A Model To Predict the Lifetime of Pneumatic Conveyor Bends

Doctoral Thesis

Centre for Numerical Modelling and Process Analysis

University of Greenwich,
30 Park Row,
SE10 9LS, England.

Robert Hanson

28th September 2001

Abstract

Bursts in pneumatic conveyor pipelines in industrial processes are a well-known hindrance to the smooth operation of any plant. Unanticipated stoppages can have serious financial implications for any company using pneumatic conveying technology, and health and safety factors are also paramount. This thesis describes an attempt to enable improved prediction of bend-wear and bend lifetime, so that more cost-effective survey work can be undertaken in anticipation of bursts.

This work delivers a tool that allows bend lifetime prediction to be made according to: the bend geometry and material; the material conveyed and its rate of transportation; and bend wall thickness.

Firstly, a computational model based on the coupling of CFD and particle tracking techniques is created in order to encompass the mechanics of the erosion process. This erosion process is assumed to be dominated by impact damage, and predictions of bend lifetime are made using empirical erosion algorithms gleaned from laboratory experiments, commercial CFD code (*PHOENICS*, and *GENTRA* its particle tracking sister code), and custom erosion modelling code that employs a three-dimensional toroidal geometry.

Secondly, matrices of predictions are built up using the mathematical modellng technology mentioned above. These predictions are collated behind a friendly interface to produce a far more accessible piece of PC software that an engineer can employ quickly and easily. More general bend-life predictions are interpolated from this fundametal dataset using behaviours established in the course of this work, according to the particular conveying conditions input by the user.

Predictions in the desktop tool are calibrated to actual bend lives as established by experimentation on a full-scale pneumatic conveyor. This experimental work was an integral part of this EPSRC-funded project, and allows some estimation of error magnitude in the predictive tool.

Acknowledgements

Thanks go to my supervisors for their help in driving the project forward: Mayur Patel and Mike Bradley; also to Ian Hutchings, the other senior investigator involved, and David Allsopp.

Thanks also to my many friends and colleagues at the University of Greenwich who managed to share my workload in some small way, and especially the staff of the Wolfson Centre who facilitated the collaboration, and were helpful in the extreme often saving me much time: Richard Farnish, Andrew Pittman, Lynne Whaley and others.

Most fulsome praise is reserved for the experimentalists at the Wolfson Centre: David Smith and Tong Deng. I salute you.

Author's Note

All of the work in this thesis is the sole and original work of the author, except where indicated by acknowledgement or reference.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis is a thorough description of a technique for the mathematical modelling of erosion. The different softwares used for the purpose are discussed. Some initial considerations and difficulties are addressed before the erosion model itself is described. Results are also presented, with empirical validation wherever possible.

1.1 Pneumatic Conveyors and Erosion

A pneumatic conveyor is an industrial transportation device whereby fragments of the conveyed material are carried from points A to B along pipework of the appropriate diameter in a flow of air.

This form of transportation is typically less energy efficient than conveyor–belt operations, but an attraction is that the conveyed material is entirely enclosed in its own atmosphere, inside the pipework. Dust problems are consequently much less severe, and this is often an important Health and Safety consideration.

The pneumatic conveyor form of transportation is now sufficiently prevalent to warrant its own science, including research on pressure drop and bend shape, particle shape and size, and particle–particle interactions.

This thesis concentrates on one specific problem associated with the pneumatic conveyor: the erosion of bends [Hum90]. In commercial situations conveyor pipelines are often obliged to follow convoluted routes. The constant large change in the momentum of the flow at bends in the pipeline leads to these regions being hotspots for erosion and bursts.

The value of the work in this thesis lies in its potential ability to predict *where* and *when* a bend will burst. This would allow a maintenance schedule to be allocated more economically, and fewer unanticipated bursts would save proportionate amounts of time

and money.

Note that the time lost to an industrial process when a bend bursts is only part of the loss, and that the conveying pipeline is often a link in a chain of communicating processes. Add in the time lost to prior and post–processes, and the gravity of the problem becomes evident.

When one considers the time and money involved in cleaning spillages, and potential health hazards, it is clear that even a small improvement in the current typical bend–lifetime prediction accuracy of ±80% [Bur96] would have a strongly beneficial effect on Industry.

It is worth mentioning here that erosion, or wear, is a universal problem in engineering, and much research has been applied to it historically and up to the present day. In many areas, lubrication is a closely related topic, though not in a conveyor. The emphasis in pneumatic conveying is on product integrity, meaning that the transported material should not be damaged or polluted.

Attempts to quantify erosion have been ongoing since at least the 1950's and 1960's, and some of these early curve fitting efforts are still used today [Fin60] [Bit63]. Erosion rates are notoriously difficult to pin down however, with a change of equipment, or a change to a seemingly similar material often producing unexpected results. Reasons for this are attributed to the complexity of the erosion process at a microscopic level, and much more is now known on this topic than was available to the 20th century pioneers in the field. Electron micrographs can magnify surface damage thousands of times, and postulated cracking, chipping, gouging and cratering phenomena are now seen to be real enough. Modelling them is still difficult however [Hum90].

An approach taken in this work, is to assume that erosion is stochastic rather than deterministic, and unsurprisingly *Monte Carlo* methods have been used before [NS95]. Also, empirical erosion data matrices are used, to remove a potential source of error with analytic erosion models. This approach does not seem prevalent, but perhaps only because it requires access to an erosion tester and the associated time and labour.

Similar works to the present one, often using applied CFD have been undertaken in diverse fields: valves [NEL] [HKS00]; hydroelectric power [IAoHER]; helical gears [FA01]; and of course pneumatic conveyors [FA96].

Work that examines the evolution of a worn surface is less common, but a nice example is [Öqv01], where a time–stepped finite element analysis is undertaken to solve a stress field in a rubbing problem. This is necessary because of the wear model used (Archard). Work in Chapter 6 of this thesis adds to the nascent evolutionary trend in erosion modelling.

The desire with erosion modelling is to predict the rate at which damage will occur in particular circumstances. This thesis will recount such predictions and the manner in which they were arrived at, in an unbashful manner, not typical of most current work in the field.

1.2 Research Objectives

In this work, the main objective is to produce a predictive tool to aid engineers involved in pneumatic conveying. Major questions that arise out of this are:

How predictable are bend lives?

It is a common assumption that bend lives are predictable, if only sufficient data were forthcoming as to the precise nature of the conveying conditions, materials, etc. The work in this thesis does not seek to deny a possible stochastic element to wear processes. Experimental data will be forthcoming to help with this question. Any stochastic element to bend life will of course set a lower limit to the accuracy of any predictive model.

What is the best way to model bend wear?

A variety of approaches to erosion modelling will be discussed, in a variety of contexts. This is discussed in depth in Chapter 2.

Does surface evolution change the rate of wear significantly?

This is important for someone constructing a predictive model, because erosion tests are usually brief, and measure mass loss rates only while target penetration is relatively shallow. This is an area where there is minimal existing work at time of writing.

The above questions are revisited in Section 9.2.

1.3 Outcomes

The goal of this project is to produce user–friendly desktop software to allow engineers and designers to make predictions with regard to the life of their pneumatic conveyor bends. With a predictive tool of a given accuracy, expensive errors in design and maintenance could be minimized, and cost–savings hence introduced.

Such a tool might be built up from mathematical models, if a predictions matrix of sufficient dimension and scope could be obtained. Interpolatory techniques could then be

used to generalise this fundamental matrix to a wide range of bend geometries, materials, and flow conditions.

1.4 Project Partners

Several Industrial and Academic collaborators have contributed to the success of this project. ABB Combustion Services Ltd. supplied historical data from their plant, and were represented at the Steering Committee by Dr Alan W Thompson. Alstom also contributed, represented by Brian Simpson. Clyde Pneumatic Conveying Ltd. were represented by Brian Snowdon and Richard Bingham, with assistance from Roy Firth and others on site. PowerGen Plc allowed us to inspect one of their huge conveyors at Kingsnorth Power Station, and also supplied data and materials. Two pipe bends were also installed at their Power Technology Centre at Ratcliffe-on-Soar, providing more data for the *Wolfson Centre*. Dave Miller was their representative.

Sincere thanks to the above Companies and their representatives for adding to the credibility of this project.

Academic collaborators were: Dr. William Easson at the University of Edinburgh; Professor Ian Hutchings and Dr David Allsopp at the University of Cambridge; and David Smith and Dr Tong Deng at the Wolfson Centre, University of Greenwich. The author's supervisors were Dr Mayur Patel (CMS) and Dr Mike Bradley (The Wolfson Centre) at the University of Greenwich.

Chapter 2

Erosion Modelling

In this chapter, different ways of measuring and modelling erosion are discussed, and previous applications tied in with a CFD approach are reviewed. An approach is suggested that could enable the integration of particles that strike twice on the same bend into the model. Also, an outline of a simple desktop tool is given.

2.1 Previous Erosion Modelling Work

Erosion, or Tribology, is a branch of Materials Science dealing with the damage or mass loss caused by particles striking a material surface. This is distinct from Abrasion, where particles slide along a surface. For many years efforts have been made to model erosion rates analytically, using only a few empirically established constants. Some progress has been made though a general purpose model that is accurate is not forthcoming. The following discussion describes some of the variables involved, laboratory equipment used in the field, erosion models, and efforts to employ these to make wear predictions in industrial applications.

2.1.1 Erosion Variables

The problem of solid particle erosion in industry is well known, with pipe bends and valves being two of the most common problem areas. Humphrey [Hum90] has set out the problem well, pulling no punches. There are numerous variables that can affect the behaviour of a large number of particles in a flowing medium, and numerous factors again that come into play with the erosive mechanism.

At a point of erosion, the amount of material removed by a single impact on a target, or

pipe wall, is liable to vary enormously. The prime variable is impact speed, and the angle and mass are the others whose dependence is most easily quantified. The variables that make erosion so unpredictable are: the material conveyed and previous collisions it may have suffered, e.g. at a previous bend; the target material and its surface; homogeneity of either material; particle shape; and particle spin and orientation at the point of impact.

In the case of pneumatic conveyors, bends come in many different shapes and sizes, and as erosion occurs the initial geometry of the bend or component is changed (by definition). The problem from a mathematical modelling point of view is therefore seen to require considerable simplification. To preserve all of the core dependencies of the problem is however crucial. Fortunately help is forthcoming from other authors, when it comes to establishing which of the above variables are the least vital.

2.1.2 Measuring Erosion Rates

Erosion rates are usually quantified as g/kg; that is, *grams* of mass lost from the target per *kilogram* of erodent material striking, at a given angle and speed of impact. The most common type of erosion tester is the gas–blast type, where particles leave a nozzle entrained in a pressurised air–flow. This nozzle will be fixed on a bench such that the exhaust strikes the required target at the required angle, and the particle velocity is controlled using air pressure. The mean particle velocity can be measured, using PIV (Photographic Image Velocimetry) or LDA (Laser Doppler Anemometry) techniques. For calculations of the fluid velocity using Bernoulli's equation 2.1[Mas89], it is required that the velocity is known at some point in the system to establish the value of the constant. This is often the case with conveyors, but may not be so with a gas–blast nozzle.

The law states

$$\frac{p}{\rho} + \frac{u^2}{2} + gz = \text{constant} \tag{2.1}$$

along any streamline.

Another more sophisticated type of erosion tester is the rotating disc variety. This is essentially a centrifuge, with numerous radial exit paths for balance. The device is contained within a drum to collect the used erodent, along with numerous targets, all of which can be oriented to create differing angles of impact. Advantages of this are: many targets at varying angles can be tested at once; the exit velocities of the particles can be more accurately calculated [SHES81], as the stream width is narrower; and transverse components of velocity are minimised compared to particles leaving the relatively wide nozzle of a gas–blast tester.

Disadvantages of the rotating disc tester are: more erodent is required (not all will hit the targets which need some spacing due to particle rebound); particles can develop spin as they accelerate along the walls of the centrifuge tubes, and this affects erosion rate, possibly to a large extent [DBB01][Den01]; and the closed drum is not amenable to viewing if multiflash photography or similar is required. These optical techniques are invaluable for checking particle velocities, or measuring rebound velocities and angles, and are usually employed alongside the gas–blast type of tester.

Lastly, the rotating disc erosion tester has an electric motor with brushes which can spark, so it is unsuitable for testing pulverised fuel (very finely crushed coal dust) which is highly inflammable.

Following are summaries of some well known investigations into erosion.

2.1.3 Modelling Erosion Rates

Scientists keep on trying to quantify the process of erosion, but valid interpolations and extrapolations from experimental data are extremely difficult to make. Vast amounts of literature exist, but firm conclusions drawn are usually qualitative and not quantitative.

The point of an erosion model is that it would remove the need for time–consuming and expensive experimentation, but all models mentioned in this section require at least some empirical constants. The typical result from use of such a model is that fair or even good predictions can be made in the region close to given data points, and moving away from these, an order of magnitude of error quickly becomes possible.

There are multiple factors in the erosive process which contribute to its complexity [Fin72]; particle size, shape, spin, friability, hardness, velocity and impact angle, material properties of the target, and there are more. Erosion–Corrosion will not be discussed here, and ambient temperature will be assumed throughout.

With the advent of high powered microscopes in the 1970's, some of the mechanisms of erosion have been clarified somewhat but modelling these has not proved any easier. For a ductile material, the erosive process can consist of pure cutting, extrusion and subsequent cutting, and forging and fracture [Lev95]. Melting is also believed to occur in some instances [YC78] but whether these or less extreme thermal effects have a work–hardening or annealing influence on the eroded surface is still uncertain [HL89], depending very much on the kinetic energy of the particles in question.

For brittle materials cracking and chipping dominate [SF66] the erosive process, though plastic effects are also suspected [EGR78]. Other phenomena which have been investigated

as relevant are particle shape [CH83], particle–particle collisions in the incoming erodent [AH83] [OP93], and particle breakage on impact [PC97][BWBB95]. While this work is relevant, interesting, and useful, the current state of erosion modelling is such that the incorporation of this level of detail is not always possible or desirable. As computational power is growing, modellers may well look back at this material at a later date.

The first multipurpose erosion model is attributed to Finnie [Fin60] and was of this form:

$$Q = \frac{MV^2}{2P}[\sin 2\alpha - 3\sin^2\alpha] \qquad \alpha \leq 18.5^o \tag{2.2}$$

$$Q = \frac{MV^2}{24P}\cos^2\alpha \qquad \alpha > 18.5^o \tag{2.3}$$

where Q is volume of material removed, M is total particle mass at velocity V, P is plastic flow stress [Fin60] and α is angle of impingement. This fitted the shape of the angular erosion dependence of a ductile material quite well but notice with this model that the erosion rate falls to zero at an impact angle of 90^o. The underprediction of this model at higher impact angles lead to efforts to model deformation wear. Finnie had based his model on the idea that incoming particles would gouge out tracks as they penetrated and then rebounded from the impacted surface, but erosion at higher angles had been demonstrated too, for which this hypothesis could not account. Deformation wear was therefore postulated by Bitter [Bit63], where the surface was cracked and chipped by cumulative near–normal impacts.

In his model he used the term

$$W_D = M\frac{(V\sin\alpha - K)^2}{2\epsilon} \tag{2.4}$$

where: W_D is volume material lost due to deformation wear; M and V are total mass striking and velocity; α is impact angle; K depends on physical properties, representing the threshold velocity at which erosion occurs; and ϵ is the energy required to remove unit volume of material from the target surface. This deformation wear term was added to a cutting term which could be made to match Finnie's by certain assumptions, and results were now encouraging, in that the angular dependence of erosion was now well modelled over the whole quadrant, even for brittle materials.

Other models followed, notably Nielson and Gilchrist [NG68] and Tilly [Til73]. These earliest and most famous tribological models were established before high powered microscopy, and have retained their status. Some intelligent guesswork meant that the cut-

ting or gouging erosive process was modelled reasonably accurately. Using microscopes the importance of the process of extrusion became apparent, blurring the boundary between ductile and brittle types of wear. This process is not so straightforward to model, and there have been no notable successes to date.

So some success has been found in this field, particularly with ductile materials, and the explanation of ripple formation [Fin65]; but change the materials studied, vary the speed of impact, and no existing erosion model will continue to predict accurately [Bah91]. The shades of pessimism (some might say realism) in the literature range from light [Ell93], through the middle ground [Bah91] to grim [Lev95].

Leaving analytical models behind for a moment it can be seen that a sure–fire way to be able to rely on an erosion model is to ensure that it utilises empirical data and little else, and that it therefore remains unambitious at heart. The end user of any erosion model, usually an engineer of some variety, has real–world concerns away from any abstraction. If an erosion model is to be used then error estimates must be quoted, and necessarily the set of circumstances for which these estimates have been established must also be available to the end user, so that any model is not unwittingly used out of context.

Given that experimental work is always required to establish constants in all contemporary erosion models, and that the demand for erosion models has never been greater (thanks to fast computers in conjunction with CFD and particle tracking); given these points, it may be that in the future tribologists will lean more and more towards the experimental side of their work in order to get fast reliable results. That certainly was the case in the present work [HP00]. Other works previously carried out at the Wolfson Centre at the University of Greenwich, are described in [BPB98], [BBB95] and [BDSR95], and were precursors of the current project.

2.1.4 Predicting Mass Removal

Erosion models are now looked at in context; where they are used to predict mass loss to estimate the life of components such as turbine blades, valves, and bends in ducts.

Assuming a reliable erosion model has been established, it should be a relatively straightforward matter to calculate mass loss rates over a surface, again assuming that the particle impact distribution is known. Naturally CFD and particle tracking are a great help in these circumstances, although they do require experienced users and probably software licences too. This makes this particular science unattractive to the casual user, and the consequence is that actual engineers do not often use these modelling techniques, or

test them to destruction. The users or authors of the models are typically mathematical modellers with limited access to experimental data. Usually one or two cases of interest can be modelled well, but it is only known that they are modelled well because data is available for these cases. There is minimal merit gained in predicting a known outcome. Engineers will only take predictive attempts seriously when they are performed before the receipt of corroborative data.

All of the techniques reviewed here have employed CFD technology. The flow is modelled, particles are tracked, and the resulting impacts have an erosion model applied. Shirazi et al.[SMSR95] try to simplify the procedure for elbow pipe bends, but still use CFD and particle tracking. The results look good over four orders of magnitude of erosion rate, which is impressive. Whilst giving a good estimate of relative erosion rate, the location of maximum wear, or the distribution of wear is not given, so that the actual lifetime of the component is not explicitly predicted. This is often the case in the following works.

Manickam et al [MSM99] have used the standard approach of 1) CFD, 2) particle tracking and impact characterisation, and 3) erosion modelling, to produce alternatives for the design of turning vanes in a bifurcation duct. Finnie's [Fin60] early erosion model was used, and while the wear predicted was not converted into a lifetime estimate, the locations and intensity of the wear was sufficiently well modelled to suggest that one design had an advantage over another.

Probably the state of the art is recounted in Wallace et al.[WPS+00] where some complicated valve geometries have been subjected to high resolution CFD mesh. The level of detail of impact intensity is very good, and it seems that CFD and particle tracking have overtaken tribology, in that fluid flow and particle tracks can be more reliably predicted than wear. No lifetime predictions are made in this work due to pending experiments, but using on this occasion Neilson and Gilchrist's [NG68] work, it is probable that the actual erosion mass loss model would be the main source of error in this type of simulation.

Sato et al [SSY95] tracked 5000 particles through a flow duct and got good results using Finnie [Fin72] and Bitter's [Bit63] equations. Forder et al [FTH98] used the work of Hashish [Has88] and Bitter [Bit63] to predict the location and intensity of wear in oilfield valves, and this was corroborated with used components.

Lastly, Parslow et al [PSST99] modelled erosion by a cyclic recirculation in a blinded tee duct bend. The erosion model used is novel in that it is based upon the penetration rate of multiple layers of paint, both in a rotating disc tester and a full scale rig. This technique was employed because the authors felt that the CFD–particle predictions they obtained were not fully capturing the recirculation of the particles, and this was thought

due to the heavy loading of the dense phase. The algorithm used neglected particle–particle interactions, and also the two–way coupling described in Section 3.2.5.

2.1.5 Evolution of Eroded Region

Going back to the definition of the word *'erosion'* it is clear that the most realistic model of this phenomenon would be one in which the change of geometry of the boundary is taken into account. The inclusion of this factor is thought to be more critical in geometries that exhibit sharp projections, corners or asymmetries, as the curved surface of a bowl or pipe will change less in a qualitative sense, unless the erosion is extremely localised. In pipe bends the surfaces are curved, so the evolution should not be extreme, but an investigation still seems attractive.

One example [Öqv01] shows a very satisfying change in geometry due to wear, but this rubbing situation requires no fluid modelling.

If CFD is to be used for a geometrically responsive model, the mesh should ideally be adapted or extended to account for the extra space available to the flow, as a result of mass loss from the boundary. One example of an approach to this problem has been found, where Zhang [ZM96] and Minemura model the wear inside a pump casing. However, this problem was a special case such that an analytic fluid flow solution could be used in a 2D simulation. The mesh used therefore was employed to model the substrate rather than the fluid. As the wear proceeded, elements of the substrate mesh were removed to represent mass loss, and the boundary was updated regularly with a spline–fit curve.

Though CFD was not used here, in principle it could be. Cells can be blocked initially, and incorporated into a flow domain as a simulation proceeds, or *vice versa.* The technique of cell-blocking is used in the modelling of solidification or melting [PC94] and could easily be adapted to model erosion. Question marks remain though as to how fine a mesh would be required in this region, and whether the boundary layer would be realistically modelled.

Another drawback to this kind of erosion model is that it is tied in computational time to a CFD and particle tracking code. Many examples in the previous section are similar; it would be much more computationally efficient to characterise the particle impacts once, and then use that information in a separate erosion model. Then reruns would be much faster and parameters could be changed without heavy computational investment. Of course with this approach, updating the fluid flow and tracks would not be possible while the erosion model is running, but updating the eroding substrate certainly would be.

2.2 An Efficient and Practical Approach

An overview of a mathematical modelling method is now given. This will utilise CFD and particle tracking software, and custom–programmed software to model the region of the pipe wall that suffers erosion. Also, empirical data must be used, this being supplied by the project partners recounted in Section 1.4.

2.2.1 Basic Erosion Modelling Technique

Given the large numbers of particles involved in the systems to be modelled, a statistical approach is the logical way to characterize their behaviour. Using CFD and particle tracking codes, a sample in excess of 20,000 particles will be tracked, and the position and velocity of each upon impact with the pipe bend will be stored. When this data set is complete, the CFD code is not required further, unless this data is to be updated in some way.

Next, interpretation software will read this data set and characterize the impact intensity as a function of position, and also relate velocities to particles according to their position of impact. This data will be in the form of probability distribution functions, or other, such that random number generation will allow the realistic generation of as many particles as required.

At this point, the erosion code (*Erode*) can be started, and the first task is to read in the output of the previous CFD software. This establishes the region of pipe bend that suffers impact, and also supplies all the data that will allow particle generation. The virtual pipe wall can now be stored in memory, and divided into small tiles along each of its three toroidal coordinates. The mass of each tile is calculated and assigned, and a flag for each is set to *True* indicating that initially, all the tiles are present in the wall.

Given that erosion is most commonly quantified as a mass removal mechanism, the above preparation now enables the simulation of erosion via the generation of many particle impacts and vectors. Projection along these vectors, and using empirical erosion data supplied facilitates the removal of a small amount of mass from whichever tile is impacted. When a tile's mass reaches zero by dint of a sufficient number of impacts, it is removed, and a hole is formed in the wall that will now permit the passage of any particle that would previously have struck it. On impact, a simple surface is constructed to account for the profile of the remaining tiles in that region, hence the angular dependence of erosion rates is realistically employed.

The amount of material conveyed in such a virtual pneumatic conveyor can be obtained

using the number of particles generated. Using the mean flow velocity and density, a lifetime for the bend can be calculated, once the outermost layer of tiles in the virtual bend wall is penetrated. This three–dimensional geometrical model will therefore also predict the position of burst, and once such a virtual rig is working, all data on pipe wall–thickness can be dumped at any stage of the virtual–conveying.

2.2.2 Modelling Erosion Due to Second Impacts

The previous section takes account of the first impact from particles incoming. To investigate second (or third) impacts, the surfaces previously constructed at the point of impact can be used in conjunction with empirical restitution and rebound angle data. This allows the projection of the particles back to the inside wall of the pipe from inside the worn substrate, and the resulting positions and velocities can be stored for future characterization. This data can be interpreted in a similar fashion to that of the impacting particles, allowing suitable p.d.f.s to be passed to an intermediary code. This can then generate as many particles as required in the form of a file that the CFD software can use.

It is now a simple matter to track these particles emerging from the pipe wall in CFD, and as before, trap their velocities and positions when they next strike the bend wall. This second impact data can then be utilised in the most fitting manner, once read into *Erode*.

To summarize the current technique: initial particle tracking is performed to establish first impact data. Now an initial dummy run of *Erode* is performed whereby the rebounding particles are stored. This stage involves some arbitrary choice of which rebounds to store; the early rebounds, or later rebounds, as in principle these should evolve due to them striking an evolving substrate. This matter is discussed in Chapter 6.

The rebounds are passed back to the CFD code in suitable form, such that the particles can be retracked until they impact a second time. The resulting second impact data is then used in another interpreter code. This reads the previous impact data file again together with this second impact data, and produces two independent impact maps which can then be read into *Erode*. With these two impact maps in memory, *Erode* can generate first and second impacting particles in an alternating fashion, thus allowing the virtual wear to proceed more realistically.

2.2.3 Desktop Software

This is described in Chapter 8, with the aim being to produce something very straightforward to use. The end product will incorporate a matrix, or matrices of predictions, allowing

interpolations to be made between them. Each prediction will come from the full-fledged mathematical model, for a given set of conveying circumstances. The dimensions of the matrices will allow for air-velocity in the bend, and its R/d ratio. All other variables will be accounted for in some other fashion.

The user can input information about geometries and conveying conditions, and pick materials from an existing dataset. This materials dataset will be limited, but the product will be designed in such a way that extension is possible on a time scale of days. If the user has no access to an erosion tester, *The Wolfson Centre (wolfsonenquiries@gre.ac.uk)* at the University of Greenwich should be able to help.

The desktop predictive tool will run on *Microsoft* platforms, and be in the form of a spreadsheet with additional dialog boxes for data input and help. Engineers can thus obtain lifetime predictions for their own bends, under their own conveying conditions.

2.3 Chapter Summary

In this chapter a review was given of existing work in the fields of erosion measuring and modelling, and applications of this, often combined with CFD, to industrial problems. While there are plenty of instances of this kind of work, hard numerical data for predictions, as well as component lifetime, is harder to come by. CFD is a relatively new science compared to Tribology, but it may be that the nature of erosion means that this more mature science is contributing more to any modelling errors, when the two processes are combined.

Modelling the evolution of a worn substrate is starting to become more popular with the advent of powerful new computers. It can involve a substantial investment of design and programming time, as there are no commercial software packages to supply a head start. Adapting CFD codes to do this kind of work could throw up serious problems, one of which could be being tied in computational time to a particle tracker.

An outline for a new erosion modelling technique for pneumatic conveyor bends is given. This was arrived at on consideration of the benefits and drawbacks of existing work in the erosion modelling field, and seeks to separate the CFD and erosion stages. Given that more than a single impact per particle is to be considered, this separation could accelerate greatly the testing stage necessary when attempting to construct surfaces and normals that will provide realistic particle rebounds.

Further, a plan is recounted for a simple to use desktop predictive tool. This will enable engineers to obtain fast and accurate predictions of when their pipe-bends might burst,

according to their individual conveying conditions.

Chapter 3

Particle Tracking using CFD Software Tools

In this chapter are described some of the problems associated with modelling a fluid-particle flow. A summary of the science and practice of CFD, or computational fluid dynamics is then given. The use of commercial CFD software is then discussed. Initial tests with this are carried out in order to set some numerical controls, and some sample results are then shown.

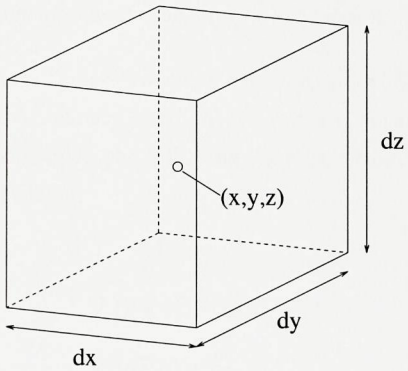
3.1 Aspects of a Fluid-Particle Flow

In a typical pneumatic conveyor, a highly turbulent flow of air is experienced (e.g. Reynolds number of 68500 using air velocity of $20ms^{-1}$, pipe diameter of $0.05m$, with kinematic viscosity of $1.46e10^{-5}m^2s^{-1}$) which is designed to entrain or carry a large number of particles along with it. The particle size must be relatively small compared to the bore of the conveying pipe in order to prevent blockages. It is assumed in this work, though this may not always be the case depending on the skill of the conveying engineers, that the conveyor will be working in lean phase. This simply means that the particles will be carried by the air, as opposed to sliding along the 'floor' of the pipework, or being pushed along in a slug or a rope.

The conveyed particles will often vary in shape and size, and also their spin will differ. They may break up as they experience collisions, either with the pipework or amongst themselves.

The extent of particle collisions depends upon the conveying density, but as pneumatic conveyors work in lean phase (air lacks the density or viscosity to carry much in its flow)

Figure 3.1: Volume Element



it is often assumed that the number of inter–particle collisions is 'low'. Work has been done in order to assess the effect of particle density on erosion rates [AH83][Bur96], which suggests that lean phase conveyor loadings are not too vulnerable to this dependence. As to how inter–particle collisions affect the flow and the turbulence, is not well established. In fact the mere presence of particles in a flow requires the use in CFD of turbulence models which are hardly well documented. Time dependent turbulence models for the presence of particles cannot at this stage be used, as there is too much computational power required in order to track simultaneously anything like the number of particles involved in a real system. Validating any such turbulence model requires the experimental tracking of large samples of particles, and this cannot be done, if at all, unintrusively.

3.2 Computational Fluid Dynamics & Particle Tracks

3.2.1 Equations of Fluid Flow

Computational fluid dynamics or CFD, is used for predicting fluid flow and other transport phenomema such as heat transfer. A variety of numerical techniques are used to solve the non-linear partial differential equations which model the physical situation. These are constructed using differential calculus, considering an infinitesimal volume element in the fluid (Fig. 3.1), and the classical conservation laws of physics:

- Mass is neither created nor destroyed
- Rate of change of energy equals the rate of heat applied plus the work done on the

system

- Rate of change of momentum equals the sum of forces on an object

Using the first two terms of a Taylor expansion over each coordinate for each variable to be accounted for (e.g. pressure, velocity components, density, enthalpy, chemical species) and the above conservation principles, the following equations can be found. For a patient and thorough derivation see [VM95].

$$\frac{\partial(\rho u)}{\partial t} + \nabla.(\rho u \underline{u}) = -\frac{\partial p}{\partial x} + \nabla.(\mu \nabla u) + S_{M_x} \tag{3.1}$$

$$\frac{\partial(\rho v)}{\partial t} + \nabla.(\rho v \underline{u}) = -\frac{\partial p}{\partial y} + \nabla.(\mu \nabla v) + S_{M_y} \tag{3.2}$$

$$\frac{\partial(\rho w)}{\partial t} + \nabla.(\rho w \underline{u}) = -\frac{\partial p}{\partial z} + \nabla.(\mu \nabla w) + S_{M_z} \tag{3.3}$$

$$\frac{\partial \rho}{\partial t} + \nabla.(\rho \underline{u}) = 0 \tag{3.4}$$

Here S_{M_i} are momentum source terms. They will incorporate gravitational or magnetic body forces, if such are present, and also surface forces,

$$P_i = \frac{\partial \sigma_{ij}}{\partial x_j} \tag{3.5}$$

where σ_{ij} is the stress (N/m^{-2}) on a plane normal to the i-axis in the j-direction. Also, ρ is fluid density (kg/m^{-3}), a scalar field defined as the mass of fluid per unit volume at each point of the solution domain. Similarly, p is fluid pressure, and the vector operator ∇, *del* or *nabla*, is defined as

$$\nabla = \frac{\partial}{\partial x_1}\,\underline{i} + \frac{\partial}{\partial x_2}\,\underline{j} + \frac{\partial}{\partial x_3}\,\underline{k} \quad \text{with} \tag{3.6}$$

$$\nabla\phi = \frac{\partial \phi}{\partial x_1}\,\underline{i} + \frac{\partial \phi}{\partial x_2}\,\underline{j} + \frac{\partial \phi}{\partial x_3}\,\underline{k} \quad \textit{or grad } \phi, \tag{3.7}$$

$$\nabla.\underline{F} = \frac{\partial F_1}{\partial x_1} + \frac{\partial F_2}{\partial x_2} + \frac{\partial F_3}{\partial x_3} \quad \textit{or div } \underline{F}, \text{ and} \tag{3.8}$$

$$\nabla \times \underline{F} = (\frac{\partial F_3}{\partial x_2} - \frac{\partial F_2}{\partial x_3})\,\underline{i} - (\frac{\partial F_3}{\partial x_1} - \frac{\partial F_1}{\partial x_3})\,\underline{j} + (\frac{\partial F_2}{\partial x_1} - \frac{\partial F_1}{\partial x_2})\,\underline{k} \quad \textit{or curl } \underline{F}. \tag{3.9}$$

The vector operator *curl* arises in Streamfunction–Vorticity methods of solving the Navier–Stokes equations, where the vorticity $\underline{w}$ is defined as $\underline{w} = \text{curl}(\underline{u})$ where $\underline{u}$ is the velocity field of the fluid, and $\underline{w}$ is a measure of its rotation at each point of the domain. These methods can be computationally beneficial when dealing with vortices in two dimensions because here, $\underline{w}$ effectively reduces to a scalar quantity. Although turbulent flow contains whole heirarchies of vortices and hence has non-zero $\underline{w}$, the stochastic nature of these means that there is loss of efficiency in applying a Streamfunction–Vorticity method more generally.

Summarising, equations (3.1–3.3) are the *Navier–Stokes* equations, and (3.4) is the continuity equation, which expresses conservation of mass. Together these describe fluid flow, but a CFD code can solve more complicated problems in general. The introduction of extra variables will result in other transport equations being added to this system, the general form of which for a scalar variable ϕ is:

$$\frac{\partial(\rho\phi)}{\partial t} + \nabla.(\rho\phi\underline{u}) = \nabla.(\Gamma\nabla u) + S_\phi \tag{3.10}$$

with Γ being a diffusion coefficient.

This equation is analogous to the *Navier–Stokes* equations, which are specifically concerned with the transport of momentum. In some sense the *Navier–Stokes* pressure terms can be considered sources of momentum, which would complete the analogy.

Note that the *Navier–Stokes* equations are derived using the assumption that the fluid modelled is Newtonian, that is that the viscous stresses are proportional to the rates of deformation. Other types of fluid (e.g. pseudoplastic, dilatant) would require different assumptions, and would lead to different equations.

So for a basic Newtonian flow problem, four equations are given in four unknowns: pressure, and three components of velocity. The problem now is to solve them, and apart from a few special cases, there are no analytical solutions. The next section describes a numerical procedure to overcome this.

3.2.2 Finite Volume Method

The most widely used method for the numerical solution of fluid flow problems is that of finite volumes. Also used but less common in this field are finite differences, finite elements and spectral methods. These other techniques are not discussed here.

The finite volume method is a weighted–residual method, with the weighting functions all set equal to unity [Pat80]. This technique is favoured because physical conservation

principles play a prominent role in the formulation, such that even using very coarse discretizations, physically plausible results are achieved. Accuracy is not guaranteed, and constant validation through experimentation is necessary. The definition of rigid rules for assessing the accuracy of CFD predictions is not straightforward due to the panoply of modelling techniques and flow conditions that might be encountered [Roe87][VSC98]. The equations of fluid flow are non–linear partial differential equations, and the solution has to be iterative whichever technique is employed.

The conservative feature of the discretization process comes from an initial integration of the PDE's over a small volume on the macroscopic scale. From here on the technique is similar to finite differences with derivatives approximated by backward differences. This leads to difference equations for each control volume in the domain, and these along with suitable boundary conditions represent a closed non-linear system.

Because of this non–linearity of flow problems, special solution procedures are necessary. To solve for a velocity field, the pressure field must be known, but in general it will not be. To overcome this, the equations are manipulated to yield a pressure–correction equation, [Pat80] which enables the velocity field to be updated bit by bit.

So, a guessed pressure field is used on the staggered grid [Pat80] which enables a solution for the velocity field. (Actually an initial guessed velocity field is also used to overcome the non-linearity in the equations.)

This solved velocity field is used to calculate an updated pressure field and coefficients, and then the procedure is iterated until convergence. Convergence is problem–specific, and can be difficult such that relaxation is usually employed. The first iterative method of this nature was called SIMPLE (semi–implicit method for pressure–linked equations) and was due to Patankar and Spalding [PS72]. Variants on this theme have followed to try and speed up convergence, including SIMPLER [Pat80], and SIMPLEC [VDR84].

So, a solution has been achieved over a discrete number of points in the domain, but the rather complicated matter of turbulence has not yet been discussed.

3.2.3 Turbulence

The Equations (3.1–3.4) describe laminar fluid flow precisely. There are many situations in real life however, where it is desired to model flows which are strongly turbulent. For a numerical definition of turbulence, the Reynolds number is used:

$$R_o = \frac{UL}{\nu} \tag{3.11}$$

where U and L are respectively characteristic velocity and length scales of the flow, and ν is the kinematic viscosity of the fluid in question. A quick calculation for a pneumatic conveying type flow ($U = 20ms^{-1}$, $L = 0.05m$, $\nu = 1.46 \times 10^{-5}m^2s^{-1}$ for air) gives $R_o = 68500$.

Figures quoted as the threshold for turbulence in pipe flows are between 2000 and 10000 [VM95], so this model is in the strongly turbulent regime. This means that there will be random fluctuations in the flow, and this can only be approximated using a time–averaged steady state. Note that the transition from laminar flow to turbulent flow is ignored in all but the most sophisticated models to date, so that the flow is considered fully turbulent.

In turbulent behaviour, eddies form in the fluid, and are extremely effective at transferring momentum across the domain, effectively accelerating the process of mixing in the fluid. These eddies tend to break up into smaller ones, move around, and break up yet again. This cascade of eddies or vortices make turbulence extremely difficult to model, as the length and time scales in which these eddies exist range across perhaps five orders of magnitude, from the scale of the domain, to the Kolmogorov scale. This is given in Equation 3.12, with ϵ the rate of dissipation of kinetic energy per unit mass, and ν kinematic viscosity. The Kolmogorov scale is an estimate of the smallest eddies expected to be observed, and is equal to 2 *microns* for the typical case quoted above. Plainly, to pick up this level of detail in a mathematical model is not yet possible, as the cells used would have to be much smaller than this.

$$L_K = (\frac{\nu^3}{\epsilon})^{\frac{1}{4}} \tag{3.12}$$

What is of practical use to engineers is to know the mean behaviour of the fluid over time, and several models are available to help make these predictions. They usually involve extra transport equations being added into the existing set, with such quantities as the rate of dissipation of turbulent kinetic energy, or the frequency of turbulent fluctations being the new variables.

Families of turbulence models are RANS (Reynolds averaged Navier–Stokes), DNS (direct numerical simulation) and LES (large eddy simulation). The presence of such choice in the field, as one might expect, indicates that no single model will produce good results for all situations. An exception to this comment is the DNS family, which allegedly give perfect results, but are computationally the most expensive thing this author knows of.

The $k-\epsilon$ (K-Epsilon) model [HN68] is well–validated for a variety of flows, is computationally efficient, and is supported by PHOENICS, the commercial CFD software employed.

It is the main candidate for use in this project.

Having outlined the computational fluid dynamics technique, a discussion is held on some relevant aspects of what is now a very broad field.

3.2.4 CFD: Flow of Air in a Pipe Bend

It is well established that CFD techniques can model accurately many complex fluid flow situations, and pipe flow in a bend is a relatively simple geometry compared to others [MSM99][WPS+00][FTH98] that have successfully been modelled.

The following references are examples of solving laminar flow in a pipe bend [Pat80][SB84], and a similar example of a turbulent flow can be found here [PPS75].

3.2.5 CFD: Predicting Particle Tracks

There are two commonly used techniques used to track particles in CFD. The first is used here and is known as Lagrangian particle tracking. The equation of motion of a particle in the fluid is used over a series of time steps, along with initial conditions, in order to calculate each new position and velocity of the particle. This is rather different to the second technique: Eulerian tracking. Indeed the Eulerian method [CW85] is not tracking at all in the sense that Lagrangian tracking is, because a mass distribution of particles is considered here, the mixture effectively being modelled as two fluids in the CFD.

