application to Earthquake Modeling

The packing fraction of the pebbles in a pebble-bed reactor can vary depending on the method of packing and the subsequent history of the packing. This

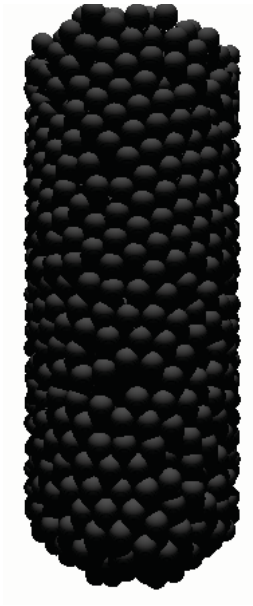


Figure 8.5. Pebbles after recirculation.

packing fraction can affect the neutronics behavior of the reactor. This is one of the most important variables for determining the neutronic behavior, since the packing fraction translates into an effective fuel density. During normal operation starting from a freshly packed bed, the packing fraction will only vary slowly, over the course of weeks and then may possibly stabilize. During an earthquake, this packing fraction can increase suddenly, which is a concern since packing fraction increase can increase the neutron multiplication and cause criticality concerns (Ougouag and Terry, 2001).

The PEBBLES code can simulate this increase and determine the rate of change and the expected final packing fraction, thus allowing the effect of an earthquake to be simulated.

8.2.1 Movement of Earthquakes

The movement of earthquakes has been well studied in the past. The magnitude of the motion of earthquakes is described by the Mercalli scale, which describes the maximum acceleration that a given earthquake will impart to structures. For a Mercalli X earthquake, the maximum acceleration is about 1 g. The more familiar Richter scale measures the total energy release of an earthquake (Lamarsh, 1983), which is not useful for determining the effect on a pebble-bed core. For a given location, the soil properties can be measured, and using soil data and the motion that the bedrock will undergo, the motion on the surface can be simulated. INL had this information generated in order to determine the

motion from the worst earthquake that could be expected over a 10,000-year period (Payne, 2003). This earthquake has roughly a Mercalli IX intensity. The data for such an earthquake are used for the simulation in this report.

8.2.2 Method Of Simulation

The code simulates earthquakes by adding a displacement to the walls of the reactor. As well, the velocity of the walls needs to be calculated. The displacement in the simulation can be specified either as the sum of sine waves or as a table of displacements that specifies the x, y, and z displacements for each time. At each time step both the displacement and the velocity of the displacement are calculated. When the displacement is calculated by a sum of sine functions, the displacement is calculated by adding vector direction for each wave, and then the velocity is calculated from the sum of the first derivative of all the waves. When the displacement is calculated from a table of data, the current displacement is a linear interpolation of the two nearest data points in the table, and the velocity is the slope between them. The walls are then assigned the appropriate computed displacement and velocity. Figure 8.6 shows the total displacement for the INL earthquake simulation specifications used in this paper.

8.2.3 Earthquake Results

The results of two simulations carried out here show a substantially safer behavior than the Ougouag and Terry (2001) bounding calculations. The methodology was applied to a model of the PBMR-400 model, and two different static friction coefficients were tested: 0.65 and 0.35. The packing fraction increased from 0.593 to 0.594 over the course of the earthquake with the 0.65 static friction model; the fastest increase was from 0.59307 to 0.59356 and took place over 0.8 seconds as shown in Figure 8.7. With the 0.35 static friction model, the overall increase was from 0.599 to 0.601. The fastest increase was from 0.59964 to 0.60011 in 0.8 seconds as shown in Figure 8.8. This is remarkably small when compared to the bounding calculation packing fraction increase rate of 0.129 sec^{-1} in free fall.¹ Both computed increases and packing fraction change rates are substantially below the free-fall bounding rate and packing fraction change of a transition from 0.60 to 0.64 in 0.31 seconds. The computed rate and the total packing fraction increase are in the range that can be offset by thermal feedback effects for uranium-fueled reactors.

During the course of the earthquake, the boundary density fluctuations (that is, the oscillations in packing fraction near a boundary) are observed to increase in amplitude. Figure 8.9 shows the packing fraction before the earthquake and after the earthquake in the radial direction. These were taken from 4 to 8 meters above the fuel outlet chute in the PBMR-400 model. All the radial locations have increased packing compared to the packing fraction before the earthquake,

¹The free-fall rate is determined by calculating the packing fraction increase if the pebbles were in gravitational free fall.

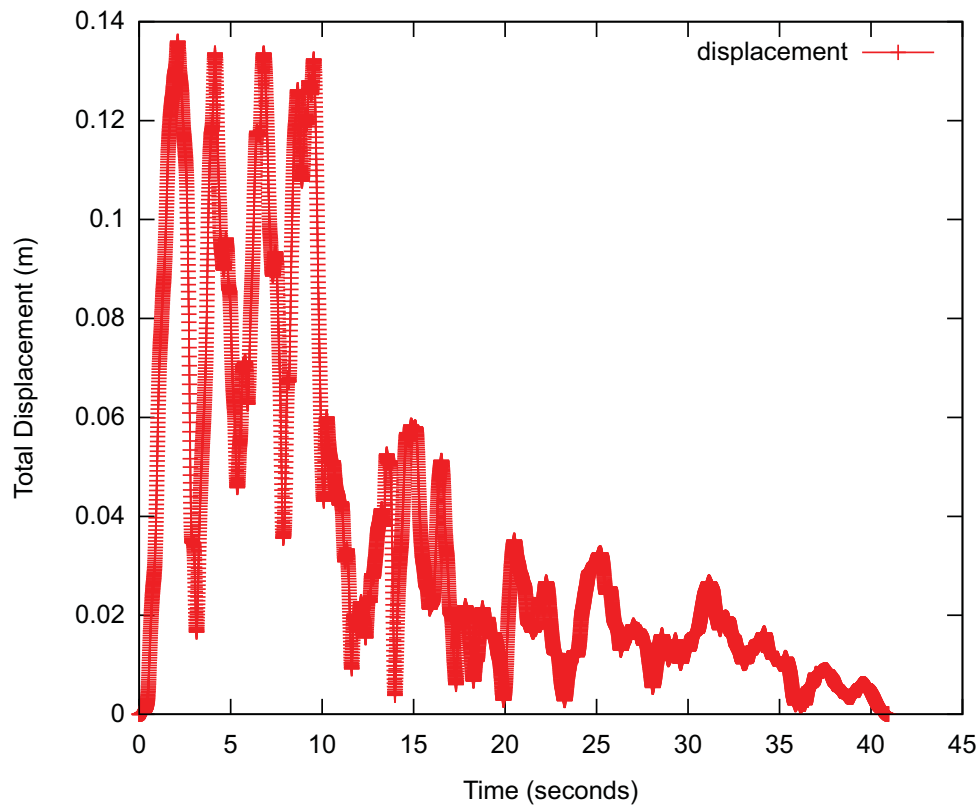


Figure 8.6. Total earthquake displacement.

but the points that are at boundary density fluctuation peaks increase the most. This effect can be seen in Figure 8.10, which shows the increase in packing fraction before the earthquake and after.

A previous version of the positional data from the earthquake simulation was provided to J. Ortensi. He used this data to simulate the effects of an earthquake on a pebble-bed reactor (Ortensi, 2009). Essentially, two factors cause an increase in reactivity: the first is the increased density of the pebbles and the second is due to the control rods being at the top of the reactor, so when the top pebbles move down the control rod worth (effect) decreases. However, the reactivity increase causes the fuel temperature to rise, which causes Doppler broadening and more neutrons are absorbed by the ^{238}U , which causes the reactivity to fall. Figure 8.11 shows an example of this.

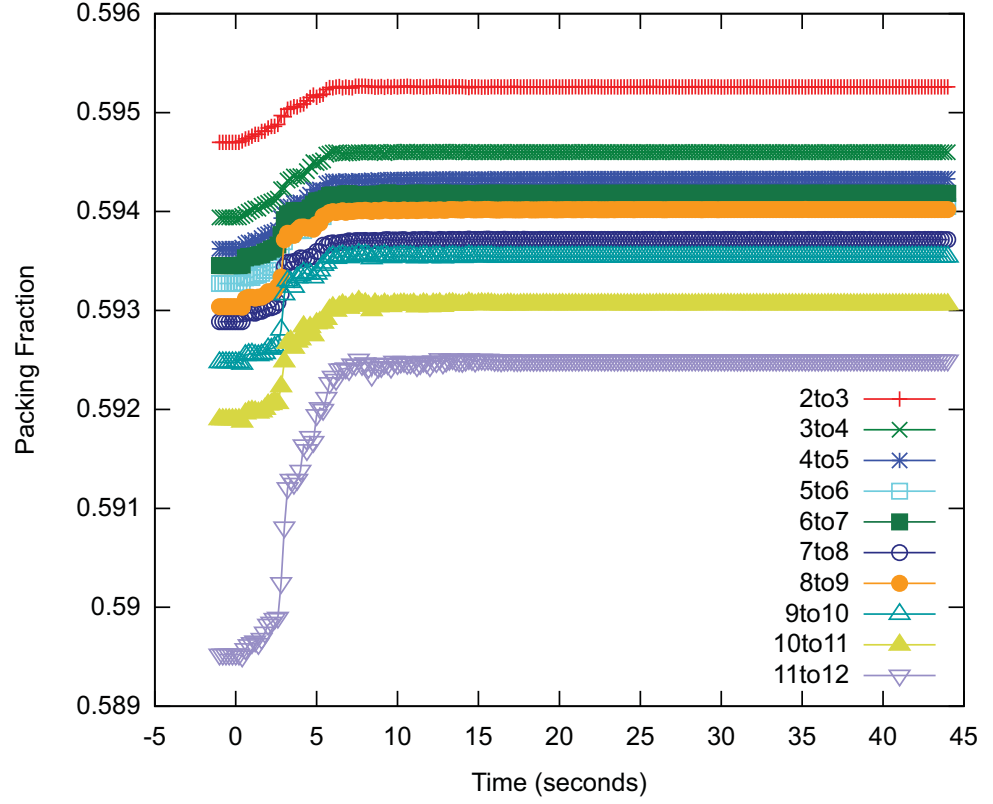


Figure 8.7. 0.65 static friction packing over time.