Note that Eulerian particle modelling requires the solution of both phases simultaneously, as the mass conservative nature of the finite volume method requires that the relative proportions of fluid to particle material are known in each cell, and that the additive total of each proportion stays constant, assuming no phase change.

With the Lagrangian method however, the fluid flow problem can be solved separately, and then this solution can be used to track particles.

The equation of motion of a particle in a fluid is of the form

$$\frac{\partial \underline{u}}{\partial t} = c(\underline{U} - \underline{u}) + \underline{g} + \underline{h} \tag{3.13}$$

where $\underline{u}$ is particle velocity, $\underline{U}$ is local fluid velocity, $\underline{g}$ is gravity, and c is a constant depending upon fluid viscosity, drag coefficient, flow Reynolds number, and particle density and diameter. The vector $\underline{h}$ contains less obvious forces which for a large difference in densities can safely be ignored. For the record, they can include the Saffman lift force, Virtual mass force, pressure gradient, Magnus lift force and Bassett history force.[WPS+00]

Equation 3.13 can be integrated twice to give an expression for particle position [CSS77], and using this, a new particle position can be estimated for each time step. In this way, particles can be tracked, but the coupling has been one-way in that the fluid has influenced the particle, but the particle has not influenced the fluid.

In reality, action and reaction are equal and opposite, and one technique to account for this is the Particle Source in Cell model (PSI-cell)[CSS77]. As the particle passes through each cell of the fluid-flow approximation, a source term is created for the transfer of momentum (heat and mass transfer can be accomodated likewise if required) from the particle, and when the tracking is finished, some more iterations of the fluid-flow solution are performed in order to incorporate these source terms into the flow solution. From here, the particle tracking is performed again using the modified flow solution, and new sources are created to reflect the influence of the particle in each cell. The fluid-flow is again iterated, and the change in the flow on this and subsequent occasions should diminish as a stable solution becomes apparent.

Summarising this Lagrangian technique, an iterative method is developed that alternates between stages of particle tracking and source creation, and then using the flow solver. A converged solution can be reached up to some tolerance, whereby the particles have been tracked through a flow that has felt their influence. This approximates two-way coupling.

One point to note about steady-state flow solutions: in the elliptic fluid-flow aspect of the above procedure, the solution is reached by considering the particle sources in each cell as if the particle were present in each cell simultaneously. A way to reconcile this with reality is to imagine that this is modelling a batch of particles that are travelling along the same streamline. It then becomes entirely plausible that there is one particle in each sourced cell in the instantaneous solution. To track one individual particle with two-way coupling, it is seen that time-dependence would be a necessary characteristic of the fluid-flow solution.

Finally, finding a turbulence model that will reflect the dispersion of particles accurately requires that the stochastic nature of both turbulence and dispersion is taken into account. There is no general-purpose validated model for particle-particle interaction, and indeed, the lack of experimental data cannot help. To measure large samples of particle velocity and position instantaneously and unobtrusively is not yet possible. Nevertheless, some particle-turbulence models have been shown to be useful and these have been used [WPS+00][MSM99] in the absence of something better. Other authors have declined to move beyond turbulence modelling of the flow, awaiting further developments in multi-

phase flows [FTH98][TSM91]. For this work, a stochastic turbulence model is used that is discussed in Section 3.3.3.

3.2.6 CFD: Particle Impact Intensities

Simple statistical techniques have been used to characterise particle impacts per unit area of flow boundary, as seen in [Yeu79]. Mean impact angle and velocity components are also used here, as these are vital quantities to know when interpreting empirical erosion models. The particle impact work in this thesis is described in Section 3.5.

3.3 Use of PHOENICS and GENTRA

Reliable commercial codes performing Computational Fluid Dynamics are now well established, and also well tested. For the duration of this project the package PHOENICS is used, along with its particle tracking sister-code, GENTRA, as this has previously been used frequently and successfully in this department. e.g. [MGHP94], [GM91].

The techniques used in PHOENICS involve CFD algorithms and solvers that are based upon public domain theory, and the same can be said of GENTRA. The specific user–options employed are here described, along with some software modifications enabled and allowed by the commercial authors.

3.3.1 Flow Modelling

For this project, it is required to obtain a description of the flow of air in many different pipe bends, at many different speeds. PHOENICS has this capability. It employs the control volume method to numerically solve problems of fluid flow, and in fairness, many other types of problem. One such example of an extended problem is particle tracking through a fluid, and this is described later as it is also required for the current work.

To reduce computational expense, the air–flow problems to be solved are assumed to be steady state, or *Elliptic* by nature. This is an approximation as the turbulent fluctuations in the flow are by no means steady. However the mean flow of the air in a pipe can be described in this way. PHOENICS allows use of a body-fitted coordinate (BFC) mesh for entirely general structured grids, and this feature has been taken advantage of. Shown in Figures 3.2 and 3.3 is a typical set up, with an extended run–in before the bend (not shown due to magnification) and some spare pipe after it. The cylindrical polar mesh feature was also tried initially, but the cyclic boundary conditions necessary seemed to affect the flow.

Figure 3.2: BFC grid, cross–section

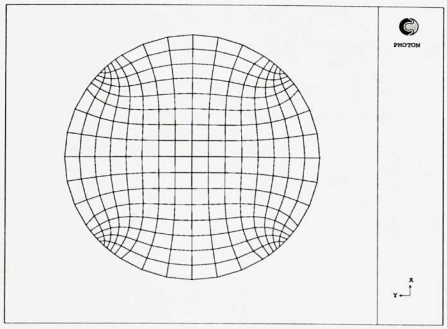
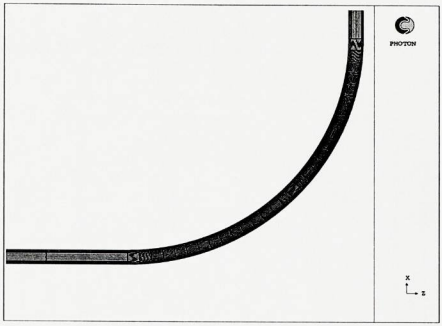


Figure 3.3: BFC grid, plan-view

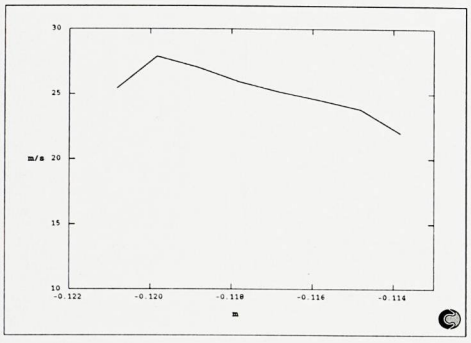


With the chosen option, the mesh cross–section starts as a square, and a Laplace Algorithm transforms the points into a circular form. The shown grid is actually built from multiple initial blocks to enable more cells closer to the boundary. A uniform grid that is passed through the Laplace Algorithm used tends to produce larger cells along the boundary.

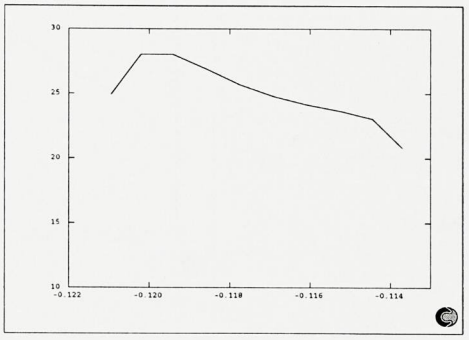
The long run–in allows the modelled flow to become fully developed due to wall friction, as it often will be in a pneumatic conveyor. This seems like an arbitrary assumption, but its opposite can be countered by the question 'Why build a conveyor over such a short distance?'. The grid construction code in the PIL (PHOENICS Input Language) was generalised from day one to enable salient features to be changed with a minimum of parameters: mesh density; pipe diameter; R/d (radius to diameter) ratio; and bend angle, though only 90^o bends are studied in this project. To help with geometrical transformations later, the pipe was set up so that the coordinate origin resides at the centre of curvature of the bend.

Some grid dependency investigations are shown in Fig.s 3.4 and 3.5, and this last was settled upon as being adequately representative. The profiles seen here can be quickly compared with the work of *Patankar et al.* [PPS75] and the experimental values they give, by quoting the maximum velocity on the profile as a percentage of the mean air–flow speed. The flow used (Fig. 3.5) in this work suggests 116%. Patankar's prediction for a helically coiled pipe is 120%, while the experiment suggested 127%. Patankar does not describe his mesh for this part of the work, but he was using 14 radial cells elsewhere in the paper. If he has a better resolution towards the boundary, it is to be expected he would get a better result. Also, the helically coiled pipe was used in order to achieve the *maximum* possible deformation of flow, and this was stated as achieved, though the point of satisfactory stability was not recounted. It is probably safe to assume that it was

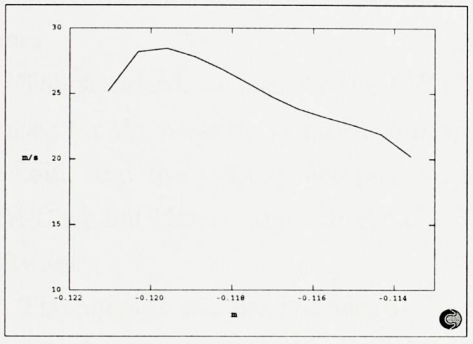
Figure 3.4: Flow Velocity Profiles at 81^o around bend



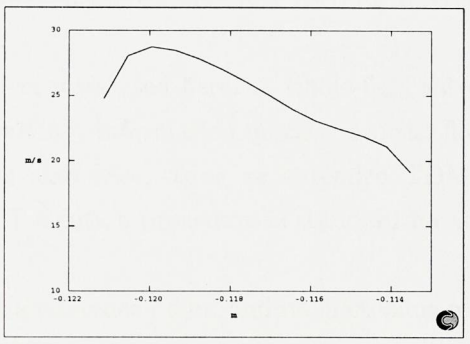
(a) Mesh: $8 \times 8 \times 54$



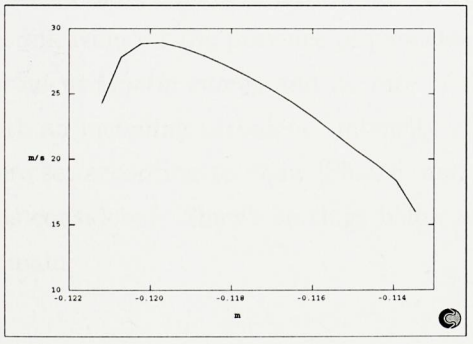
(b) Mesh: $10 \times 10 \times 75$



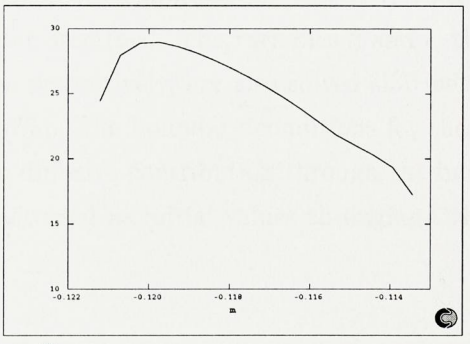
(c) Mesh: $12 \times 12 \times 100$



(d) Mesh: $14 \times 14 \times 150$

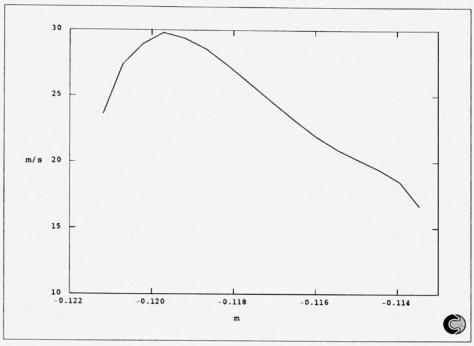


(e) Mesh: $16 \times 16 \times 196$



(f) Mesh: $18 \times 18 \times 250$

Figure 3.5: Velocity Profile at 81°: 16 × 16 × 50



considerably further around the coil than 81°.

On the kind of computer usually available to the author (sun-sparc20) a flow could be computed in around half an hour. Faster and slower machines were also used at different times.

The standard settings for the PHOENICS solver were used here. A whole-field solver is used for the pressure variable, to propagate boundary information quickly into the flow domain, and the velocity components are solved slab–wise, using an extended TDMA (Tri-Diagonal Matrix Algorithm). The SIMPLEST solution procedure is standard for the software.

The pressure was fixed at zero at the outlet for a reference point, and its inlet value was described in terms of an incoming mass flow rate. The velocity component perpendicular to the inlet boundary was set at the mean air–flow velocity, and assumed no diffusive contribution. The standard $k - \epsilon$ turbulence model was employed for the flow, with a modification for the presence of particles to be later described. The variables k and ϵ, the *turbulent kinetic energy* and its *rate of dissipation* respectively, are also solved slab–wise, with an incoming turbulence intensity $I = 0.06$ (6%). The boundary conditions for these were set according to Shaw [Sha92], and again no diffusive contribution through the inlet was considered. Shaw's settings below were in fact used as initial values throughout the domain:

$$k = \frac{3}{2} I^2 V_{inlet}^2 \tag{3.14}$$

$$= 0.0054\, V_{inlet}^2 \tag{3.15}$$

and

$$\epsilon = \frac{0.09k^2}{\nu_T} \tag{3.16}$$

$$= 944\ Jm^{-3}s^{-1} \tag{3.17}$$

with V^2_{inlet} the mean air–flow velocity, and ν_T the effective turbulent viscosity. Wall friction conditions appropriate to the k - ϵ model are set automatically when this turbulence model is switched on.

For the outflow, the appropriate variable sources are sized to whatever values are found in the last cell. This affects no variable values or gradients, ensuring that the last plane of cells is influenced only from upstream.

One thousand sweeps were set as the default number of iterations performed, though most flows converged in less than 300. The default residual criteria used was 1.1% of the initial value, though few variables would reach this. The main convergence criteria used by this author were the stability of values at a reference point, combined with stability of residuals. Visual checks were also performed for each flow for at least the main flow component of velocity. Linear relaxation for the pressure variable was set at 0.3, and false time steps of between 0.01 and 0.001 were used for velocity and turbulence variables, depending on the speed of flow. Convergence difficulties were only experienced at the faster speeds, and these were relatively easy to rectify. Pipe flow is simple in CFD terms. A sample flow is shown in Fig. 3.6 for an R/d ratio of 14, at $27ms^{-1}$. The components of velocity plotted are the local BFC ones, so the axis guide is at times misleading, referring as it does to the absolute Cartesian axes. The BFC components rotate with the mesh, so the x and z components differ by 90^o from one end of the domain to the other.

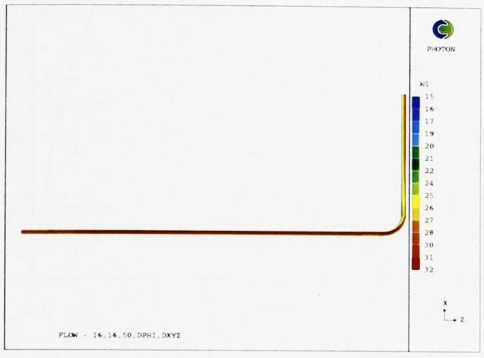
The cross–sections shown in Figure 3.6 are arranged such that the outside of the bend is at the left–hand side of the pictures. The expected recirculation as described in [PPS75] is clearly visible.

Flows once solved are stored and they can be restarted at a later date, typically to track particles.

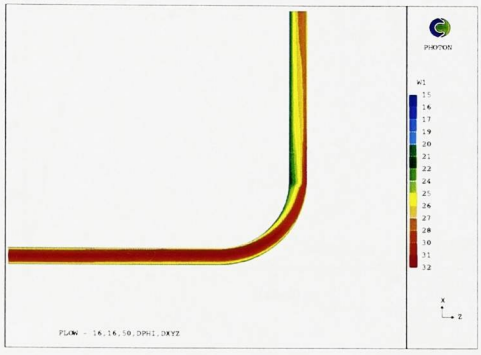
3.3.2 Initial Particle Data

Particle tracking requires initial conditions for the particles; their positions and velocities, as well as physical properties such as size and density. In the current work this data has

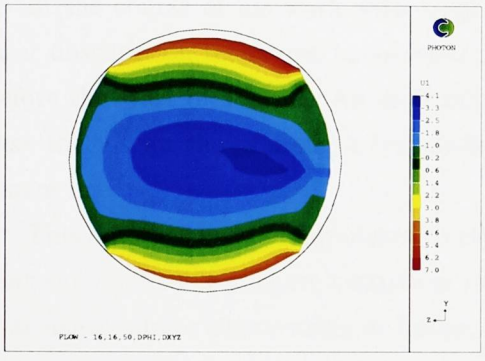
Figure 3.6: Pipe Flow at $27ms^{-1}$



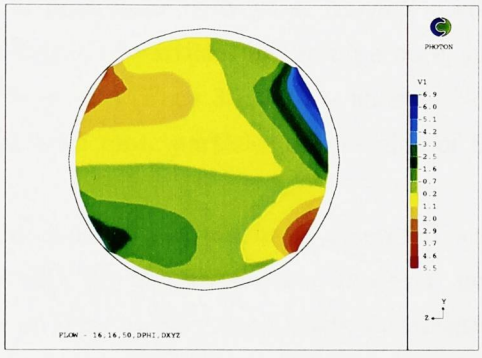
(a) Flow Velocity: overview



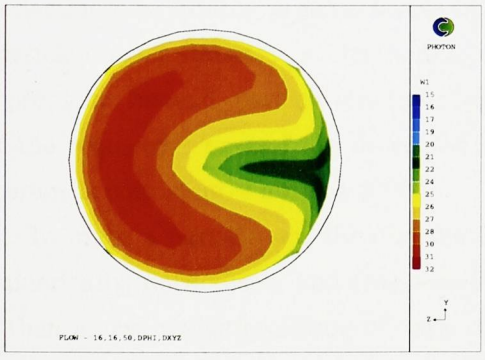
(b) Bend magnified



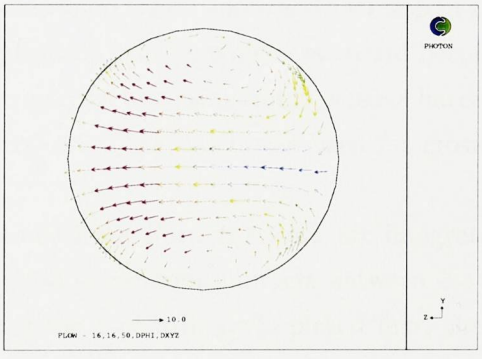
(c) u-component at 87^o



(d) v-component at 87^o



(e) w-component at 87^o



(f) Velocity vectors at 87^o

Figure 3.7: Burnett's x-Scan

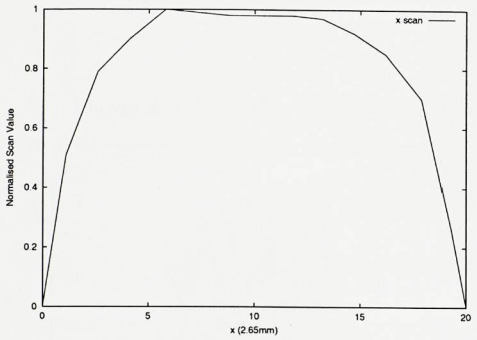
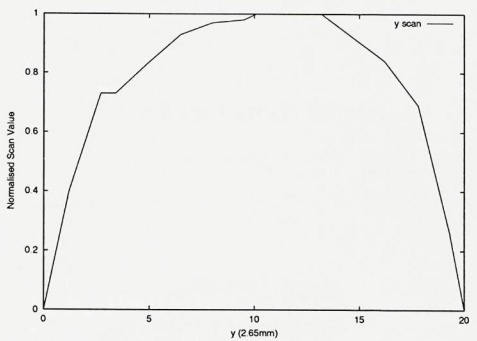


Figure 3.8: Burnett's y-Scan



been generated using data found in the Thesis of *A.J.Burnett* [Bur96], a precursor of the current author at *The Wolfson Centre, University of Greenwich.*

In the course of his work with bend erosion in pneumatic conveyors, *Burnett* used a laser obscuration technique to measure the distribution of particles in the pipe bore, just before the start of a bend. An interpretation of this data (Fig.s 3.7, 3.8) is shown. Note that the y-scan shows a slight bias under gravity, with more particles on one side of the centre line.

This data has been manipulated to allow a random set of particles to be generated, such that the final set will share statistical properties with the measured data. However, note that using a laser obscuration technique supplies no data on the particle size distribution in the flow. The particle set has therefore been generated with a homogeneous particle size. This is also important with regard to the statistical characterisation to be described in Section 3.5. Adding in particle size as a variable would require another dimension in the particle impact statistics, and as well as the complication, machine memory would become a problem. Instead, particle size investigations are carried out using homogeneous batches of the required size, and are described in Section 3.4.2. The batch size used for erosion studies is consistently around 20000.

To make a particle set, the distributions obtained from *Burnett's* work are integrated numerically, normalised, and transposed. Generation of random numbers between 0 and 1 then allows realistic picking of data. However, if the x coordinate is picked first, the y coordinate has to be scaled according to the available height of the pipe at this x coordinate. If the y coordinate is picked first, then similarly the x coordinate must be scaled to the available width. The upshot of this is that the final distribution is not independent of the first coordinate chosen, but exhibits a slight bias. This is visible in Fig. 3.9 as slightly more

Figure 3.9: Pipe Cross–Section

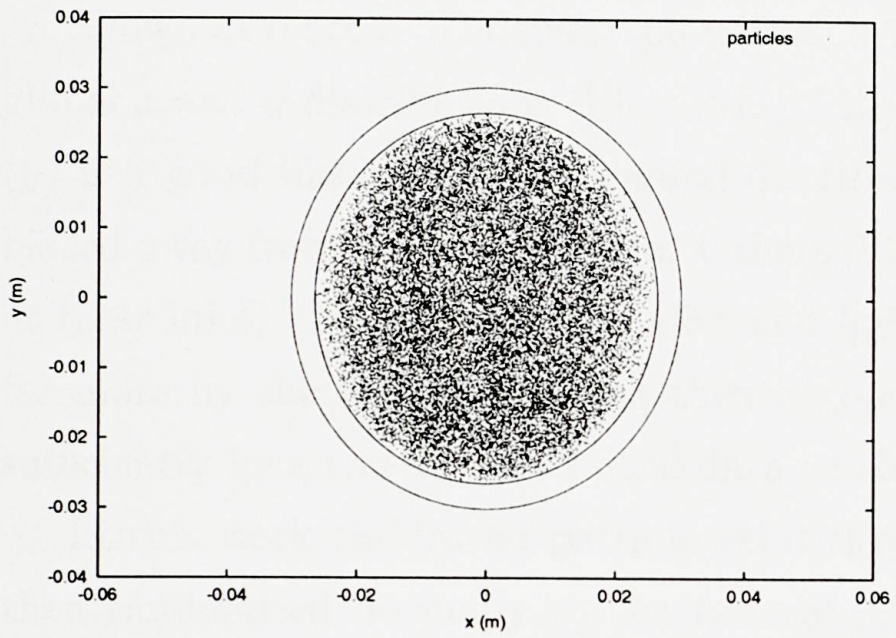


Figure 3.10: Particle Density

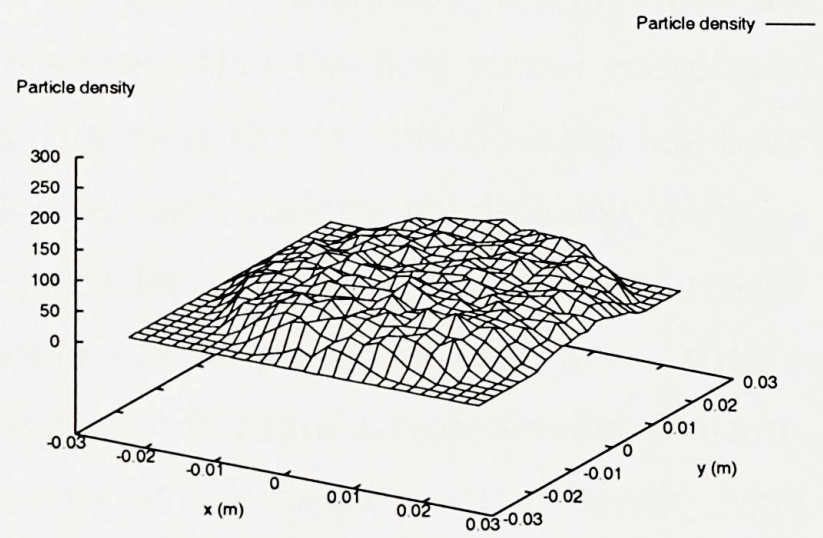


Figure 3.11: X Distribution

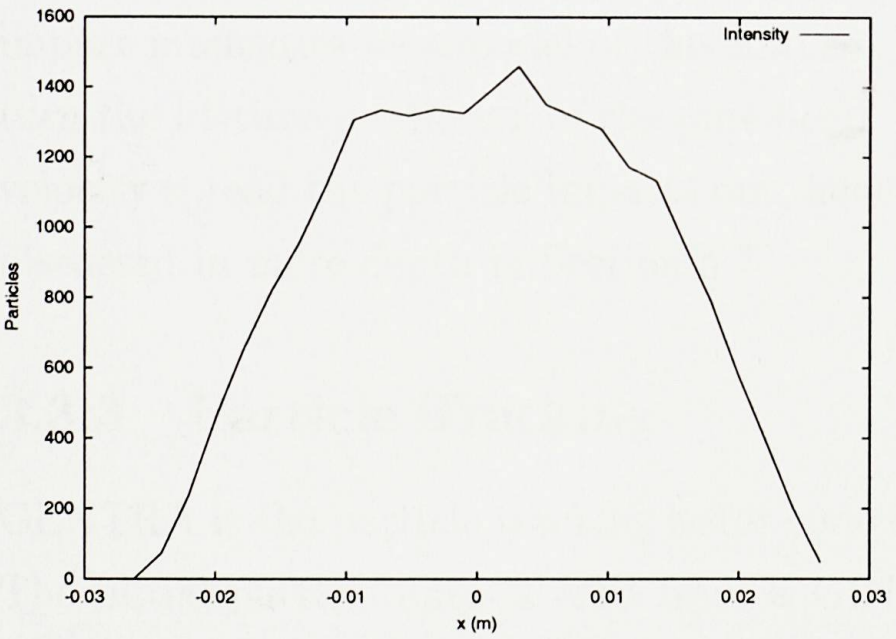
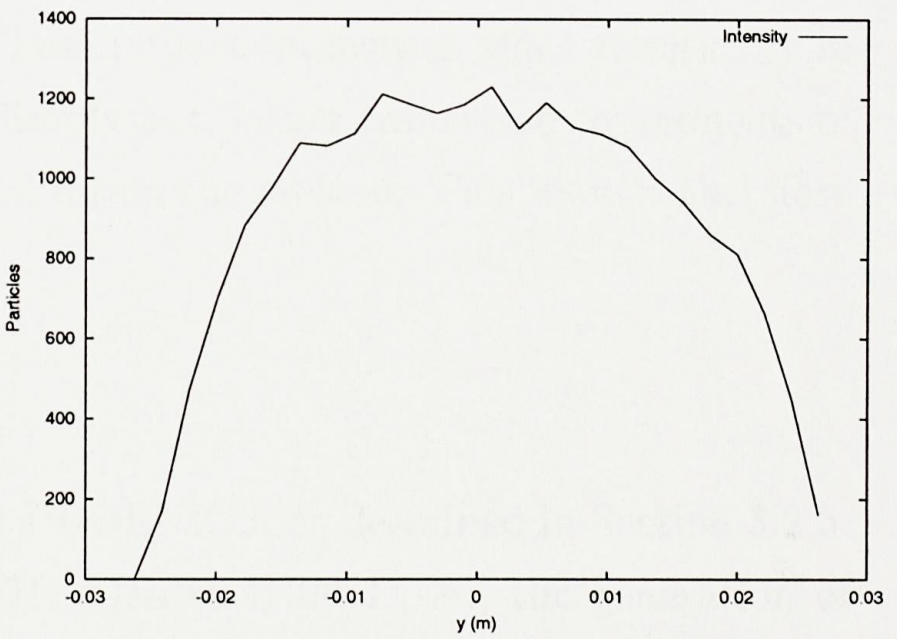


Figure 3.12: Y Distribution



space to the left and right of the particles, as opposed to more space above and below if the x coordinate is picked first. This is no doubt due to the fact that the obscuration data takes in the entire depth of field rather than a single cross-section, but when generating the second of the two particle coordinates, ideally information about only the current plane should be used. The effect does not look major however.

Shown also are a 3D density plot (Fig. 3.10) of the generated particle set, and also the global x and y distributions (Fig.s 3.11, 3.12). Note again that the first picked coordinate (y) is a good likeness of the original distribution, but that the (x) distribution has been biased away from its original form. Others [LM98] have used uniform particle distributions at their inlet, but these were smaller and lighter particles than here, and were influenced far more by the flow. Assuming then a good model was employed in that work, with a sufficiently long run–in, any initial data would eventually fall into a realistic distribution.

In this work the initial particle velocities were set at the mean air-flow velocity, and then randomised normally to the tune of $\pm 2\%$. This value was found in the Thesis of S.R.Woodhead [Woo92] and is an upper bound for the reliability of his measurements, using laser doppler velocimetry. Note that the mean slip velocity he measured was $0m/s$, and around half of his data points suggested that the particles were travelling fractionally faster than the air. His particles were pulverised fuel, which is relevant to this project. In the absence of data for olivine sand, the same values have been used. Similarly, no data has been found for transverse velocity components, so these were initially randomised at $\pm 2\%$ of the main component.

Having experimented though with different values of these transverse velocity components, it became evident that they play possibly *the* major role in shaping the particle impact intensities in a model of this nature. These impact intensities affect drastically in turn the lifetime prediction of the pipe bend. Simply put, larger transverse components of velocity spread the particle impacts out, hence diluting the erosion. This issue is therefore discussed in more depth in Section 5.7.

3.3.3 Particle Tracking

GENTRA is the particle tracking sister code of PHOENICS, as described in Section 3.2.5. The initial particle data is read from a GINFIL (Gentra INlet FILe), the generation of which was described in the previous section. The initial data required for each particle are: x, y, z positions; u, v, w components of velocity; and diameter, density, and mass flow rate along this particular stream–line. The last of these is required because the PSI-cell method

tracks 'packets' rather than individual particles. The diameter and density are constants for each GINFIL, for reasons described in section 3.3.2.

There are numerous switches that control GENTRA, and these are input at the bottom of the *q1* file, which is the same input file used to set up the flow in PHOENICS. GENTRA can be used in conjunction with the facility to restart a flow calculation in PHOENICS. Thus previously calculated flows can be re–used as many times as required, by simply naming in the *q1* the files on which the required flow is stored.

Switches are used to specify: that particles are in an isothermal flow; that the stochastic turbulence model is on; and the name of the output files.

Numerical controls can be adjusted to control: which of the flow iterations are the first and last particle tracking sweep; the frequency of particle tracking sweeps; the maximum time–step; the minimum number of time–steps per cell; the maximum total time–steps, or the maximum time elapsed before tracking is abandoned; linear relaxation of the particle variables; and the restitution value for collisions.

The stochastic turbulence model used is the built–in tool for PHOENICS, and it is an extension of the standard $k - \epsilon$ two-equation model. Indeed it will default to this in the absence of particles. It is described in [GI83], and uses the (randomly generated) local turbulent velocity fluctuation in the flow added on to the mean value obtained by the standard CFD solution. The time of influence of this fluctuation is taken as the minimum of the following two quantities: the life of the local eddy; or the residence time of the particle in this eddy. A simplified version was used in [WPS+00], where the time of influence was assumed always to be the life of the eddy.

For the particular purpose of this work, the GENTRA code was modified such that impact positions were dumped once particles had reached the pipe wall. The modular nature of PHOENICS allows this peripheral type of change to be made, while preserving the security of the proprietor's code in areas considered more commercially sensitive.

Each particle impact provides one line of data in a file, and this information can later be passed to a customised interface for interpretation. This stage of the modelling process is described in Section 3.5.

3.4 Testing the Particle Tracker

Investigations were performed in order to establish how to get the best out of the particle–tracking software.

Table 3.1: Mean Impact angle with S.D. (degrees)

Sweep Frequency	Mean Impact Angle	S.Deviation
1	15.24	8.15
2	27.90	13.04
3	26.62	10.01
5	32.29	11.30
7	24.41	7.89
10	24.83	9.67
20	25.13	11.66
30	26.18	11.27
40	24.76	11.03

3.4.1 Numerical Controls

Here, some of the main controls available to the GENTRA user are discussed, and parameters are investigated before the modelling proper is begun.

For the initial tests, particles with strong coupling with the air were desired, so as to fully explore the two-way coupling numerical behaviour. This point is raised because a lot of the other work on this project used olivine sand particles, (specific gravity 3.28, diameter 320 μm) which exhibit very little coupling with air as will be seen. Used here are pulverised fuel particles (coal crushed up to promote efficient combustion) of diameter 30 μm and specific gravity of 1.3. This last value is central in the accepted density range [PG84] of the wide variety of coals used in power generation, covering bituminous and lignite varieties (specific gravity 1.1–1.5), but neglecting the slightly denser and more expensive anthracite.

Sweep Ratio

The first series of tests used the p.f. (pulverised fuel) whilst varying the number of sweeps of the fluid solver used in–between each GENTRA pass. Recall that the sources due to the presence of particles are applied and held constant between each GENTRA pass, such that the flow can adjust to them over several sweeps. For this study, five GENTRA iterations were consistently used. Results are shown in Table 3.1 and Figures 3.13, 3.14, and they suggest that allowing more iterations between each produces a more stable result. It can be recommended that no fewer than ten fluid iterations be performed between each GENTRA pass. The samples in this section were batches of twenty particles. For a visual account of what these numbers mean, see Figures 3.15 to 3.20.

Figure 3.13: Mean Impact Angle

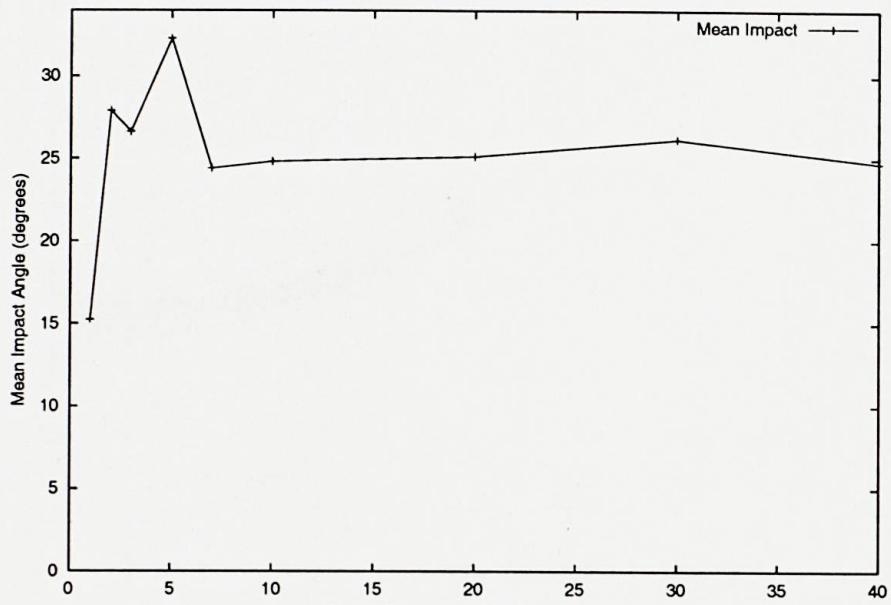


Figure 3.14: St. Dev. of Impact Angle

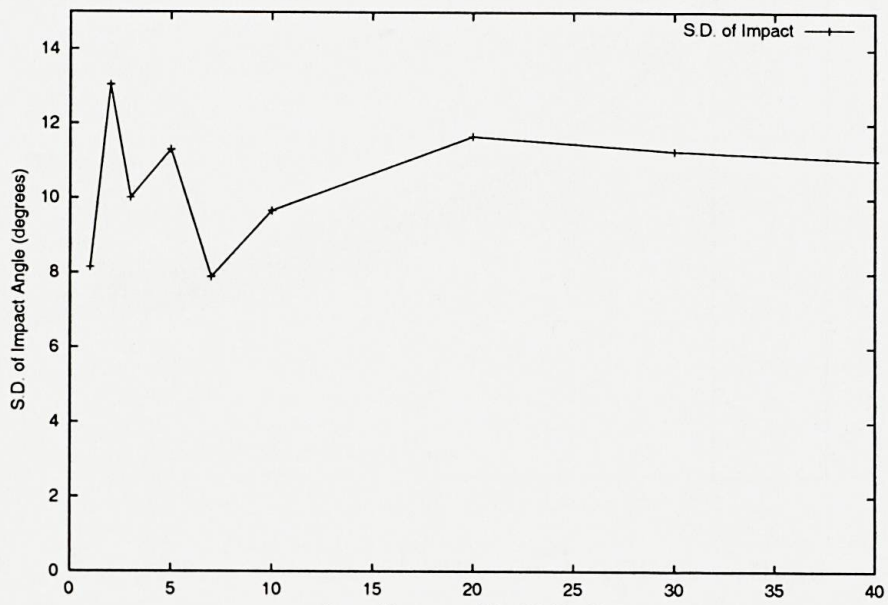


Figure 3.15: PF: one-way coupling

Figure 3.16: PF: sweep freq=5

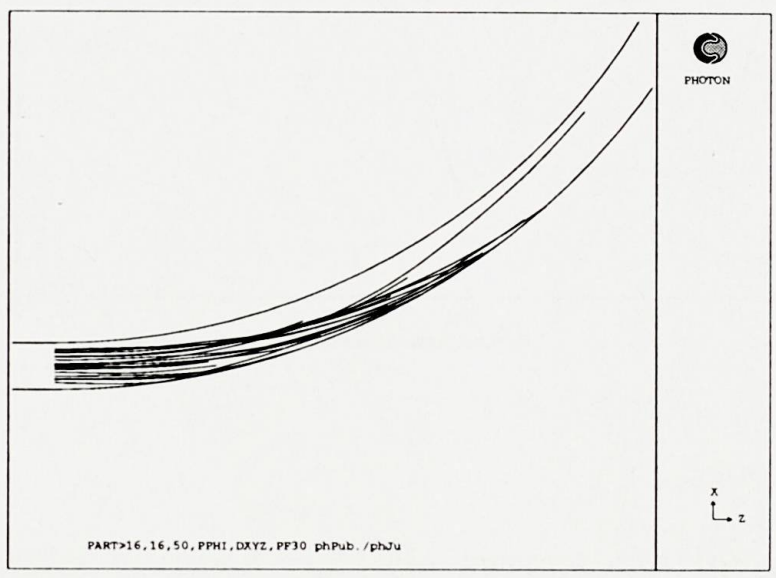


Figure 3.17: PF: sweep freq=10

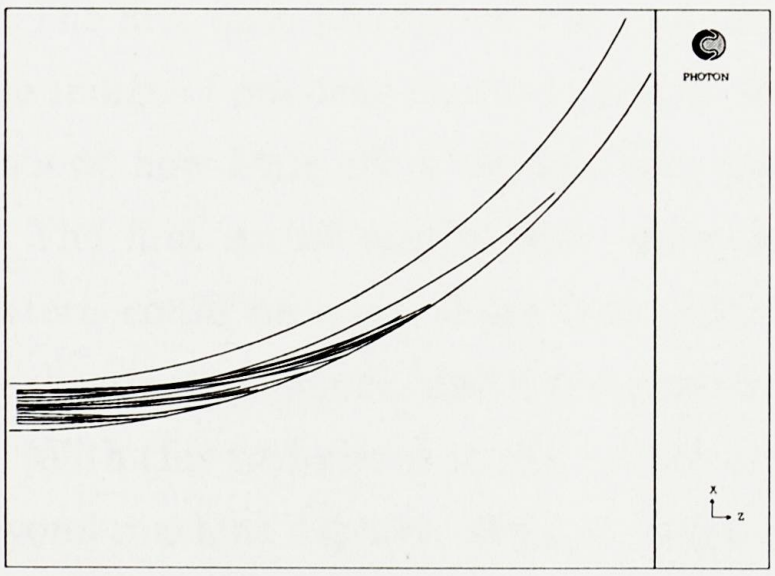


Figure 3.18: PF: sweep freq=20

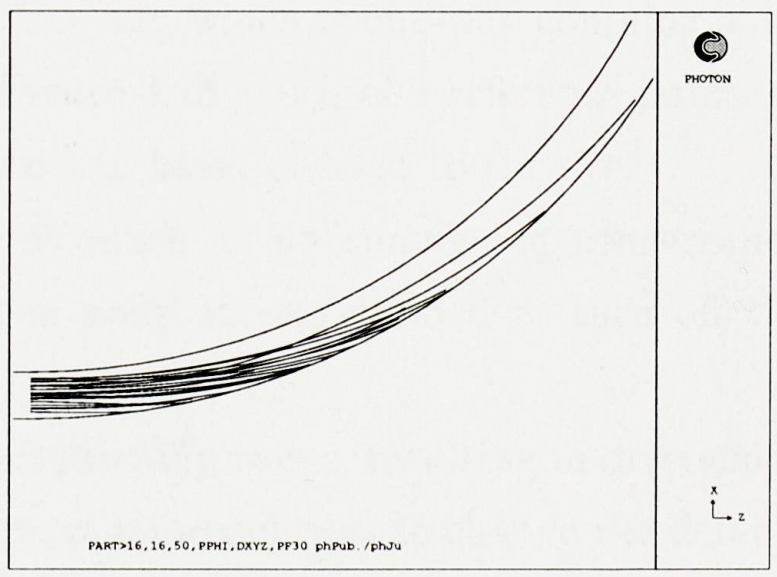


Figure 3.19: PF: sweep freq=30

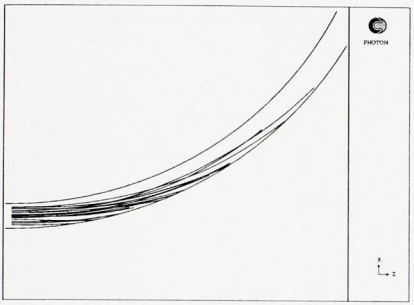


Figure 3.20: PF: sweep freq=40

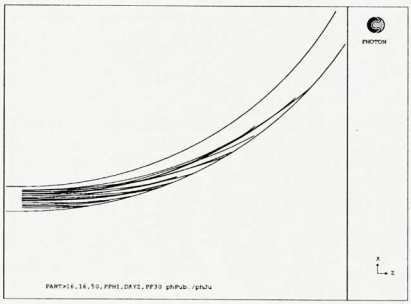


Figure 3.21: Mean Impact Angle

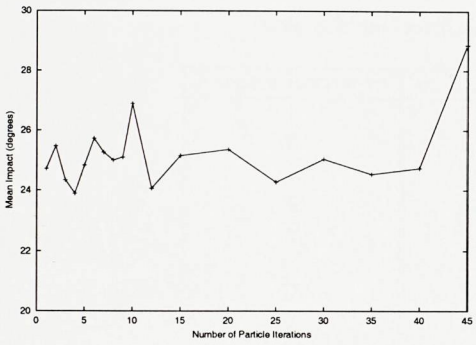
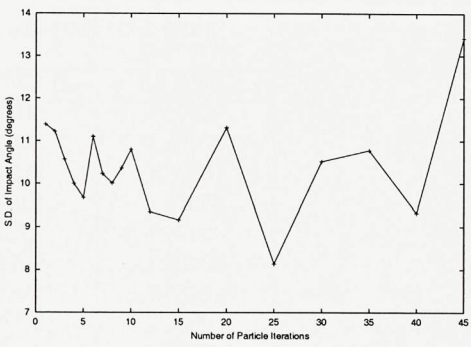


Figure 3.22: St. Dev. of Impact Angle



Iterations Required

With the previous assumption of 10 fluid iterations between particle sweeps, an investigation of the number of particle iterations to be used was undertaken. Obviously the recommended values in this section may vary with physical factors such as particle size, but here the modest aim is to illustrate the behaviour for p.f.