8.2.4 Earthquake Equations

For each time-step, the simulation calculates both a displacement and a wall velocity. For the sum of waves method, the displacement is calculated by:

$$\mathbf{d} = \sum_i \mathbf{D} \left[\sin \left((t - S) \frac{2.0\pi}{p} + c \right) + o \right] \quad (8.1)$$

where t is the current time, S is the time the wave starts, p is the period of the wave, c is the initial cycle of the wave, o is the offset, and \mathbf{D} is the maximum displacement vector.

The velocity is calculated by:

$$\mathbf{m} = \sum_i \frac{2\pi\mathbf{D}}{p} \cos \left((t - S) \frac{2\pi}{p} + c \right) \quad (8.2)$$

For the tabular data, the displacement and velocity are calculated by:

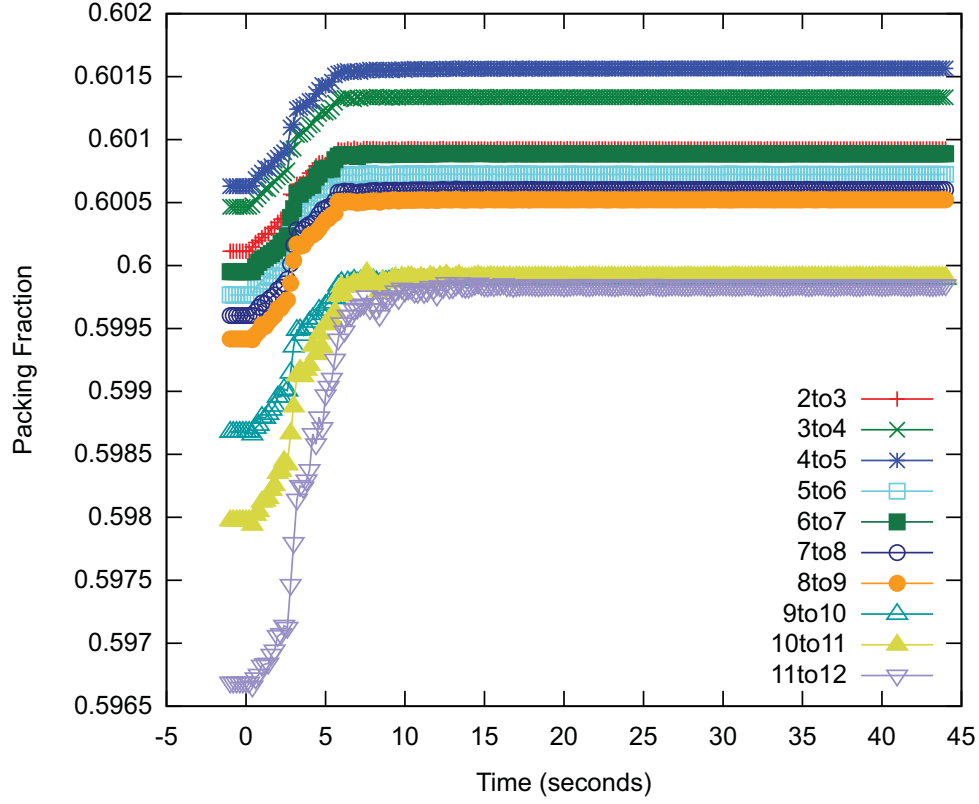


Figure 8.8. 0.35 static friction packing over time.

$$\mathbf{d} = (1 - o)T_k + oT_{k+1} \quad (8.3)$$

$$\mathbf{m} = \frac{1}{\delta}(T_{k+1} - T_k) \quad (8.4)$$

where T_i is the displacement at the i 'th time-step, o is a number between 0 and 1 that specifies where 0 is the start of the time-step and 1 is the end, and δ is the time in seconds between time-steps.

With these displacements, the code then uses:

$$\mathbf{p}' = \mathbf{p} + \mathbf{d} \quad (8.5)$$

$$\mathbf{v}' = \mathbf{v} + \mathbf{m} \quad (8.6)$$

as the adjusted position and velocity.

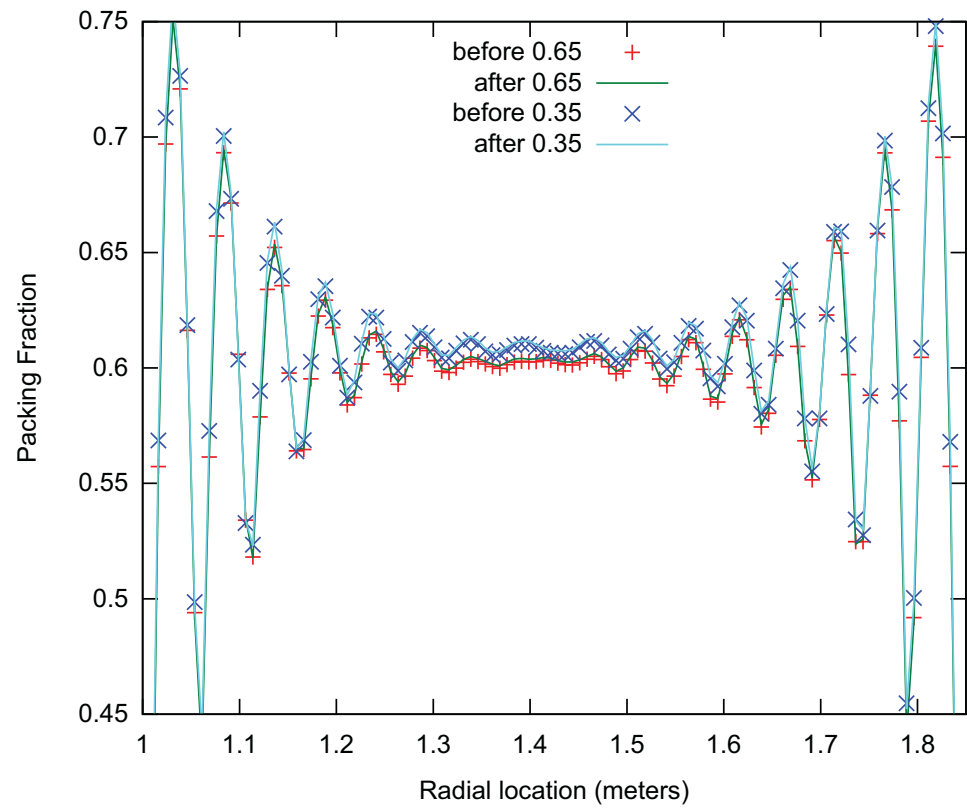


Figure 8.9. Different radial packing fractions.

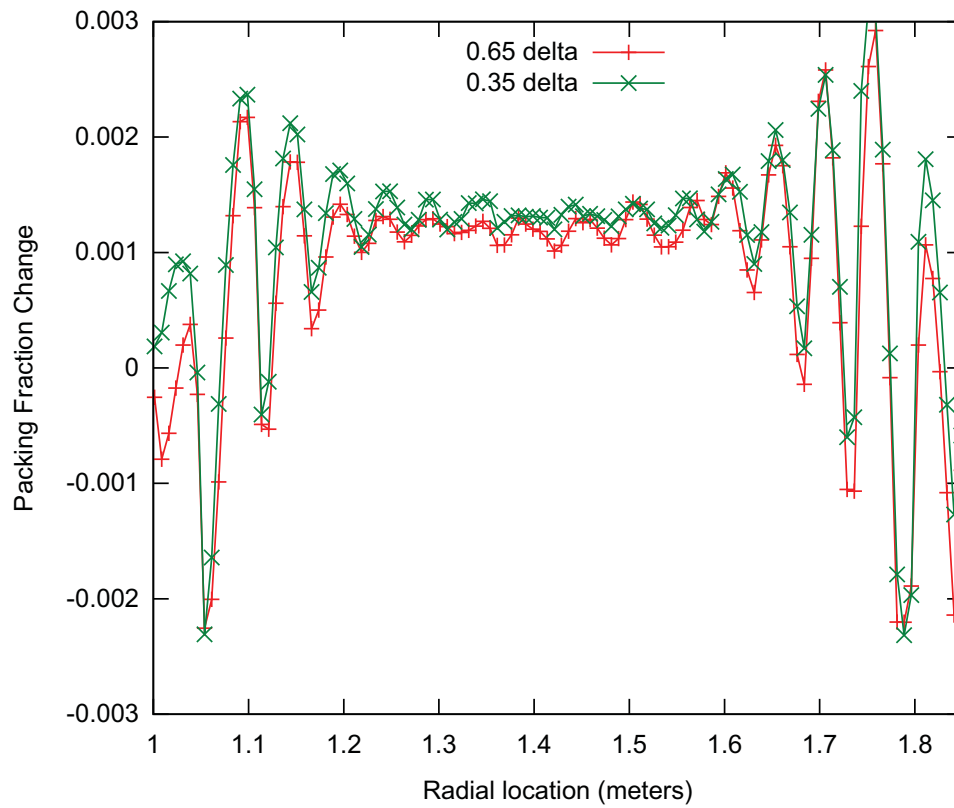


Figure 8.10. Changes in packing fraction.

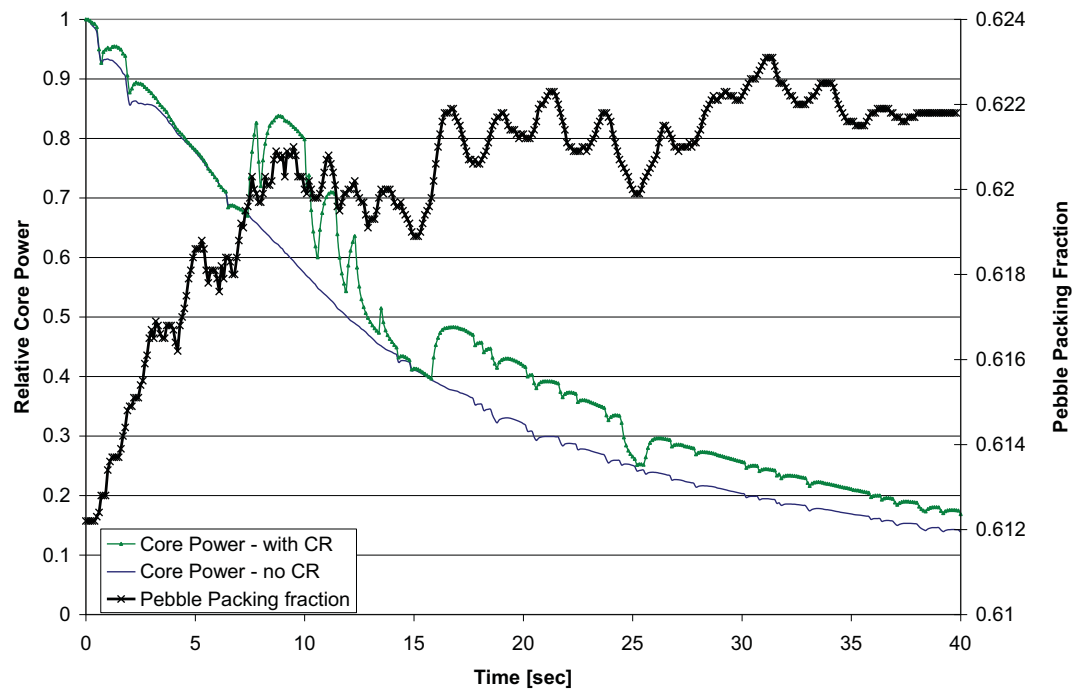


Figure 8.11. Neutronics and thermal effects from J. Ortensi.

Chapter 9

Construction of a Dust-Production Framework

With the creation of the PEBBLES simulation, one issue that was examined was using the simulation to attempt to predict the volume of dust that would be produced by an operating pebble-bed reactor. This is an important issue that could affect the choice of a pebble-bed reactor versus a prismatic reactor for process heat applications. However, as this chapter and Appendix B will discuss, while the PEBBLES code has the force and motion data required for this simulation, the coefficients that would allow this information to be used have not been sufficiently robustly determined experimentally yet.¹

With the data provided by PEBBLES, equations to link the dust production to PEBBLES-calculated quantities were examined. As shown in *Equation (B.1)*, the volume of dust produced can be approximated if the normal force of contact, the length slide and the wear coefficients are known. The force of contact and the length slide are calculated as part of the PEBBLES simulation, so this method was used to calculate dust production for the AVR reactor. This resulted in an estimate of four grams of graphite dust produced per year as compared to the measured value of three kilograms of dust produced per year. Several possible causes of this were identified in the paper documenting this work (Cogliati and Ougouag, 2008). A key first issue as described by this report's previous work section is that there are no good measurements of graphite wear rates in pebble-bed reactor relevant conditions (especially for a reactor run at AVR temperatures). A second issue is that the previous model of AVR was missing features, including the reactor control rod nose cones and wall indentations. A third issue, identified after the paper's publication, is that significant portions of the length traveled were not due to motion down through the reactor. Instead,

¹For a discussion on open questions related to the wear coefficients, see Cogliati et al. (2011).

much of the length that was tallied was due to pebbles vibrating. In the model used in the 2008 paper, this problem was about four times more severe than the current model, due to the new addition of slip correction via *Equation (5.2)*.

In order for the wear lengths and wear amounts to be correctly simulated, some care needs to be taken. First, sufficient recirculation of the pebbles must be done. The forces on the wall also need to be converged, and before sufficient recirculation has been achieved, there can be some artifacts from the filling method in the distribution of forces, and the static friction will not be fully loaded. Second, the cone at the top needs to have close to the correct angle of repose, or there will be extra vibrations from the pebbles impacting an unrealistically flat top of the cone. For these reasons, about one sixth of the pebbles were recirculated before calculating dust production. Finally, the time-step needs to be sufficiently small that unrealistic vibrations do not cause strong effects. Essentially, if the time-step is too large, the simulation will contain large numbers of pebbles that oscillate in one time-step between a positive and negative relative velocity; thus, the calculation incorrectly measures velocity, which will result in incorrectly calculating the length of wear. These two considerations would result in very long computational times since small time-steps would be needed for large numbers of time-steps used for partial recirculation. Instead, what was done was that the pebbles were recirculated with longer time-steps, and then 200 pebbles were recirculated with the smaller time-steps necessary, followed by the dust production calculations, which were performed with 200 more pebbles recirculated. At this point, the dust-production results were converged.

9.1 HTR-10 Simulation Results

A simple model of the HTR-10 reactor was created. This model used a 0.9-m cylinder with no indentations. The pebbles were all assumed to be fueled (that is, only one pebble type was simulated). Figure 9.1 shows the wall pressure in the reactor, while Figure 9.2 shows the wear-length weighted wall force and the wear-length weighted pebble-to-pebble force. Basically, the force spikes in two places. The first is located where the outlet chute starts on the wall. At this point in the reactor, a partial arch forms, the base of which is at the top of the outlet chute. This causes the outlet chute spike. The second spike is at the top. This one is caused by the impact of the pebbles falling onto the top of the reactor. Since Figure 9.2 is wear-length weighted and these are the fastest moving pebbles, they appear in Figure 9.2, but are absent from Figure 9.1.

The top of the reactor core will have a cone from the pebbles that land in the center. This can be seen in Figure 8.3. The slope of this cone is 0.5. Velocity profiles are tallied to show how the pebbles move on average. The pebbles will come out of the inlet chute and roll down the cone at a fairly fast rate, as Figure 9.3 shows. Once they reach the bottom of the reactor-core vessel, the pebbles' average velocity increases because there is less area. This phenomenon can be seen in Figure 9.4.

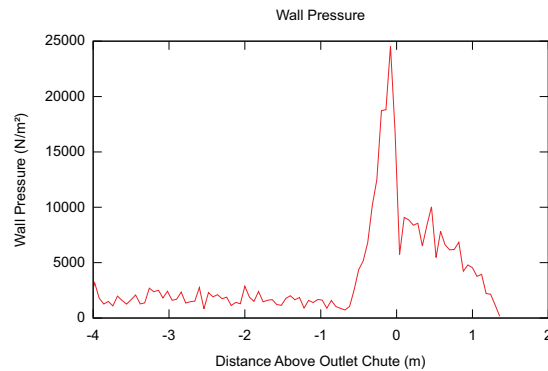


Figure 9.1. Wall pressure.

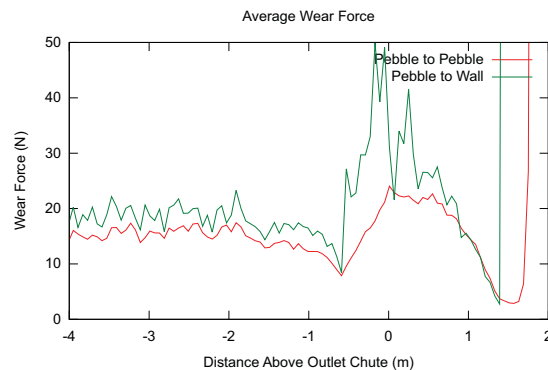


Figure 9.2. Average wear force.

The wear length is the next tally to be observed.² The wear length is tallied separately for pebble-to-pebble contact since the wear coefficient will likely be different because of the different curvature and potentially different types of graphite used. These wear lengths are tallied every 6 cm (a pebble diameter) and have been divided by the number of pebbles that were recirculated during the tally time. The wear lengths in the reactor calculated by PEBBLES are shown as a function of height in Figures 9.5 and 9.6. These are normalized to per pebble pass, so when they are above 0.06 cm, the wear lengths are stating that more wear is occurring than the distance that the pebble is traveling.

What range of wear length values could plausibly occur? If all the pebbles travel as a group, with no internal displacement, only wall wear would occur—no internal wear. On the other extreme, a pebble can be in cubic close packing

²In PEBBLES, the wear lengths (and wear amounts as well) only include the wear on the pebble side. So if a pebble rubs another pebble for 1 m, then the wear length will be 2 m, but if a pebble rubs the wall for 1 m, then the wear amount will be only 1 m.

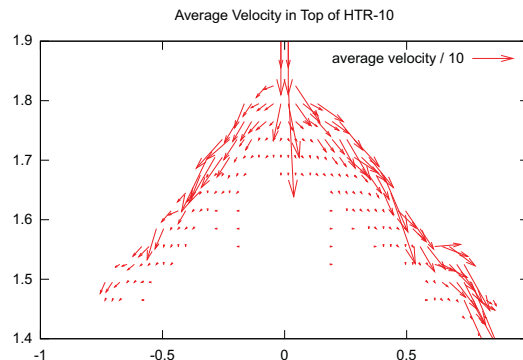


Figure 9.3. Velocity of pebbles at top.

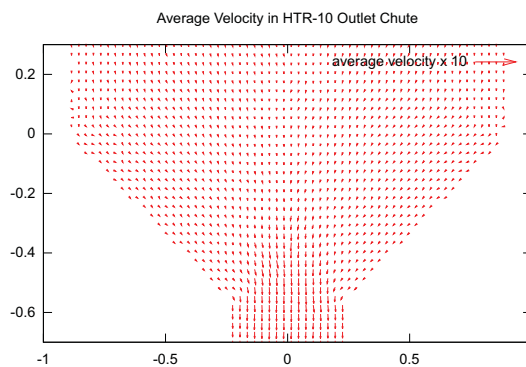


Figure 9.4. Velocity of pebbles in chute.

where each pebble is surrounded by 12 other pebbles. In this case, when the pebble rotates it will cause wear on itself and 12 other pebbles. Because of the possibility of rotation, it is hard to determine an upper bound on the amount of wear per unit length of vertical travel. From the PEBBLES code simulation, the pebble-to-pebble wear length in the roughly 2 meters of the core is 14 meters per pebble pass. This number could be higher than realistic. Three factors cause the wear length to be higher than the linear distance traveled. The first cause is due to vibrations from the pebbles impacting the top of the bed. The second is caused when the pebbles follow non-straight paths through the reactor. The third is wear produced from regular rotations. There are two peaks in the wear length, and each has somewhat different causes. The peak around 1.5-m comes from effects related to the falling pebbles. These falling pebbles will cause vibrations on impact, and will roll and slide down the cone, causing a higher amount of wear length for a typical pebble. The peak near 0 m comes from the

motion necessary to rearrange the pebbles into the outlet chute. During this portion of the journey, the pebbles will be forced to have horizontal motion as well as vertical motion. Before the outlet chute, the pebbles can move together approximately as a group with less internal wear. Switching to the smaller diameter outlet chutes causes these types of groups to break up. Figure 9.7 shows how the wear lengths are distributed by the contact speed. The fastest speeds (0.1 m/s to 1 m/s) have two peaks: one near 1.5 m and the other near 0 m. The 1.5 m peak comes from the falling pebbles, since they are traveling fairly fast when they impact, and usually continue to roll down the top cone. The 0-m peak comes from the constriction near the outlet chute and the accompanying rearrangement. The lowest speeds (less than 0.001 m/s) have their highest peak at the top of the reactor pile, and this peak comes from vibrations from the impacting pebbles.