The first data points came from a single–iteration run, which is one-way coupling only. The image of one-way coupled particle tracks in Figure 3.15 is a useful reference point, as it shows how little effect the two-way coupling seems to have, at least to the eye.

The first set of results were inconclusive, in as much as no convincing convergence pattern could be seen, (Fig.s 3.21, 3.22) so at this point it was decided to turn off the Stochastic Turbulence Model temporarily.

With this turbulence model off, results were considerably worse, resulting in divergence beyond machine capacity after 15 iterations. The next approach was to change the default setting of the GENTRA linear relaxation factor from 0.7 to 0.3, with results shown in Fig.s

Figure 3.23: Mean Impact Angle

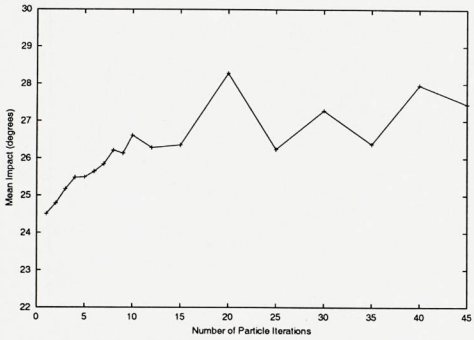


Figure 3.24: y-Coord.s of Particle 1

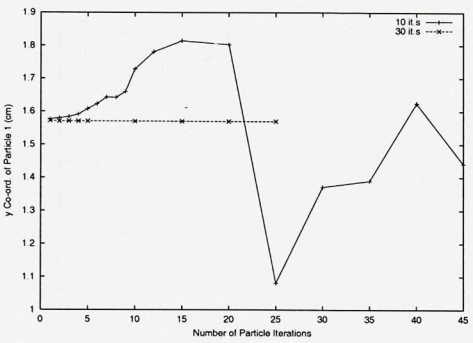


Table 3.2: y-Coordinates of Particle 1 (cm)

Particle Iterations	y: 10 Fluid It.s	y: 30 fluid It.s
1	1.57643	1.57239
2	1.58008	1.57018
3	1.58338	1.57037
4	1.59094	1.57032
5	1.60778	1.57023
10	1.72791	1.56999
15	1.81387	1.56992
20	1.80256	1.56991
25	1.08097	1.56991
30	1.37162	-

3.23 and 3.24. This factor influences the size of the sources applied to the fluid equations, but still no convergence is seen. Revising the initial assumption of 10 fluid iterations per particle sweep however, seemed to solve all problems.

Also shown in Figure 3.24 is data obtained using 30 fluid iterations per particle sweep, and the convergence is now readily seen. Plotting these on the same scale, the fine detail is not visible, so Table 3.2 is also supplied.

With the addition of the Stochastic Turbulence model, this also now achieves convergence, (Fig.s 3.25 and 3.26) but to a different y-point due to the nature of the model. Here, the mean standard deviation of impact angle becomes 11.3^o compared to a value of 8.75 when the model is not active. The particle dispersion has therefore increased, as expected, by 29%. Convergence of the stochastic case is slower it seems.

Concluding, for $30\mu m$ p.f. the recommended number of particle iterations to use is

Figure 3.25: Mean Impact Angle

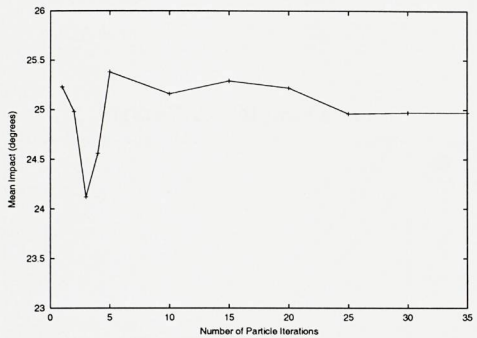


Figure 3.26: y-Coord. of Particle 1

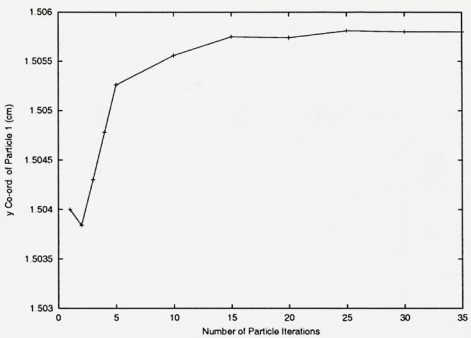


Figure 3.27: 30 μm P.F

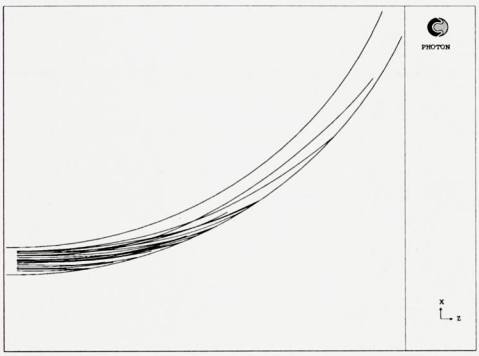
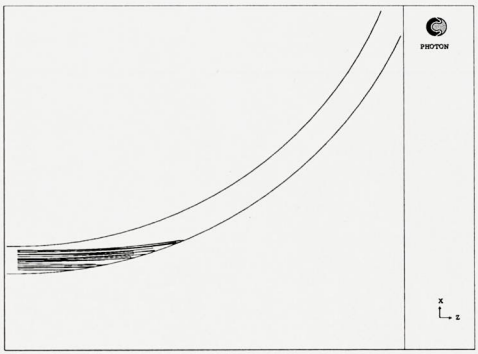


Figure 3.28: 75 μm P.F



10, with 30 fluid iterations between each of these. This should supply good numerical convergence, but for practical purposes, two or three iterations should suffice.

3.4.2 Particle Size Investigation

The results in this section are straightforward. Greater particle inertia results in a corresponding reduction of the air's ability to influence particle motion. The densities of the two particles used were 3280 $kg\,m^{-3}$ for Olivine Sand, and 1300 $kg\,m^{-3}$ for the Pulverised Fuel. The factor of 2.5 difference between these makes a visible difference, comparing directly Fig.s 3.27 and 3.30, or Fig.s 3.28 and 3.31.

The particle sizes where air coupling starts to take serious effect can be roughly stated as 30 μm for p.f, and probably 15 μm for Olivine. Particles at these sizes are following the streamlines to a large though not total extent. The sizes quoted are particle diameters.

Figure 3.29: 20 μm Olivine

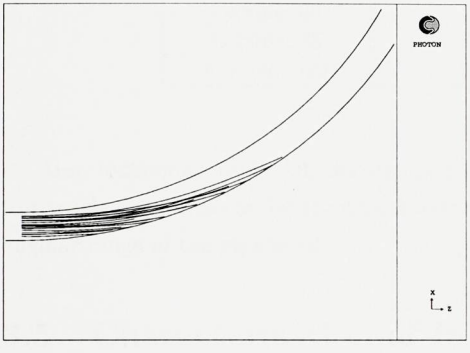


Figure 3.30: 30 μm Olivine

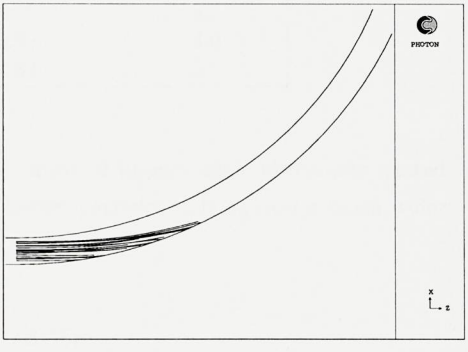


Figure 3.31: 75 μm Olivine

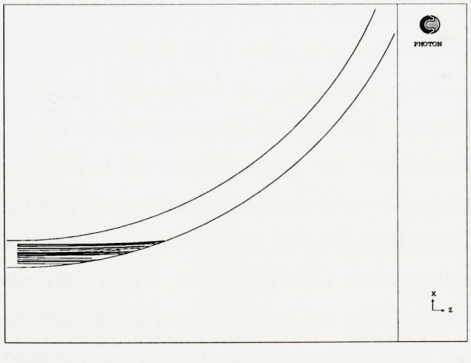


Figure 3.32: 320 μm Olivine

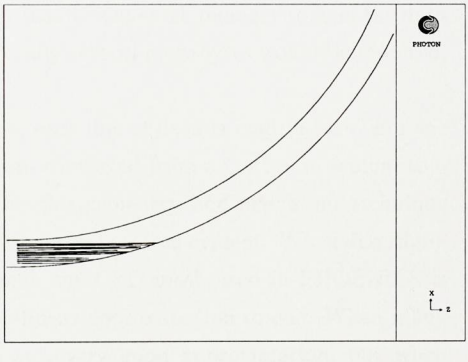


Table 3.3: Mean Impact Angle (degrees)

Particle size (μm)	Mean Impact Angle	St. Deviation
P.F, 30	25.16	12.1
P.F, 75	16.48	4.6
Olivine, 20	21.08	7.5
Olivine, 30	18.16	5.5
Olivine, 75	15.33	4.0
Olivine, 320	14.83	3.7

Also, looking at Table 3.3, the standard deviation of impact angle also seems related to particle size. This can be attributed to the lighter particles striking over a much wider angular range of the pipe bend.

3.5 Characterisation of Impacts

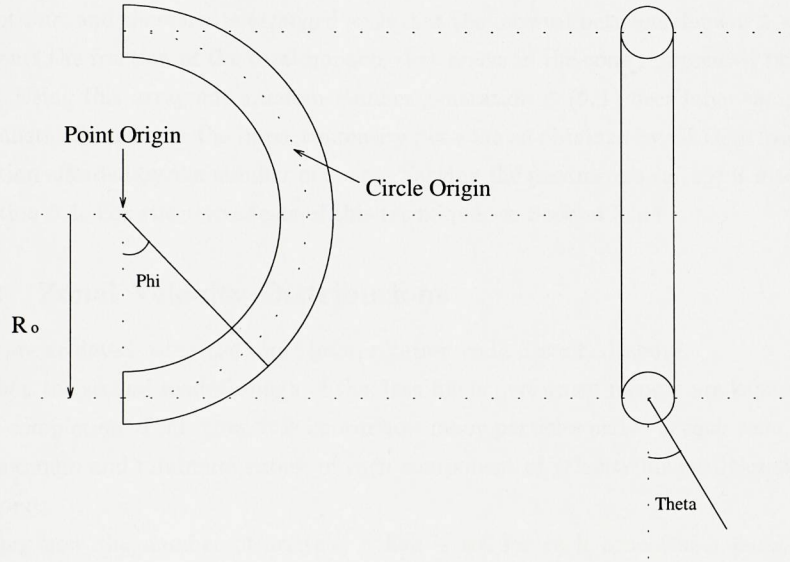
Given particle impact data that has been obtained at some considerable computational expense, accurate methods of characterizing this data must now be found.

3.5.1 Impact Intensity

To characterize the particle impacts the file containing this data is read three times by an interpretation code. The first two reads allow the impact intensity to be quantified, and the third read–through is required for velocity characterisation as described below. These multiple reads are necessary as there is currently insufficient stack memory to load the data into arrays inside the interpreting software; typically several megabytes would be required, plus that required for the program's execution.

On the first read–through as with the others, each line of data is read in turn. For the position of the particle strike, the coordinates are converted from a Cartesian system to a toroidal one. This is the most natural system for this geometry, and makes our technique possible, though note that an earlier approximation now becomes evident. When describing the pipe wall in a body–fitted coordinate system, the CFD mesh used in PHOENICS is actually not a perfect toroid but a piecewise bi–linear approximation to one. When many cells are used (here, 90 for a 90^o bend) this is a very good approximation, but when converting the Cartesian impact points into toroidal coordinates, the radial value will only be exactly equal to the radius of the inner wall of the pipe, if the particle in question has

Figure 3.33: The Toroid and its Coordinates



struck a node of interpolation of the BFC mesh. In short, the radial value is therefore not calculated, but at this stage *set equal* to the bend radius *rmin.*

Returning to the data file, the magnitude of the θ and ϕ coordinates (Fig. 3.33) of each particle is compared to those for the previous particle, and the maximum and minimum values observed are therefore recorded. This information is then used to divide up the domain of impact into zones, by dividing the θ and ϕ extent by specified integer parameters. Here, the actual maximum ϕ value is often substituted by a smaller one. One stray particle can result in the modelling of an extra 10^o of bend, which is plainly not going to contain interesting regions of erosion, and is computationally wasteful. The cut–off value used can result in perhaps 5% of particle impacts being 'ignored' by the characterisation process, so to compensate, the mass flow rate calculated from the number of particles generated in *Erode* is multiplied by a factor. This prevents any bias towards the impact intensity of the particles that are used for statistical generation.

When the extent of each zone in this two–dimensional array is established, the second read–through is performed, and records are kept such that it is known how many particles struck in each zone.

Each of these zonal counts are then scaled by the total number of impacts, and stored

cumulatively in a $(1 + zi * zj)$ one–dimensional array, where (zi, zj) are the number of zones in the (θ, ϕ) coordinates respectively. Said array has first element *zero* and last element *one*, and therefore is arranged such that the interval between element k and $k+1$ represents the fraction of the total impacts that arrive in the zone represented by element $k+1$. Using this array and random number generation $\in (0, 1)$ over large samples, this can realistically recreate the impact intensity per zone as obtained by CFD, at least to the resolution afforded by the number of zones. Varying the parameters (zi, zj) is investigated in Section 5.4. For statistical tests of this technique see Section 3.5.4.

3.5.2 Zonal Velocity Distributions

These are achieved using the same interpretation code described above.

When the second read–through of the data file is performed records are kept such that on the completion of this pass, it is known how many particles arrive in each zone, and also the maximum and minimum extent of each component of velocity for particles striking in each zone.

Using now the number of arrivals, a flag is set for each zone which represents the decision made as to how to characterise the information. This is also made available to the erosion code such that it will know which method to use to interpret the rest of the data in a given file, and there will be three files for each heavily impacted zone representing *(u,v,w)* components of velocity.

The first flag indicates that there were no impacts in a zone. The second indicates that there were less than 160, and hence a simple mean is taken of each velocity component. The third flag means that in excess of 160 particles struck, and in this case a distribution is created using twenty divisions of the range of the variable in question. The number 160 is chosen so that distributions will be created only in areas of intense activity.

The third pass of the data is used to fill each of the intervals in each distribution made. The dumped distributions now have to be integrated into probability distribution functions, and for this paths have to be set up to refer to each file containing a distribution, and also to each file onto which will be written the relevant integral.

3.5.3 Zonal Velocity PDFs

To obtain a probability distribution function for each component of velocity in each zone where a distribution was made, a separate piece of software is called from within the main interpretation code. This starts with generic character strings representing the path to the

files to be read, and also those to be written to. Loops are organised such that the strings account for (u, v, w) components of velocity in each zone. If simple data is encountered, it is simply copied into the p.d.f. file, but if the relevant flag is found, a distribution has been made and must be integrated. In this case the read and write paths are both concatenated onto the name of a custom integration code. The resulting string is sent out as a system call, and the program thus called reads the contents of the first file as a piecewise linear function.

The integration code is a simple finite–element method that solves the two point boundary problem

$$-u''(x) = f(x) \qquad \text{for } x \in [a, b] \tag{3.18}$$

$$\text{with} \quad u(a) = 0 \tag{3.19}$$

$$\text{and} \quad u'(b) = g(x) \tag{3.20}$$

Given the desire to integrate our distribution only once, ie. starting from $u'(x)$ and needing $u(x)$, first differentiate it to obtain a piecewise constant function to be called $u''(x)$. Putting this equal to $-f(x)$ in Equation 3.18 allows the solution of a problem that will supply the integral of $u'(x)$.

Also, integrating a distribution which tends to zero value outside a finite range, the derivative at the extremes tends to zero, and for the boundary values set the Dirichlet condition $u(a) = 0$, and the Neumann condition $u'(b) = 0$.

Around 80 nodes are used to pick up the curve of the function, and increasing this further gives no discernable change, whilst increasing the amount of storage required when these p.d.f.s are read into the erosion code.

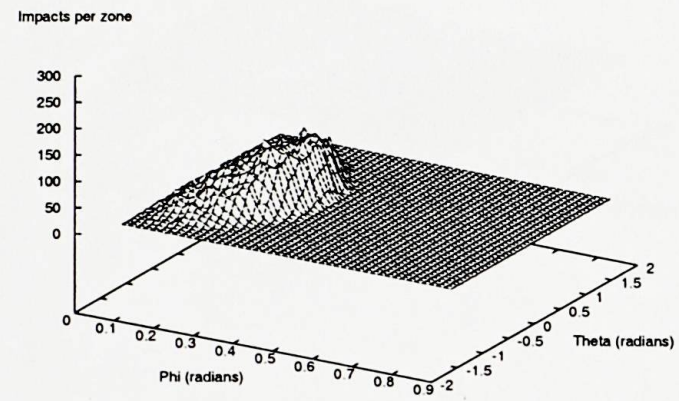
The integrated function is dumped in piecewise linear form on a file, the name of which comes from the second argument to the call. The whole process of interpretation of the impact data file is fully automated such that the above work takes around a minute on a *sparc-20*.

3.5.4 Statistical Tests

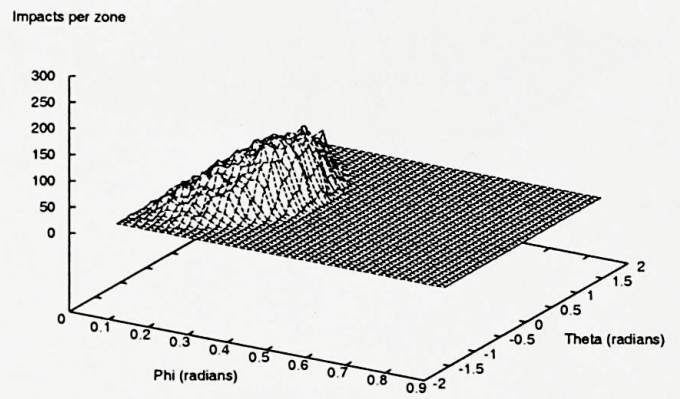
To confirm that the statistical techniques work adequately, some representations of impact intensity are shown in Figure 3.34. The sample sizes were matched exactly here, so the diagrams are directly comparable. Experience suggests that 20000 particles for the sample size will produce a smooth match with the original data.

The generated data was obtained from the *Erode* code. A modification was made so

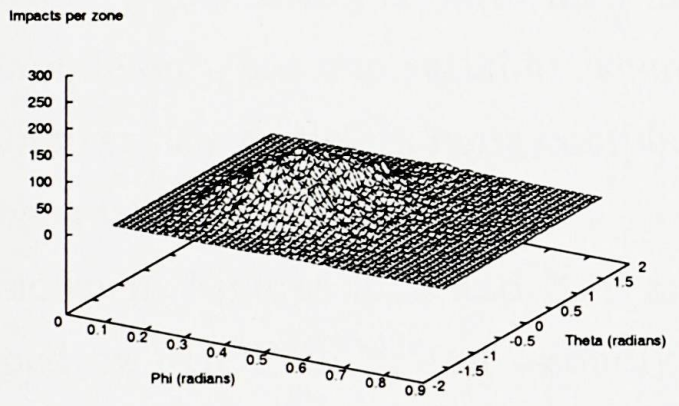
Figure 3.34: Impact Intensities: Original and Randomly Generated



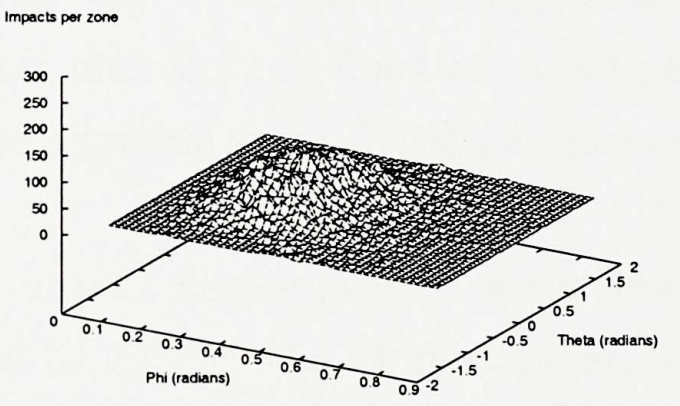
(a) 1st: Original



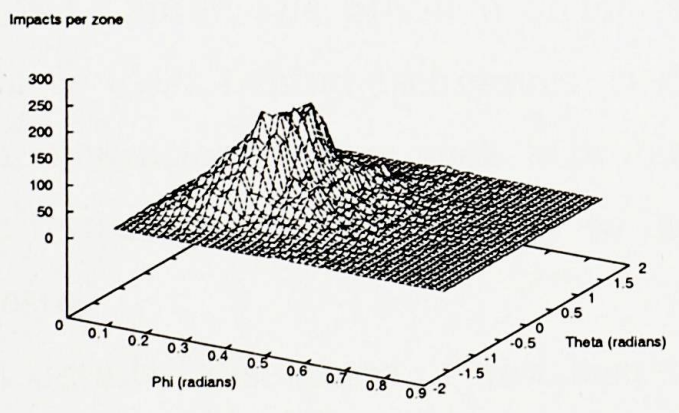
(b) 1st: Generated



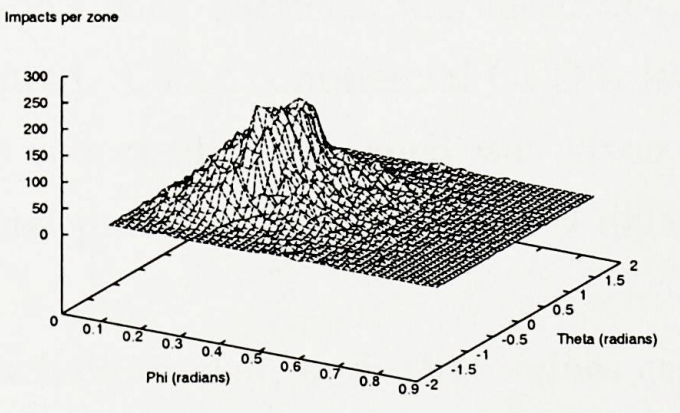
(c) 2nd: Original



(d) 2nd: Generated



(e) Original Impacts Combined



(f) Generated Impacts Combined

Figure 3.35: Original Impact Angles

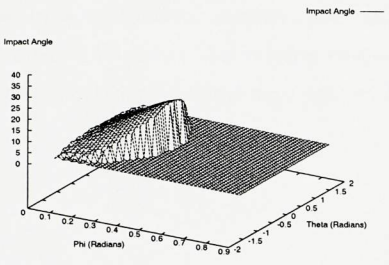
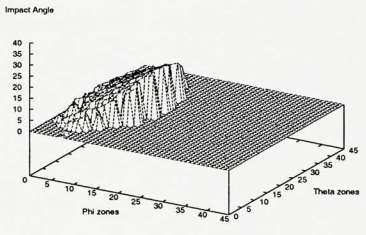


Figure 3.36: Generated Imp. Angles



that the required number of particles was generated using the statistical data made from the original impact data. These were then dumped as PHOENICS dumps the impact data, so that they could be passed through the filter, and pictures etc. could be obtained. The second impacts shown are discussed in Chapter 6.

Velocity generation is performed in a very similar way. It is a simpler matter as here the sample only has one variable, whereas the impact intensity is a function of both θ and ϕ. However, the coding is more complicated because a separate distribution or mean value has been used for each zone.

Shown in Figures 3.35 and 3.36 are the mean impact angles in each zone; firstly as dumped by PHOENICS, and secondly as generated by *Erode*. The likeness confirms that velocities, surfaces and normals are all being generated correctly in *Erode*.

3.6 Chapter Summary

In this Chapter, the problem of modelling fluid flows with particles was described, and the basic ideas behind techniques used were recounted. Using commercial CFD software, a demonstration of how such a problem could be successfully modelled was given, and some results were shown. A way to initialise tracked particles using laboratory data was suggested.

A detailed discussion of how best to use the particle tracker, and what values certain parameters should be set to was held. A conclusion, at least for particles similar to 30 μm diameter p.f., was that there should be 30 fluid iterations between each particle tracking sweep under the conditions used here. The number of particle tracking sweeps could be limited to three, though ten were required for exact convergence. Particle size had a

predictable effect with regard to the amount of air–solids coupling experienced.

The manner in which CFD particle impact results could be used, manipulated and passed to a customised erosion code was also detailed, and some statistical tests were performed to illustrate the validity of the *Monte Carlo* approach used here. Statistically generated particles matched very well with the original dataset.

Chapter 4

Mathematical Aspects of Erosion

4.1 Overview: Structure of the *Erode* Model

This chapter concentrates on the *Erode* code itself, but an indication of where this custom software might fit into the overall modelling mechanism is given in the dataflow chart in Figure 4.1. The main pieces of software are the CFD code and *Erode*, and these communicate via data dumped in files, which are manipulated into a form ready for the other by interpretation software. It is intended to exploit the cyclic nature of this relationship in order to attempt to model series of particle impacts around the bend.

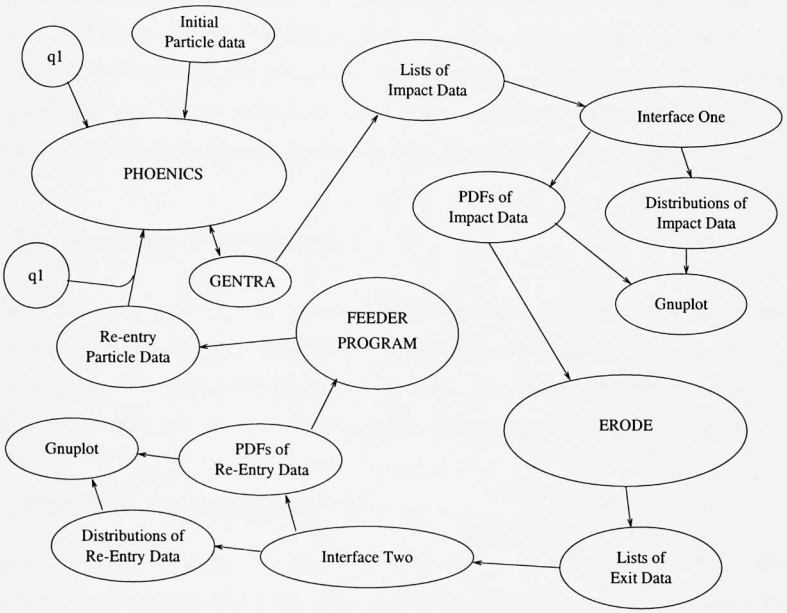
The computational techniques used to model erosion will now be described.

This mathematical model focusses on the section of pipe wall which suffers the most significant amounts of erosion; typically the first 30^o – 50^o of the bend, but with the capability to model up to and beyond 90^o, which can be necessary for very short radius bends. The modelled substrate is bombarded with virtual particles in three–dimensional geometry, and mass is removed according to empirically derived erosion algorithms. The particles are generated using random numbers, and weighted statistically through probability distribution functions. These are made from CFD particle impact data by an interpretation code, as described in the previous chapter.

4.2 Representation of Pipe Wall

The most natural system for modelling the pipe wall is toroidal geometry, and this is what is used. The extent of the impacted domain has already been established using CFD and particle tracking, and this is read into the erosion code from an updated *ErodeGeometry* file, which is dumped by the filter. The domain of interest is now divided into many small

Figure 4.1: Overview of Suggested Modelling Structure



tiles using three integer parameters. In toroidal geometry this is simple, as the extent of the domain in each dimension is simply divided by the relevant integer. Because of the ordered nature of this tile–grid no tile corner positions need to be stored in memory as they would have to be in body-fitted coordinate systems.

A three-dimensional array matching the number of tiles present is initialised, with each element being a data structure. This data structure contains a floating point number that is initialised with the mass that would be contained in the tile's volume, and note that this increases with its radial position. In addition a short integer keeps count of the number of impacts that the tile experiences, for statistical purposes. A one-to-one relationship exists between the abstract tiles, and the array.

Material properties such as density of pipe wall material are read in from a file. For calculations to establish the volume of a torus and hence of a single tile, see Appendix A.

It is certainly the case that the memory intensive technique of storage described is far from optimal, in that only perhaps 7% of tiles modelled will be removed by the time of burst. To introduce the complication of more efficient memory use is currently deemed unnecessary for two reasons: a *Stack* size of six megabytes is not prohibitively large at

this time of rapid expansion of computing power; and secondly, the tile array behaves in a truly three-dimensional manner, in that it cannot be assumed that all tiles removed due to mass loss will be on the 'top' of their respective piles. Tiles can be struck on their sides, and bottom, as well as their top, and hence cavities and overhangs can theoretically form, though given the statistical techniques used, this is unlikely to happen on a large scale. But any storage technique economies would have to allow for this complication.

4.3 Incoming Particles

The bulk of the data provided by the interpretation code is now read into *Erode*. Recall that this describes the impact intensity on the domain of interest, and also characteristic velocities varying with position within that domain. The technique requires the breakdown of the domain surface into smaller zones, parametrised by two integers, (zi, zj).

4.3.1 Particle Generation

The impact intensity is stored in an array with $zi.zj + 1$ elements, the first of which is assigned *zero*, and the last of which *one*. The other elements contain numbers increasing between zero and one, and the difference between the first element and the second reflects, as a fraction of *one*, the proportion of impacts that occurred in the first zone. In this fashion, the intervals between each contiguous pair of array elements represent relative impact intensities, and taking a large sample of random numbers $\in (0, 1)$ allows the realistic reproduction of the recorded intensities, at least to the resolution of the zones.

When the zone of impact is thus established using a random number, the particle's position within the extent of that zone is randomly created using the two coordinates θ and ϕ. Hence the number of zones used will influence the amount of data smear, and this is examined in Section 5.4.

For the generation of a velocity for this particle, data particular to its host zone is employed. This will be in one of two forms, according to the sample size of impacts that occurred in this zone at the particle tracking stage. For less than 160 particles, a mean value is used for each component of velocity. This is not considered a very accurate technique, but it is unlikely that this would be a region critical to the pipe burst. For more than 160 particles, distributions are made for each component, then integrated and scaled between $(0, 1)$. Using the inverse of this single–valued function (probability distribution function), random numbers $\in (0, 1)$ can be used to generate velocity components that over a large

sample will accurately reflect the initial distributions.

These velocity generation techniques do not take account of linkage between components, in that if for a particular particle a certain component is larger than average, one or more of the others may tend to be smaller than their average value. It is considered that the standard deviations measured over each zone are sufficiently small that generating the three components separately is still a good representation of reality, and that the magnitude of the vectors produced is not unduly biased.

4.3.2 Projection

Now that both the geometry of the problem and realistic particles can be generated, the particle's velocity vector must be followed until the point of impact with the pipe wall is found. This entails a computationally intensive process whereby the equation of a line is, if necessary, repeatedly solved for its point of impact with planes. The line is that of the velocity vector, and the planes used represent the faces of the tiles. Starting from the inside of the pipe wall, if it has been removed, the vector is projected through the host tile until the point of exit from this tile is found. At this point attention is switched to the neighbouring tile encountered, and if this neighbour is still present the impact point has been found. If not, the projection is continued until impact does occur. This projection process is the undoubted engine inside the erosion model, and novel techniques are used.

4.4 Vectors in a Toroidal Coordinate System

Rather than using Cartesian vectors in this model, a local approximation to a straight line is used. The technique and the motivation are outlined below.

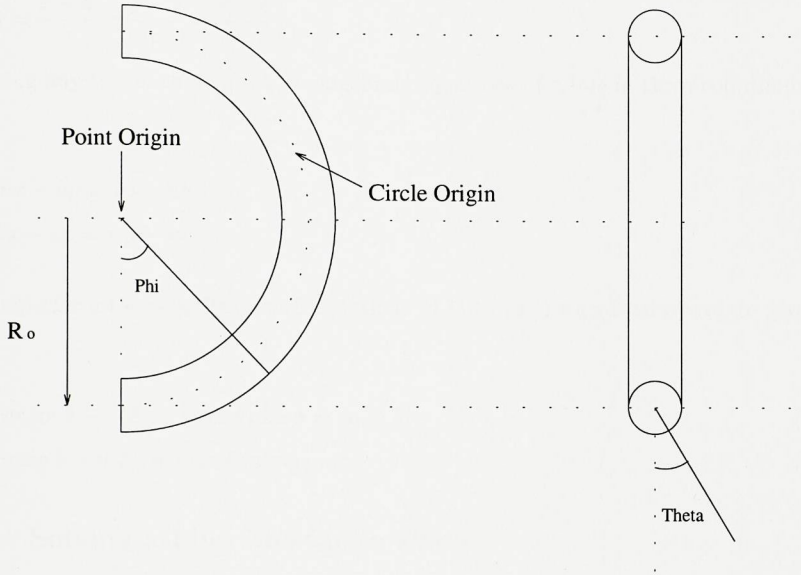
4.4.1 Equations of a Line

In the version of toroidal coordinates used, there are two origins: a point origin at the centre of the torus from which stems the angle ϕ, (Fig. 4.2) and a line or circle origin of radius R_o, at every point of which there is a unique (r, θ) plane as in plane polar coordinates.

It is possible to describe a general straight line in this coordinate system, but the equations take a form which defy easy solution. Given the transformations

$$x = r \sin \theta \tag{4.1}$$

Figure 4.2: Coordinate System



$$y = (R_o + r\cos\theta)\sin\phi \tag{4.2}$$

$$z = (R_o + r\cos\theta)\cos\phi \tag{4.3}$$

substitute these into the Cartesian equations of a line to obtain the equations of a line in toroidal geometry. Starting in Cartesian geometry with a point and vector (impact point and velocity) allows the construction of the parametrised equation of a line:

$$(x, y, z) = (a, b, c) + \lambda(u, v, w) \tag{4.4}$$

From this it is ascertained that

$$x = a + \lambda u \tag{4.5}$$

$$y = b + \lambda v \tag{4.6}$$

$$z = c + \lambda w \tag{4.7}$$

and from this that

$$\lambda = \frac{x-a}{u} = \frac{y-b}{v} = \frac{z-c}{w} \tag{4.8}$$

Taking any two of these gives the algebraic equations of a line in three coordinates.

$$vx - uy = va - ub \tag{4.9}$$

$$wx - uz = wa - uc \tag{4.10}$$

Now perform the substitution of Equations (4.1) to (4.3) suggested above, to give

$$vr\sin\theta - u(R_o + r\cos\theta)\sin\phi = va - ub \tag{4.11}$$

$$wr\sin\theta - u(R_o + r\cos\theta)\cos\phi = wa - uc \tag{4.12}$$

4.4.2 Solving a Line Meeting a Plane

The above (4.11,4.12) are the equations of a straight line in toroidal geometry, and for current purposes it is desired to solve a line meeting a plane as the vector passes through the array of tiles. Fortunately, the equations of the planes will always be easily obtained in this toroidal situation, because the tile faces are all defined by either r, θ, or ϕ equals a constant. Note that the term *plane* is used in the sense of a two-dimensional surface, which will not necessarily be flat in the Euclidian sense.

So to investigate this method, let an attempt be made to find where a vector strikes the top of a particular tile, i.e. where the line passes through $r = r_c$. Putting this value in the equations, look for the two unknowns θ and ϕ. After some manipulation it is seen that

$$\sin\phi = \frac{\alpha\sin\theta - \gamma}{(R_o + r_c\cos\theta)} \tag{4.13}$$

$$\cos\phi = \frac{\beta\sin\theta - \delta}{(R_o + r_c\cos\theta)} \tag{4.14}$$

where

$$\alpha = \frac{vr_c}{u} \tag{4.15}$$

$$\beta = \frac{wr_c}{u} \tag{4.16}$$

$$\gamma = \frac{va - ub}{u} \tag{4.17}$$

$$\delta = \frac{wa - uc}{u} \tag{4.18}$$

are all constant and known. Now to eliminate ϕ, use Pythagoras' Theorem on the unit circle, namely

$$\cos^2 x + \sin^2 x = 1 \tag{4.19}$$

giving now

$$(\alpha \sin\theta - \gamma)^2 + (\beta \sin\theta - \delta)^2 = (R_o + r_c \cos\theta)^2 \tag{4.20}$$

To solve this, multiply out, and substitute $(1 - \sin^2\theta)$ for the $\cos^2\theta$ term on the *RHS*. However, there will also be a $(\cos\theta)$ term present, and to perform the same substitution for this it would have to be isolated and the whole equation squared again. This would generate a quartic equation in $\sin\theta$.