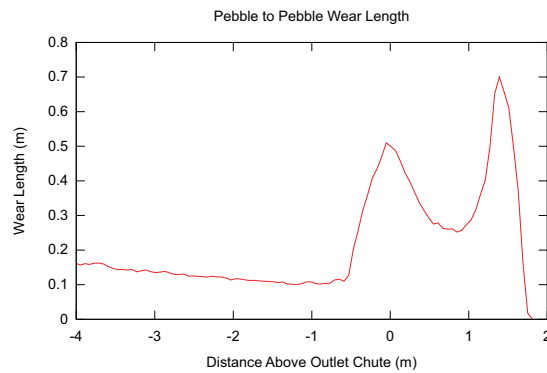


Figure 9.5. Pebble-to-pebble wear length.

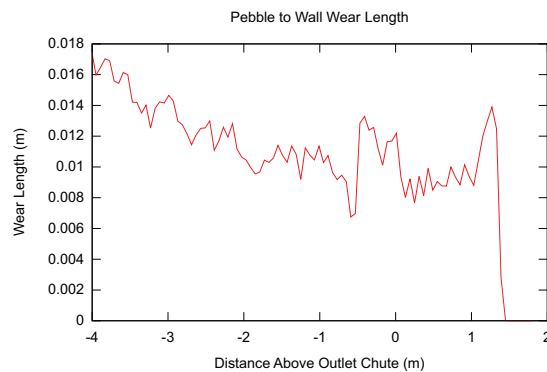


Figure 9.6. Pebble-to-wall wear length.

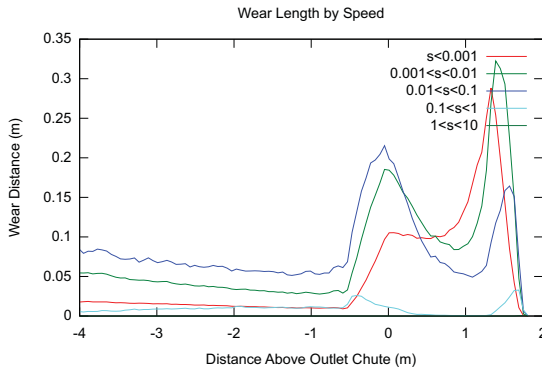


Figure 9.7. Pebble-to-pebble wear length by speed.

Concurrently with calculating the wear lengths, the wear amount is calculated by multiplying each wear length by the normal force. Figures 9.8, 9.9, and 9.10 show these wear amounts. The wear tends to follow similar patterns as the wear length, but because of the force component, the wear is much lower the higher one goes in the reactor. However, deeper in the reactor where the pressures are greater, the wear amount will be higher. One particular point of peak wear is at the start of the outlet chute. At that point, the bottom of the pseudo-arch forms and concentrates the weight from the pebbles above in the reactor. The wear amount simulated increases below -1 m in the outlet chute. This is likely an artifact of approximations in the simulation since the force shown in Figure 9.2 saturates in the outlet chute, so the wear amount should saturate as well. One possible cause of this increased wear is by additional horizontal motion caused by pebbles being removed at random locations at the bottom of the outlet chute.

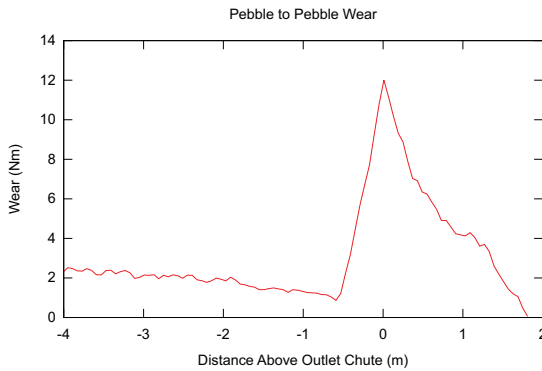


Figure 9.8. Pebble-to-pebble wear amount.

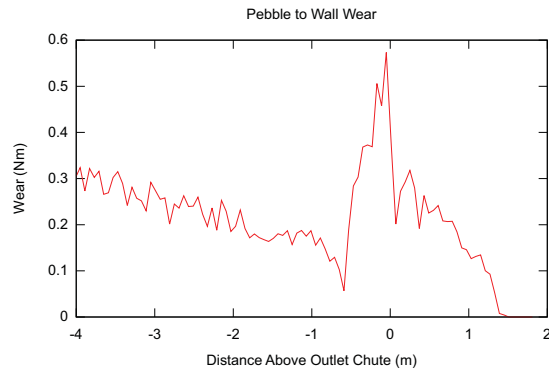


Figure 9.9. Pebble-to-wall wear amount.

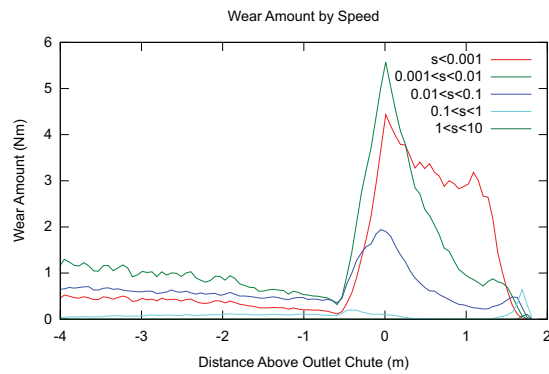


Figure 9.10. Pebble-to-pebble wear amount by speed.

Chapter 10

Future Work

The dust-production simulation requires both proper dust-production wear coefficients and properly determining the correct method of dealing with vibrational issues. It would be useful to determine the number of pebbles that need to be simulated to provide a correct representation of a full NGNP-600-sized reactor. Since the middle portions are geometrically similar, determining the amount of recirculation required to reach a geometrically asymptotic state might allow only a portion of the recirculation to be done. Those two changes might allow quicker simulation of full-sized reactors. Finally, in order to allow sufficiently fast simulations on today's computer hardware, many approximations to the true behavior are done. In the future, some of these approximations maybe relaxed.

Chapter 11

Summary and Conclusions

Research results presented in this report demonstrate a DEM that provides high fidelity and yet has reasonable run-times for many pebble fuel element flow simulations. The new static friction test will be useful for evaluating any implementation of static friction for spheres. The PEBBLES code produced for this report has been able to provide data for multiple applications including Dancoff factor calculation, neutronics simulation and earthquake simulation. The new static friction model provides expected static friction behavior in the reactor including partial matching of the Janssen model predictions and correctly matching stability behavior in a pyramid. The groundwork has been created for predicting the dust production from wear in a pebble-bed reactor once further experimental data is available. Future work includes potentially relaxing some of the physical approximations made for speed purposes when faster computing hardware exists and investigating new methods for allowing faster simulations. This project has provided significant enhancements in simulation of the mechanical movement of pebbles in a pebble-bed reactor.

Bibliography

- Mpi: A message-passing interface standard, version 2.2. 2009. URL <http://www.mpi-forum.org/docs/mpi-2.2/mpi22-report.pdf>.
- Openmp application program interface, version 3.0. 2008. URL <http://www.openmp.org/mp-documents/spec30.pdf>.
- Oprofile - a system profiler for linux. 2009. URL <http://oprofile.sourceforge.net>.
- The commissioning of the thtr 300. a performance report, 1989. D HRB 129288E.
- Hkg hochtemperatur-kernkraftwerk gesellschaft mit beschränkter haftung, 2009. <http://www.thtr.de/> Accessed Oct 27, 2009. Technical data on: <http://www.thtr.de/technik-tht.htm>.
- Atomwirtschaft-Atomtechnik-atw. Avr-atomversuchskraftwerk mit kugelhafen-hochtemperatur-reaktor in juelich. *Atomwirtschaft*, May 1966. Sonderdruck aus Heft (Available as part of NEA-1739: IRPhE/AVR <http://www.nea.fr/abs/html/nea-1739.html>. Pages particularly useful are 230 and 240).
- R. Bäumer, H. Barnert, E. Baust, A. Bergerfurth, H. Bonnenberg, H. Bülling, St. Burger, C.-B. von der Decken, W. Delle, H. Gerwin, K.-G. Hackstein, H.-J. Hantke, H. Hohn, G. Ivens, N. Kirch, A. I. Kirjushin, W. Kröger, K. Krüger, N. G. Kuzavkov, G. Lange, C. Marnet, H. Nickel, P. Pohl, W. Scherer, J. Schöning, R. Schulten, J. Singh, W. Steinwarz, W. Theymann, E. Wahlen, U. Wawrzik, I. Weisbrodt, H. Werner, M. Wimmers, and E. Ziermann. *AVR: Experimental High Temperature Reactor; 21 years of successful operation for a future energy technology*. Association of German Engineers (VDI), The Society for Energy Technologies, 1990. ISBN 3-18-401015-5.
- D. Bedenig, W. Rausch, and G. Schmidt. Parameter studies concerning the flow behaviour of a pebble with reference to the fuel element movement in the core of the thtr 300 mwe prototype reactor. *Nuclear Engineering and Design*, 7: 367-378, 1968.

- R.F. Benenati and C. B. Brosilow. Void fraction distribution in beds of spheres. *A. I. Ch. E. Journal*, 8:359–361, 1962.
- J. D. Bernal, J. Mason, and G. David Scott. Packing of spheres. *Nature*, 188: 908–911, 1960.
- Bharat Bhushan. *Modern Tribology Handbook*. CRC Press, Boca Raton, Florida, USA, 2000. Chap. 7.5.
- I. Bratberg, K.J. Maløy, and A. Hansen. Validity of the janssen law in narrow granular columns. *The European Physical Journal E*, 18:245–252, 2005.
- J. J. Cogliati and A. M. Ougouag. Pebble bed reactor dust production model. Proceedings of the 4th International Topical Meeting on High Temperature Reactor Technology, 2008. Washington, D.C., USA, September 28 – October 1.
- Joshua J. Cogliati, Abderrafi M. Ougouag, and Javier Ortensi. Survey of dust production in pebble bed reactor cores. *Nuclear Engineering and Design*, 241: 2364–2369, 2011. doi:10.1016/j.nucengdes.2011.03.023.
- Jonathan D. Cohen, Ming C. Lin, Dinesh Manocha, and Madhav K. Ponamgi. Icollide: an interactive and exact collision detection system for large scale environments. Proceedings of the 1995 Symposium on Interactive 3D Graphics, 1995. Monterey, CA, April 9-12, pp. 19-24.
- P. A. Cundall and O. D. L. Strack. A discrete numerical model for granular assemblies. *Géotechnique*, 29:47–65, 1979.
- Ulrich Drepper. What every programmer should know about memory. 2007. URL <http://people.redhat.com/drepper/cpumemory.pdf>.
- Jacques Duran. *Sands, Powders, and Grains: An Introduction to the Physics of Granular Materials*. Springer, New York, New York, USA, 1999. ISBN 978-0387986562.
- Hannsjörg Freund, Thomas Zeiser, Florian Huber, Elias Klemm, Gunther Brenner, Franz Durst, and Gerhand Emig. Numerical simulations of single phase reacting flows in randomly packed fixed-bed reactors and experimental validation. *Chemical Engineering Science*, 58:903–910, 2003.
- A. J. Goodjohn. Summary of gas-cooled reactor programs. *Energy*, 16:79–106, 1991.
- Keishi Gotoh, Hiroaki Masuda, and Ko Higashitanti. *Powder Technology Handbook, 2nd ed.* Marcel Dekker, Inc, New York, New York, 1997.
- Hans D. Gougar, Abderrafi M. Ougouag, and William K. Terry. Advanced core design and fuel management for pebble-bed reactors. 2004. INEEL/EXT-04-02245.

- J. M. Haile. *Molecular Dynamics Simulation*. John Wiley & Sons, Inc, New York, 1997.
- A. M. Ougouag J. L. Kloosterman. Computation of dancoff factors for fuel elements incorporating randomly packed triso particles. Technical report, 2005. INEEL/EXT-05-02593, Idaho National Laboratory.
- K.L. Johnson. *Contact Mechanics*. Cambridge University Press, 1985. ISBN 0-521-34796-3. Section 4.2.
- Rémi Jullien, André Pavlovitch, and Paul Meakin. Random packings of spheres built with sequential models. *Journal Phys. A: Math. Gen.*, 25:4103–4113, 1992.
- Andrew C. Kadak and Martin Z. Bazant. Pebble flow experiments for pebble bed reactors. Beijing, China, September 22-24 2004. 2nd International Topical Meeting on High Temperature Reactor Technology.
- Martin P. Kissane, Fan Zhang, and Michael W. Reeks. Dust in htrs: its nature and improving prediction of its resuspension. Proceedings of HTR 2010 Prague, Czech Republic, October 2010.
- G. A. Kohring. Studies of diffusional mixing in rotating drums via computer simulations. *Journal de Physique I*, 5:1551, 1995.
- J.R. Lamarsh. *Introduction to Nuclear Engineering 2nd Ed*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1983. pp. 591-593.
- J. K. Lancaster and J. R. Pritchard. The influence of environment and pressure on the transition to dusting wear of graphite. *Journal of Physics D: Applied Physics*, 14:747–762, 1981.
- J.K. Lancaster and J.R. Pritchard. On the ‘dusting’ wear regime of graphite sliding against carbon. *J. Phys. D: Appl. Phys.*, 13:1551–1564, 1980.
- James W. Landry, Gary S. Grest, Leonardo E. Silbert, and Steven J. Plimpton. Confined granular packings: Structure, stress, and forces. *Physical Review E*, 67:041303, 2003.
- Zi Lu, Mohamed Abdou, and Alice Ying. 3d micromechanical modeling of packed beds. *Journal of Nuclear Materials*, 299:101–110, 2001.
- Xiaowei Luo, Lihong Zhang, and Yu Suyuan. The wear properties of nuclear grade graphite ig-11 under different loads. *International Journal of Nuclear Energy Science and Technology*, 1(1):33–43, 2004.
- Xiaowei Luo, Yu Suyuan, Sheng Xuanyu, and He Shuyan. Temperature effects on ig-11 graphite wear performance. *Nuclear Engineering and Design*, 235: 2261–2274, 2005.

- Jerry B. Marion and Stephen T. Thornton. *Classical Dynamics of Particles and Systems, 5th Ed.* Saunders College Publishing, 2004. ISBN 0-534-40896-6.
- G. K. Miller, D. A. Petti, and J. T. Maki. Development of an integrated performance model for triso-coated gas reactor particle fuel. High Temperature Reactor 2002 Conference, April 2002. URL <http://www.inl.gov/technicalpublications/Documents/3169759.pdf>. Chicago, Illinois, April 25-29, 2004, on CD-ROM, American Nuclear Society, Lagrange Park, IL.
- R. D. Mindlin and H. Deresiewicz. Elastic spheres in contact under varying oblique forces. *ASME J. Applied Mechanics*, 20:327–344, 1953.
- Rainer Moormann. A safety re-evaluation of the avr pebble bed reactor operation and its consequences for future htr concepts. 2008a. URL <http://hdl.handle.net/2128/3136>. Berichte des Forschungszentrums Jülich; 4275 ISSN 0944-2952.
- Rainer Moormann. Fission product transport and source terms in htrs: Experience from avr pebble bed reactor. *Science and Technology of Nuclear Installations*, 2008, 2008b. doi:10.1155/2008/597491.
- Von R. Nieder. Schlußfolgerungen für die htr-chemie aus 21 jahren betrieb des avr-reaktors. Chemie im Kraftwerk, 1990.
- Javier Ortensi. *An Earthquake Transient Method for Pebble-Bed Reactors and a Fuel Temperature Model for TRISO Fueled Reactors*. PhD thesis, Idaho State University, 2009.
- Abderrafi M. Ougouag and William K. Terry. A preliminary study of the effect of shifts in packing fraction on k-effective in pebble-bed reactors. Proceeding of Mathematics & Computation 2001, September 2001. Salt Lake City, Utah, USA.
- Abderrafi M. Ougouag, Hans D. Gougar, William K. Terry, Ramatsemela Mphahlele, and Kostadin N. Ivanov. Optimal moderation in the pebble-bed reactor for enhanced passive safety and improved fuel utilization. PHYSOR 2004 The Physics of Fuel Cycle and Advanced Nuclear Systems: Global Developments, April 2004.
- S. J. Payne. Development of design basis earthquake (dbe) parameters for moderate and high hazard facilities at tra. 2003. URL <http://www.inl.gov/technicalpublications/Documents/2906939.pdf>. INEEL/EXT-03-00942.
- Frederik Reitsma, 2005. (PBMR, Pty. Ltd.) private communication to A. M. Ougouag, August 18.
- Gerald H. Ristow. Flow properties of granular materials in three-dimensional geometries. Master's thesis, Philipps-Universitt Marburg, Marburg/Lahn, January 1998.

- Chris H. Rycroft, Martin Z. Bazant, Gary S. Grest, and James W. Landry. Dynamics of random packings in granular flow. *Physical Review E*, 73:051306, 2006a.
- Chris H. Rycroft, Gary S. Grest, James W. Landry, and Martin Z. Bazant. Analysis of granular flow in a pebble-bed nuclear reactor. *Physical Review E*, 74:021306, 2006b.
- X. Yu Sheng, X. S., Luo, and S. He. Wear behavior of graphite studies in an air-conditioned environment. *Nuclear Engineering and Design*, 223:109–115, 2003.
- Leonardo E. Silbert, Deniz Ertas, Gary S. Grest, Thomas C. Halsey, Dov Levine, and Steven J. Plimpton. Granular flow down an inclined plane: Bagnold scaling and rheology. *Physical Review E*, 64:051302, 2001.
- W. Soppe. Computer simulation of random packings of hard spheres. *Powder Technology*, 62:189–196, 1990.
- Matthias Sperl. Experiments on corn pressure in silo cells – translation and comment of janssen’s paper from 1895. *Granular Matter*, 8:59–65, 2006.
- O. M. Stansfield. Friction and wear of graphite in dry helium at 25, 400, and 800° c. *Nuclear Applications*, 6:313–320, 1969.
- William K. Terry, Soon Sam Kim, Leland M. Montieth, Joshua J. Cogliati, and Abderrafi M. Ougouag. Evaluation of the htr-10 reactor as a benchmark for physics code qa. ANS Topical Meeting on Reactor Physics, 2006. URL <http://www.inl.gov/technicalpublications/Search/Results.asp?ID=INL/CON-06-11699>.
- L. Vu-Quoc, X. Zhang, and O. R. Walton. A 3-d discrete-element method for dry granular flows of ellipsoidal particles. *Computer Methods in Applied Mechanics and Engineering*, 187:483–528, 2000.
- Loc Vu-Quoc and Xiang Zhang. An accurate and efficient tangential force-displacement model for elastic frictional contact in particle-flow simulations. *Mechanics of Materials*, 31:235–269, 1999.
- Dr. Wahsweiler. Bisherige erkentnisse zum graphitstaub, 1989. HRB BF3535 26.07.1989.
- R. Wait. Discrete element models of particle flows. *Mathematical Modeling and Analysis I*, 6:156–164, 2001.
- D. M. Walker. An approximate theory for pressures and arching in hoppers. *Chemical Engineering Science*, 21:975–997, 1966.
- Zongxin Wu, Dengcai Lin, and Daxin Zhong. The design features of the htr-10. *Nuclear Engineering and Design*, 218:25–32, 2002.