The solution of this quartic, if obtainable, would result in four different values of $\sin\theta$, which in turn would generate eight cosines of θ. Each of these eight solutions would then have to be substituted back into Equations 4.11 and 4.12, to obtain non-unique values for ϕ, and then the correct solution has to be found from all of those on offer by testing against the limits of the tile. As shown in Figure 4.3 for example, it is required that

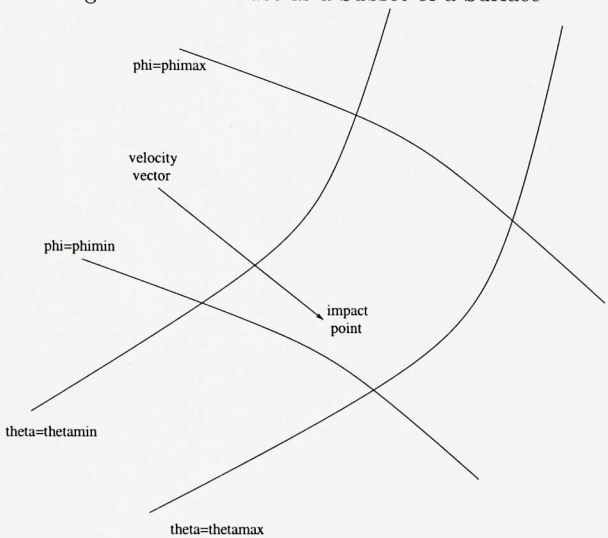
$$\theta_{min} < \theta \leq \theta_{max} \tag{4.21}$$

and

$$\phi_{min} < \phi \leq \phi_{max} \tag{4.22}$$

This adds up to a lot of computation, and it must be stressed that the above task of solving where a line meets a plane has to be performed many times; as soon as some tiles have been removed from the substrate, this technique would be used to track each particle through, one tile at a time, until the impact point is found. For each case, the exit face of the tile has to be established by trial and error, and while it can be assumed that the *top* and *south* faces of the tiles are those most often struck, so testing these cases first, to follow the particle track precisely all other eventualities must be covered. At basic tile resolutions e.g. 120x40x20, there may be the need to pass through 40 tiles to reach an impact point as the erosion modelled reaches the halfway stage. The algorithm would therefore require

Figure 4.3: Tile Face as a Subset of a Surface



application two or three times as often as this. Any simpler alternative would therefore be welcome.

4.4.3 Cartesian Projection Technique

A purely Cartesian technique is available, but does not inspire. First, the tile grid is defined easily in toroidal coordinates, but requires extra computation to obtain the Cartesian values. Secondly, the use of four points to define the plane of the tile face presents a problem. As far as the author is aware, there is no equation of a surface that can be made to fit four arbitrary points in space; a Cartesian plane is defined by three unique points. The technique available then is to ignore one of each of the four points in turn, and to construct planes through the other three. Each of the four points of contact with the vector is calculated, and the average coordinates of the resulting points are used as the strike. Admittedly this level of accuracy would suffice in this and presumably many other cases. Thirdly, whilst this seems an interesting technique, the point of impact now has to be tested for its presence on the tile face. This is simple in a toroidal coordinate system as shown in Figure 4.3, but becomes an unresolved problem here.

Now an investigation of the equations to be solved is conducted. Given three unique

points in Cartesian space, the parametrised equation of the plane thus defined is

$$(x, y, z) = \vec{A} + \lambda \vec{AB} + \mu \vec{AC} \tag{4.23}$$

and using say the following,

$$\vec{AB} = (d_x, d_y, d_z) \tag{4.24}$$
$$\vec{AC} = (e_x, e_y, e_z) \tag{4.25}$$

gives

$$x = a_x + \lambda d_x + \mu e_x \tag{4.26}$$
$$y = a_y + \lambda d_y + \mu e_y \tag{4.27}$$
$$z = a_z + \lambda d_z + \mu e_z \tag{4.28}$$

Eliminating λ and μ achieves the algebraic form of the equation of a plane,

$$px + qy + rz = s \tag{4.29}$$

where

$$p = \frac{d_z}{d_x} - \frac{d_y}{d_x} t \tag{4.30}$$
$$q = t \tag{4.31}$$
$$r = -1 \tag{4.32}$$
$$s = a_x \frac{d_z}{d_x} - a_z - \frac{(d_y a_x - d_x a_y)}{d_x} t \tag{4.33}$$

with the following introduced to simplify the above:

$$t = \frac{(d_x e_z - d_z e_x)}{(d_x e_y - d_y e_x)} \tag{4.34}$$

Now, to solve for the general line meeting this general plane, take the linear system

$$fx + gy = h \tag{4.35}$$

$$ky + mz = n \tag{4.36}$$

$$px + qy + rz = s \tag{4.37}$$

where the constants in the Equations (4.35) and (4.36) can be established as in Section 4.4.1. Solving this yields

$$y = \frac{(nfr + mph - sfm)}{(kfr + mpg - qfm)} \tag{4.38}$$

$$\text{with} \quad x = \frac{h - gy}{f} \tag{4.39}$$

$$\text{and} \quad z = \frac{n - ky}{m} \tag{4.40}$$

and recall that this process must be performed four times to achieve an average value. Recall also that a strike on the plane in question may have been achieved, but establishing whether that point is within the confines of the tile face is just as vital. No method for achieving this will be put forward by this author.

It is not the aim here to compare *flops* (floating point operations) in an exact sense, but to convey the kind of calculations involved in this problem. A strict comparison of *flops* would be made more complex by the presence in the toroidal method of the trigonometric functions, and Taylor series approximations would have to be accounted for, though it may not be known to how many terms. The best comparison would be to run both programs and time them, but this is problematic too. One, different architectures may perform better with the Taylor series, and two, programming and using both techniques would defeat the object of choosing the best to begin with.

In light of the above, if there is a solution which is fast, accurate, and simple to program, it cannot be ignored. Next the actual technique used in this model will be described. It uses an approximation to the toroidal equations of a line, and takes advantage of all the convenient features of this geometry.

4.4.4 Approximation Technique

In toroidal geometry it is possible to make an approximation to a line that is very accurate over short distances, shown later to be at least a few percent of the radius of the torus. At several radii out from the central region, this technique of approximation becomes valid over far greater distances, but conversely, the area close to the torus' circle origin is that where most care is required. For an erosion model where the points of interest are consistently removed from the toroid's circle origin by the radius of the pipe bore, and where the distances to be projected are at most the thickness of the pipe wall, this method can be successfully applied.

These are the equations of the line approximation in toroidal geometry,

$$r\cos\theta + R_o = \frac{D}{\cos(\phi - \beta)} \tag{4.41}$$

$$r\cos(\theta - \alpha) = d \tag{4.42}$$

with some constants to be established. R_o is the radius of the circle origin; d is the perpendicular distance of the line to the pipe centre in the (r, θ) plane and α is the angle of that $\perp$; D is the $\perp$ distance of the line to the point origin in the ϕ plane, and β is the angle of that $\perp$.

The first of these (4.41) is an expression of the fact that the line is uniquely defined in the plane of ϕ, and it is not an approximation. This equation is effectively that of a straight line (that does not pass through the origin) in plane polar coordinates.

The second (4.42) makes the approximation that projected onto the (r, θ) plane, the path of the vector is a straight line. This is only the case in certain special cases, and the usual result is a hyperbola type curve. Recall that this is because there is a different (r, θ) plane for every value of ϕ.

Because however it is desired only to project along this line a very short distance (as a percentage of the radius of the bend or torus), take two Cartesian points on the line that are 'close', and project them onto the (r, θ) plane in toroidal coordinates. Treating these points as if they were in plane polar coordinates, they can then be used to obtain the second approximating equation. This effectively provides a tangent in the limit as the base points approach one another, and describes a straight line in that locale. Globally this would not be accurate, but over a sufficiently local range, it is more than adequate.

This mathematical model benefits from the technique which is easy to implement and

is computationally efficient. For more details and magnitudes of error, see Appendix B.

4.4.5 Motivation for Toroidal Coordinates

The reasons for using a toroidal geometry are the following: the tiles are most naturally ordered and arranged using this system; and the equations of the faces of these are described most simply as either r, θ, or ϕ equals a constant.

Detecting void tiles as the wear proceeds necessitates the intensive solving of velocity vectors in the form of the equations of a line, with the planes that make up the faces of the tiles. For this, the equations to be solved (4.41, 4.42) are quickly formulated by simply setting the relevant coordinates to their constant value. Set one of the three coordinates to a constant value, and this leaves two equations in two unknowns. Finally, when the point of contact with the plane is established, the limits of the tile face are easily described and tested against this point's position in the plane.

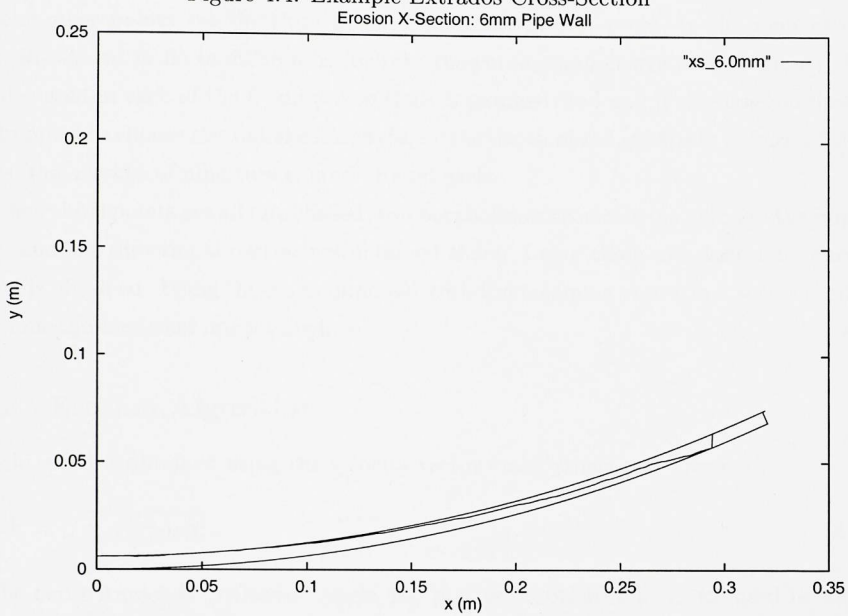
There is a small overhead whereby the constants required for these equations of a line are calculated, but this is performed only once per particle projected. Towards the end of a simulation, the velocity vector may have to pass through forty empty tiles before finding one present, and per tile it will have to test at worst all six faces before it finds the exit. Indeed this constitutes the bulk of the computations performed for this model.

No claim for the relative speeds of the above techniques can be made, as the alternatives have not been used. Inverse trigonometric functions are not single–valued, but this results in managable ambiguity when the domain of interest lies in mainly one quadrant in ϕ and mainly two quadrants in θ. While it is true that the use of the toroidal equations requires repeated use of trigonometric functions, which may or may not slow computation dramatically according to machine architecture, it is the view of the author that establishing and coding the alternative systems would have been harder and more time consuming.

Indeed, both the other systems described in the previous sections were incomplete. In the true toroidal system, a quartic equation would have to be solved, and only one of a multiplicity of solutions can be correct. In the Cartesian system, a method of establishing whether or not a point is within certain limits in an entirely general plane would be required.

The author has not put forward solutions to these problems.

Figure 4.4: Example Extrados Cross-Section



4.5 Point of Impact

Having established that a particular particle has struck a particular spot on our tile substrate, we now require its speed and angle at impact in order to use our supplied erosion data.

4.5.1 Surface Fitting and Normal Generation

The section of pipe wall modelled is made up of little blocks approximating cuboids, where the diameter of the incoming particles will be far smaller than the dimensions of an individual tile. Clearly to take the normal vector to the surface of one tile will not produce realistic angles of impact, as one of the assumptions of this model is that as the surface erodes, it stays 'smooth'.

What the tiles do represent well is the global picture of erosion over the whole domain, and viewed like this the eroded surface will appear to curve in places, but limited by the tile resolution (Fig. 4.4). In order to obtain realistic angles of impact, information about what is going on in the tile substrate at a distance must be used in conjunction with the

depth of wear at the impact point.

Four other points are therefore taken around the impact point, in the manner of a *five point stencil* in finite differences (or take the corners and centre of a rhombus). The distance used in each of the θ and ϕ directions is parametrised and is discussed in Section 5.5. In order to choose the radial coordinate, or the depth of the substrate at each of these points, the average of nine tiles is used around each.

When these points are all established, two parabolas are constructed through the impact point, and the directional derivatives obtained there. Using these in a vector product, a normal is obtained. Using this in conjunction with the incoming vector in a scalar product ascertains the modelled impact angle ψ.

4.5.2 Erosion Algorithms

Particle speed is obtained using the velocity vector and Pythagoras' Theorem.

$$V = \sqrt{v_x^2 + v_y^2 + v_z^2} \tag{4.43}$$

The two parameters ψ (impact angle, see previous section) and V are used in simple functions that are based on experimental data, to obtain an erosion rate in *grams lost per kilogram striking.* This erosion rate is in turn used in order to calculate an amount of mass to be removed from the struck tile in the model. This mass loss is simply defined by

$$n_{ijk} = m_{ijk} - q \tag{4.44}$$

where n_{ijk} is the updated mass of tile (i, j, k) after the strike, m_{ijk} is the mass of the same tile before the impact, and q is the mass loss. This is calculated using

$$q = m_p \epsilon(\psi, V) s \tag{4.45}$$

where ϵ is the erosion rate as a function of ψ and V, m_p is the mass of the impacting particle, and s is a scalefactor, typically around 10^4 to 10^8, used to avoid the need to generate billions of particles, as might be present in a real system.

The erosion rates employed use algorithms based on empirical data, obtained by *Dr.David Allsopp* and *Dr.Ian Hutchings* at the *Department of Materials Science, University of Cambridge* as a part of this EPSRC project. A matrix of experiments were performed, to ascertain the erosion rate of a stream of erodent particles impinging at a constant speed and at a constant angle upon a flat target of the sample material.

The erodent used for this specific matrix was olivine sand. For each of the four target materials tested (mild steel, Ni–Hard, basalt, Alumina) the matrix of experiments was of three angles of impact (20^o, 30^o, and 90^o) at three different speeds, approximately 20, 30, and 40 ms^{-1}. These values were chosen by the project team in consultation with industrial collaborators, in order to reflect the kind of working situations that engineers face. Lower angles of testing are not practical with a gas–blast erosion tester.

Comparing this gas–blast tester with historical rotating disc data (*Burnett*, [Bur96]), and using a reference point of impact 30^o at 25 m/s, the RDET gave a value 6% higher, so an error of $\pm 6\%$ might be assumed for the erosion data.

Piecewise linear interpolation is used within the erosion data matrix, to retrieve erosion rates at whichever velocity and angle occur. It is assumed the zero point takes zero value for both variables, i.e. that an impact angle of zero, or a velocity of zero, results in no erosion. This is particularly necessary for the angular dependence as despite the difficulties in making laboratory measurements at low angles, in the model this type of collision certainly does occur.

4.6 Rebounding Particles

The erosion model described in this work has multi–impact capability, and can utilise the impact data of two waves of strikes.

The CFD and particle tracking codes find the first impact locations, and *Erode* models the wear due to these impacts. Each particle can if necessary rebound from the tile substrate several times until it leaves the pipe wall, this being defined by the original unworn value of the radius of the inside pipe bore.

Information about the exiting particle is dumped at this point, so that the CFD engine can retrack the particles, until they strike the wall again, as dense particles undoubtedly will. This second wave of impacts can be read by *Erode* in conjunction with the original wave, and used together, weighted 50%–50%, they make for a more realistic measure of the actual number of impacts occurring in the bend.

This section describes how the particle rebounds are modelled. The empirical data referred to was obtained by *Dr.David Allsopp* and *Dr.Ian Hutchings* at the *Department of Materials Science, University of Cambridge* as a part of this EPSRC-funded project.

The empirical data supplied was of a statistical nature, and was sourced from video recordings of an experiment that used incoming particles impinging on a target, and the consequent rebounds. The particles used were Ballotini and olivine sand, and the targets

Figure 4.5: Power Law Rebound Algorithm

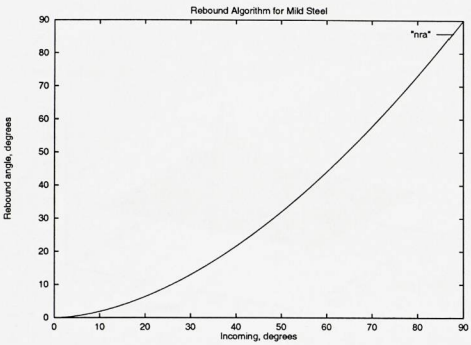
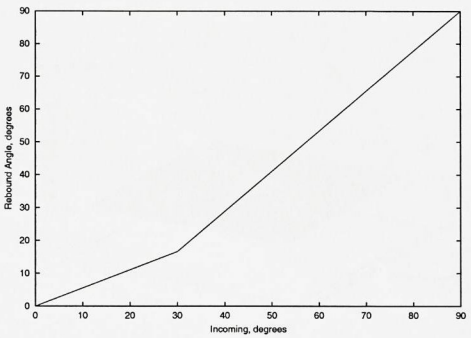


Figure 4.6: Piecewise Linear Rebound Alg.



were of four commonly used pipe materials found in pneumatic conveyors: mild steel, Ni-hard, Alumina, and cast basalt. Around a hundred particles were sampled in each experiment, and these experiments were performed at a variety of impact angles and speeds for the target materials recounted above, similar to the matrix used for the erosion data described in Section 4.5.2.

4.6.1 Rebound Angle

Using the data mentioned in Section 4.6, rebound angles were initially implemented using a power–law fit, with equations of the form $y = Ax^e$ (Figure 4.5). However, having investigated the types of angles involved in the model, it was found that many impacts occur at the lower end, around 5^o. Using a power–law fit seems reasonable at the higher angles, but the size of the rebound diminishes rapidly below 5^o. Realistically, considering the irregular shape of both the particles and the worn surface, rebound angles of 0.5^o just seem too low, and indeed, they were being lost computationally, because so many bounces were required before the particles in question returned to the flow. The present model has therefore reverted to a piecewise–linear fit (Figure 4.6) but still assuming zero rebound at 0^o, and normal rebounds at 90^o.

As well as a core value obtained with this algorithm, each particle rebound is randomised to some degree, dependent on the measured standard deviation in the associated experiment. Note that the experimental data was two-dimensional and that therefore the rebounds were initially implemented in this manner, using the plane that contains both the incoming velocity and the normal. Any scatter which occurred outside this plane was absent from this model.

Figure 4.7: No Transverse Smear

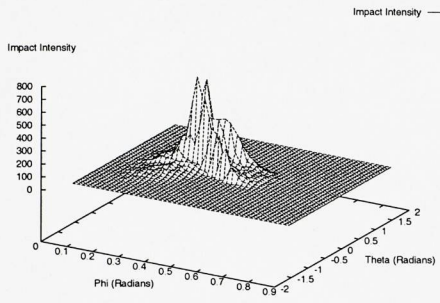
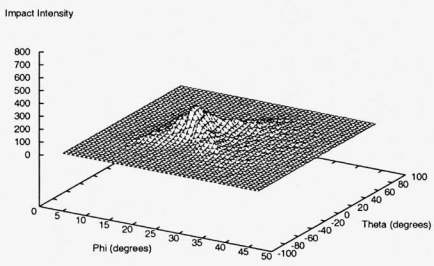


Figure 4.8: Circular Transverse Smear



The resulting second impacts can be seen in Figure 4.7 and they look extremely focussed. After some thought it was decided that the assumption that all rebounds remained in the plane of the incoming and normal vectors could be contributing to the sharp peaks seen.

If some transverse smear (randomisation out of the above-mentioned plane) is added in to the rebound velocities, a great difference in the impact intensities can be found. There is a strong dependence on this smear, so an experiment should ideally be performed to establish the kind of range that occurs. For the time being, a value equivalent to the planar randomisation is used (Figure 4.8) which would produce a rebound 'beam' of circular cross–section.

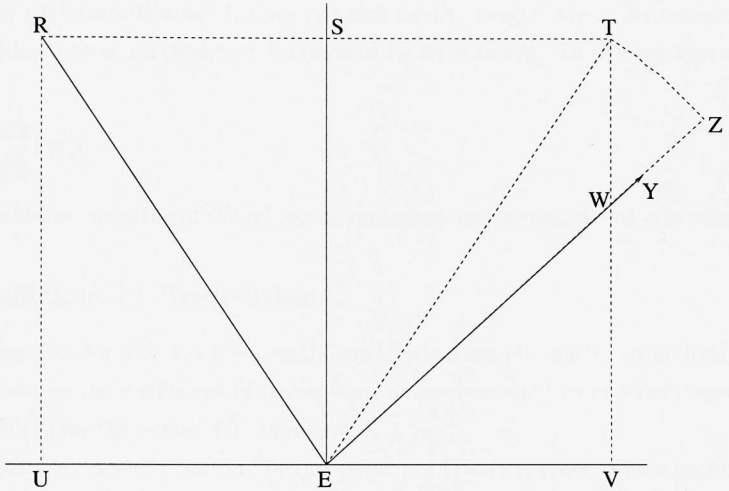
The lifetime dependence on this varying impact data is not quite as extreme as the two images suggest, because it is only the second wave impacts that are affected by this change. The first wave impacts stay constant and occur at higher speeds, so that most of the wear is due to these. Nevertheless, lifetime almost doubles in the second case, where the peak is absent.

4.6.2 Rebound Angle: Implementation

As described in previous sections the velocity (u, v, w) (4.3), the impact point $\vec{E}$ (4.4), the normal to the surface $\vec{n}$ (4.5.1), and the angle of impact ψ (4.5.1) of the particle in question have all been established. Let it also be assumed for now that the new rebound angle χ and the coefficient of restitution R to be applied to the rebounding velocity, have been found using algorithms described in Sections 4.6.1 and 4.6.4 respectively.

The rebounds are implemented in Cartesian geometry, and then converted back to the toroidal for projection through the tile substrate. In Figure 4.9, the impact point is

Figure 4.9: Rebound Guide



represented by the vector $\vec{E}$. The point $\vec{R}$ is chosen by projecting back along the velocity vector by one second, and then the point $\vec{U}$ can be found by solving for where the normal through $\vec{R}$ meets the plane defined by $\vec{E}$ and its normal. For details on how to construct planes around a normal, see Appendix C.

The point $\vec{T}$ is found by adding $2\vec{UE}$ to the point $\vec{R}$, so that a specular reflection has been achieved, in the vector $\vec{ET}$. The angle of this vector now has to be manipulated, and its magnitude corrected, and then finally the restitution can be applied.

In order to divert this outgoing vector, consider the parallel and perpendicular components, $\vec{EV}$ and $\vec{VT}$ respectively. These have already been obtained, as $\vec{EV}$ is the same as $\vec{UE}$, and $\vec{VT}$ is the same as $\vec{UR}$ $(= -\vec{RU})$. It is known that

$$\tan\psi = \frac{|\vec{VT}|}{|\vec{EV}|} \tag{4.46}$$

so to find the point $\vec{W}$, use

$$\tan\chi = \frac{|\vec{VW}|}{|\vec{EV}|} \tag{4.47}$$

to find the modulus of $\vec{VW}$, as the other two quantities are known; χ being the desired rebound angle. The vector $\vec{VW}$ is then given by scaling $\vec{VT}$.

$$\vec{VW} = \frac{|\vec{VW}|}{|\vec{VT}|}\vec{VT} \tag{4.48}$$

Now establish the vector $E\vec{W}$ by the addition of $E\vec{V}$ and $V\vec{W}$, and this is the path that the rebounding particle will take. It only remains for the magnitude to be corrected, as the vector will have been shortened or lengthened by its rotation. To achieve this scaling correction use

$$E\vec{Z} = \frac{|E\vec{T}|}{|E\vec{W}|} E\vec{W} \tag{4.49}$$

For the transverse smearing of the rebound, analogous vector manipulation is employed.

4.6.3 Coefficient of Restitution

Once the outward vector $E\vec{Z}$ has been established, it is a simple matter to multiply each of its components by the coefficient of restitution to simulate the loss of kinetic energy in the impact. This gives the vector $E\vec{Y}$ as required.

The restitution is currently taken to be independent of particle speed, so for particularly friable materials, or faster conveying conditions, this may have to be revised. The values used were experimentally established, as described below.

4.6.4 Restitution Algorithm

As a part of the Cambridge experiments (Section 4.6), three restitution values were measured at different angles for olivine sand on mild steel. These were very close to collinear (Fig. 4.10), and indicate that a linear fit is a good approximation to the angular dependence of restitution, at least in the range 20^o to 45^o. In the absence of other data, this line was extrapolated over the whole quadrant (Fig. 4.10).

For the other materials initially covered by this model, namely Alumina, basalt and NiHard, only one restitution data point was obtained, at 30^o. In these cases, a linear extrapolation from one point was employed, using the gradient of the mild steel algorithm. The data is shown in Fig. 4.11, with extrapolations added at impact angles of 0^o and 90^o degrees. Where extrapolation would have resulted in coefficients greater than one, the maximum value was capped at 0.98, e.g Fig. 4.12, this being an arbitrary choice, satisfying *Conservation of Energy.*

Some of the data in the Cambridge set can be directly compared with similar found in the literature. Ballotini (glass spheres, diameter $7\mu m$) was also investigated at Cambridge and this data (impact velocity $31.0ms^{-1}$) has been plotted against measurements of a similar test material (steel, at $26.5ms^{-1}$) made by Kleis, Frishman and Pappel [KFP95] in Fig. 4.13. It is a good match, considering the velocities are different.

Figure 4.10: Restitution for Mild Steel at $29ms^{-1}$

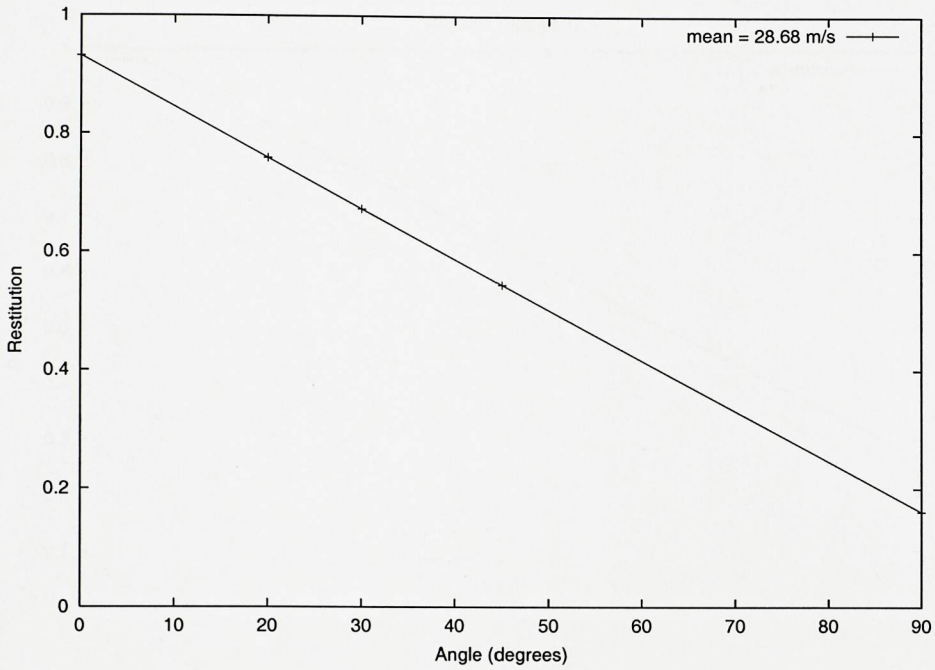


Figure 4.11: Restitution for Different Materials

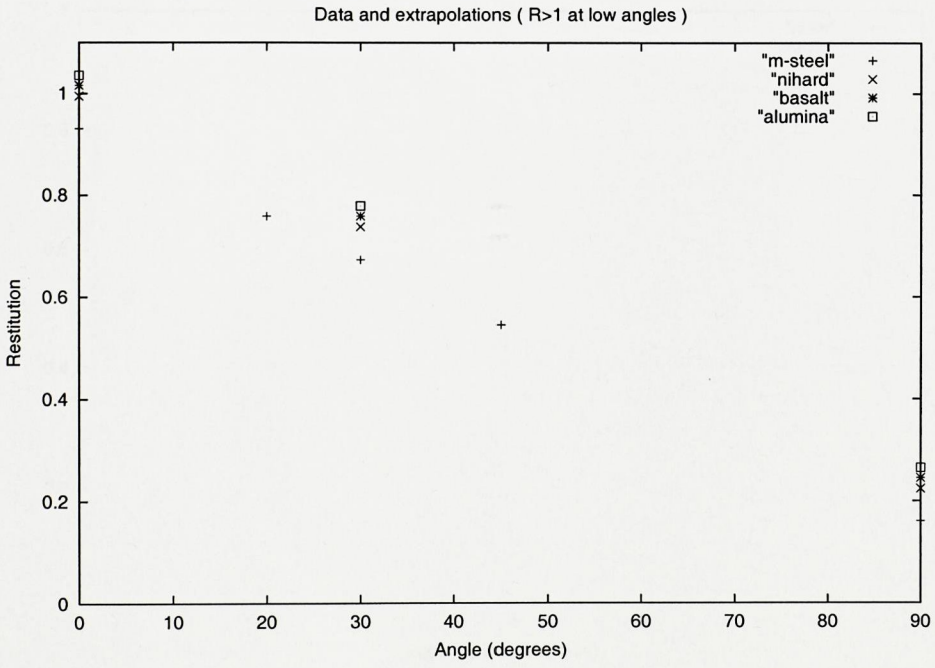


Figure 4.12: Capped Algorithm for Alumina

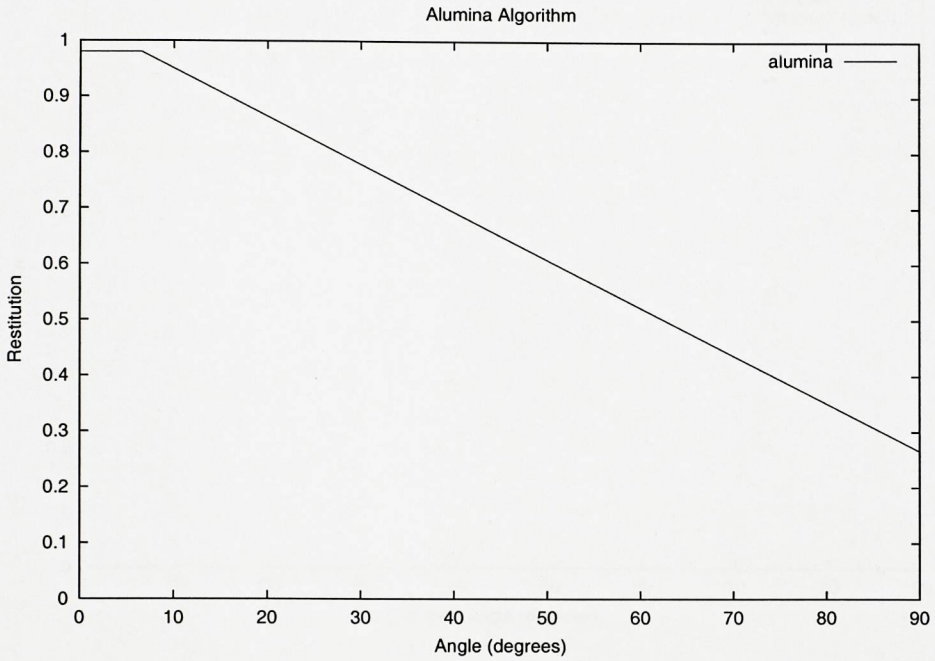


Figure 4.13: Ballotini Comparison

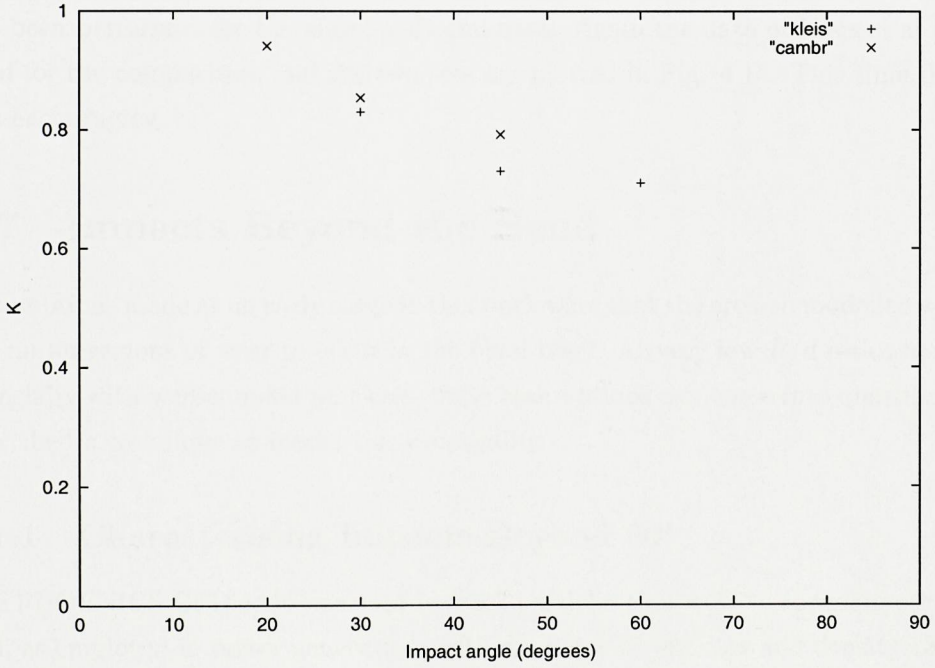
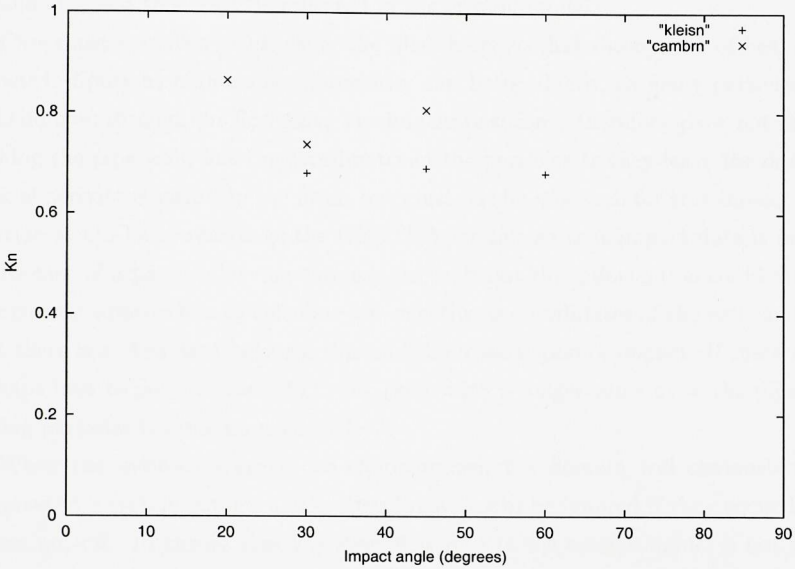


Figure 4.14: Ballotini Comparison; Normal Component



A popular practise in erosion studies is to separate the normal components of the impact and rebound velocities, and use these to obtain a normal restitution coefficient. This is not necessary in the current model, but in the interests of making comparisons the process has been performed for the above Ballotini data. Again the data of Kleis et al [KFP95] is used for the comparison, and the two sets are plotted in Fig. 4.14. This time the result is less encouraging.

4.7 Impacts Beyond the Bend

Assumptions made at an early stage in this work were that the erosion modelled would cause the major regions of wear to occur in the bend itself. At very low R/d ratios however, and especially with well-coupled particles, these assumptions can come into question. Below is described a technique to model this eventuality.

4.7.1 Characterising Impacts Beyond 90^o

The PHOENICS CFD code was used to track particles that were to approximate pulverised fuel, as employed in power generation, with diameter 75 *microns* and density 1300 kg/m^3.

In a short radius bend (R/d=1.5) some 49% of these particles struck the pipe bend at beyond 90^o, and this must be reflected in the erosion model.

Care must be taken when using the filter here, so that the regions of heavy wear are detected. Spurious high impact intensities can be produced, as many particles leave the CFD domain through the flow exit. The impact position is therefore given not as a point of striking the pipe wall, but the coordinates of the particles as they leave the domain. This peak of activity is visible in the filter, but must not be mistaken for true impact data. One solution would be to rearrange the GENTRA module so that impact data is not dumped in the case of a particle leaving through the exit, but this information could then be lost. The current approach is to calculate the effective ϕ–coordinates of the exit, and to ensure that there is a clear area between this and the main region of impact. If there is not, it is perhaps best to perform the CFD runs again with a longer run–out to the pipe, until the exiting particles become unmistakable.

When the intensities are to be characterised, the domain will obviously have been trimmed to a certain extent, and certain impacts will be ignored if they occur beyond the chosen cut-off. To ensure that the mass flow rate in the erosion model is not affected by this, the proportion of particles in the CFD batch that hit the substrate is recorded and passed to *Erode*.

The particle velocities are characterised as normal at this stage.

4.7.2 Modelling Erosion Beyond 90^o

Rather than introduce a cylindrical polar segment of tile substrate into the *Erode* code, the existing toroidal geometry has been approximated to this, to model erosion which occurs beyond the far end of a 90^o bend. To justify this: it is a large undertaking to interface two different geometries at a relatively late stage of this project; only relatively few of the cases in the prediction matrix will experience wear at this extreme position; the extension of wear scars beyond 90^o does not exceed 30^o; the extension of burst points beyond 90^o does not exceed 10^o.

The coordinates used in recording particle impacts in the pipe run–out are ϕ and θ, when they should in fact be θ and z, as in cylindrical polars. Note that the θ coordinates in each system are entirely equivalent, and so too are the r coordinates. The radius r is not used at this stage because it is fixed at $r = r_{min}$ for each impact on the pipe wall.

Because of the approximation made in projecting impacts on a straight pipe onto a curved pipe, some compression of impacts will be experienced as the domain curves further

away from the straight pipe. These impacts will not be near the critical region though, if burst points do not exceed 100^o.

In order that erosion rates are not unrealistic in the pipe run-out, a modification was made to the algorithm that generates the impact angle. If the point of impact exceeds $\phi = 90^o$ around the bend, the impact angle ψ is reduced by $(\phi - 90^o)$. This approximates the corresponding impact angle if the particle were to strike the run-out, and hence the corresponding amount of wear.

When the erosion algorithm is applied, the calculated mass loss to a particular tile is scaled by the hit rate that was counted and passed on by the filter. Recall that not all particles in a CFD batch will always strike the bend, and batch size is sometimes increased in order to give a sufficient sample of strikes in the area of characterisation.

For a first impact, the erosion is scaled by: *hit rate 1*, and for a second wave impact: *hit rate 1 * hit rate 2*. None of the particles that missed the domain the first time, will be around to strike it in the second wave. This scaling ensures that the mass flow in the virtual rig is increased in proportion to the number of strikes predicted by CFD. This is a general routine; if all particles strike the modelled domain, as they will in some geometries, the scaling factors are equal to one.

4.8 Chapter Summary

In this chapter, custom software is described which models the erosion element of this simulation, and this is given the name *Erode*. This virtual pneumatic conveying rig employs a novel and computationally efficient way of tracking vectors through a toroidal geometry. A three-dimensional representation of pipe wall is stored in memory, and is divided into many individual cells or tiles in all three dimensions. *Monte Carlo* techniques are used to randomly generate incoming particles, according to statistical data previously described in Chapter 3.

Details of the erosion model described herein include: detection of particle impacts by a general three-dimension tracking algorithm; surface normal generation and calculation of particle impact angle; mass loss algorithms created from experiments carried out by project partners; consequent tile removal to mimic mass loss on a macroscopic scale; and generation of rebound angle, restitution factor and consequent rebound vector.

Finally a way of accounting for very large diameter short radius bends is given, as in some simulations of this type particles will strike the pipe wall beyond 90^o of the bend, and burst points can even occur in this region. Due to these initially unexpected results,

modifications were introduced to this toroidal geometry based model in order to avoid the need to introduce a cylindrical polar section of tile mesh.

Chapter 5

Erode Behaviour: Parameter Sensitivity Analysis

This chapter investigates the behaviour of the custom erosion software, *Erode*. Though not exhaustive, this chapter will establish the viability of the predictive model, and the ranges of values that must be used for the parameters. Careful study of this chapter will increase greatly the reader's understanding of the way in which this mathematical model works.

5.1 Scale–Factors and Mass Conservation

In outline, recall that this model works by modelling the pipe wall or substrate as an array of tiles (Figure 5.1) in a toroidal geometry. Particles are randomly generated using statistical techniques, they are tracked to an impact point on a particular tile, and using erosion algorithms, an amount of mass is subtracted from the impacted tile.

At this stage it is noted that the mean amount of mass lost due to the impact of each individual particle (currently uniform size and density) can be calculated from erosion algorithms, but if this practice were followed strictly, it would require the generation of many billions of particles, possibly taking longer to compute a prediction than it would take to perform an experiment.

A way around this is to use a scale-factor, and to multiply the amount of mass removed by, for instance, one million. It is a simple matter to allow for this when calculating the mass-flow rate modelled, and as long as a large statistical sample of particles is still used, there should be no loss of accuracy.

In Fig. 5.2 a variety of scalefactors were used to find a range where there was as little dependence as possible, and indeed at values around $2x10^5$ the fluctuations die down

Figure 5.1: Substrate

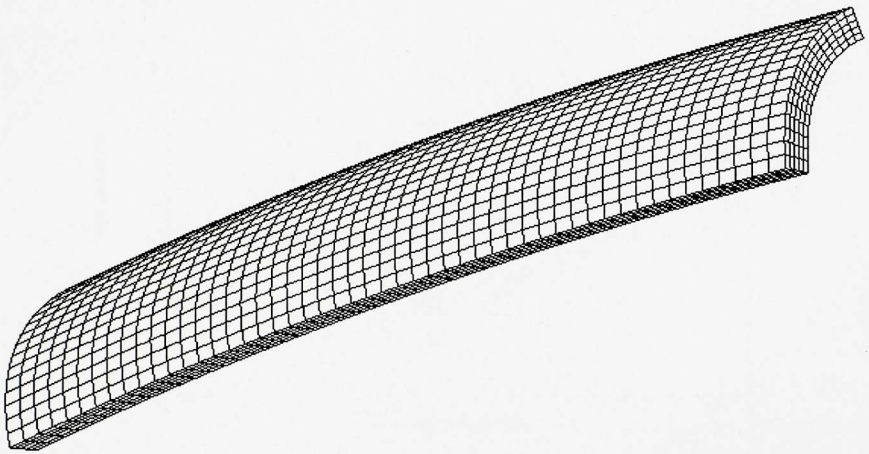


Figure 5.2: Scalefactor Dependence

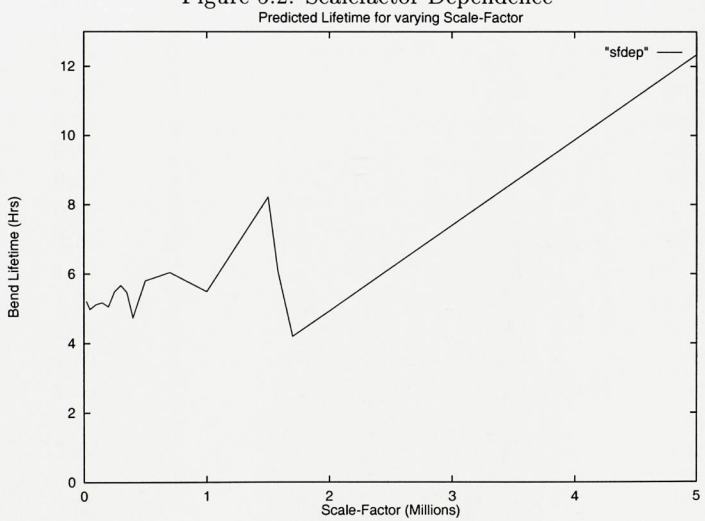


Figure 5.3: Scalefactor Dependence with Mass Conservation

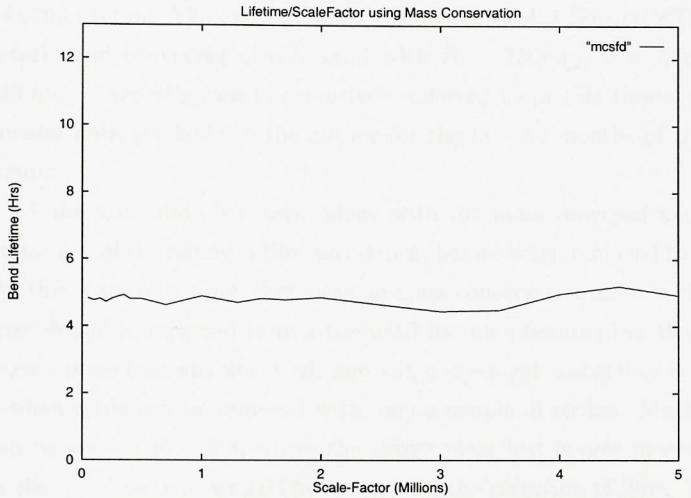


Table 5.1: Variation of Predicted Lifetime with Scale–Factor

Scale–Factor(M)	Conveyed(Tonnes)	Predict.Life (Hrs)	Hits per Tile
0.02	4.9	5.21	125.66
0.05	4.6	4.98	50.43
0.10	4.8	5.12	25.56
0.15	4.8	5.17	17.28
0.20	4.7	5.05	13.06
0.25	5.1	5.49	10.71
0.30	5.3	5.67	9.07
0.35	5.1	5.47	7.81
0.40	4.4	4.74	6.70
0.50	5.4	5.80	5.72
0.70	5.6	6.04	4.23
1.00	5.1	5.49	3.04
1.30	6.6	7.13	2.62
1.50	7.7	8.22	2.49
1.70	3.9	4.20	2.06
2.00	4.6	4.93	1.74
2.50	5.8	6.17	1.45
3.00	6.9	7.40	1.33
4.00	9.2	9.87	1.24
5.00	11.5	12.34	1.20

considerably. The lowest scale–factor of $2x10^4$ resulted in a 43 minute run using $1.8x10^6$ particles; that of $5x10^4$ required 17 minutes, and the highest scale–factor of $5x10^6$ required one minute of computation. The case modelled was the one in *A.J.Burnett's* Thesis [Bur96]; a 90^o mild steel bend conveying olivine sand with $R = 750mm$, $d = 53mm$ and wall thickness $2.39mm$. *Burnett's* case is extensively referred to in this thesis, as it was the only experimental data available to the author for the first 30 months of this three year Phd. programme.

In Table 5.1 the same data is shown, along with the mass conveyed and the average number of times each of the removed tiles was struck, before being removed to simulate hole formation. At this stage of testing, there was no mass conservation used in the algorithm, such that mass would be removed from a tile until its mass became less than or equal to zero. Any excess mass loss was not used, and the consequent instability is clear at high scalefactors, when a tile can be removed with only a couple of strikes. Much more stable behaviour can be seen in Fig. 5.3, where the excess mass lost is now passed to the next neighbour in the $+\phi$ direction, which corresponds to the direction of flow. This is not so much *Conservation of Mass* as it is *Conservation of Mass Loss* according to the erosion algorithms.

Such an implementation eliminates the mass inaccuracies that accrue due to the discrete nature of the tiles, but tile resolution may still affect the predictions made via the path of surface-fitting, to be discussed later in this chapter.

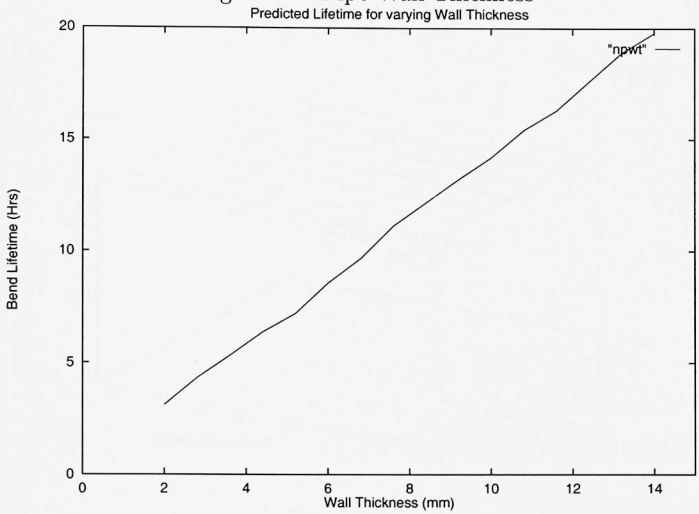
Though any scalefactor now looks to give a reasonable result, the author's advice when using the model is to find one such that an average of around 12-16 strikes per tile is required for their removal. Any more than this, and excess computations are being performed. Any less, and the statistical nature of the way the particles are randomly generated is not used to best effect. Statistical samples that exceed 20000 particles are used to characterize the impacts at the CFD stage, and the number of particles generated by the model should ideally exceed this to ensure that the random numbers smoothly reproduce the data.

5.2 Pipe–Wall Thickness

Some different wall thicknesses were used to see how the model reacted. The case is again *Burnett's.* [Bur96]

In Fig. 5.4 dependence appears to be linear, though this model does not account for the potential deposition of conveyed material within deep cavities. Any such material would presumably have a shielding effect and increase bend life. No experimental data has yet

Figure 5.4: Pipe-Wall Thickness



been found to enable comparison with these predictions.

5.3 Grid Dependence

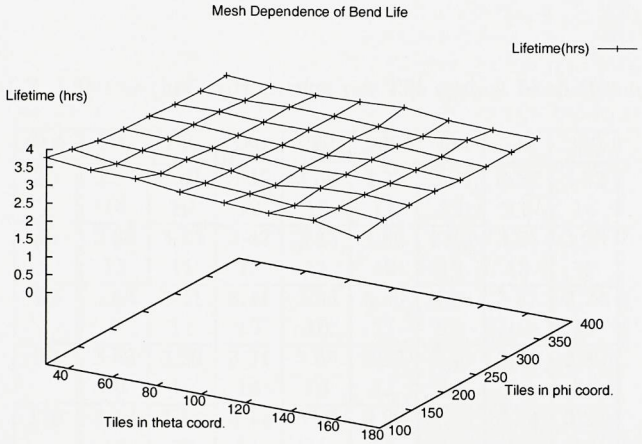
Depending on the number of tiles used in the tile-array, or mesh, used to model the substrate, lifetime predictions can vary. This section investigates the behaviour, first by keeping the radial resolution constant (20 tiles), and varying both the theta and phi resolutions, and then by keeping fixed theta and phi values, and varying the radial resolution. The percentage of tiles removed from the substrate in forming a burst point will also be looked at in this manner.

5.3.1 Theta- and Phi-Mesh Lifetime Dependence

The ranges used in this study were chosen firstly with consideration to the likely memory availability on the user's computer, and secondly at the other end of the scale, in the belief that there should be at least several tiles in each zone used to characterize impact position and velocity. The radial resolution was fixed at 20 tiles, and the scalefactor varied over a wide range, in order that the number of strikes per tile was held at a reasonable value, ranging here between 9 and 15.

Table 5.2 shows the results; the lifetime in hours above the number of strikes per tile

Figure 5.5: Bend Life against Tile Resolution



rounded to an integer. The first comment is that the lifetime decreases as the resolution increases, but not very steeply (Fig. 5.5).

To see this in more detail, some plots have been produced in Fig.s 5.6 and 5.7, where the lifetime is plotted primarily against one coordinate, but biased in the other too. In each figure, there are eight groups of points. If plotted in the usual manner, each group would be in a vertical line, as e.g. in Fig. 5.6 each shares the same θ value. However, in order to show the trend as the table's rows are traversed, the points are spread out a little in θ, with the lowest ϕ values at the left of each group, increasing towards the right. A similar approach is used for the ϕ dependence.

The first conclusion to be drawn here is that the predictions are reasonably stable. They would be expected to be more accurate at higher resolutions, given the statistical approach used. The variation over the parameter ranges used is a maximum of $\pm 9.5\%$, falling to $\pm 4.3\%$ if the coarsest resolutions are ignored, and falling to $\pm 2.2\%$ if the next two coarse resolutions are ignored. This would suggest a minimum mesh of (90,220) tiles.

Given that the predictions appear to be settling asymptotically with increasing mesh, what are the finite resolutions that give adequate results? Looking again at Figure 5.6 even at high resolutions in θ, there still seems to be a benefit in increasing the ϕ resolution, whereas at higher ϕ resolutions (Figure 5.7) the scatter in the points is less and looks more random. This indicates that the ϕ resolution is more critical, and that if computing resources are limited, the θ mesh is where savings should be made.

Table 5.2: Lifetime (hrs) with Strikes per Tile against Mesh Resolution

$\theta \backslash \phi$	100	140	180	220	260	300	340	380
30	3.77	3.61	3.45	3.38	3.35	3.29	3.27	3.32
	14	10	12	12	11	12	11	14
50	3.66	3.44	3.42	3.35	3.36	3.33	3.25	3.26
	11	11	13	11	12	10	13	12
70	3.66	3.41	3.41	3.33	3.30	3.28	3.31	3.24
	9	11	10	10	11	11	12	11
90	3.53	3.39	3.34	3.33	3.27	3.30	3.29	3.30
	10	15	14	13	12	12	11	11
110	3.47	3.33	3.17	3.25	3.22	3.30	3.24	3.28
	10	10	11	11	11	11	11	10
130	3.42	3.24	3.33	3.32	3.34	3.32	3.20	3.25
	11	11	11	11	11	12	11	10
150	3.46	3.41	3.39	3.31	3.29	3.26	3.24	3.22
	12	11	11	11	12	11	11	10
170	3.21	3.31	3.33	3.33	3.24	3.22	3.23	3.22
	12	11	11	12	12	12	12	11

Table 5.3: Percentage of Tiles Removed at Burst

$\theta \backslash \phi$	100	180	260	340
30	6.2	5.8	5.5	5.5
70	6.3	5.8	5.6	5.6
110	5.8	5.5	5.6	5.4
150	5.6	5.6	5.5	5.5

Figure 5.6: Bend Life against Theta Tile Resolution

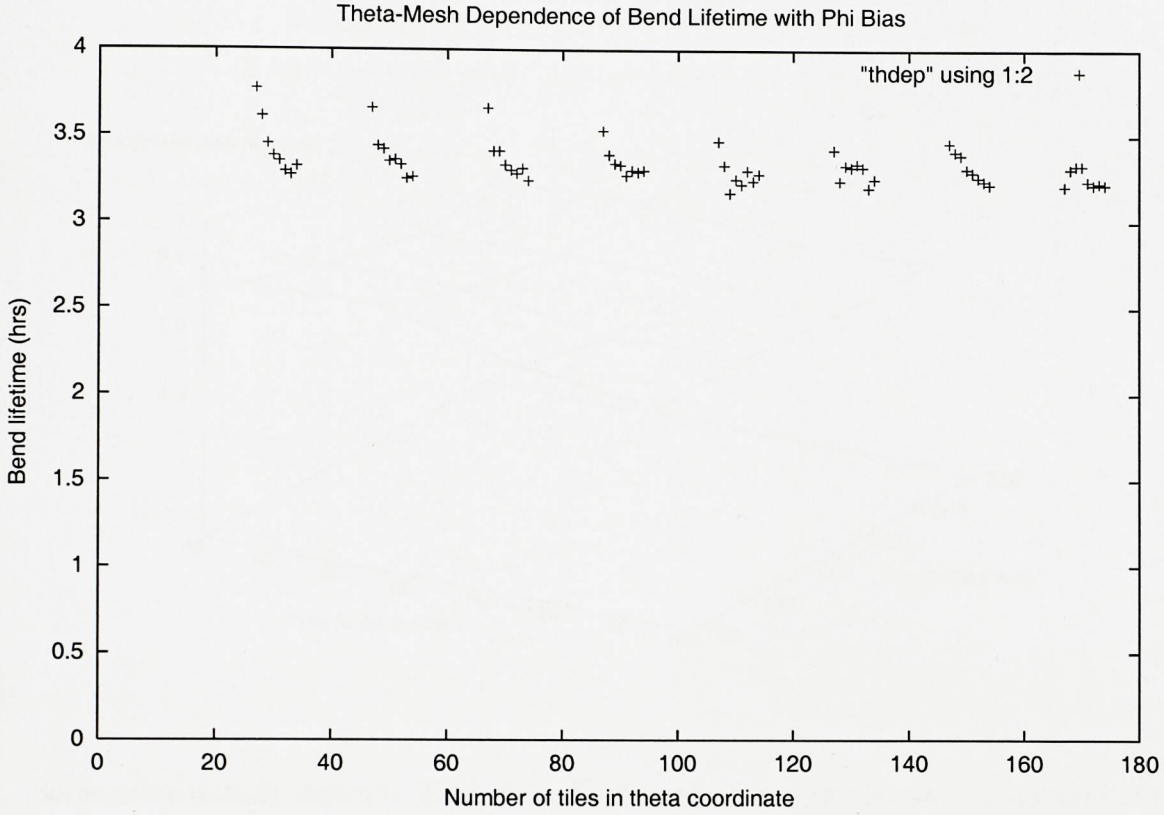


Figure 5.7: Bend Life against Phi Tile Resolution

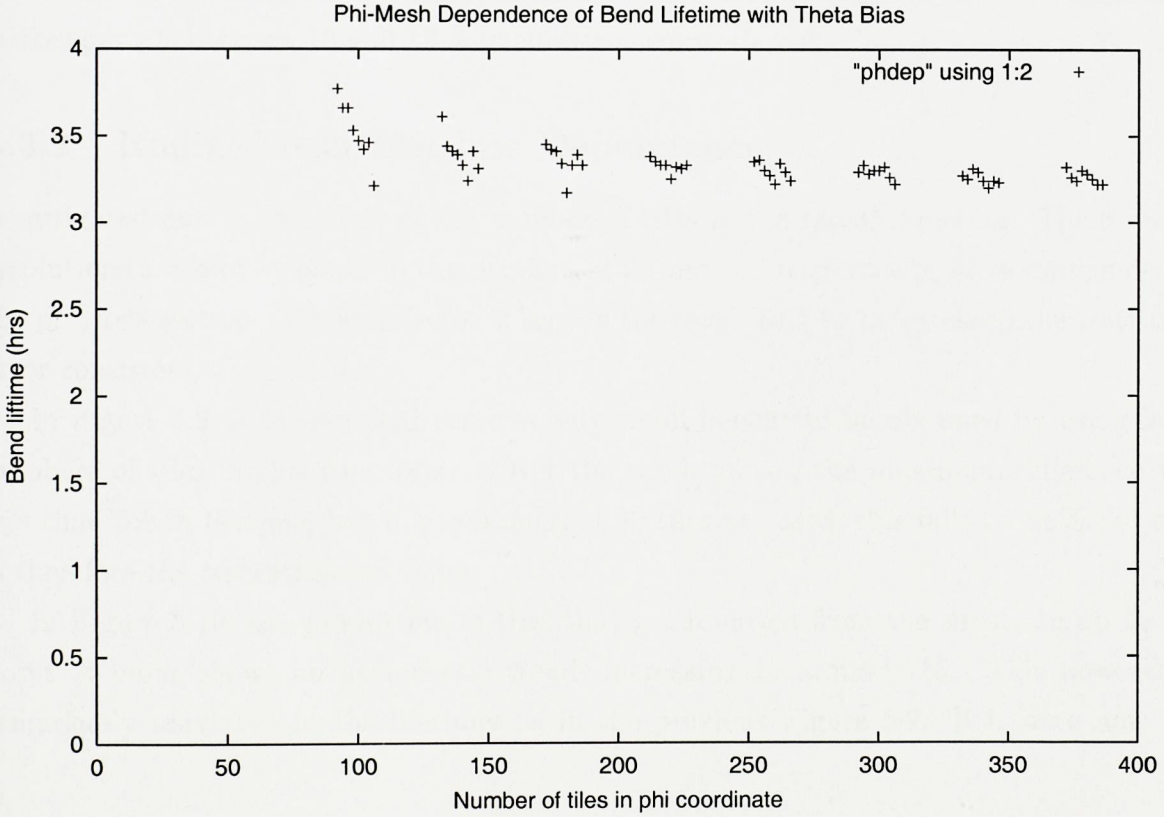
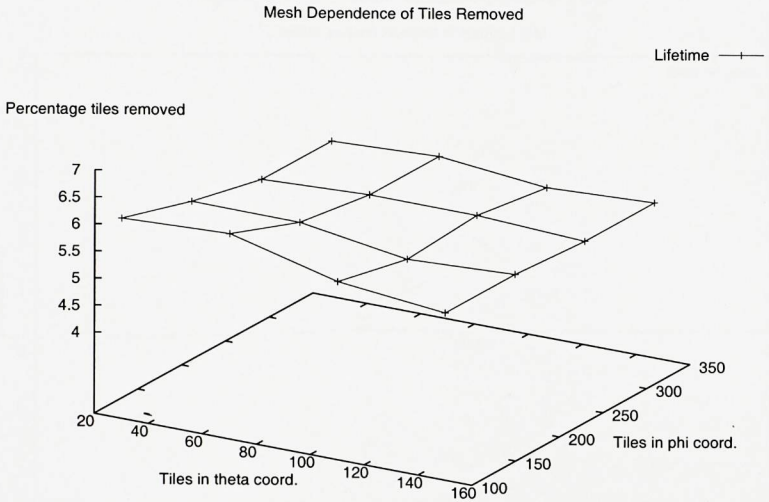


Figure 5.8: Percentage of Tiles Removed



Some more data is shown in Table 5.3. At the end of the simulation, the number of tiles that have been removed from the substrate to simulate wear are shown as a percentage of the initial total. Again, differing scalefactors were used in order to keep the number of strikes per tile between 10 and 12, and similar asymptotic behaviour is seen, as in Fig. 5.8.

5.3.2 Radial-Mesh Lifetime Dependence

Investigated now is the effect of the number of tiles in the radial direction. The θ and ϕ resolutions are kept constant in this section, at 90 and 220 respectively, as recommended in the previous section. The scalefactor is kept in the range 10.5 to 11.7 to keep the statistical error consistent.

In Figure 5.9, it is seen that there is only small benefit to be obtained by using large numbers of tiles in this coordinate. Over the whole range, the maximum difference was less than 5% in lifetime, but if a minimum of 30 tiles are used, this falls to 1.8%. 30 tiles is therefore the recommended value.

In Figure 5.10, the proportion of tiles that get removed from the substrate up to the point of burst shows an asymptotic trend, increasing to around 7%. This however is remarkably unrelated to the lifetimes as in the previous Figure 5.9. If it were not, the

Figure 5.9: Lifetime against Radial Tiles

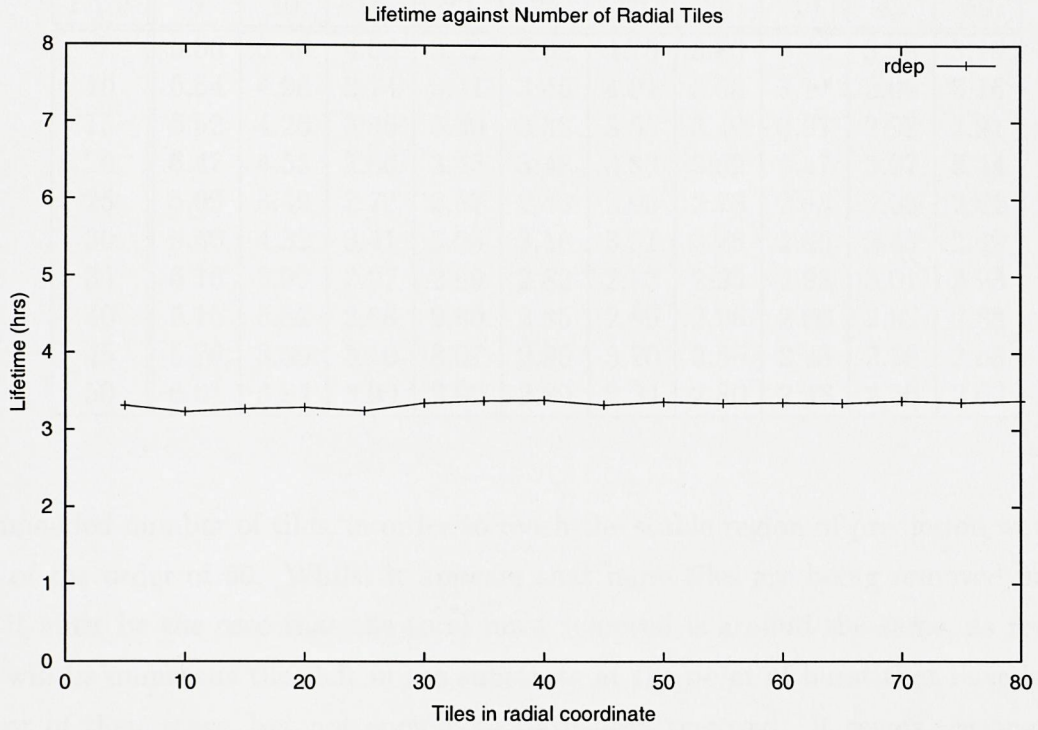


Figure 5.10: Proportion of Tiles Removed at Radial Resolutions

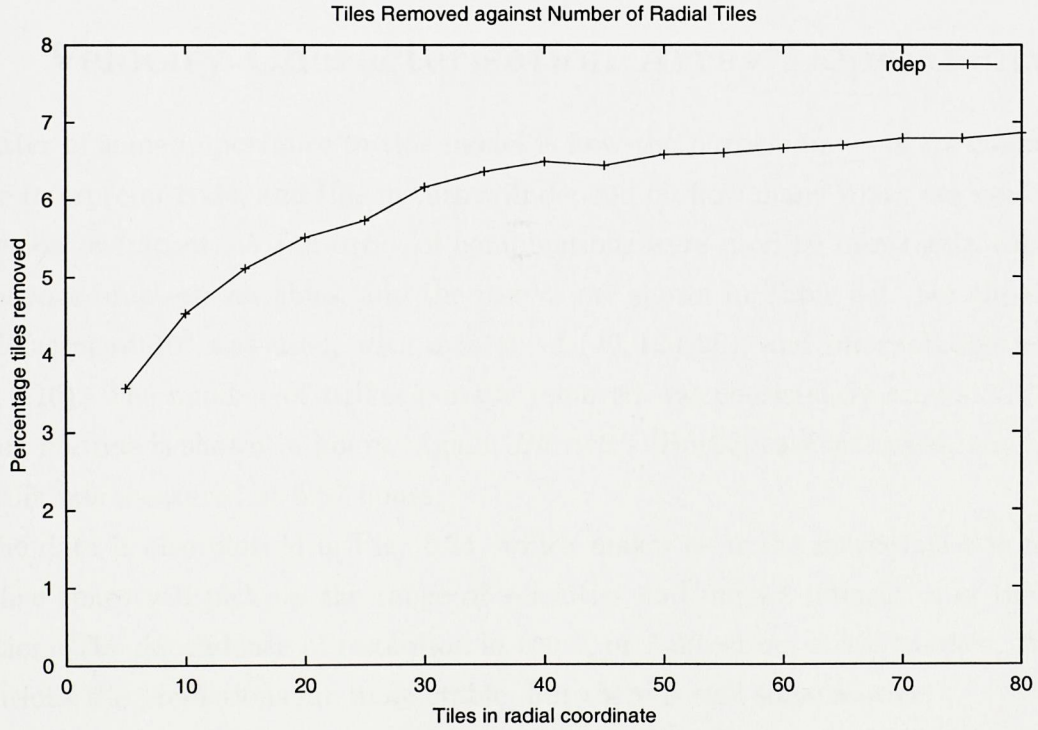


Table 5.4: Lifetime against Zonemap Resolution

I \ J	5	10	15	20	25	30	35	40	45	50
5	6.56	5.40	4.06	3.72	3.62	4.15	3.80	3.48	3.30	3.19
10	6.54	4.96	3.74	3.41	3.45	4.01	3.38	3.17	3.09	3.16
15	6.52	4.26	3.45	3.39	3.38	3.55	3.49	3.37	2.98	2.91
20	6.47	4.55	3.66	3.45	3.48	3.88	3.62	3.47	3.37	3.44
25	5.69	3.49	2.72	2.57	2.63	2.95	2.78	2.53	2.58	2.61
30	6.40	4.32	3.41	3.09	3.16	3.51	2.75	2.66	2.48	2.49
35	6.16	3.99	2.97	2.99	2.82	2.93	2.95	2.98	3.01	2.93
40	6.15	3.82	2.98	2.80	2.85	2.89	2.96	2.86	2.87	2.88
45	5.76	3.89	3.10	3.07	2.90	3.20	2.56	2.43	2.58	2.68
50	6.01	3.84	3.00	2.90	2.89	2.94	2.80	2.48	2.56	2.63

recommended number of tiles, in order to reach the stable region of prediction would have to be of the order of 50. Whilst it appears that more tiles are being removed in certain runs, it must be the case that the total mass removed is around the same, as recall that there will be numerous tiles left in the substrate at the point of burst that have lost some or most of their mass, but not enough to have been removed. It seems reasonable that with lower tile resolutions, it takes longer to erode large tiles, and hence at a simulation's end, more of the lost mass is stored in tiles only partially worn away.

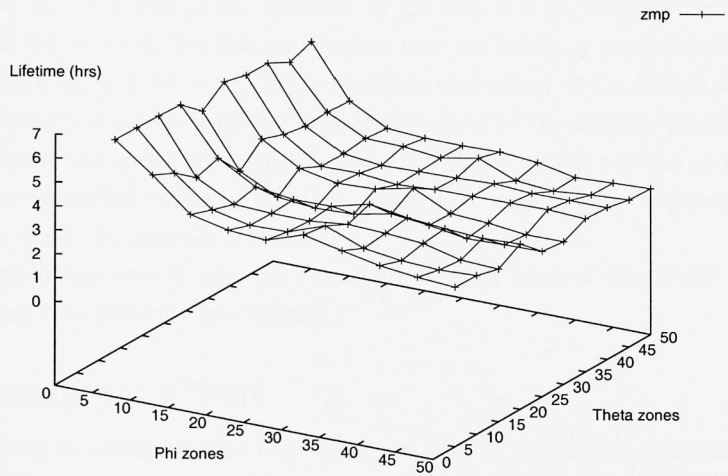
5.4 Velocity Characterisation Array Dependence

A matter of some importance to this model is how the particle impacts are characterized by the interpreter code, and this in turn will depend on how many zones are used to cover the region of impact. A full array of combinations were used to investigate the lifetime dependence on these variables, and the results are shown in Table 5.4. For these results, a scalefactor of 10^6 was used, with a mesh of $(40, 120, 20)$, and interpolation lengths of $(0.20, 0.10)$. The number of strikes per tile removed was consistently around 13, and the varying lifetime is shown in hours. Again, *Burnett's* [Bur96] case was used, and recall his bend life was measured at 5.87 hours.

The data is also plotted in Fig. 5.11, which makes clear the inaccuracies in assuming only five zones will pick up the range of velocities and impact intensities in the ϕ, or J direction. The dependence of resolution in the θ, or I direction, is less marked. At higher resolutions the predictions are more stable, but there is still some scatter.

Taking a minimum of 40 zones in each coordinate, the mean prediction is 2.66 hours,

Figure 5.11: Lifetime against Differing Zonemaps



with variance of $\pm 9\%$. Taking a minimum of 45 tiles in each, the mean lifetime is 2.61 hours, with a much smaller variance of $\pm 2.5\%$.

The recommended minimum is therefore 45 zones in each direction. Taking more would not increase computation significantly, but would require more memory.

Note that if a high zonal resolution is used to benefit from the stable region of prediction, the bendlife predictions achieved are much lower than some of those results already obtained. This indicated that the model was initially underpredicting bendlife rather more than at first thought, typically from -50% to -70%.

5.5 Interpolation Length Dependence

Recall that in order to establish an impact angle for each particle that strikes the substrate in this model, some way of generating a realistic normal to the surface is required. The technique used is to use a five-point stencil as recounted in Section 4.5.1, but remember that there is a seemingly arbitrary choice to make as to where to choose the four points around that of the impact.

Given our coordinates, it is reasonable to take them as displacements in the θ and ϕ directions respectively, but the question is, how far from the impact point should they be?

The considerations are firstly, that the points be sufficiently separated so that they fall upon different tiles in the substrate than that which the impact occurs on. This ensures that the behaviour of the 'gradient' of the tiles will be reflected. Secondly, the points must not be so distant from the impact that the resulting parabolas smooth out the local distribution of the tiles. Thirdly, because the θ extent of the domain approaches π radians, but the ϕ extent is limited to around $\frac{\pi}{4}$ radians, the different curvatures may require different interpolation lengths. Fourthly, so that the code remains as general as possible, the interpolation lengths should be stated and input in terms of a proportion of the domain size in the relevant coordinate.

The tests in this section were performed with a fixed mesh of 40x120x20, using the physical data from *Burnett's* case.[Bur96]

5.5.1 Interpolation Tests

The first thing to examine is the underlying accuracy of the surface generated. To do this, the erosion in the software is turned off, and the natural inward pointing normal to the unworn substrate or toroid, as established geometrically, is compared with the normal generated using our technique.

The natural normal is given by:

$$i = -\sin\theta \tag{5.1}$$

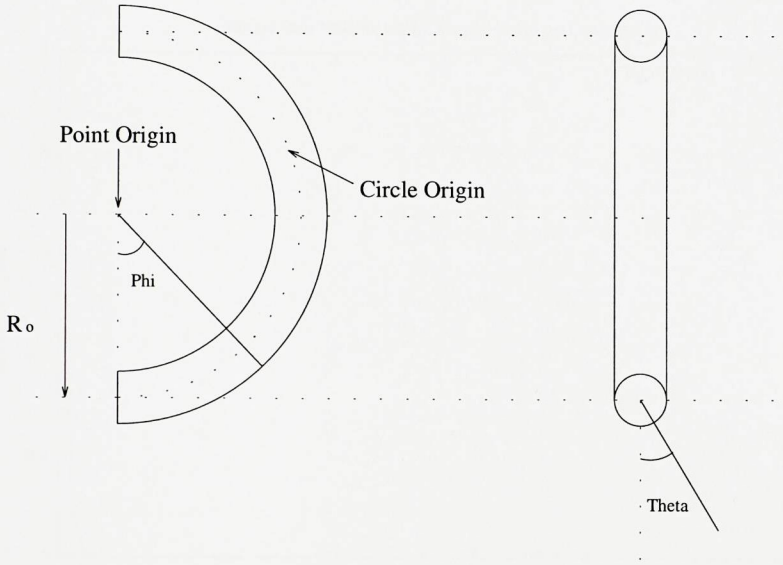
$$j = -\cos\theta\sin\phi \tag{5.2}$$

$$k = -\cos\theta\cos\phi \tag{5.3}$$

where the coordinate system is shown in Fig. 5.12.

Taking some very small interpolation lengths, i.e. those that don't extend beyond the impacted tile, it is seen in Fig. 5.13 that 98% of the generated normals are within 3^o of the natural normal. The untypical values of 20^o and 35^o were on closer inspection seen to be occurring at the boundary of the substrate, when the impact had occurred on a boundary tile. This may have been a *starting from one or zero?* bug. Certainly, interpolation points extending beyond the edge of the sustrate were being handled well enough in general (Fig. 5.14, notice the change in *abscissa* scale), and the previous anomalies did not justify the investment in time, for the peripheral effect they might have caused. Recall that conveyor bends burst close to the extrados of the bend in general. (The extrados is the line on the

Figure 5.12: Coordinate System



outside of the bend at either 3 or 9 O'clock, assuming the bend is in a horizontal plane.)

5.5.2 Lifetime Dependence-First Impacts Only

The next investigation is to look at the predicted life of the bend, whilst varying the interpolation lengths. Constructing different surfaces might result in different impact angles, and hence erosion rates, so how great is this effect?

It is quick and easy to do these comparisons given the nature of the erosion software, as only *Erode* need be rerun once the interpolation values have been altered in the input *e1* file. On a fast machine (DEC alpha 878) with still plausible parameter settings, each run can be performed in around a minute. The scalefactor used was $1.2\text{x}10^6$.

Looking at the results in Table 5.5 it is seen that there is hardly any dependence on the variation in the θ interpolant. This is understandable, as the main velocity component in the incoming vectors moves along the $+\phi$ direction, and consequently the ϕ dependence is more visible. Changing the angle along this coordinate would have a more direct effect on the impact angle, and hence the erosion rate, and lifetimes are seen to vary by a maximum of 6.8% at the higher θ values.

The step-function nature of these predictions cannot be explained by the relative coar-

Figure 5.13: Difference Between Normals:Very Short Interp. Lengths

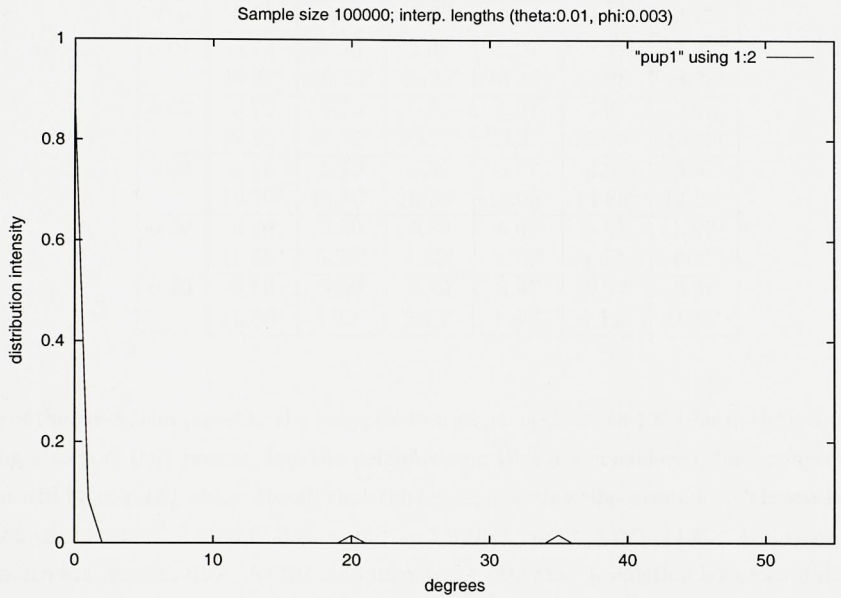


Figure 5.14: Difference Between Normals:Very Long Interp. Lengths

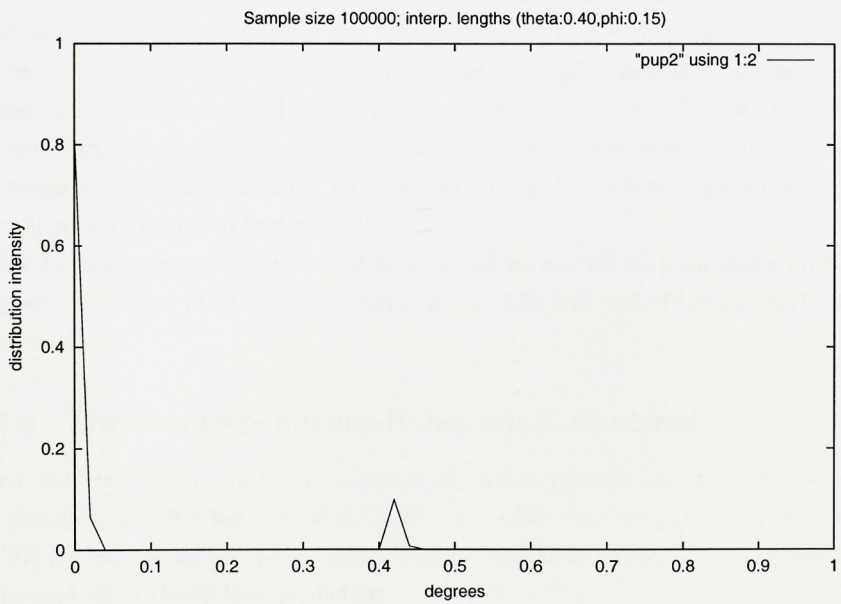


Table 5.5: Lifetime (hrs) with Max. Deviation from Natural Normal

$\theta \backslash \phi$	0.01	0.03	0.05	0.07	0.10	0.15
0.01	5.70 48.41^o	5.70 48.29^o	5.89 48.28^o	5.97 48.28^o	5.97 48.28^o	5.97 48.28^o
0.02	5.70 29.98^o	5.70 29.31^o	5.89 29.27^o	5.97 29.27^o	5.97 29.26^o	5.97 29.26^o
0.04	5.70 16.70^o	5.70 14.40^o	5.89 15.36^o	5.97 15.03^o	5.97 14.85^o	5.97 14.85^o
0.20	5.70 16.59^o	5.59 5.29^o	5.89 4.85^o	5.97 4.73^o	5.97 4.69^o	5.97 4.66^o
0.40	5.70 16.59^o	5.59 5.32^o	5.89 2.79^o	5.97 1.65^o	5.97 1.11^o	5.97 0.98^o

sity of the mesh, compared to the interpolation steps used. With 120 tiles in the ϕ direction, using a step of 0.01 means that the neighbouring tiles are considered, but many of those used will be counted twice. Recall that the average of nine tiles around the chosen interpolation tile is taken. A step of 0.03 would be 3.6 tiles from the impact tile, and there would be much less overlap now. As the step increases further the prediction becomes stable, and this is the behaviour required of a mathematical model.

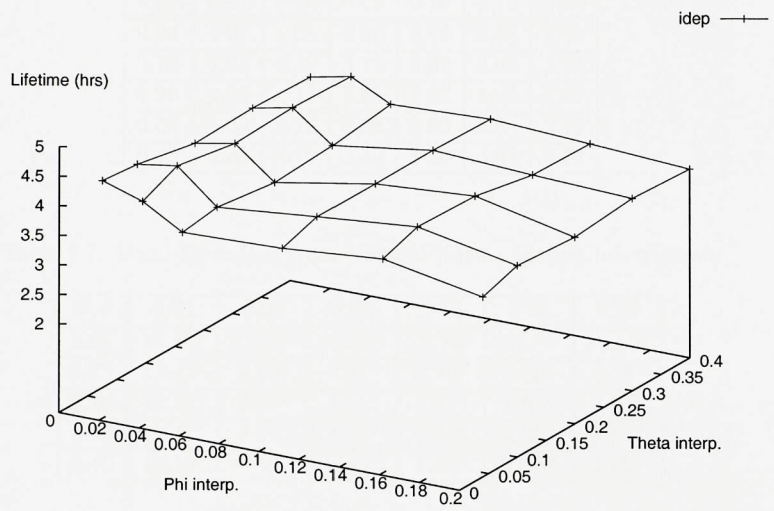
When looking at the deviations from the natural normals in Table 5.5, one must bear in mind that these maximum values may be unrepresentative, as witnessed in Section 5.5.1. However, the trend is clear: using short interpolation lengths, some steep tile gradients are picked up, and the contours of the worn substrate can be accounted for with regard to the erosion effect of later impacting particles; using longer interpolation lengths, the contours of erosion are smoothed, and the angle of impact will be a closer approximation to the natural normal defined in Section 5.5.1.