- Luo Xiao-wei, Yu Su-yaun, Zhang Zhen-sheng, and He Shu-yan. Estimation of graphite dust quantity and size distribution of graphite particle in htr-10. *Nuclear Power Engineering*, 26, 2005. ISSN 0258-0926(2005)02-0203-06.
- Luo Xiaowei. Personal correspondence, January 2011a. The units for the upper and lower sample are 10^{-3} mg/m or (10^{-6} g/m).
- Luo Xiaowei. Personal correspondence, February 2011b.
- Yuanhui Xu and Kaifen Zuo. Overview of the 10 mw high temperature gas cooled reactortest module project. *Nuclear Engineering and Design*, 218:13–23, 2002.

Appendix A

Calculation of Packing Fractions

For determining the volume of a sphere that is inside a vertical slice, the following formula can be used

$$a = \max(-r, \text{bot} - z) \quad (\text{A.1})$$

$$b = \min(r, \text{top} - z) \quad (\text{A.2})$$

$$v = \pi \left[r^2(b - a) + \frac{1}{3}(a^3 - b^3) \right] \quad (\text{A.3})$$

where

r = the pebble radius

bot = the bottom of the vertical slice

top = the top of the vertical slice

z is the vertical location of the pebble center.

To determine the area inside a vertical and radial slice, two auxiliary functions are defined: one that has the area inside a radial 2-D slice, and another which has the area outside a radial 2-D slice.

Figure A.1 shows the area inside both a circle of radius c and a radial slice of I . The circle is r from the center of the radial circle. Auxiliary terms are defined, which include f , the distance from the intersection of the segment of the radial circle perpendicularly to the center line, j the distance to the intersection of f , θ the angle of segment, and ϕ the angle from the segment intersection on the circle side. The `area_inside` function has the following definition:

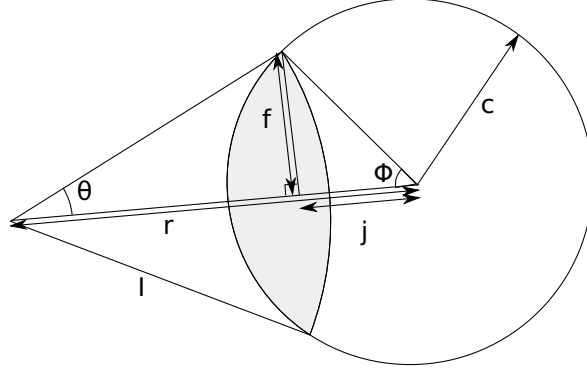


Figure A.1. Area inside geometry.

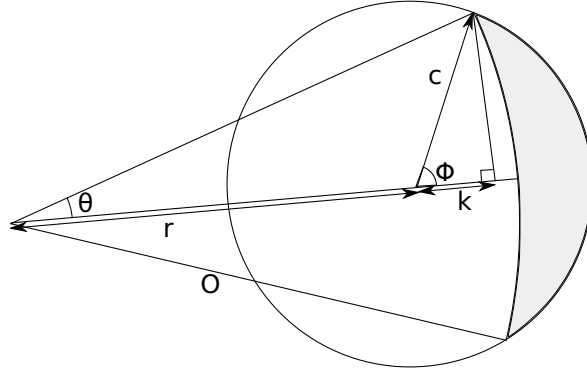


Figure A.2. Area outside geometry.

$$a_i = 0.0 \quad \text{if } I < r - c \quad (\text{A.4})$$

$$a_i = \pi c^2 \quad \text{if } r + c < I \quad (\text{A.5})$$

$$a_i = \pi I^2 \quad \text{if } I < r + c \text{ and } I < c - r \quad (\text{A.6})$$

$$\text{otherwise} \quad (\text{A.7})$$

$$j = \frac{r^2 + c^2 - I^2}{2r} \quad (\text{A.8})$$

$$f = \sqrt{c^2 - j^2} \quad (\text{A.9})$$

$$\theta = 2 \arccos \frac{I^2 + r^2 - c^2}{2Ir} \quad (\text{A.10})$$

$$\phi = 2 \arccos \frac{r^2 + c^2 - I^2}{2rc} \quad (\text{A.11})$$

$$a_i = \frac{1}{2}c^2\phi + \frac{1}{2}I^2\theta - fr \quad (\text{A.12})$$

Figure A.2 shows the area that is outside the radial slice, but inside the circle. The radial slice has a radius of O . The new auxiliary term k is the distance from the circle's center to the perpendicular intercept. The `area_outside` function has the following definition:

$$a_o = 0.0 \quad \text{if } O > c + r \quad (\text{A.13})$$

$$a_o = \pi c^2 \quad \text{if } c - r > O \quad (\text{A.14})$$

$$a_o = \pi c^2 - \pi O^2 \quad \text{if } O < r + c \text{ and } O < c - r \quad (\text{A.15})$$

$$\text{otherwise} \quad (\text{A.16})$$

$$k = \frac{O^2 - r^2 - c^2}{2r} \quad (\text{A.17})$$

$$m = \sqrt{c^2 - k^2} \quad (\text{A.18})$$

$$\theta = 2 \arccos \frac{k + r}{O} \quad (\text{A.19})$$

$$\phi = 2 \arccos \frac{k}{c} \quad (\text{A.20})$$

$$a_o = \left(\frac{1}{2}c^2\phi - mk\right) - \left(\frac{1}{2}O^2\theta - m(k+r)\right) \quad (\text{A.21})$$

Then, the total volume in a radial slice can be determined from the computation:

$$a = \max(-r, \text{bot} - z) \quad (\text{A.22})$$

$$b = \min(r, \text{top} - z) \quad (\text{A.23})$$

$$v_t = \pi \left[R^2(b - a) + \frac{1}{3}(a^3 - b^3) \right] \quad (\text{A.24})$$

$$v_i = \int_a^b \text{area_inside}(c = \sqrt{R^2 - z^2}) dz \quad (\text{A.25})$$

$$v_o = \int_a^b \text{area_outside}(c = \sqrt{R^2 - z^2}) dz \quad (\text{A.26})$$

$$v = v_t - v_i - v_o \quad (\text{A.27})$$

Appendix B

Determination of dust-production coefficients

One potential use of the PEBBLES code is to predict the dust production of a pebble-bed reactor. This section discusses the features that make this possible and work that has been done to determine the necessary coefficients. Unfortunately, the following literature review discovered a lack of robust wear coefficients, which prevents prediction of dust production.

There are essentially four contact wear mechanisms: (1) adhesive wear arises from the contacting surfaces adhesively bonding together and part of the material being pulled away, (2) abrasive wear occurs when one of the contacting materials is harder than the other and plows (or shears) away material, (3) fatigue wear takes place when the surfaces repeatedly contact each other, causing fracture of the material, and (4) corrosive wear occurs when chemical corrosion causes the surface to behave with increased wear (Bhushan, 2000). In a pebble-bed reactor, adhesive wear will occur between pebbles, abrasive wear could occur between different graphite types or between graphite and other materials (such as between graphite and steel in the fuel-handling system), fatigue wear might be possible from the impact of the pebbles on the top of the bed when they are dropped in, and corrosive wear might be possible from impurities in the helium gas.

As a first-order approximation, the adhesive dust-production volume is:

$$V = K_{ad} \frac{N}{H} L \quad (\text{B.1})$$

Where V is the wear volume, K_{ad} is the wear coefficient or wear rate for adhesive wear, L is the length slid, and $\frac{N}{H}$ is the real contact area (with N the normal force and H the hardness) (Bhushan, 2000). Typically, the hardness and the wear coefficient for adhesive wear are combined with the units of either mass or volume over either force times distance or distance. The different wear units can be converted provided that the material density and the contact force

are known. For two blocks, the length slide is the distance that one of the blocks travels over the other while in contact. Note that this formula is only an approximation, since the wear volume is only approximately linear with respect to both the normal force and the distance traveled. Abrasive wear also can be approximated by this model, but fatigue and corrosive wear will not be modeled well by *Equation (B.1)*. To the extent that corrosive and fatigue wear mechanisms are present in the pebble-bed reactor, this equation may also be less valid.

The wear coefficient is typically measured by grinding or stroking two pieces of graphite against each other and then measuring the wear (Sheng et al., 2003; Stansfield, 1969). The details of the experiment, such as the contact shape and the orientation of the relative motion, affect the wear coefficient.

The wear that occurs with graphite depends on multiple factors. A partial list includes the normal force of contact (load), temperature of the graphite, and past wear history (since wear tends to polish the contact surfaces and remove loose grains). The atmosphere around the graphite affects the wear rates because some molecules chemically interact with the carbon or are adsorbed on the surface. Reduction of the adsorbed gases can produce high wear (Stansfield, 1969). Neutron damage and other radiation effects can damage the structure of the graphite and affect the wear. The type and processing of the graphite can affect wear rates. During the manufacturing process of the pebbles, graphite is mixed with a binder and heated. To avoid damaging the silicon-carbide of the TRISO particles, the heating is kept below 2000°C. Temperatures of >2800°C are required for full graphitization (Kissane et al., 2010). As a related effect, if hard and soft graphites interact, the harder one can “plow” through the softer and increase wear rates.

For graphite on graphite, depending on conditions, there can be over three orders of magnitude difference in the wear. Table B.1 lists different wear rates mentioned in this article. For example, graphite on graphite in air at room temperature can exhibit wear rates of $3.3\text{e-}8$ g/(Nm) (Sheng et al., 2003), but in the dusting regime at 200°C the wear rate can be $2\text{e-}5$ g/(Nm) (Lancaster and Pritchard, 1980), which is about a thousand times greater. This dusting regime occurs in air when the contact temperature is above a critical value and causes much greater wear rates. When dusting occurs, large amounts of finely divided wear debris are produced. Changes in the speed of contact, the normal force of contact and the partial pressures of gases affect the onset of dusting (Lancaster and Pritchard, 1981). The dusting regime may not occur in primarily helium atmospheres. However, to explain dust amounts seen in some past pebble-bed reactors with a primarily mechanical production theory, wear rates of roughly the same magnitude as in the dusting regime would be required. Because of the wide range of wear rates of graphite, and the sensitivity to temperature and the composition of gases, conditions as close to the in-core conditions as possible are needed for determining a better approximation of the wear coefficients.