So, to pick up a certain amount of contour, whilst benefitting from stable predictions, suggested values for (θ, ϕ) might be (0.20,0.10), or 20% and 10% of the size of the domain respectively.

5.5.3 Lifetime Dependence-Rebounds Considered

When modelling second or higher numbers of particle impacts, the interpolation lengths are also going to affect the way particles rebound. After retracking of these particles using a CFD engine, the second wave impact intensity would be expected to vary, and this in turn could affect the lifetime prediction.

Figure 5.15: Lifetime for Differing Interpolants



Some predictions are shown in Table 5.6. The values chosen for the interpolations are in ranges that broadly satisfy the criteria discussed above. The scalefactor used here was $1.5\text{x}10^6$. The rebounds were stored from an eroding surface, and the rebound sample used was from the period of 0% to around 40% wear. The data is also represented in Fig. 5.15.

Comments are that once again the ϕ dependence seems more critical. Again the more stable region of prediction occurs at the higher ends of the scales, but this is not directly attributable to the statistics of the model, as it so often is elsewhere in this chapter. In this case, as the interpolation lengths increase, the tangential parabolas become flatter, and the stability of the predictions presumably stems from the fact that a flat curve can only become flatter to a limited extent. Hence, to pick up the evolution of the substrate in the erosion algorithms, the interpolants should not be too long, and the values of $(0.20, 0.10)$ suggested in the previous section might well be, as these appear to be in stable regions of prediction.

Recall also the concept of the *natural normal* defined in Section 5.5.1. The maximum deviation from this is plotted in Table 5.7 and Figure 5.16, and clearly some of the larger interpolants are reaching the point of flatness beyond which the parameters are not sensitive. In order to pick up the gradients of the tiles, the author recommends the use of

Table 5.6: Lifetime (hrs) against Varying Interpolations

$\theta\backslash\phi$	0.01	0.03	0.05	0.10	0.15	0.20
0.04	4.27	4.05	3.66	3.72	3.87	3.55
0.10	4.21	4.32	3.75	3.92	4.08	3.75
0.20	4.01	4.14	3.61	3.92	4.04	3.67
0.30	4.05	4.19	3.68	3.93	3.84	3.77
0.40	4.02	4.16	3.82	3.90	3.81	3.71

Table 5.7: Max. Deviation from Natural Norm. against Interpolants

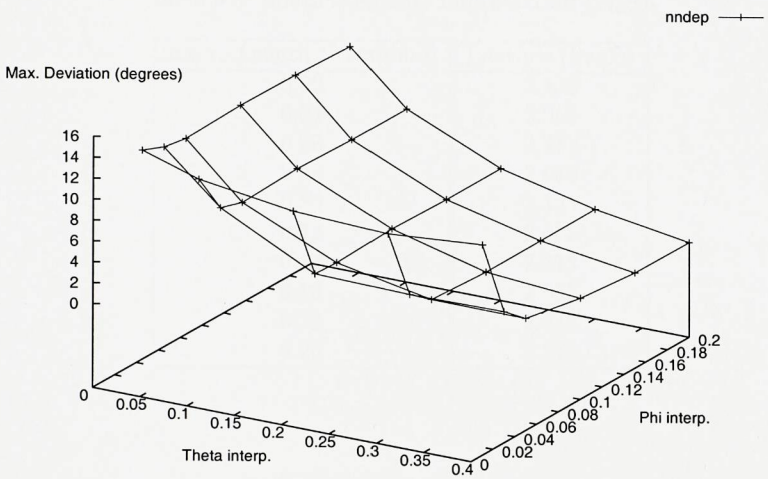
$\theta\backslash\phi$	0.01	0.03	0.05	0.10	0.15	0.20
0.04	14.80^o	13.92^o	13.55^o	13.78^o	13.72^o	13.50^o
0.10	13.11^o	9.19^o	8.51^o	8.78^o	8.60^o	8.58^o
0.20	11.85^o	4.66^o	4.55^o	4.82^o	4.65^o	4.65^o
0.30	11.47^o	4.47^o	2.79^o	2.44^o	2.47^o	2.51^o
0.40	12.19^o	4.58^o	2.75^o	1.69^o	1.15^o	1.09^o

maximum interpolation values of 0.10 for theta and 0.04 for phi, as these values are in the region where there is still some sensitivity to the interpolations, but not as much as could be achieved. A maximum deviation from the natural normal of around 9^o does not seem excessive even for a thin pipe-wall, when in Section 5.5.1 it was shown that there may be up to 3^o of error in the constructed normal vector anyway.

Whichever values are recommended, the inbuilt error has to be quoted as $\pm 10\%$, so given that this model was constructed with the aim of modelling the evolution of the substrate, it is felt that the full resolution of the tile array should be utilised. Hence the interpolation points should be taken only one or two tiles away from the impact point. This general rule can be used to calculate interpolation lengths according to tile resolution, and for a mesh of (90,220) would provide values of around (0.02,0.01).

For the modelled case, recall that the pipe-wall is very thin (2.39mm) compared to the radius of the bend (750mm). As the model must be general enough to cope with much thicker walls, e.g. those made from cast Basalt, the interpolation lengths should be kept sufficiently large to avoid excess sensitivity, which might occur with the steep and deep cavities likely to be experienced in this material.

Figure 5.16: Max. Deviation from Natural Norm. against Interpolants



5.5.4 Higher Resolution Runs

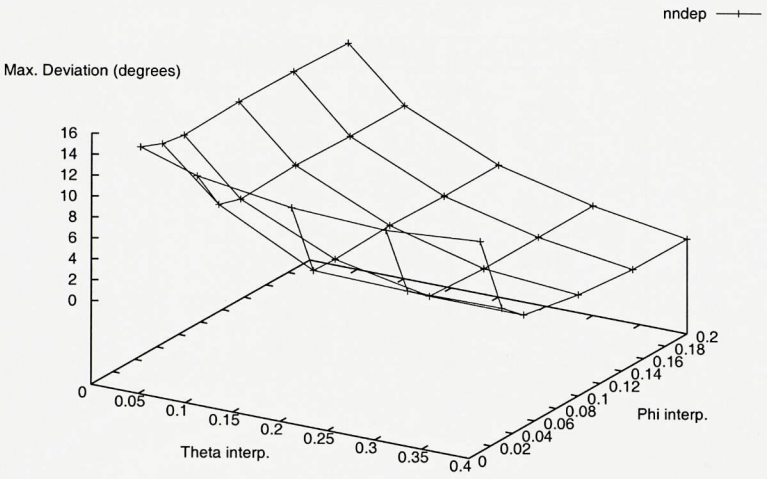
The error quoted above of $\pm 10\%$ would be the single largest contribution to the stability of the erosion model, so it was decided to investigate it further with a higher resolution mesh. This made a significant difference, and the results are shown in Figure 5.17 and Table 5.8. These results were obtained using the recommended parameter settings or ranges, as established earlier on throughout this section.

In principle each of the parameters recounted in this chapter could now be revisited, and the resulting error estimates perhaps whittled down after an iterative fashion. The interpolation lengths however were the priority, as the initial results showed such large variance.

There are ten images associated with the obtained ten different batches of second wave impact data, but these are not shown as they are identical to the eye. Regarding prediction of bendlife, the mean value is 2.14 hours $\pm$ 8.1%. Using only interpolation lengths of 0.12 or less however gives a mean value of 2.09 hours $\pm$ 3.5%.

Note that the (I, J) interpolation lengths were both set to the same value in these tests, to save time by investigating only a main diagonal of the whole matrix. With regard to the previous section's discussion, it is recommended to use interpolation lengths of $(0.02, 0.02)$

Figure 5.16: Max. Deviation from Natural Norm. against Interpolants



5.5.4 Higher Resolution Runs

The error quoted above of $\pm 10\%$ would be the single largest contribution to the stability of the erosion model, so it was decided to investigate it further with a higher resolution mesh. This made a significant difference, and the results are shown in Figure 5.17 and Table 5.8. These results were obtained using the recommended parameter settings or ranges, as established earlier on throughout this section.

In principle each of the parameters recounted in this chapter could now be revisited, and the resulting error estimates perhaps whittled down after an iterative fashion. The interpolation lengths however were the priority, as the initial results showed such large variance.

There are ten images associated with the obtained ten different batches of second wave impact data, but these are not shown as they are identical to the eye. Regarding prediction of bendlife, the mean value is 2.14 hours $\pm$ 8.1%. Using only interpolation lengths of 0.12 or less however gives a mean value of 2.09 hours $\pm$ 3.5%.

Note that the (I, J) interpolation lengths were both set to the same value in these tests, to save time by investigating only a main diagonal of the whole matrix. With regard to the previous section's discussion, it is recommended to use interpolation lengths of $(0.02, 0.02)$

Table 5.8: Lifetime against Interpolation Length

Interp.Length (fractional)	Lifetime (hours)
0.02	2.132
0.04	2.068
0.06	2.117
0.08	2.018
0.10	2.125
0.12	2.082
0.14	2.313
0.16	2.063
0.18	2.212
0.20	2.273

Figure 5.17: Lifetime against Interpolation Length

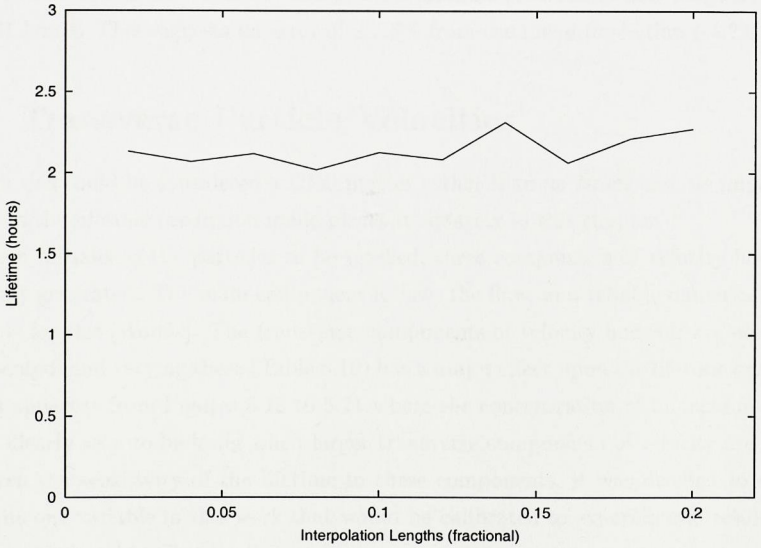


Table 5.9: Random Seed with Lifetime (hours)

Seed	Lifetime
2	4.242
3	4.162
5	4.301
7	4.128
11	4.268
13	4.257

with quotable error of $\pm$ 3.5%.

5.6 Random Number Seeding

The random number generator used in the model is built into the C programming language. Used in its default state, the generator will produce the same set of (random) numbers every time a program is run, and this is useful for debugging. There is however a facility to seed these random numbers such that a different set is produced each time. Table 5.9 shows lifetime predictions produced by *Erode* for different seeds. These seeds were taken to be the first few prime numbers, and produce lifetime predictions that range from 4.128 to 4.301 hours. This suggests an error of $\pm$2.3% from the mean prediction (4.226 hrs).

5.7 Transverse Particle Velocities

Though this could be considered a *CFD* matter rather than an *Erode* one, its import with regard to the lifetime prediction made places it squarely in this chapter.

When initialising the particles to be tracked, three components of velocity have to be randomly generated. The main component follows the flow, and reliable numerical data is available for this [Woo92]. The transverse components of velocity however are not so well documented, and varying these (Table 5.10) has a major effect upon the lifetime prediction. This is apparent from Figures 5.18 to 5.21 where the concentration of impacts in a single area is clearly seen to be lower when larger transverse components of velocity are used.

Given the sensitivity of the lifetime to these components, it was decided to consider them the one variable in this work that would be calibrated to experimental results. This work can be found in Chapter 7.

Table 5.10: Lifetime against Transverse Particle Velocity

S.D of ratio as %	Lifetime (hrs)	% err	Total Mass Loss (g)
0.97	1.981	-62	69.51
3.88	3.214	-45	102.77
7.76	5.085	-13	168.03
8.73	5.746	-2	185.38

Figure 5.18: Tracks: S.D = 0.97 %

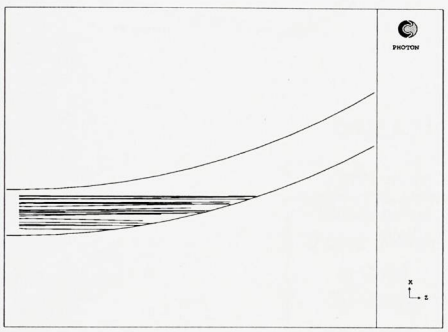
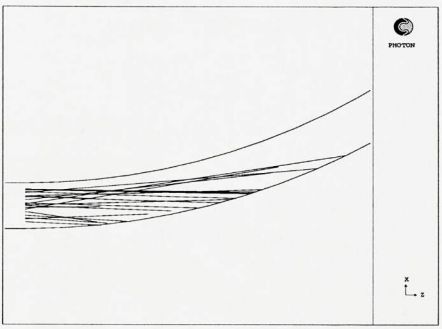


Figure 5.19: Tracks: S.D. = 7.76 %



The transverse velocity components will be described in terms of the standard deviation of a sample of 300 particles, of the ratio of the transverse components to the main component. Circular beam randomisation will be assumed, so that a single parameter can control the particle spread.

5.8 Parameter Summary: Inherent Error Estimates

The inherent errors discussed in this Chapter are an assessment of the reproduceable nature of the predictive tool *Erode*. They indicate how much variance can occur in *Erode's* predictions, if any of the internal parameters are adjusted while the same prediction problem is solved. The parameters considered and the error bounds established are shown in Table 5.11. The total given results from the addition of all of these in quadrature, as they are independent sources of error.

This total error of $\pm 6.7\%$ represents the likely stability of any given prediction. It represents fundamental inaccuracies due to the modelling techniques used, and it says nothing about how accurate predictions may be when compared to lifetime measurements from a real pneumatic conveyor. These other errors will have to be added in also when

Figure 5.20: 1st Impacts: S.D.=0.97%

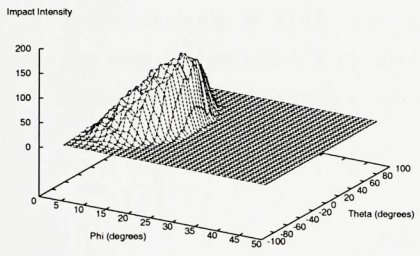


Figure 5.21: 1st Impacts: S.D.=7.76%

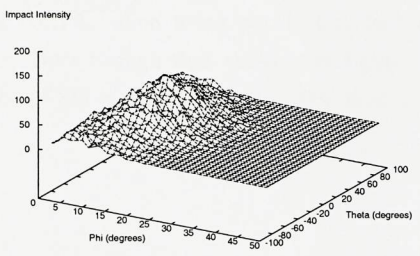


Table 5.11: Error Collation

Parameter	Error Estimate
Scalefactor	$\pm$ 2.5%
θ and ϕ Grid	$\pm$ 2.2%
r Grid	$\pm$ 0.9%
Zonemap	$\pm$ 2.5%
Interpolation	$\pm$ 3.5%
Random Seed	$\pm$ 2.3%
Erosion Tester	$\pm$ 3.0%
Total	$\pm$ 6.7%

established. For a discussion of this topic, which should be of considerable interest to the reader, please see Chapter 7.

5.9 Chapter Summary

In this chapter the behaviour of the virtual rig *Erode* is investigated, by a process of experimentation with each of the parameters in turn.

The erosion model works on simple physical principles and has been seen to behave as rational thought might predict. The lack of serious surprises indicates that *Erode* has potential to be a stable and plausible prediction model, perhaps a *virtual pneumatic conveyor.*

One difficulty described herein was the extreme bendlife dependence noted on the initialised transverse velocity components of the particles used in the CFD stage of this model.

This lifetime variation dwarfed any such stemming from the internal parameters of *Erode*. The decision was therefore taken to calibrate these initial transverse velocity components to experimental data, as described in Chapter 7.

The overall stability of *Erode* was quoted as $\pm 6.7\%$, when using parameters in the appropriate ranges as established in this Chapter. Note though that the quoted figure is not representative of *Erode's* ability to predict bend life in the real world. For this, see Chapter 7.

Chapter 6

Modelling Further Impacts

The erosion prediction model *Erode* was designed to have the capability to model rather more than the obvious first impact of the conveyed particles as they reach a bend in a pipe. A description of how this was achieved follows.

6.1 Motivation

Due to the work of *Burnett* [Bur96] it has been an assumption throughout this work that the point of greatest wear and burst in a pneumatic conveyor bend does not necessarily arise in the region of first impact. It is suspected that bursts that occur further around the bend than the projected line of flight from the incoming straight, are more likely to occur in long-radius bends, where the initial impacts are at relatively low angles. For shorter radius bends, the first impacts are at higher angles, and perhaps due to a greater loss of energy in collision, the burst point usually arises at around the line of sight.

In this work it will be attempted to model a series of impacts, such that the possibility of burst points beyond the line of sight is allowed. With regard to predicting the life of the bend, if the wear due to second wave impacts is severe and it is not modelled, the predicted life could be overestimated.

6.2 Numerical Technique for Multiple Impacts

The reader should be familiar with the *Erode* model described in Chapter 4 to get the most from this section. Herein are described extensions to the basic concepts of using CFD to obtain particle impact data, and then using this data in custom software to wear away at a virtual pipe wall.

6.2.1 Initial Rebound Algorithm

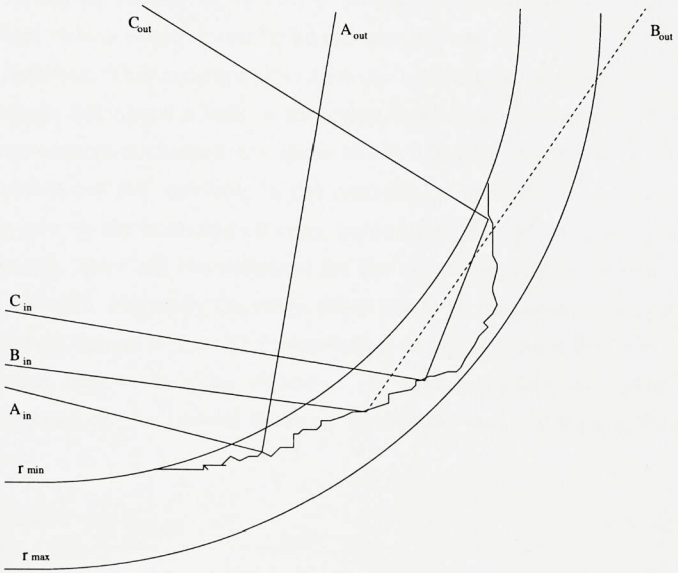
The erosion code was designed with rebound capability in mind, so that obtaining the positions and velocities of rebounding particles seemed on the surface straightforward.

The interface, or dividing line between the CFD and custom parts of this modelling technique, is the surface $r = r_{min}$, as shown in Figure 6.1, and this is the inside surface of the pipe wall, before any wear has occurred. Particles to be tracked with PHOENICS and GENTRA must be started on or inside this surface, as otherwise they are initialised outside the BFC (Body-Fitted Coordinate) CFD mesh, and will be rejected. Hence, a particle which rebounds inside the worn *Erode* substrate has to be projected back to this surface before its details are worth storing.

The initial algorithm to do this simply solved for where the vector struck this surface. Soon after, a pair of related problems arose with this. These can both be seen in Figure 6.1. While this early algorithm worked for many particles, as the wear proceeded this became less likely. Rebound angles were generally lower than might have been expected, and so the phenomena of line B results in many lost particles; there is no solution to the equations if the line never meets the surface $r = r_{min}$. The other problem, happily ignored as a first approximation, is that even if the vector can be solved as meeting the surface, it may have passed through a solid tile to reach this point. Given that an investigation into surface and rebound evolution was desired, the only solution to obtaining realistic rebounds at the later stages of wear was to implement a multi-impact capability into *Erode*, such that particle vectors such as line C could be realistically accounted for.

Here, the possibility of a fluid recirculation in the flow in the physical system is raised, in those regions of deeply worn pipe wall. Certainly it would not be expected in regions of shallow wear any more than it would be expected at any point on the inside pipe wall, but materials such as basalt do allow for miniture caverns to form under erosion. It must be highly likely that in extreme cases these offshoots from the main flow can play host to recirculation. However with no known experimental observations of this, or how it could affect particle velocities and hence erosion, the phenomenon is not accounted for in this general purpose predictive model. It is certainly true that at pneumatic conveying velocities, particles would spend only the briefest time in traversing such small spaces, and their capacity for being accelerated would be minimal.

Figure 6.1: Example Trajectories on Eroded Surface



6.2.2 *Erode* Multi-Impact Capability

One of the attractions of the initial algorithm described above was that it was computationally cheap; particles in the updated algorithm have to be tracked perhaps twice as far through the tile substrate as they did previously, and hence program execution has slowed, perhaps by 80%.

Implementing multiple impacts was not too time-consuming, as the modular nature of the code allowed many routines to be reused.

The most complicated aspect was how to handle the first step of a general tracking routine, one of which already existed for the *incoming* particles. The difficulty arose because for incoming particles, it is always known that the start point of the particle is on the top face of a tile (on the inside pipe-wall). At a general point of rebound, it could be on any of six. Also, will the rebound velocity vector pass through this present tile, or does it immediately leave it and enter another?...

The solution to this was to deal with the first tile as a special case. If the entry face was the top, then it was assumed that the rebound would bounce off this tile rather than enter it. This is correct about 99 times out of 100, and facilitated the following: the value

of the *EntryFace* variable was set to zero, corresponding to *no* face, as opposed to one of the usual *north, south, east, west, top,* or *bottom.* This allowed the entry point on the tile (for the incoming vector) to be also if necessary the exit point on the tile (for the outgoing vector), which would normally be inhibited by the algorithm, by its checking that *EntryFace*$\neq$*ExitFace.* This measure allows for the case where the rebound bounces off the tile it just struck, but opens a leak in the other case where the vector changes direction somewhat, but continues through the same tile it has just struck. In this case, the exit position algorithm can fail, resulting in the rebound being aborted.

Returning now to the bouncing off case, because the *EntryFace* is not specified for the rebounding vector, there are two solutions for the algorithm; the entry point, and the exit point actually sought. Normally the entry point would be discarded as a possibility by the algorithm, giving a unique result. Here, depending on which it finds first, the particle might be sent travelling backwards along its vector. A solution to this was found by using the other test employed by the general tracking algorithm: that having converted the points to Cartesians,

$$\frac{y_{Exit} - y_{Entry}}{v} >= 0 \tag{6.1}$$

to ensure the particle is not projected backwards. Note that u or w velocity components could equally as well be used here with their corresponding coordinates.

This condition is not enough alone, though it does look as though it will suffice. If the entry point *is* the exit point, and the particle is effectively retracing its steps, the LHS of Equation 6.1 will be zero. What was initially happening though (it was discovered) was that the value used was not zero, due to machine precision; typically

$$\pm 1.23456e - 211 \tag{6.2}$$

or similar would be used, and in 50% of cases, with *negative* random noise (for +ve v), this test would fail. A solution to this problem could not involve moduli, as the sign of the factor was what was of import. Instead, a small decrement was made to zero before the test was performed, e.g.

$$\frac{y_{Exit} - y_{Entry}}{v} > -0.000001 \tag{6.3}$$

This would have negligible effect on the general tracker, and backward vectors would still be caught. It would also allow the entry and exit points to be the same in our special circumstance.

Table 6.1: Bounce Breakdown

No. of Bounces	Freq. $R/d = 14.15$	%	Freq. $R/d = 1.5$	%
1	138868	93.00	341619	91.59
2	5676	3.80	19554	5.24
3	4070	2.73	7063	1.89
4	613	0.41	2538	0.68
5	76	0.05	1094	0.29
6	6	0.00	537	0.14
7	5	0.00	592	0.16

Another problem with multiple impacts was that on perhaps 10% of occasions, the normal generated for a second impact would result in an impact angle of less than zero. This is due to the fact that the second impacts typically occur at much lower angles than the first impacts, and that there is around ± 3 degrees of variance in the generated normals. It was decided that the normal had to be changed under these circumstances, hence a routine was produced to rotate this by around 3 degrees towards the flow. This routine is called repeatedly in the code, (up to six times if necessary) until a positive impact angle is produced. The rebound algorithm is then entered in the usual fashion.

An idea of how many of the impacts involve multiple collisions is given in Table 6.1. These numbers are for R/d ratios of 14.15 and 1.5, to show the variation. Note that in the *seven bounces* row are also all the particles that would have bounced eight times *or more*, if they were not terminated at this point. It is unclear why different R/d ratios should produce differing rates of multiple bounces.

Note now that many of these particles have bounced twice or more, and so to call the next set of impacts 'second impacts' is erroneous. The term 'second wave' will be preferred.

6.2.3 Handling Multiple Sets of Impact Data

With the above code in place, obtaining files of rebound data in the desired form becomes straightforward. Software is in place in *Erode* to dump batches of rebound data at any stage of the erosion, up until burst.

Particle data can be dumped directly in the form required by PHOENICS and GENTRA to track particles. The tracking is straightforward to carry out, restarting the previously solved and stored flow, with the second wave of particles seemingly coming out of the wall of the pipe-bend, where *Erode* had released them at $r = r_{min}$. These particles are tracked until they hit the pipe-wall again, and the relevant impact data stored just as with

the first impacts.

Both files of impact data now have to be characterized. The conclusion drawn here was that they should be used together only to establish the extent of the domain to be modelled. When this is established, two different zonal maps are set up, but with the same boundaries, and they are loaded independently with impact intensities and velocities, as described in Section 3.5. The software required for this is a modification of the original filter, where two impact data files are read instead of one. Code was reused from the earlier single-impact file filter to ease the development of this.

Similarly, in order to read in both of these zonemaps, *Erode* itself had to be updated, and no serious problems were encountered. The idea is now to generate, in alternating fashion for each nominal particle, a first impact followed by a second wave impact. This should truthfully represent the number of impacts a single particle makes as it bounces around the bend. Note that the second wave impact generated will be unrelated to the preceding first impact, unlike in the real system, but over large statistical samples, this technique should reproduce real world behaviour.

6.3 Surface Evolution

In this section the aspects of surface evolution specifically related to rebounds and second wave impacts are examined. For more general erosion results, see Chapter 7.

6.3.1 Specular Rebound Tests

First the rebound capability of *Erode* was tested by turning off all manipulation, and dumping only specular rebounds. These can then be compared to impact data obtained using PHOENICS and GENTRA, which also uses specular rebounds by default. Note that the *Erode* rebounds were modelled from an unworn, pristine virtual pipe bend, and that they were then retracked in PHOENICS to obtain further impact data.

There are two main differences as seen in Figures 6.2 and 6.3. The first is that the *Erode* data is less smeared (the sample sizes were the same). The second is that there are some isolated patches of impact data predicted by GENTRA which are not produced by *Erode*. These isolated patches are a little surprising, and the difference cannot be explained by the author. The two sets of data are broadly the same however.

It is not known how the proprietary code GENTRA works, but it is possible that the surfaces used for rebounds are interpolated from the CFD BFC mesh. This would be a

Figure 6.2: Specular: PHOENICS

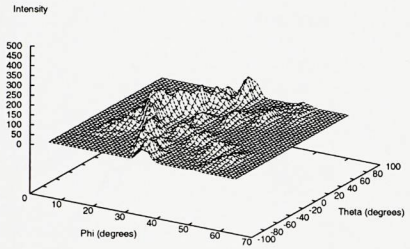
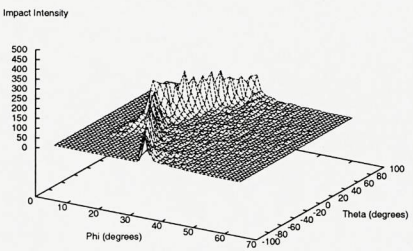


Figure 6.3: Specular: *Erode*



different technique from that used in *Erode*, which uses a true toroid for the pipe wall.

6.3.2 First Wave Impact and Rebound Angles

Looking for evidence of surface evolution, the first things to examine are the impact and rebound angles of the first impacts. Shown in Figures 6.4 and 6.5 are distributions of particle impact and rebound angle for the same in *Burnett's* [Bur96] (*R/d=14.15*) case. As would be expected from the rebound algorithms used, the rebound angles are lower than the incoming angles. Batches of six impacts and rebounds were collated using *Erode*, at increasing stages of the virtual wear process. Only two of the twelve images are shown, as there was no visible change in these distributions.

A much different geometry was also investigated, to look for any variation. Using a Basalt bend of the same bore but with *R/d=4.0* and pipe wall thickness of 12.0 mm, Figures 6.6 and 6.7 show some corresponding distributions of first impact angle. There is slightly more evolution in this series, so the first and last of the six distributions are shown. The impact angle spreads over a wider range, and changes shape appreciably.

Two of the rebound distributions are also shown in Figures 6.8 and 6.9. Again there is a tendency for the rebound angle to spread over a wider range as the wear proceeds, and the mode (peak) moves down from 10^o to 8^o.

It would be expected that this fatter pipe bend should exhibit more in the way of evolutionary traits, due to the much thicker wall and the potential for deeper cavities to form.

Figure 6.4: First Impact Angles

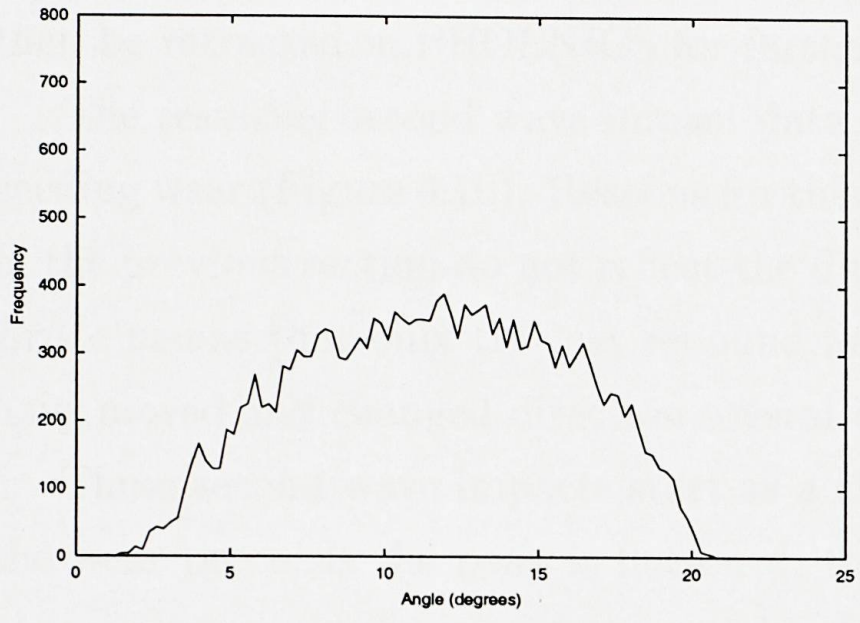


Figure 6.5: First Rebound Angles

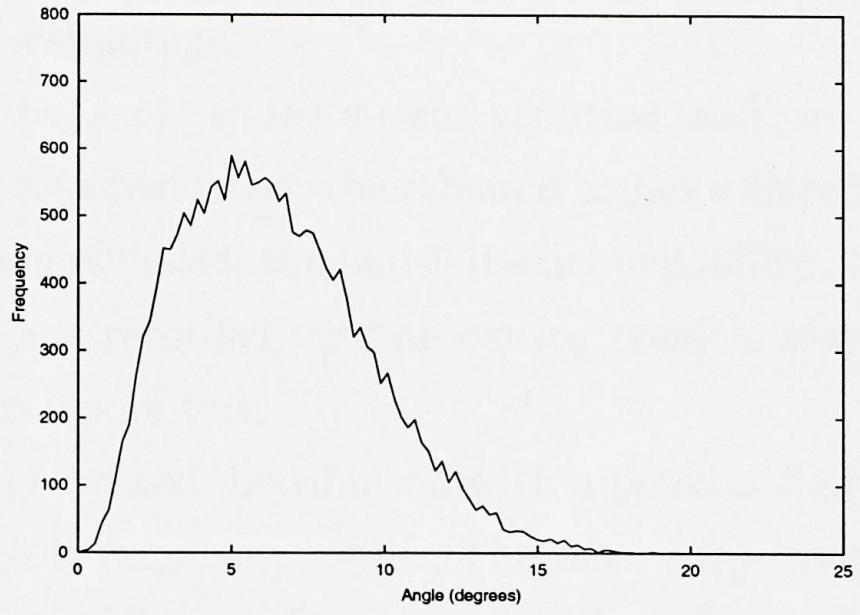


Figure 6.6: R/d=4.0: Imp.Batch a

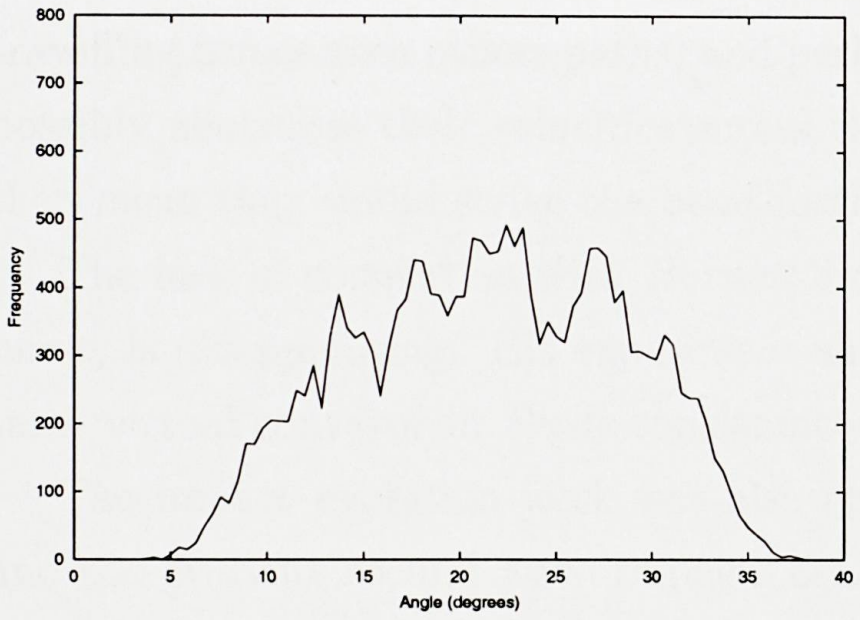


Figure 6.7: R/d=4.0: Imp.Batch f

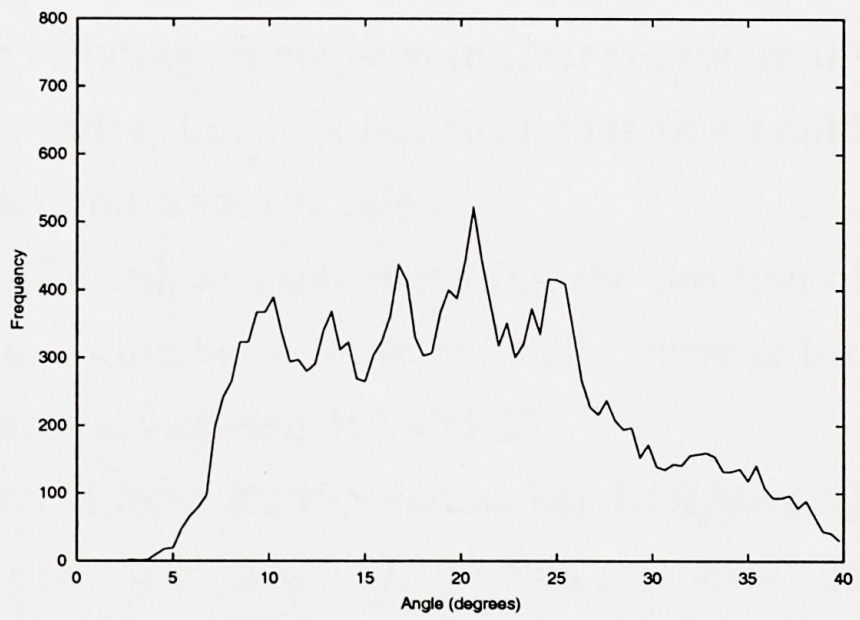


Figure 6.8: R/d=4.0: Reb.Batch a

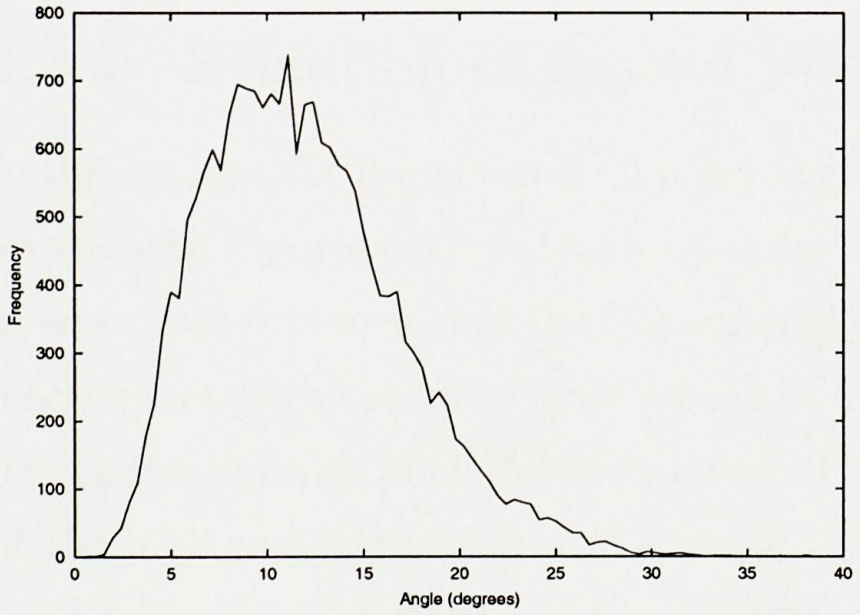
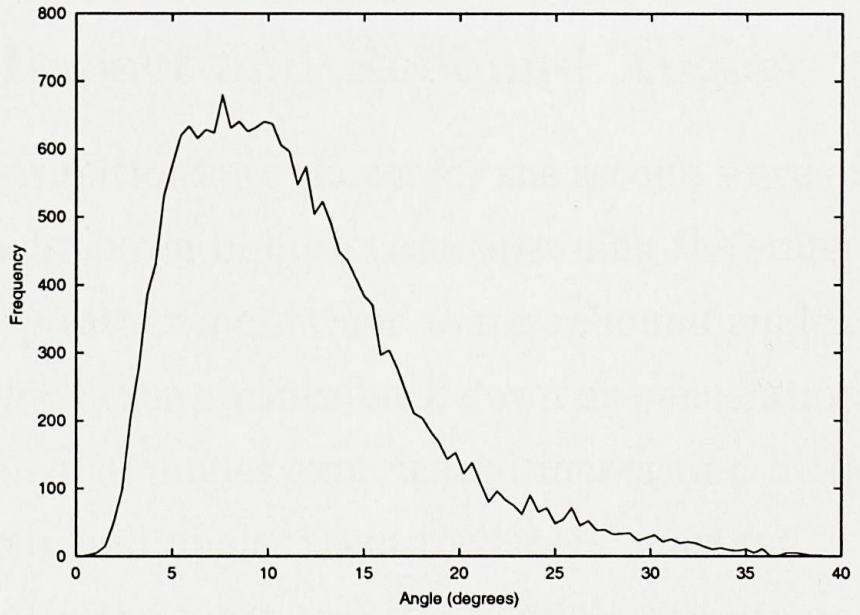


Figure 6.9: R/d=4.0: Reb.Batch f



6.3.3 Evolution of Second Wave Impact Intensities

Dividing the rebounding particles from any given run into six batches as above, they can then be retracked in PHOENICS for further investigation.

The resulting second wave impact data (R/d=14.15) shows a clear variation with proceeding wear (Figure 6.10). Reasons for this are assumed to be: the rebound angles collated in the previous section do not reflect the direction of travel; the multi–impact capability of *Erode* means that only the last rebound angle was recorded, and an exiting particle may have moved and changed direction several times before this.