For tests using nuclear graphite near in-core conditions, the best data available to the authors is from two independent sets of experiments. One dataset emerged from Stansfield (1969) experiments and the other is from a series of

experiments performed at Tsinghua University (Sheng et al., 2003; Luo et al., 2004, 2005).

O. M. Stansfield measured friction and wear with different types of graphite in helium at different temperatures (Stansfield, 1969). In the experiments, two pieces of graphite were slid against each other linearly with a 0.32-cm stroke. The stroke is the distance traveled one way in one cycle, which is half the length slid in each cycle. Two different loads were used: a 2-kg mass and an 8-kg mass. The data for wear volumes is only provided graphically, not tabulated; therefore, only order of magnitude results are available. The wear values were about an order of magnitude higher at 25°C than at 400°C and 800°C. A reduction of friction occurred with increased length slide, but no explanation was provided.¹ Typical values for the wear rates are $10\text{e-}3 \text{ cm}^3/\text{kg}$ for the 25°C case and $10\text{e-}4 \text{ cm}^3/\text{kg}$ for the 400°C, and 800°C for 12,500-cm distance slide for wear between MHLM versus MHLM graphite with an 8-kg load. With a density of 1.82 g/cm^3 , these correspond to $1.5\text{e-}6 \text{ g/(Nm)}$ and $1.5\text{e-}7 \text{ g/(Nm)}$. These are roughly an order of magnitude above room-temperature wear.

Tsinghua university performed a second set of experiments. In the first, researchers measured the wear coefficient of graphite KG-11 by pressing a static specimen against a revolving specimen. The wear was measured by weighing the difference in mass of each specimen before and after the experiment. In room-temperature air, researchers measured wear rates of $7.32\text{e-}9 \text{ g/(Nm)}$ with 31 N load with surface contact, $3.29\text{e-}8 \text{ g/(Nm)}$ with 31 N load with line contact, and $3.21\text{e-}8 \text{ g/(Nm)}$ with 62 N load. In the surface contact, the end of a rotating cylinder was pressed against a flat specimen with a contact area of 804 mm^2 , and in the line contact experiment, the flat specimen was pressed against the cylinder's rounded side with a contact length of 30 mm (Sheng et al., 2003).

In the second Tsinghua University experiment, researchers measured the wear coefficient of graphite IG-11 on graphite and steel at varying loads ranging from 30 to 50 N (Luo et al., 2004). There are inconsistencies in the units used in the multiplier to the data tables in this paper and the following paper that were clarified with the author (Xiaowei, 2011a) to be $1.0\text{e-}3 \text{ mg/m}$. The 30 N of load upper specimen wear coefficient for the first 30 minutes is listed as $1.4\text{e-}3 \text{ mg/m}$, which corresponds to $4.7\text{e-}7 \text{ g/(Nm)}$. It is worth noting that the wear does not increase linearly with the increasing load in the measured values.

In the third Tsinghua University experiment, researchers measured the temperature effects in helium (Luo et al., 2005; Xiaowei, 2011a). This third experimental setup is similar to the second, but with a helium atmosphere, temperatures of 100°C to 400°C and a load of 30 N. In Figure 2 of that paper, it can be qualitatively determined that as the temperature increases, the amount of wear increases. Furthermore, the wear tends to have a higher rate initially, which then decreases. Since the wear experiment was performed using a 2-mm-long stroke, it is plausible that wear rates in an actual pebble-bed are closer to the initially higher rates because the pebble flow could expose more fresh surfaces on the pebbles to wear. Figure B.1 does not reveal a clear trend in the wear

¹Possibly this was due to a lubrication effect or the removal of rough or loose surfaces.

as a function of temperature. This makes it difficult to estimate wear rates since pebble-bed reactor cores can reach temperatures over 1000°C in normal operation. The highest wear rate in Table 2 of the paper is 31.3e-3 mg/m at 30 N at 200°C in the lower sample for the first 30 minutes, so the highest wear rate measured is 1.04e-6 g/(Nm). This is roughly 20 times lower than wear in the high-wear-rate dusting regime. Since the average amount of wear between 200°C and 400°C roughly doubles in the upper specimen and increases by approximately 35% in the lower specimen, substantially higher wear rates in over 1000°C environments are hard to rule out. Note, however, that the opposite temperature trend was observed in the Stansfield paper.

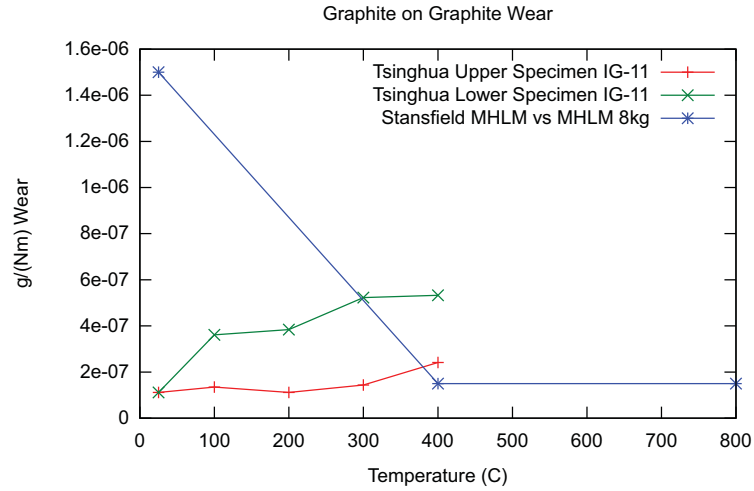


Figure B.1. Wear compared to temperature (Luo et al., 2005; Stansfield, 1969).

B.1 Calculation of Force in Reactor Bed

In order to calculate the dust produced in the reactor, the force acting on the pebbles is needed. Several different approximations can be used to calculate this with varying accuracy. The simplest (but least-accurate) method of approximating the pressure in the reactor is assuming that hydrostatic pressure can be used, or

$$P = \rho fgh \quad (\text{B.2})$$

where

P = the pressure at a point

ρ = the density of the pebbles

Table B.1. Different wear coefficients. Because these use different grades of graphite and experimental setups, they are not always directly comparable.

Wear Coefficient	Notes	Ref
2e-5 g/(Nm)	200°C dusting, air	Lancaster and Pritchard (1980)
1.5e-6 g/(Nm)	25°C, helium	Stansfield (1969)
1.5e-7 g/(Nm)	400° and 800°C, helium	Stansfield (1969)
7.32e-9 g/(Nm)	31 N load, surface contact, air	Sheng et al. (2003)
3.29e-8 g/(Nm)	31 N load, line contact, air	Sheng et al. (2003)
4.7e-7 g/(Nm)	30 N load, air, upper, first 30 minutes	Luo et al. (2004)
1.04e-6 g/(Nm)	30 N load, helium, 200°C, lower, first 30 minutes	Luo et al. (2005)
7.0e-8 g/(Nm)	30 N load, helium, 300°C, upper, 120–150 minutes	Luo et al. (2005)
1.17e-7 g/(Nm)	30 N load, helium, 400°C, upper, 120–150 minutes	Luo et al. (2005)
2.77e-7 g/(Nm)	30 N load, helium, 300°C, lower, 120–150 minutes	Luo et al. (2005)
3.47e-7 g/(Nm)	30 N load, helium, 400°C, lower, 120–150 minutes	Luo et al. (2005)
3.23e-7 g/(Nm)	30 N load, helium, 400°C, graphite-steel, 120–150 minutes	Luo et al. (2005)

f = the packing fraction of the pebbles (typical values are near 0.61 or 0.60)

g = the gravitational acceleration

h = the height below the top of the pebble-bed.

With knowledge of how many contacts there are per unit area or per unit volume, the pressure can be converted into pebble-to-surface or pebble-to-pebble contact forces. This formula is inaccurate when static friction occurs since the static friction allows forces to be transferred to the walls. Therefore, *Equation (B.2)* over-predicts the actual pressures in the pebble-bed.

In the presence of static friction, more complicated calculations are required. The fact that static friction transfers force to the wall was observed by German engineer H. A. Janssen in 1895 (Sperl, 2006). Walker (1966) derived formulas for the pressure on the wall for cylindrical vessels with conical exit chutes. The conical exit chute will concentrate the force around an outer ring to the extent that static arches form (Walker, 1966). For a cylinder, essentially, when the upward force on the wall from static friction for a given segment matches the downward gravitational force from the additional pebbles in that segment, the pressure stops increasing.

For a cylinder, the horizontal pressure equation is (Gotoh et al., 1997):

$$P_h = \frac{\gamma D}{4\mu_w} \left[1 - \exp\left(\frac{-4\mu_w K}{D} x\right) \right] \quad (\text{B.3})$$

where γ is the bulk weight (or $f\rho g$), D is the diameter of the cylinder, μ_w is the static friction coefficient between the pebbles and the wall, K is the Janssen coefficient, and x is the distance below the top of the pile.

The Janssen coefficient depends on the pebble-to-pebble static friction coefficient and can be calculated from:

$$K = \frac{1 - \sin \phi}{1 + \sin \phi} \quad (\text{B.4})$$

where $\tan \phi = \mu_p$ and μ_p is the pebble-to-pebble static friction. Since $\tan^{-1} \mu = \sin^{-1} \left(\frac{\mu}{\sqrt{\mu^2 + 1}} \right)$ then K can also be written as:

$$K = 2\mu_p^2 - 2\mu_p \sqrt{\mu_p^2 + 1} + 1 \quad (\text{B.5})$$

The Janssen formula derivations make assumptions that are not necessarily true for granular materials. These include assuming the granular material is a continuum and that the shear forces on the wall are at the Coulomb limit (Bratberg et al., 2005). The static friction force ranges from zero at first contact up to μN (the Coulomb limit) when sufficient shear force has occurred. If the force is not at the Coulomb limit, then an effective μ could be found and used instead. In general, this assumption will not be the case when the pebbles are freshly loaded because they will not have slid against the wall enough to fully load the static friction. Even after the pebbles have been recirculated, they may

not reach the Coulomb limit, and effective values for the static friction constant may be needed instead for predicting the wall pressure. Finally, real reactors have more complicated geometries than a smooth cylinder above a cone exit chute.