These second wave impacts start as a cleanly defined distribution with a peak, and as the wear proceeds the peak is flattened, though it remains present. The most surprising aspect perhaps is the way that particles are sent well away from the extrados. The lack of any focussing effect that might cause a burst at around 28^o may suggest that an inter–particulate effect is required here. For particles to bounce so far from the extrados, they must be crossing from the other side of the pipe. This means that some particles will be travelling across each others paths, and perhaps colliding. A collision in their course would possibly neutralise their velocities across the extrados, but it is not clear that this would then mean they would strike the bend further around from the inlet.

The lack of correlation with *Burnett's* case [Bur96], at least regarding the position of burst, is disappointing. His experiment showed a pipe bend burst at 28.5^o, whereas the same virtual conveyor in *Erode* consistently bursts at between 19^o and 22^o.

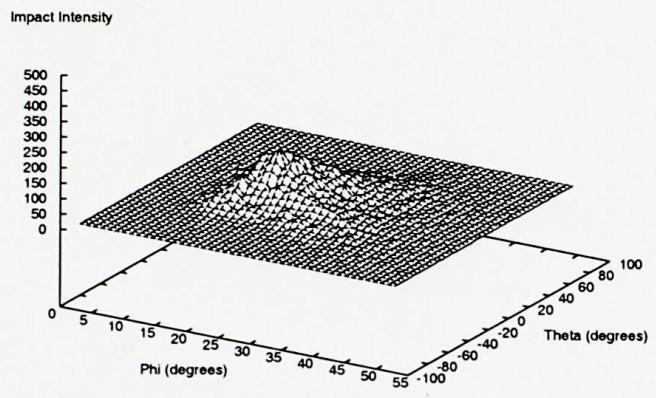
The impact–evolution work was also performed for a shorter radius bend (R/d=4.0) and the evolving second wave impacts can be seen in Figure 6.11. In this geometry the initial peak is also blunted as the wear proceeds, though perhaps not as much. Here, large numbers of particles do not get so far from the extrados, perhaps because they have a shorter distance to travel before striking the bend again.

6.3.4 Evolution of Second Wave Impact and Rebound Angles

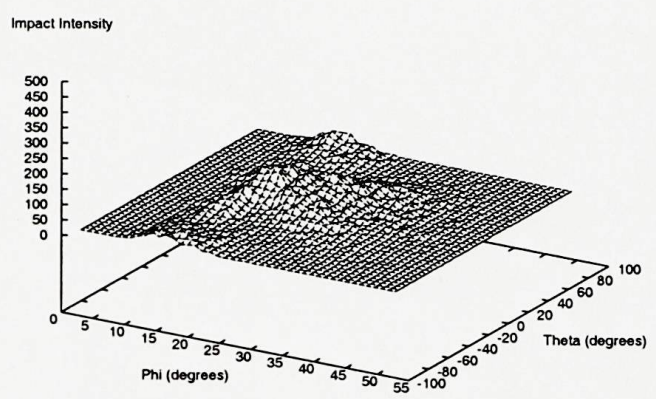
In this section the impact and rebound angle distributions are shown for the second wave of impacts, in Figures 6.12 and 6.13. A series of six is shown in both cases spanning the states of wear, and it is seen that the impact angles vary rather more than do the rebound angles. The impact angles seem to grow a peak at 9^o, which then shrinks back down as penetration passes the halfway point towards burst. The range of angles experienced increases a little at the high end, perhaps from 12 to 13^o. The rebound angles seem to remain constant.

These angle distributions were obtained under the most realistic possible conditions,

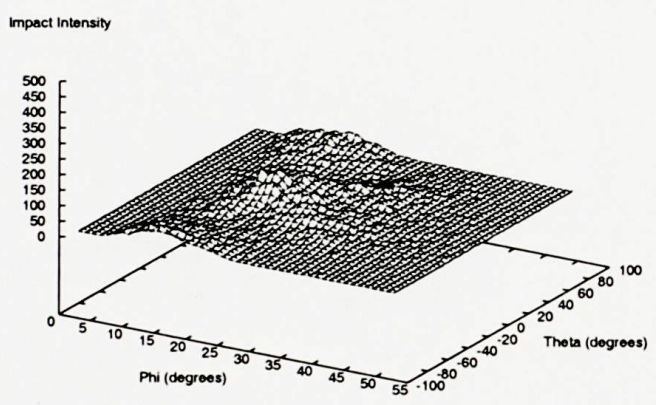
Figure 6.10: Evolving Impact Intensities: R/d = 14.15



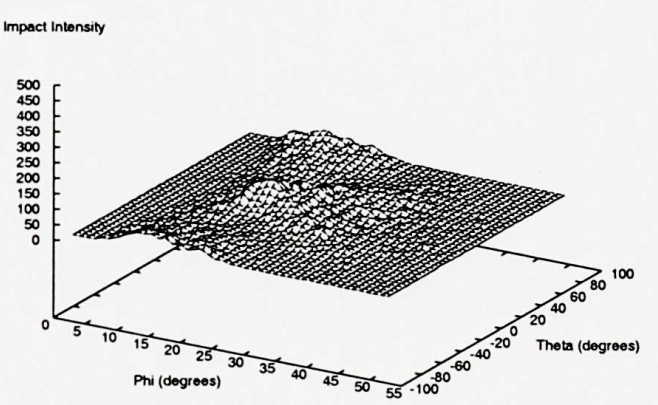
(a) First Batch



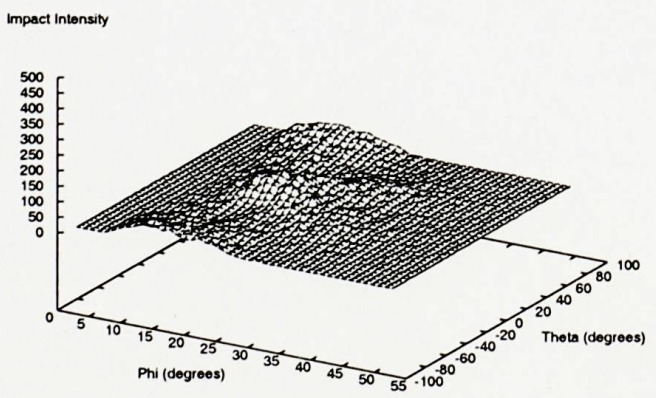
(b) Second Batch



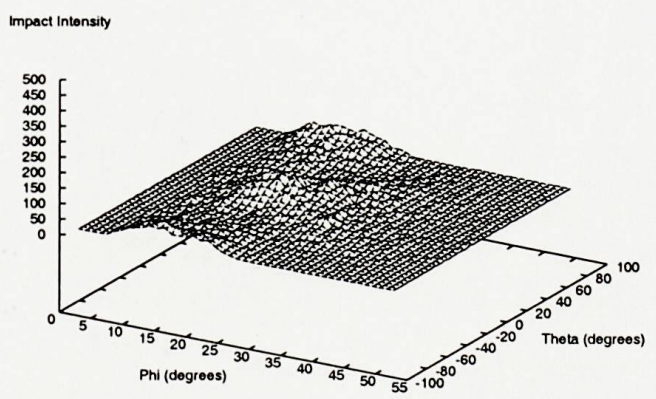
(c) Third Batch



(d) Fourth Batch

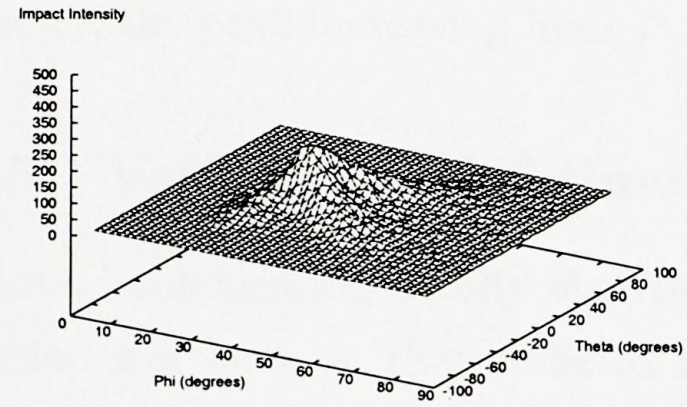


(e) Fifth Batch

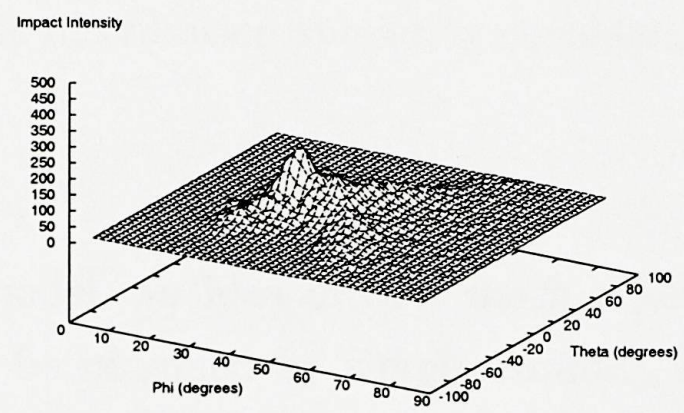


(f) Sixth Batch

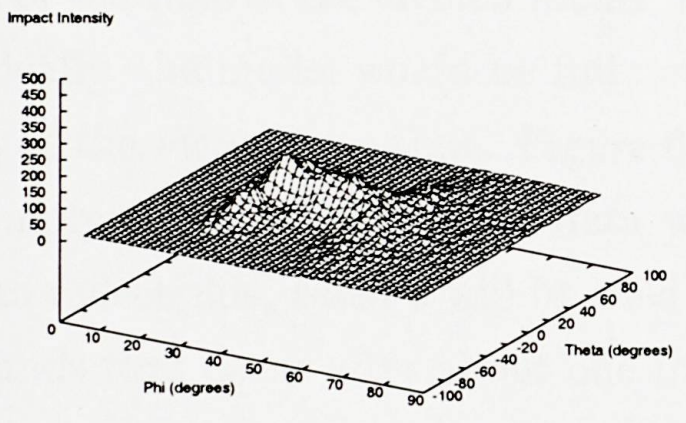
Figure 6.11: Evolving Impact Intensities: R/d = 4.0



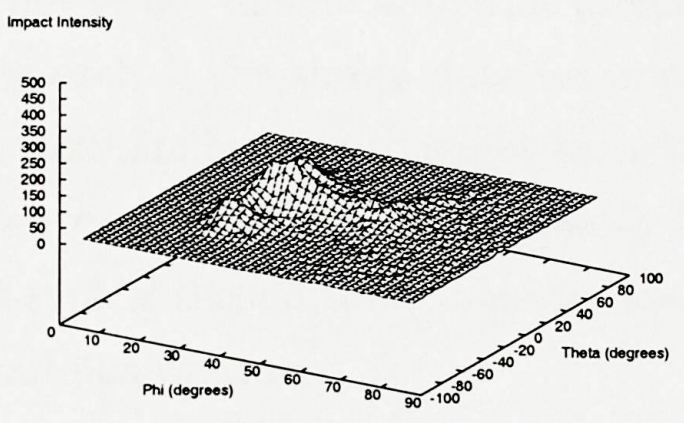
(a) First Batch



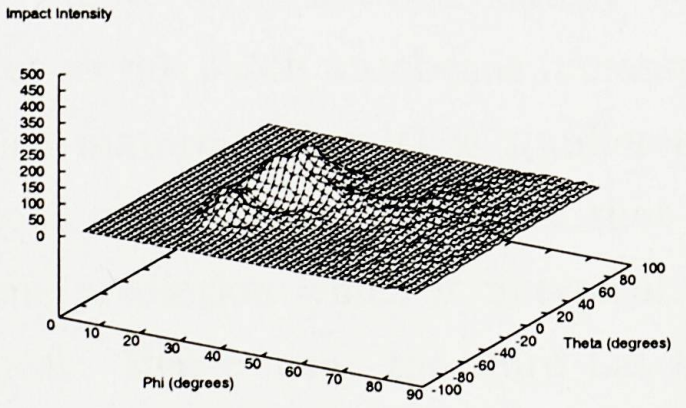
(b) Second Batch



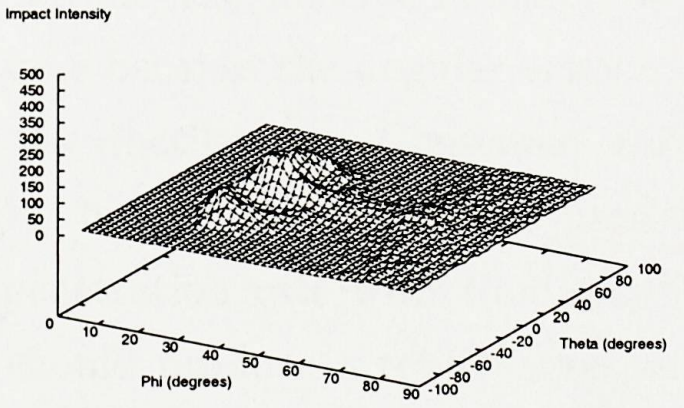
(c) Third Batch



(d) Fourth Batch



(e) Fifth Batch



(f) Sixth Batch

with the erosion model using both first and second wave impacts, to enable the recording of data from a realistically evolving surface.

Using a different bend geometry (R/d=4.0) with thicker pipe walls, the evolution is shown in Figures 6.14 and 6.15. The peak impact angle increases from 12^o to 18^o, while the whole distribution spreads over a wider range. The rebound angles also evolve in this geometry, the peak increasing from 7^o to 9^o and the distribution spreading considerably.

6.3.5 Variation of Lifetime

Without implementing a fully evolving erosion model, an idea of how much effect the evolution will have on the predicted lifetime can be obtained by simply running *Erode* with each of the evolved second impact data files.

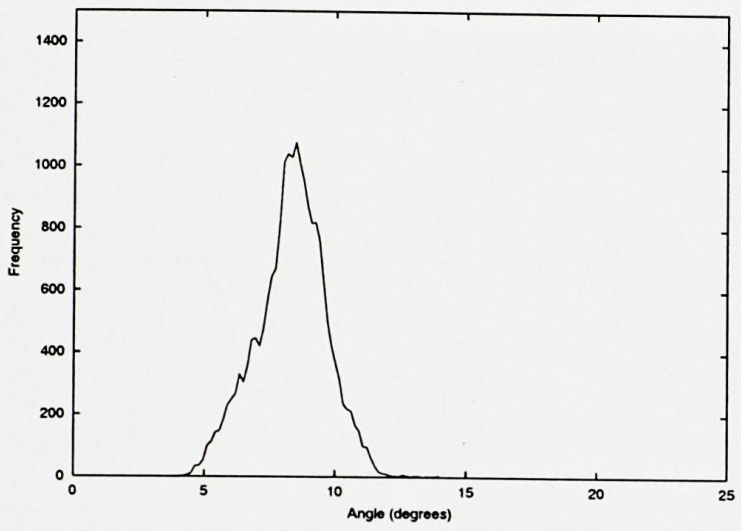
As might be expected from looking at the impact data in Figure 6.10, the lifetime prediction using the data with peaks is shorter than when using flatter impact distributions. Given that the initial second wave data (*batch a*) is unrepresentative of what is occuring for most of the time in the erosion model, it is sensible not to use this for a lifetime prediction.

Ideally, the model would be fully evolving, using each of the shown data for around a sixth of the virtual run-time. Figure 6.16 suggests that similar results might be achieved by simply using second impact data which has been gleaned from a slightly worn bend, and to reflect this, *batch c* will be used to represent typical second wave impacts, meaning rebounds that occur after about one third of the wall penetration.

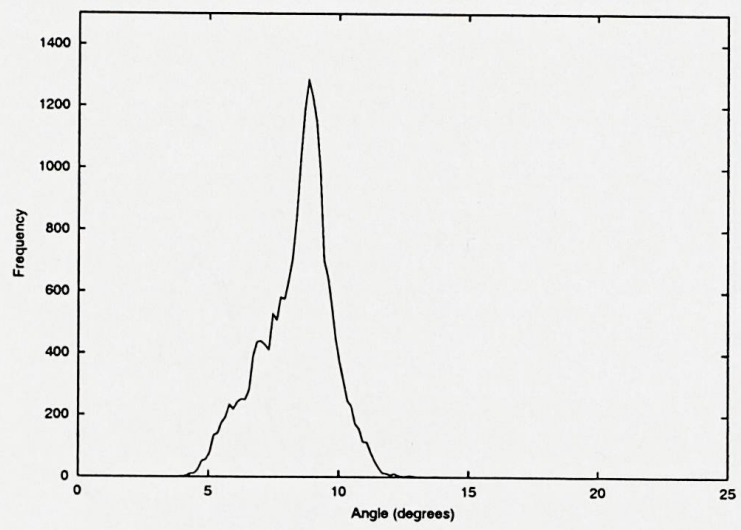
The penetration in each of the above cases was very close to linear with passing time, so it would be expected that a fully evolving model would predict a life bounded by the upper and lower predictions shown in Figure 6.16.

Performing similar tests for the R/d=4.0 geometry, although the peak of the impact data (Figure 6.11) flattens slightly during the evolution, the lifetime actually becomes shorter as the batch number is increased. This may be because the angular erosion curve for this material (Basalt) is a different shape to the ductile one. Consistent with the previous case though is the fact that using the first batch of rebound data produces a lifetime prediction which is untypical. Also, the penetration rate with time stays close to linear. Hence using the third batch of the six should produce a result closer to that achieved by a fully–evolving model, and therefore a more realistic result.

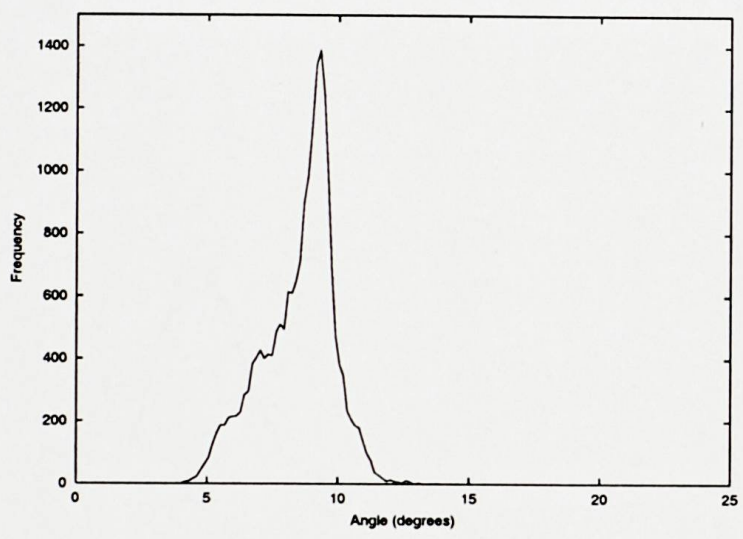
Figure 6.12: Evolving Second Wave Impact Angles (R/d=*14.15*)



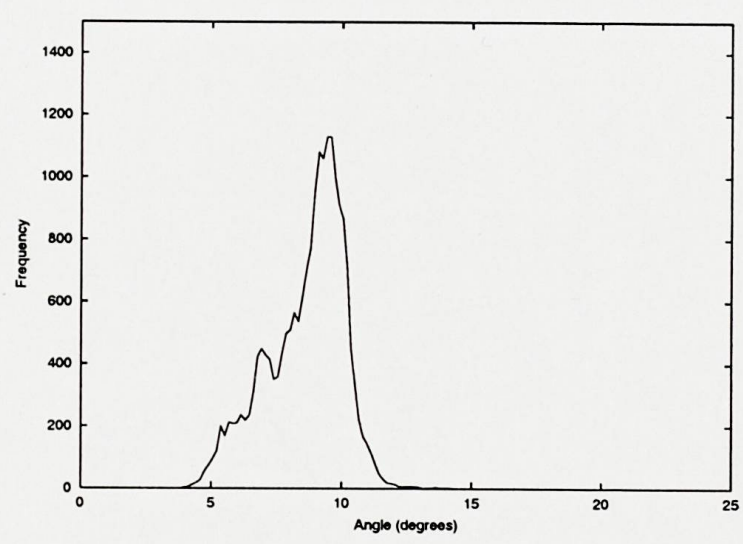
(a) First Batch



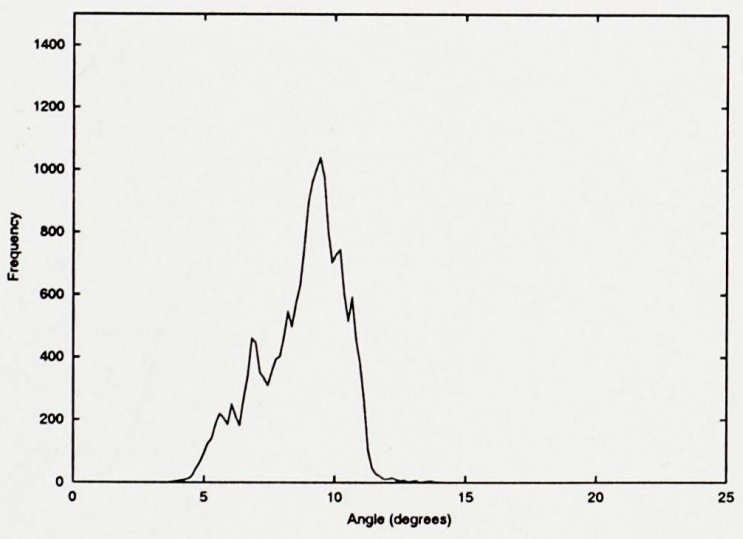
(b) Second Batch



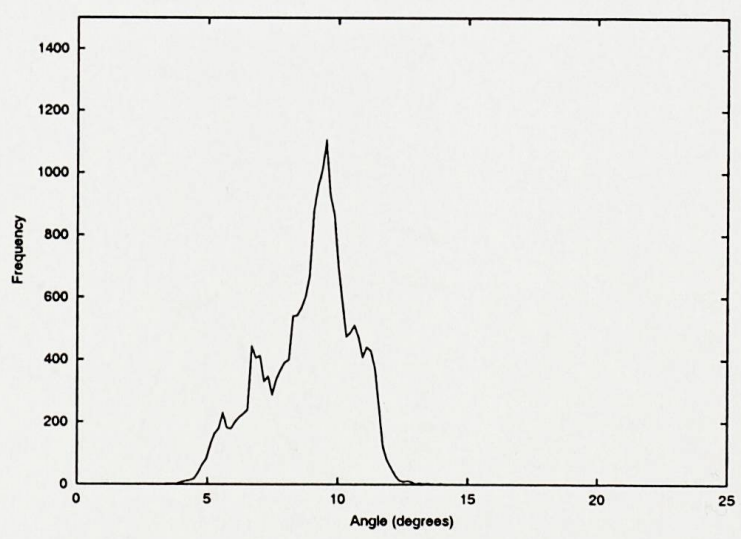
(c) Third Batch



(d) Fourth Batch

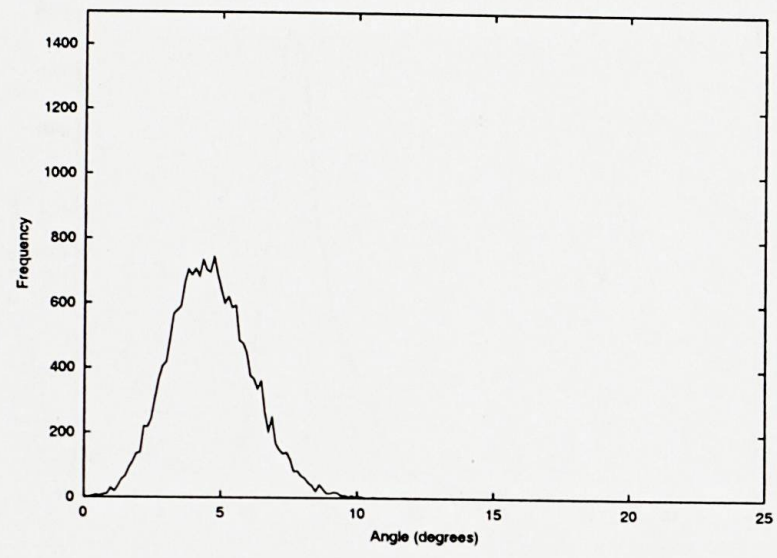


(e) Fifth Batch

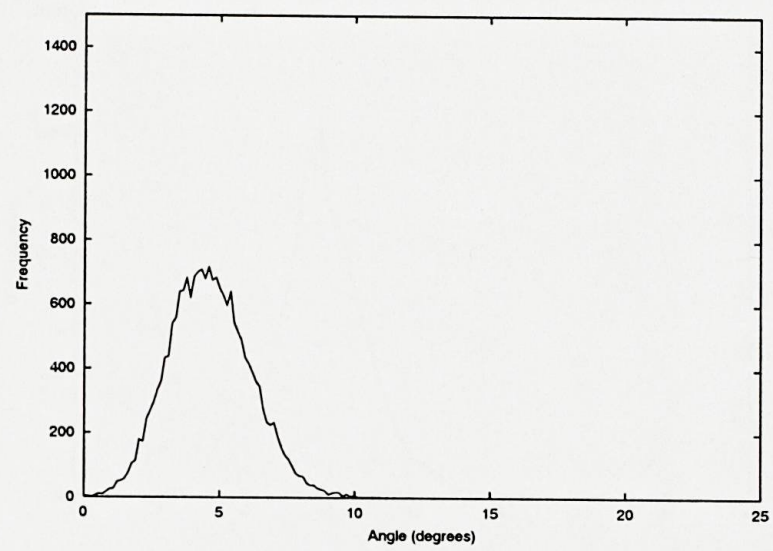


(f) Sixth Batch

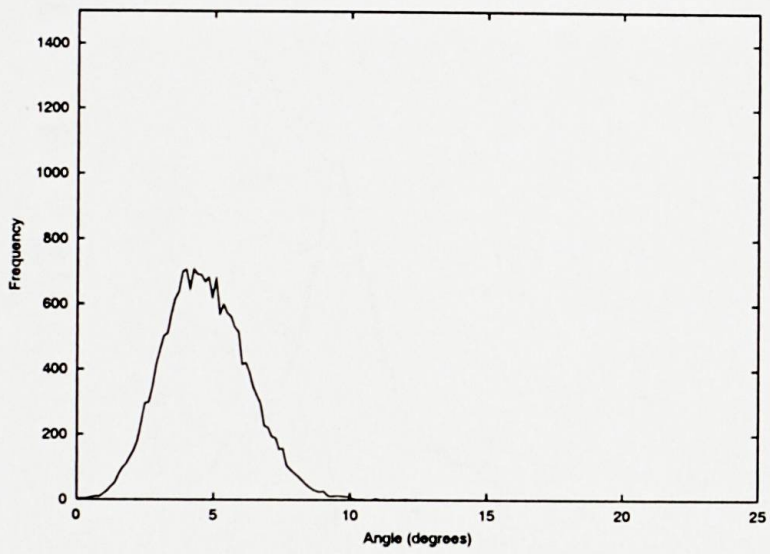
Figure 6.13: Evolving Second Wave Rebound Angles (R/d=*14.15*)



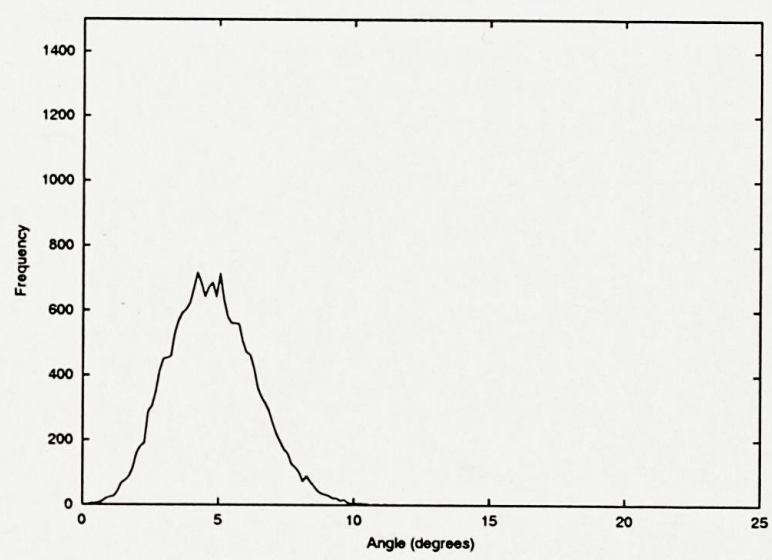
(a) First Batch



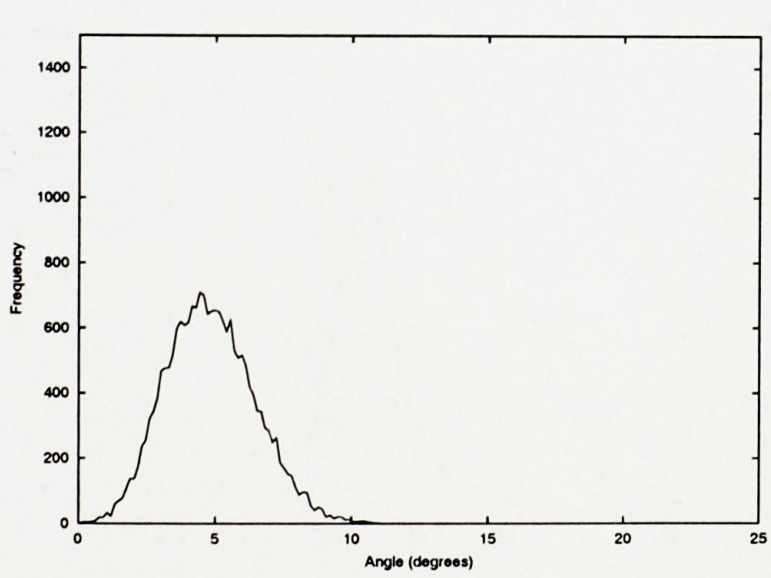
(b) Second Batch



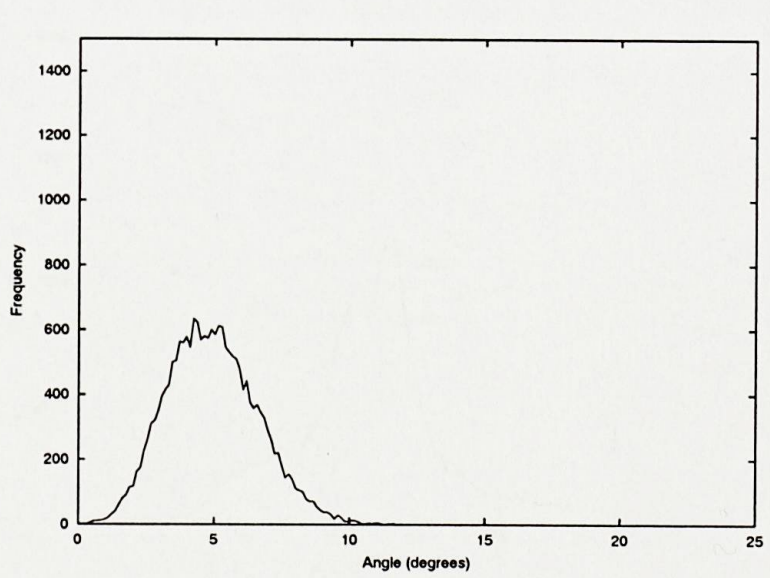
(c) Third Batch



(d) Fourth Batch

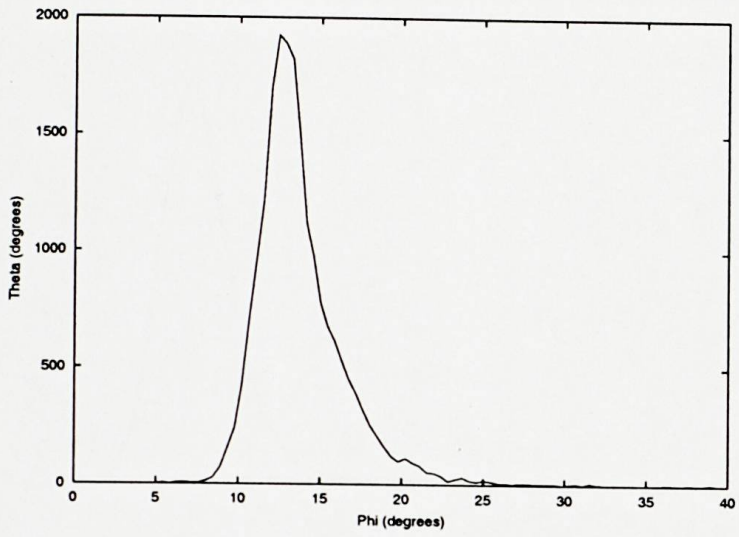


(e) Fifth Batch

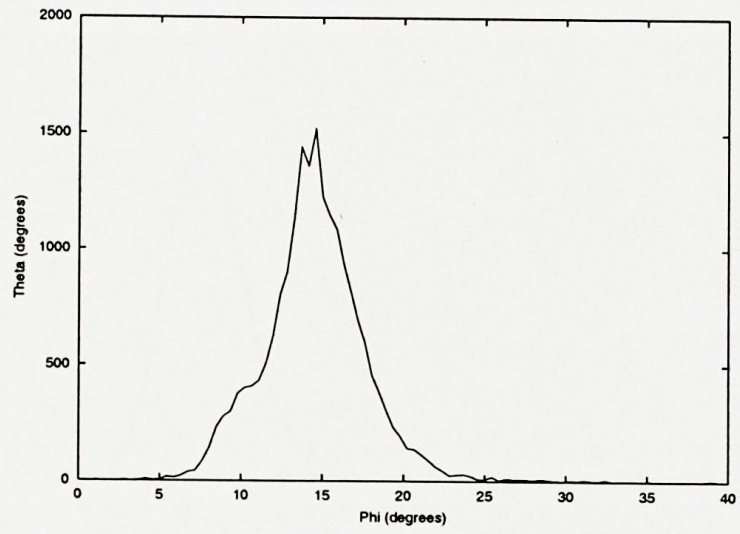


(f) Sixth Batch

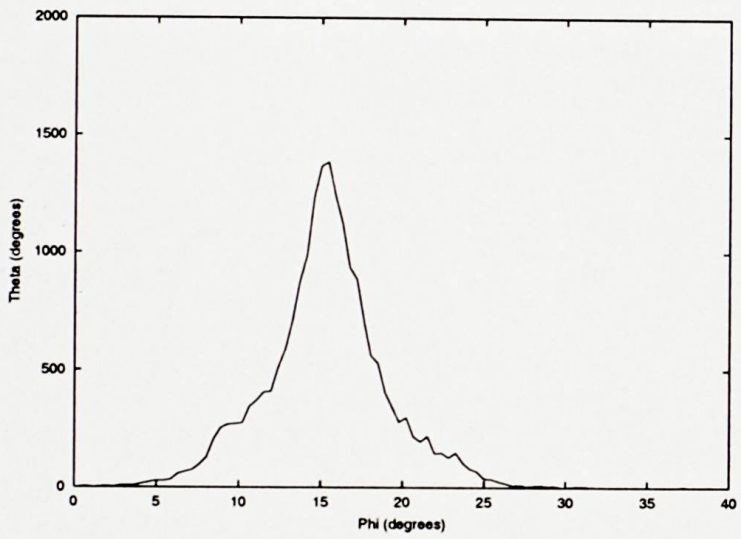
Figure 6.14: Evolving Second Wave Impact Angles (R/d=4.0)



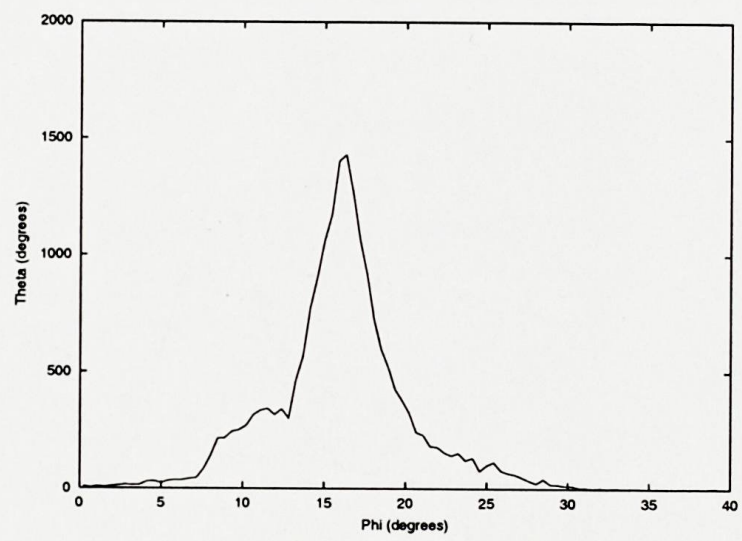
(a) First Batch



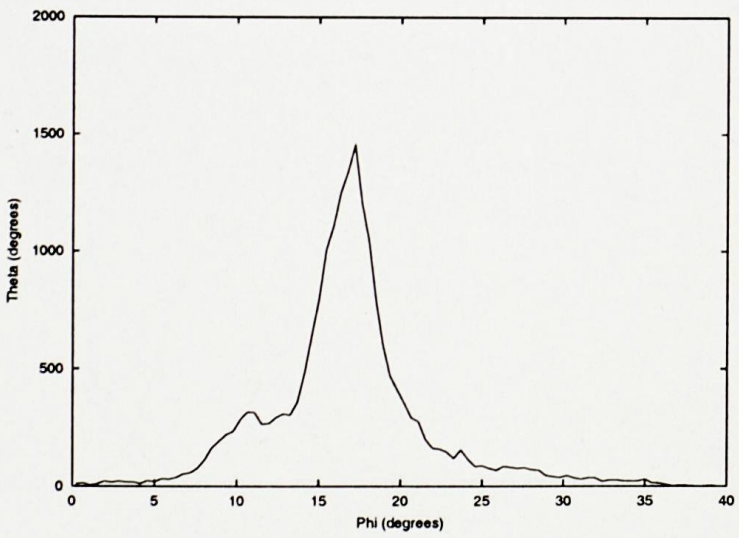
(b) Second Batch



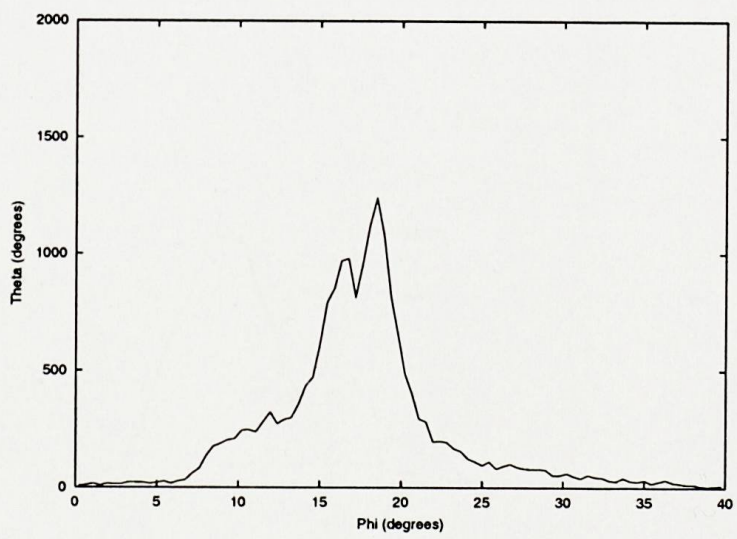
(c) Third Batch



(d) Fourth Batch

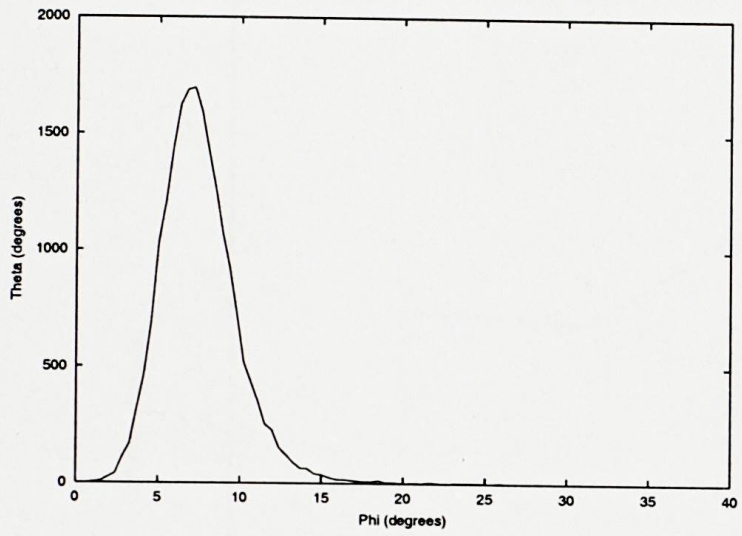


(e) Fifth Batch

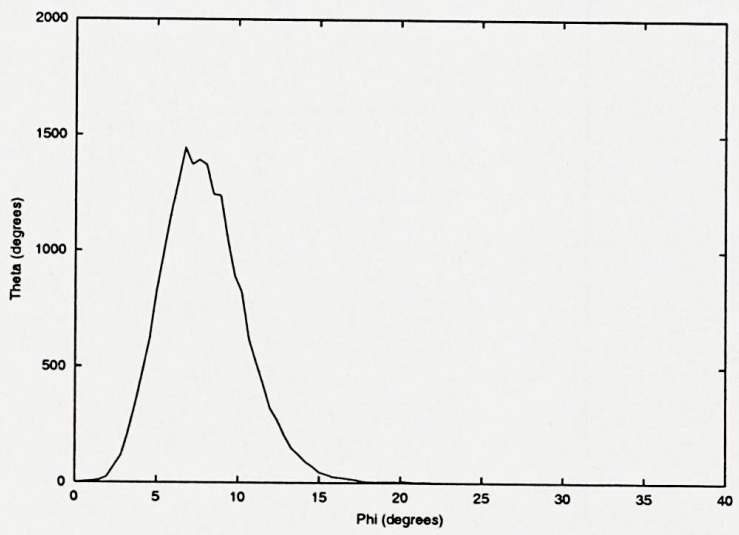


(f) Sixth Batch

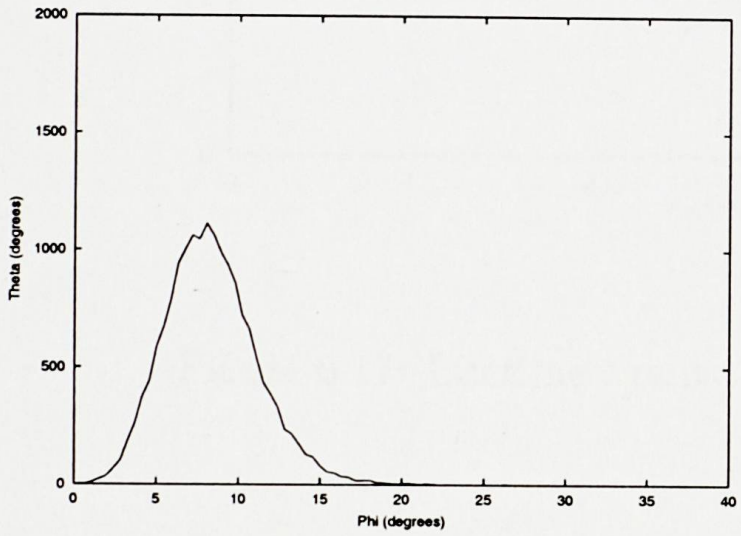
Figure 6.15: Evolving Second Wave Rebound Angles (R/d=4.0)