B.2 Prior data on dust production

The 46-MWt thermal pebble-bed reactor AVR was created in the 1960s in Germany and operated for 21 years. The pebbles were added into the reactor through four feeding tubes spaced around the reactor and one central feeding tube at the top of the reactor. There was one central outlet chute below. In the reactor cavity were four noses of U-shaped graphite with smooth sides for inserting the control rods. The cylinder walls contained dimples about 1/2 a pebble diameter deep that alternated location periodically. All structural graphite was needle coke graphite. Dimensions are shown in Figure B.2, and design and measured data is provided in Table B.2. The initially measured dust-production rate was 3 kg per year but had to be increased after inspections performed during dismantling to 100 to 200 kg total (Moormann, 2008b; Bäumer et al., 1990), or about 10 kg per year. R. Moormann estimates that 5 kg of this yearly production is from mechanical abrasion. Several sources of this dust have been proposed in addition to wear. There was a 0.12-m³ oil ingress in 1982 that could have produced up to 75 kg of carbon dust (Moormann, 2008b). Due to the high concentration of hydrogen and carbon monoxide impurities in AVR (see Table B.4) the chemical reactions of $C + H_2O \rightleftharpoons H_2 + CO$ and $C + CO_2 \rightleftharpoons 2CO$ were proposed as a possible method of producing about 60 kg of carbon dust (Nieder, 1990). It is difficult to draw solid conclusions about the quantity of dust produced in AVR because of a water ingress, an oil ingress, the uncertainty in the composition of the dust (i.e., metallic components), and the uncertainty of the location of dust production (Bäumer et al., 1990; Atomwirtschaft-Atomtechnik-atw, 1966). In addition, the interior of pebbles in the AVR reactor reached over 1280°C as determined by melt-wire experiments (Moormann, 2008a).

Germany's Thorium Hochtemperatur Reaktor (THTR-300) was a thorium- and uranium-powered pebble-bed reactor that first went critical in 1983 and ran through 1988 for 16 months of full power operation. THTR-300 produced 16 kg of dust per full power year (FPY), and an estimated 6 kg of that was produced in the core of the reactor (Wahsweiler, 1989). The control rods in the THTR-300 actually pushed into the pebble-bed. On a per-pebble basis, the amount of dust produced in the THTR-300 was lower than in the AVR. Further data on the THTR-300 is summarized in Table B.3 (tth, 1989, 2009; Moormann, 2008b).

Table B.2. AVR data.

Name	Value
Average inlet temperature	250°C
Average outlet temperature	950°C
Pebble circulation rate	300–500 per day
Dust produced	10 kg per year
Pebbles in reactor core	100,000
Reactor radius	1.5 m
Outlet chute radius	0.25 m
Angle of outlet cone	30°
Control-rod nose thickness	0.3 m
Radius of control-rod nose	0.15 m
Feed tube to outlet chute	2.83 m
Total pebbles ruptured	220

Table B.3. THTR data.

Name	Value
Average inlet temperature	250°C
Average outlet temperature	750°C
Core height	6.0 m
Pebble circulation rate	3561 per day
Reactor radius	2.8 m
Pebbles in reactor core	657,000
Total dust produced	16 kg per FPY
Estimated in-core dust	6 kg per FPY
Total pebbles ruptured	8000

Table B.4. Helium impurities in AVR and THTR (Nieder, 1990).

Impurity	AVR	THTR 300
H ₂ O μ bar	≤ 0.5	≤ 0.5
H ₂ μ bar	50–100	15–35
CO μ bar	300–800	<12–15
CO ₂ μ bar	<5–20	<4–6
CH ₄ μ bar	2–10	<4–8
N ₂ μ bar	50–300	<4

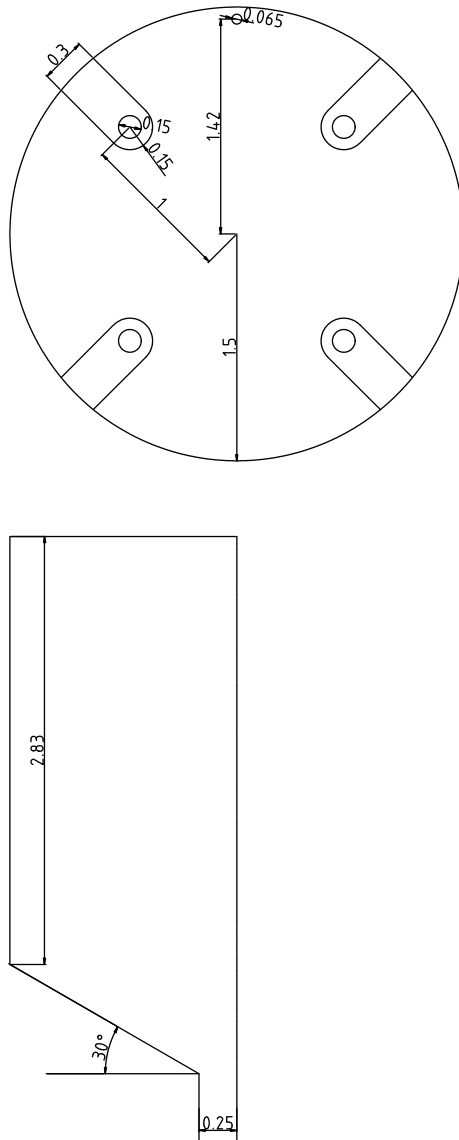


Figure B.2. AVR dimensions.

B.3 Prior Prediction Work

Two papers attempt to predict the in-core pebble dust production. The first, “Estimation of Graphite Dust Quantity and Size Distribution of Graphite Particle in HTR-10,” (Xiao-wei et al., 2005) estimates the dust production that the core of the HTR-10 reactor would produce. The second, “Pebble Bed Reac-

tor Dust Production Model,” (Cogliati and Ougouag, 2008) is by this report’s authors and attempts to estimate the dust that the AVR reactor produced.

The HTR-10 paper (Xiao-wei et al., 2005) started by calculating from the hydrostatic pressure the force between the pebbles at the bottom of the reactor. The force was approximated to be 30 N. The remainder of the paper uses 30 N as the force for conservatism. Note that the HTR-10 paper is in Chinese, so this literature review may contain mistakes in understanding due to language differences. As well, the paper contains a number of simplifying assumptions that the authors are working on correcting in a future paper (Xiaowei, 2011b), so the predictions should be considered rough estimates.

The dust production is calculated in three regions: the core of the reactor, the outlet chute of the reactor, and the fuel loading pipe.² The calculations are done with the data from the IG-11 structural graphite wear experiments.

For the core of the reactor the temperature used is 550°C with pebble-to-pebble wear rates of 4.2×10^{-3} mg/m extrapolated from 400°C data. The pebble-to-wall wear rates are extrapolated to 480°C to 12.08×10^{-3} mg/m from the 400°C data. The pebble-to-pebble wear is estimated to occur for³ 2.06 m, and 3.85% of pebbles are estimated to wear against the wall. From this data the average pebble dust production per pass in the core is determined to be 8.65×10^{-3} mg for pebble-to-pebble wear and 0.99×10^{-3} mg from pebble to wall. The total in-core graphite dust produced per pebble pass is 9.64×10^{-3} mg.

The outlet chute wear is estimated to occur for 2.230 m in the graphite portion and 1.530 m in the stainless steel portion, and that 44.16% of the pebbles wear against the chute. Both these portions are estimated to be at 400°C. Wear rates of 3.5×10^{-3} mg/m are used for the pebble-to-pebble wear, 10.4×10^{-3} mg/m for the pebble-to-graphite chute, and 9.7×10^{-3} mg/m for pebble to steel. Thus, for the outlet chute, the upper portion has 18.05×10^{-3} mg of dust produced per average pebble and the lower portion has 11.91×10^{-3} mg produced, for a total outlet chute amount of 29.96×10^{-3} mg per pebble pass.

The fuel loading pipe is approximately 25 m long and the temperature is 200°C, which gives a wear value of 2.1×10^{-3} mg/m and 52.50×10^{-3} mg per pebble pass. Thus, for an estimated average pebble pass, 10.5% of the dust is produced in-core, 32.5% is produced in the outlet chute, and 57.0% is produced in the loading pipes. Note that the authors consider the loading pipe estimate to be an oversimplification (Xiaowei, 2011b), so the actual loading pipe percentage could be significantly different. The HTR-10 paper estimates that 50% of the outlet chute graphite dust enters the core and that 75% of the graphite dust produced in the fuel loading pipes enters the reactor core, for a total amount of graphite dust entering the core of 64.0×10^{-3} mg per pebble pass.⁴ Since 125 pebbles per day enter the reactor, at 365 days a year, this works out to 2.92

²For consistency with the wear papers (Luo et al., 2004, 2005; Xiaowei, 2011a), μg is replaced with mg.

³This is the length slide and is multiplied by 4.2×10^{-3} mg/m to get per-pass dust production.

⁴ 9.64×10^{-3} mg + 29.96×10^{-3} mg*.5 + 52.50×10^{-3} mg*.75 = 64.00×10^{-3} mg

g/year of pebble dust ⁵(Xiao-wei et al., 2005).

HTR-10 has 27,000 pebbles and a rate of 125 pebbles per day compared to AVR's 100,000 pebbles and a rate of 400 pebbles per day. Multiplying these into crude scaling factor⁶ gives an estimate of 35 grams of dust per year that would be produced in AVR. Measured values of dust generation rates from HTR-10 would provide valuable information on pebble-bed reactor dust production but appear to be unavailable.

⁵Reported in the paper as 2.74 kg/year due to a precision loss and unit errors

⁶ $(100,000 \text{ pebbles}/27,000 \text{ pebbles}) \cdot (400 \text{ pebbles per day}/125 \text{ pebbles per day}) \cdot 2.92 \text{ g/year}$
= 34.6 g/year