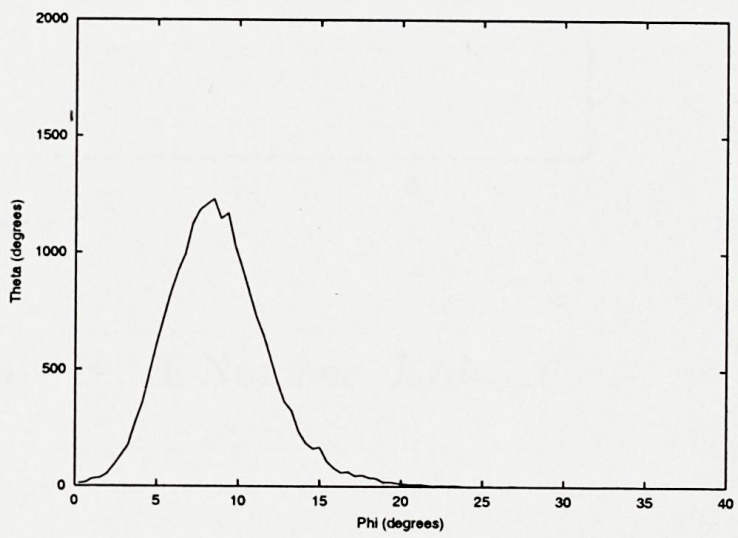
(a) First Batch



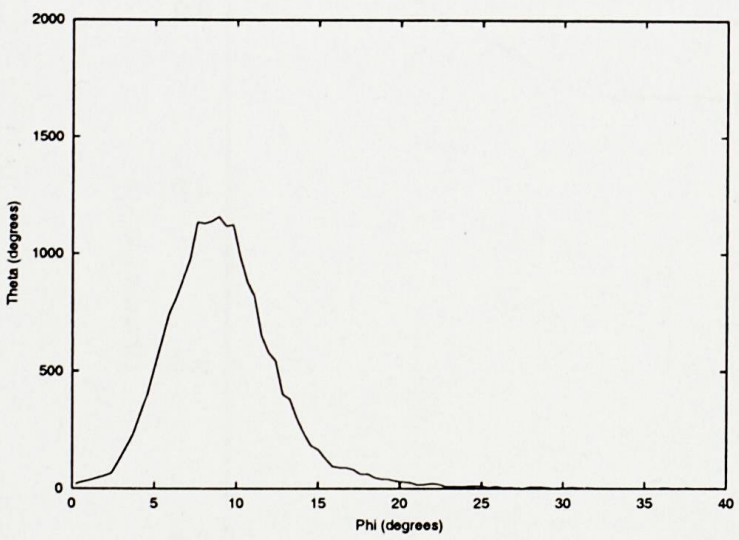
(b) Second Batch



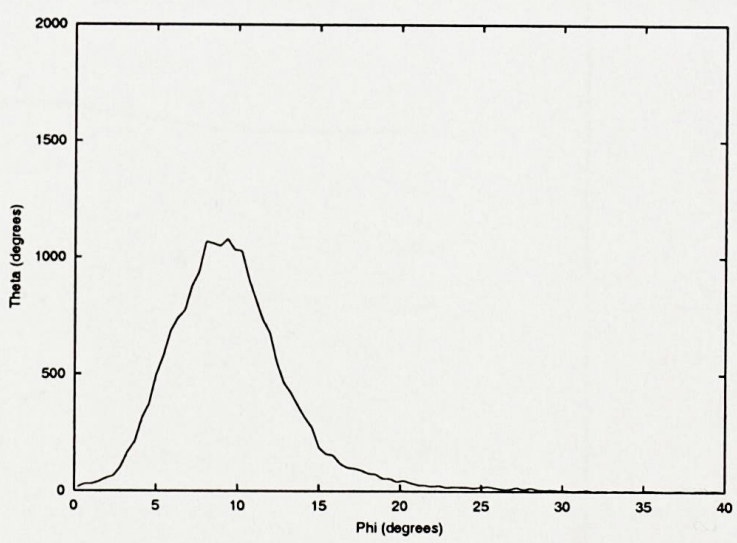
(c) Third Batch



(d) Fourth Batch



(e) Fifth Batch



(f) Sixth Batch

Figure 6.16: Lifetime against 2nd Wave Batch Number: *R/d=14.15*

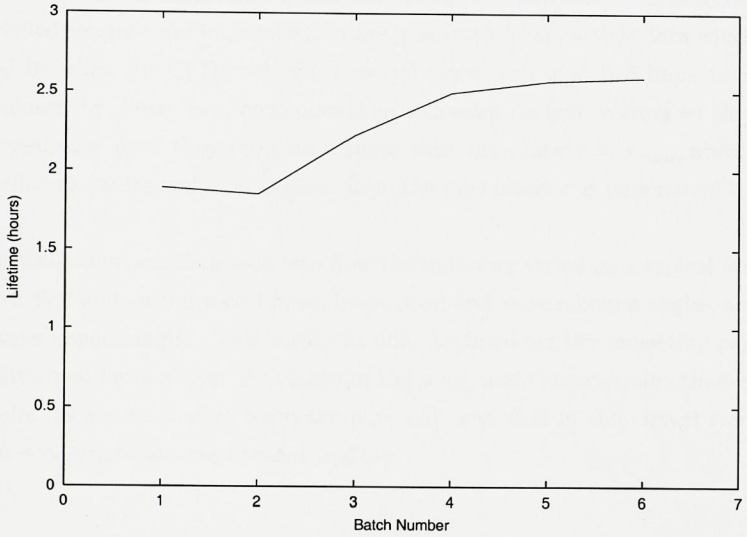
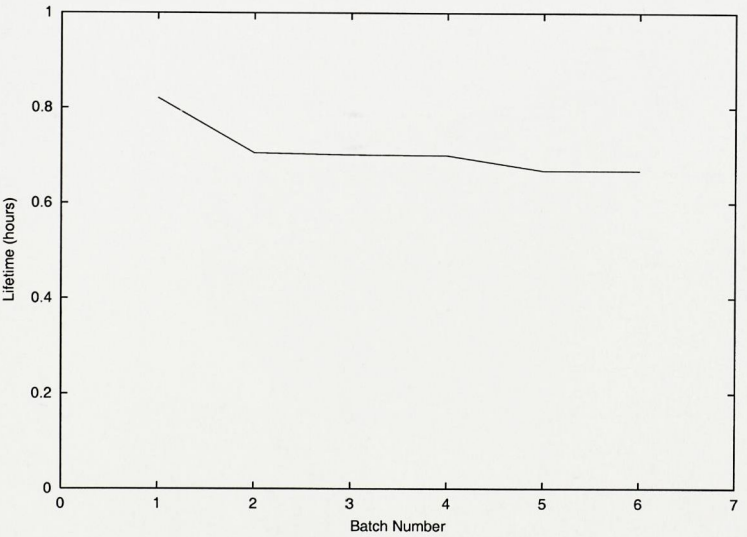


Figure 6.17: Lifetime against 2nd Wave Batch Number: *R/d=4.0*



6.4 Chapter Summary

The motivation behind a multiple impact erosion model was given in this chapter, and the manner in which it was implemented was described. The different kinds of second impacts are: modelled second wave impacts which are generated from particle data which has been generated by using the CFD part of the model twice; and modelled impacts which have been produced by *Erode* as a consequence of following particle vectors as they rebound along a wear scar until they meet once more with the plane $r = r_{min}$, which separates the modelled substrate and wear cavities from the pipe interior as understood by the CFD code.

An investigation was then held into how the following varied as a typical wear process proceeded: first and second wave impact intensities; first wave rebound angles; and first and second wave impact angles. This work was done by breaking the impacting particles into six roughly equal batches over the course of the wear, and characterizing them separately. More evolution was seen when a thicker pipe wall was used in the virtual conveyor, but any other conclusions are not obvious to draw.

Chapter 7

Predictive Results and Parameter Calibration

In this chapter the sensitivity of the virtual rig *Erode* to particle transverse velocities is illustrated, and the need for and technique of calibration is described. Results are then given.

7.1 Introduction

The behaviour of the virtual rig *Erode* has been found to be stable with regard to its own parameters (Chapter 5), but when varying the transverse components of the initial particle velocities, enormous variation in the predicted life is experienced. In Table 7.1 the standard deviation quoted characterises how far from a parallel beam the initial particle velocities are. Specifically, for a large sample this is the standard deviation of the ratios of the transverse components of velocity divided by the main component.

Clearly, if particles diverge to a greater extent, the area of pipe worn is larger, and the penetration rate will be slower. Shown in Figures 7.1 and 7.2 are the variations in lifetime and total mass loss in graphical form. Shown in Figures 7.3 and 7.4 are resulting virtual wear scars for S.D.s of 0.97 and 8.73 respectively, and the higher concentration of impact intensity gives clearly visible deep gouges in Figure 7.3.

Given that the ratio used is critical, the best approach as regards making accurate predictions is to calibrate the value used from measurements of actual bend lives. Section 7.3 describes how this is done using predictions made by *Erode* for two different sets of initial particle data, and the experimental data points.

Note that not all of the bends experimented on were actually burst, but the assumption

Figure 7.1: Bend Life vs. S.D. (%)

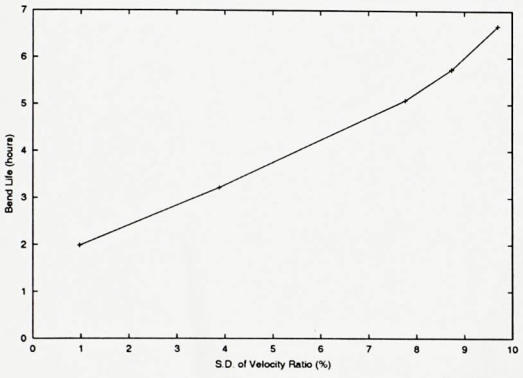


Figure 7.2: Mass Loss vs. S.D. (%)

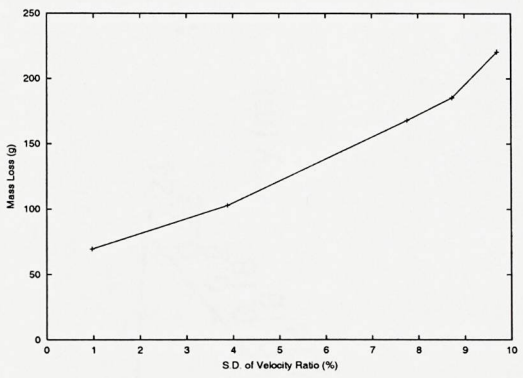


Figure 7.3: Virtual Wear Scar, S.D.=0.97

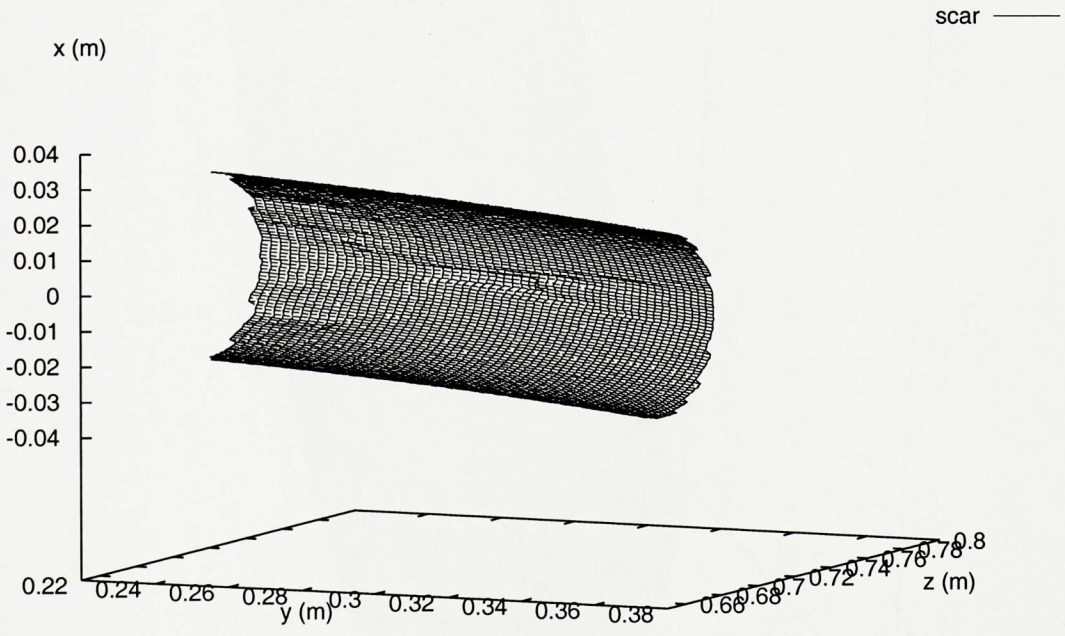


Figure 7.4: Virtual Wear Scar, S.D.=8.73

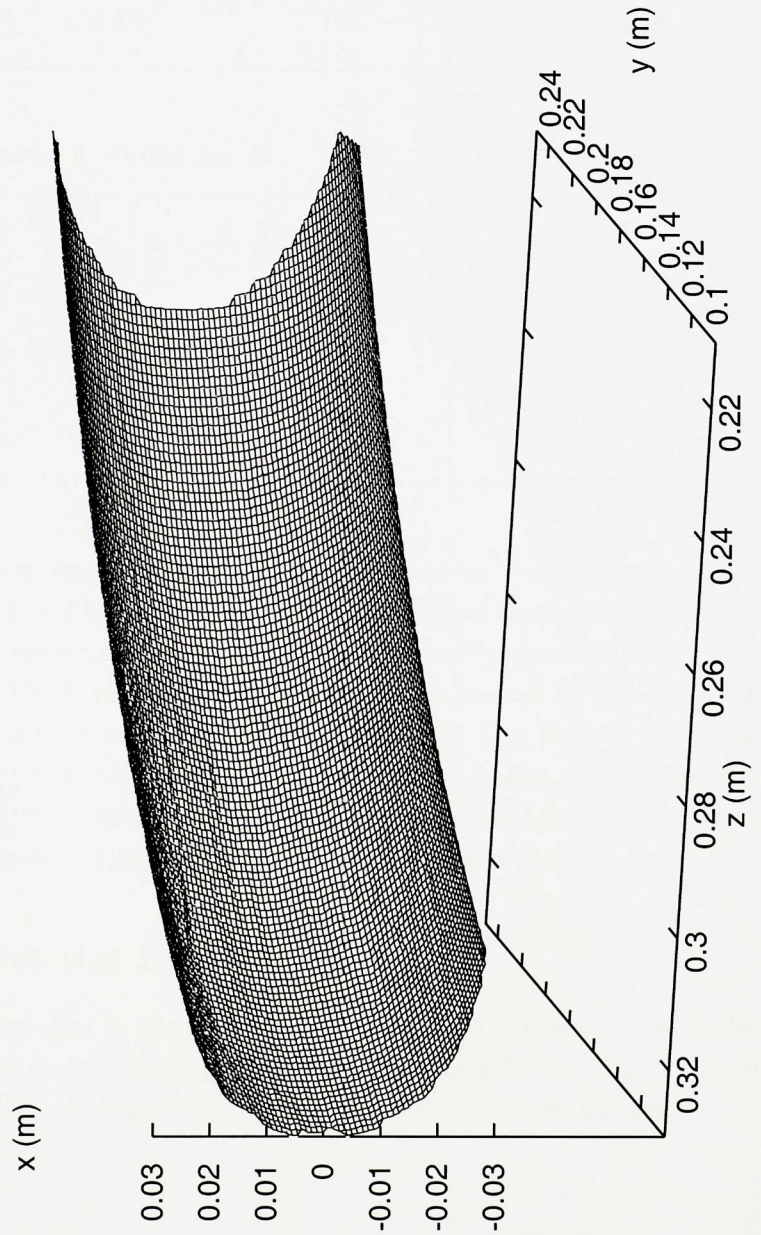


Table 7.1: Lifetime against Transverse Particle Velocity

S.D of $(u, v)/w$ (%)	Lifetime (hrs)	% err	Total Mass Loss (g)
0.97	1.981	-62	69.51
3.88	3.214	-45	102.77
7.76	5.085	-13	168.03
8.73	5.746	-2	185.38
9.70	6.675	+14	220.56

Figure 7.5: Virtual History

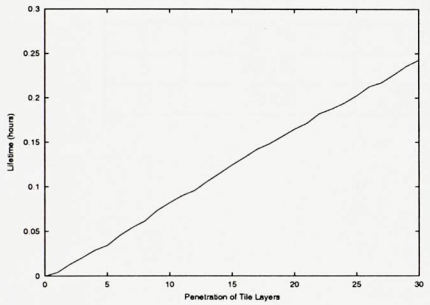
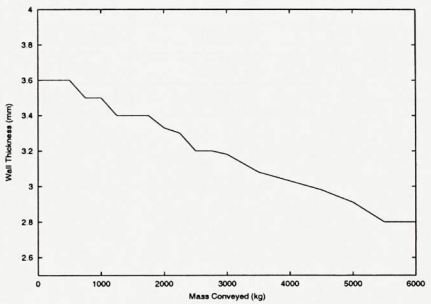


Figure 7.6: Measured: Bend 2



of linear penetration with time was used to calculate the bend life from the deepest point of wear at the end of the experiment. This assumption of linear penetration has been borne out by both the virtual rig *Erode* and the experiments (Figures 7.5, 7.6).

Details of each bend and the conveying conditions can be found in Tables 7.2 and 7.3. The data for bends one to five was gathered by *Tong Deng* at *The Wolfson Centre, University of Greenwich*, as a part of this *EPSRC* funded project. Bend 2R was the replacement for the original bend 2, and the data for bend B is historical, it being found by *Tony Burnett* in 1996 [Bur96], also at *The Wolfson Centre*.

7.2 Test Rig Results

The numerical data in this section was obtained by *Tong Deng* at the *Wolfson Centre, University of Greenwich* as a part of this *EPSRC*-funded project. That recounted here is by no means all of *Tong's* work, but it is the most relevant to this thesis.

Table 7.2: Bend Details

Bend	Air Vel (m/s)	R/d Ratio	Material	Massflow (g/s)
1	22.12	10	Mild Steel	1020.3
2	23.34	6	Mild Steel	1020.3
2R	23.34	10	Mild Steel	1020.3
3	25.76	10	Mild Steel	1020.3
4	26.70	10	Mild Steel	1020.3
5	36.03	6	Ni-Hard	1020.3
B	25.60	14.15	Mild Steel	259.0

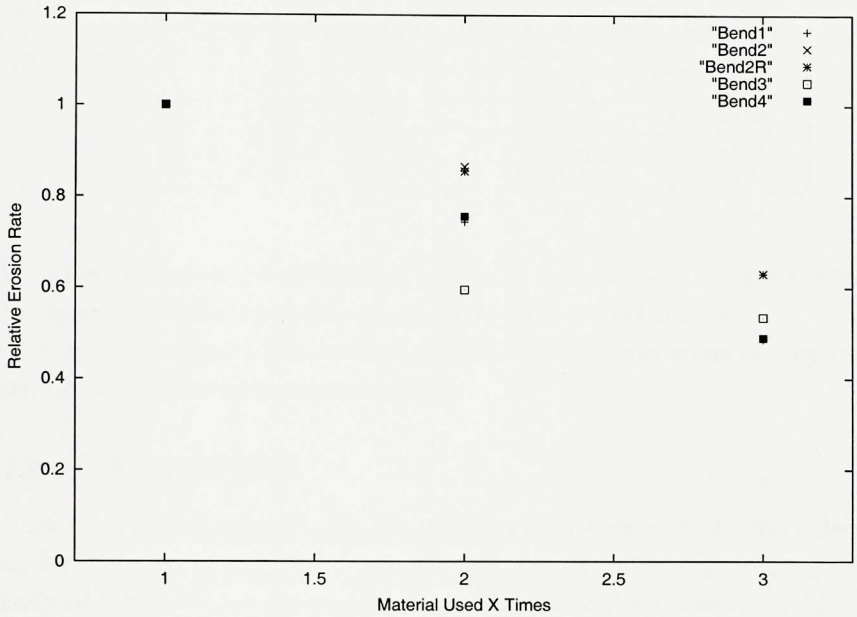
Table 7.3: Further Bend Details

Bend	Wall Thickness (mm)	Bend Orientation
1	3.4	Up to Horizontal
2	3.7	Horizontal Plane
2R	3.94	Horizontal Plane
3	3.3	Horizontal to Down
4	3.4	Horizontal to Up
5	16.3	Horizontal Plane
B	2.39	Horizontal Plane

Table 7.4: Test Rig Runs

Run	Mass (kg)	Σ (kg)	Delivery	Usage
1	250	250	1	1st
2	278	528	1	1st
3	261	789	1	1st
4	277	1066	1	1st
5	264	1330	1	1st
6	247	1577	1	1st
7	252	1829	1	2nd
8	264	2093	1	2nd
9	264	2357	1	2nd
10	253	2610	1	2nd
11	269	2879	1	2nd
12	269	3148	1	2nd
13	504	3652	2	1st
14	500	4152	2	1st
15	500	4652	2	1st
16	523	5175	2	1st
17	494	5669	2	2nd
18	495	6164	2	2nd
19	543	6707	2	2nd
20	501	7208	2	2nd
21	1002	8210	2	3rd
22	1022	9232	2	3rd
23	1005	10237	3	1st
24	1038	11275	3	1st
25	1004	12279	3	2nd
26	935	13214	3	2nd

Figure 7.7: Relative Erosion for Re-used Material



7.2.1 Material Conveyed and its Re–Use

Table 7.4 illustrates the mass conveyed for each run of the *Wolfson Centre* rig. Three batches of material were delivered, and these were used at least twice, and sometimes three times. Given that the erosivity of a re–used material will diminish, due to the loss of asperities, some way of compensating for this re–use is required.

It was thought that a mean value might suffice to reflect the degradation of particles, but given the scatter for different bends, each bend's rate was individually used. These rates were used to weight the amount of material conveyed for a second or third use, to give a more accurate portrayal of the bend life. This is shown in Table 7.5, where the *Suggested Mass* is obtained by using the weighting formula:

$$\frac{(5647kg)}{1.0} + \frac{(5543kg)}{\text{Rate2nd}} + \frac{(2024kg)}{\text{Rate3rd}} \tag{7.1}$$

where the numerators used are the amounts of mass conveyed for the first, second and third times, and Rate2nd, Rate3rd are the relative erosion rates (Figure 7.7) on re–using the material. The *Life Factor* (Table 7.5) is obtained by dividing the *Suggested Mass* by

Table 7.5: Scaling Conveyed Mass with Relative Erosivities

Bend	Rate2nd	Rate3rd	Suggested Mass (kg)	Life Factor
1	0.745	0.487	17243.3	1.305
2	0.866	?	15823.8	1.198
2R	0.856	0.631	15330.1	1.160
3	0.596	0.535	18730.5	1.417
4	0.757	0.490	17099.9	1.294

the actual mass conveyed, and this can be used to scale the bend lifetimes to what would be expected if all the material used in the tests was conveyed once only.

Note that for bend 2, no 3rd use of material occurred, so to obtain the *Suggested Mass* the mean of the other four values was used.

7.2.2 Bend Weights

Much interesting data (Table 7.6) has arisen from the decision to weigh the bends used on the experimental rig at the *Wolfson Centre*, and thanks must go to *Tong Deng*, who performed this extra work at short notice, under considerable pressure of work.

Table 7.7 shows the mass lost by each bend for mass conveyed (first use of material only), and the mean erosion rate. Bend 5 (Ni-Hard) could not be weighed as it was too heavy, and there is no data on the weight of AJB's bend.

The virtual rig *Erode* can very easily calculate the total mass lost by the virtual bend, and hence direct comparisons of mass loss rate can be made in Table 7.8. Bend 2R is the replacement for bend 2, which was changed due to odd thickness measurements.

Looking at these results, it can only be assumed that erosion testers produce an exaggerated picture of the damage done in pneumatic conveyors (at least, this conveyor). The erosion rates quoted from the *Erode* results are in the range expected for the velocity used at each bend, using results from two erosion testers which were only 6% different. The *Erode* results are maybe 40% higher than the erosion tester values, but this should be expected, as in the virtual rig each particle strikes at least twice and possibly more. Hence the damage done per unit mass will definitely be greater than in an erosion tester, as these are designed to measure one impact of each particle.

A factor of four can only be explained by differences in the physical situations. One suggestion is that on an erosion target, eroded substrate and particle fragments are cleaned away by the airstream, whereas in an enclosed pipe there is a uniform layer of this dust all

Table 7.6: Test Rig Bend Mass Loss (grams)

Run	Bend 1	Bend 2	Bend 2R	Bend 3	Bend 4
1	-	-	-	-	-
2	-	-	-	-	-
3	-	-	-	-	-
4	-	-	-	-	-
5	-	-	-	-	-
6	-	-	-	-	-
7	-	-	-	-	-
8	-	2.5	-	-	-
9	-	3	-	1	-
10	-	2	-	1	3
11	-	2.5	-	1.5	3
12	-	2.5	-	2	2
13	4	4	-	8	8
14	5	4	-	7	6
15	4	4	-	6	6
16	4	4	-	7	7
17	3	2	-	4	5
18	3	3	-	5	5
19	3	-	3	6	5
20	3	-	3	5	5
21	4	-	5	7	6
22	4	-	5	8	7
23	8	-	8	12	13
24	8	-	8	-	6
25	6	-	8	-	-
26	6	-	6	-	-

Table 7.7: Test Rig Mass Loss Rates

Bend	Mass Loss (g)	Conveyed (kg)	Loss Rate (g/kg)
1	33	4070	0.0081081
2	16	2027	0.0078934
2R	16	2043	0.0078316
3	40	2888	0.0138504
4	46	3511	0.0131017

Table 7.8: Mass Loss Rates

Bend	*Erode*(g/kg)	*Expt.*(g/kg)	x/y
1	0.03176	0.00811	3.92
2	0.04138	0.00789	5.24
2R	0.03461	0.00783	4.42
3	0.04359	0.01385	3.15
4	0.04813	0.01310	3.67
Mean	-	-	4.08

Table 7.9: Virtual Predictions with Measured

Bend	S.D (%)	Virtual Life (hrs)	Measured Life (hrs)
1	5.82	5.410	4.927
	15.52	11.444	
2	5.82	2.946	6.339
	15.52	5.892	
2R	5.82	5.736	9.621
	15.52	13.056	
3	5.82	3.558	3.894
	15.52	8.507	
4	5.82	3.199	3.775
	15.52	7.368	
5	5.82	5.251	9.457
	15.52	10.751	
B	5.82	17.401	5.872
	11.64	32.742	

along the pipe. This is certainly true, having seen a glass section of pipe in the *Wolfson* rig, but it is not obvious that this dust can shield the pipe wall from incoming particles to such a large extent.

A concern here is that such a discrepancy has not been observed elsewhere as far as this author knows, though finding published numerical data in erosion modelling is difficult when so many authors tend towards the qualitative in their work.

Another possibility is that the conveyed particles are not actually travelling as fast as the air-stream, though experiments seem to refute this (Section 3.3.2).

This is certainly one of the most interesting results in this project. For the purposes of the current project, the above difficulty can be surmounted by simply assuming that erosion testers over-estimate erosion by a factor of 4.08, which is the average of the above. This factor is now inserted into *Erode*, so that the transverse components of initial particle velocity can be calibrated. The data set of predictions by *Erode* is updated, and compares with the experimental data points as in Table 7.9. Note the large variance of *Burnett's* data point (B), to be discussed in Section 7.3.

7.2.3 Penetration Rates and Lifetimes

Using extensive data obtained at the *Wolfson Centre*, Table 7.10 summarises the amount of wear experienced by each bend in the test rig. The pipe wall thickness measurements were painstakingly obtained by *Tong Deng*, using a commercial tool employing the reflection of ultrasound waves. A grid of points was used on each pipe, but only the point of deepest wear has been used here. Also, the amount of mass conveyed in Table 7.10 has been scaled to compensate for the fact that material was reused. This was performed as discussed in Section 7.2.1.

Recall that bend 2R was the replacement for bend 2, and B refers to *Burnett's* case, which was measured with different equipment [Bur96]. The bends 3B and 4B refer to the bursts obtained on the test rig, whereas the bends 3 and 4 refer to these same bends, but with their penetration measured with the ultrasound tool. Note that even though these bends had burst, at the last measurement of the pre-defined measurement points only around 70 % penetration was being picked up. This is because wear is local, and may not fall conveniently. Experimental technique requires a time and work budget and the amount of data points gathered must be finite. To avoid excessive error here, it will be assumed that in all unburst cases the pentration has been undermeasured by 30%. This gives the lifetimes shown in Table 7.11, and these will be used for calibration.

Table 7.10: Bend Penetration

Bend	Scaled Mass (kg)	% Wear	Life (hrs)
1	17244	66.7	7.039
2	7384	22.2	9.055
2R	8178	16.2	13.744
3	14506	72.7	5.432
4	14590	65.9	6.028
5	16848	48.5	9.457
3B	14302	100.0	3.894
4B	13867	100.0	3.775
B	5475	100.0	5.872

Table 7.11: Bend Life Assuming Deepest Wear Missed

Bend	Life (hrs)
1	4.927
2	6.339
2R	9.621
3B	3.894
4B	3.775
5	6.200
B	5.872

Table 7.12: Bend Burst Points (Degrees)

Bend	Modelled	Experimental
1	26	-
2	32	-
2R	25	-
3	25	22
4	28	8
5	46	-
B	20	29

7.2.4 Burst Points

These are shown in Table 7.12 and were most useful when assessing which range of transverse velocity component standard deviations to use. When these S.D.s became too large (typically over 20%) the spread in particle impacts would cause the burst point to appear in unexpected places, often around 70^o from the extrados, and less than 5^o around the bend. Disregarding these unrealistic points, other modelled burst points fell within 10^o of the extrados of the bend, and experimentally they occurred out to around 15^o.

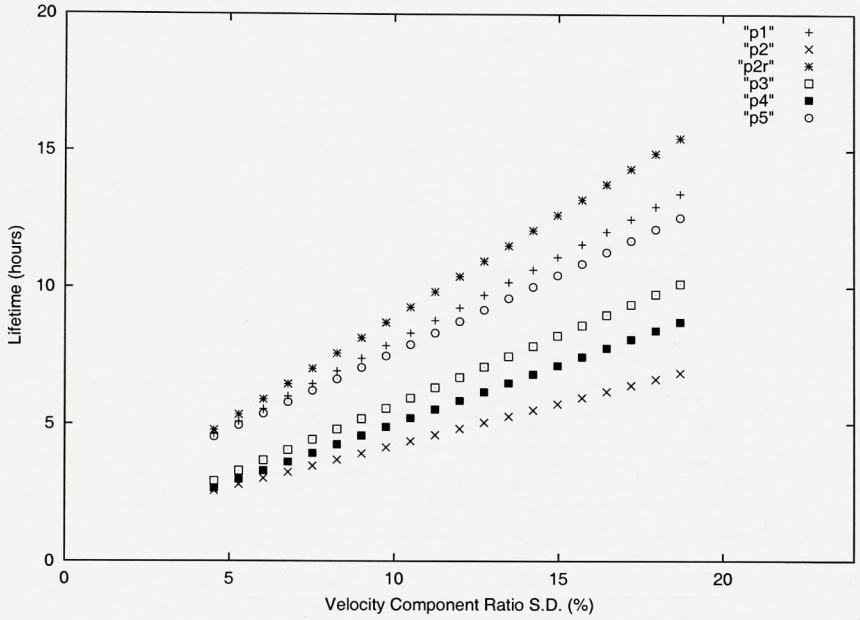
Modelling *Burnett's* [Bur96] bend one such odd burst point occurred at an S.D. of 15.52%, so this is why a lower value of 11.64% was used for the second interpolation point, as seen in Table 7.9.

7.3 Calibration of Model

Recall from Section 5.7 that an extremely sensitive factor for lifetime dependence was the transverse components of velocity initialised for the tracked particles.

Some experimental observations on this were found [Eas98][Woo92] but other works usually concentrate on the main component of velocity in the pipe. The numbers suggested in the above works were either 2.0 or 2.5 % of the stream velocity. The first figure was achieved with water droplets, which might stick to a lengthy pipe rather than bouncing and randomising to the greatest extent, so this might be a lower limit. The second figure was achieved with sand particles in a conveyor, but with equipment of quoted tolerance of around the same. This therefore might be an upper limit, but it seems unlikely all or most values will fall into this range, given the irregular shape of e.g. sand particles. Early runs with *Erode* suggested that a normal distribution with standard deviation of around 8% is more appropriate, and this is why the decision to calibrate was taken.

Figure 7.8: Bend Life vs. S.D. (%)



However, changing the initial particle conditions in the CFD part of this model requires that the particles must be retracked in order to find their new trajectories, before rerunning the *Erode* code. As this is computationally expensive, it is not possible to change the transverse velocity components quickly or easily.

Instead, having established that the dependence on velocity component ratio is close to linear (Figure 7.1), two points bounding the desired lifetime prediction will be used to make interpolations, or possibly extrapolations.

Having established the maximum and minimum of this ratio required to bound all the lifetime measurements, the resulting range was divided by an integer number of points, and a lifetime prediction was interpolated for all used bends at each of these points (Figure 7.8). For each of these interpolations, a least squares type of error estimate was made and summed over all the bends. These error estimates were then plotted against the ratio used, and in Figures 7.9 and 7.10 a minimum is clearly seen. In the magnified figure, this is seen to occur at an S.D. of 8.37%.

This ratio giving the minimum error was then used for filling the matrix of predictions discussed in the next chapter, and definitive predictions for the six calibration bends (and

Figure 7.9: Bend Life Error vs. S.D. (%)

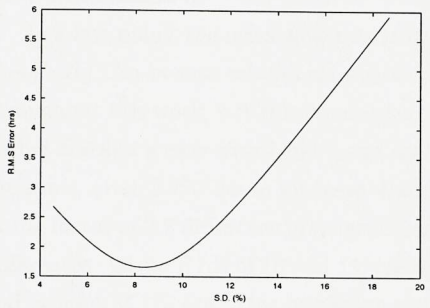


Figure 7.10: Bend Life Error vs. S.D. (%)

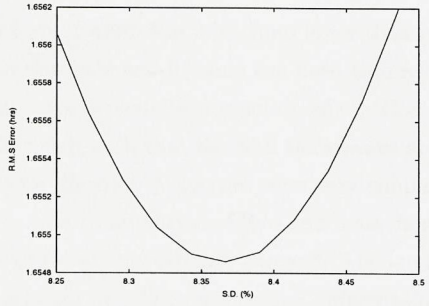


Table 7.13: Virtual Predictions after Calibration

Bend	Measured Life (hrs)	Virtual Life (hrs)	Error (%)
1	4.927	6.991	42
2	6.339	3.718	-41
2R	9.621	7.654	-20
3	3.894	4.855	25
4	3.775	4.291	14
5	6.200	6.692	7.9
AJB	5.872	24.100	310

Burnett's bend) are shown in Table 7.13.

It was eventually decided to discard *Burnett's* [Bur96] data point for this process, as it was so far removed from the others. This is an acknowledgement of the fact that each conveying rig might have its own keyfactor, which links its erosion rate to erosion testers. Without more conveyor data not much can be said on this subject, but as regards making life predictions, it is important that this subject be broached in the desktop tool associated with this project. Users could then potentially scale predictions once they had obtained their own keyfactor.

The results shown in Table 7.13 suggest a standard deviation of error of 31%, this by ignoring *Burnett's* point. If the whole process of calibration is run including this last point, then the s.d. of error comes out as 58%.

Aside from the reduction in error, there are good reasons for not using *Burnett's* point. There were no mass measurements on this pipe, and the erosivity keyfactor of 4.08 used to scale *Erode's* predictions came purely from *Tong Deng's* rig. It would be inconsistent to perform a second calibration using different data. Also, if there is a rig–specific factor

at play as regards bend life, smearing the calibration process across two diverse rigs will not help to isolate it.

As a last point, the mass–flow rate in *Burnett's* rig [Bur96] was 3.94 times lower than in *Deng's* rig. An inverse relationship between mass-flow rate and lifetime has been assumed throughout this work, but this assumes no serious particle-particle interaction effect. Comparing *Burnett's* case with bend 4 and scaling the result such that the wall thicknesses are the same, gives 3.775 hours for bend 4 and 8.353 for *Burnett.* Velocities were very similar so the factor of 2.2 difference may be accounted for by a combination of R/d and mass–flow differences. Given R/ds of 10 and 14, use a factor of 1.4 to approximate *Burnett*: 5.966 hrs, and Bend 4: 3.775 hrs. This might suggest that a factor of 1.58 comes from a difference of mass-flow rate of 3.94 (at concentrations of *Deng*: $18.1 kg/m^3$ and *Burnett*: 4.48 kg/m^3), but with just one data point, there can be little confidence in this.

7.4 Blowtank Conveyors

The Blowtank type of conveyor is currently the most prevalent one used in industry, and this was the kind used for the experimental validation recounted in this thesis.

With this type of conveyor, material is transported in a series of short bursts, rather than in a more continuous fashion. The reason this is relevant here is that a significant proportion of actual conveying time is spent by the system stabilising to a steady state. When this is achieved, it is not long before the solids loading ratio starts to diminish, and the air velocity starts to increase again as the batch is finished. Figure 7.11 shows the air velocity at bend 2 in the *Wolfson* rig over a 500 second period. This and other data used here was supplied by *Tong Deng*, and was obtained using pressure transducers, arrayed along the rig pipework.

As material is introduced to the flow, the air velocity slows dramatically, and similarly it returns to its original value when the screw feeder runs out of material. Also shown in Figure 7.12 is the mass of the destination hopper over the same period. Note that the initial value is not zero, and that the scale can only produce a value every few seconds.

Given these data, and the known power-law dependence of erosion upon impact velocity, the question arises: is it accurate to model a pneumatic conveyor using one mean conveying velocity, as is done in this work?

To investigate this a typical erosion against velocity dependence was used (Figure 7.13), in this case data supplied by *David Allsopp* from the Cambridge data set. This was used over a series of time steps in conjunction with the appropriate mass-flow rate (Figure

Figure 7.11: Conveyor Air Velocity at Bend 2

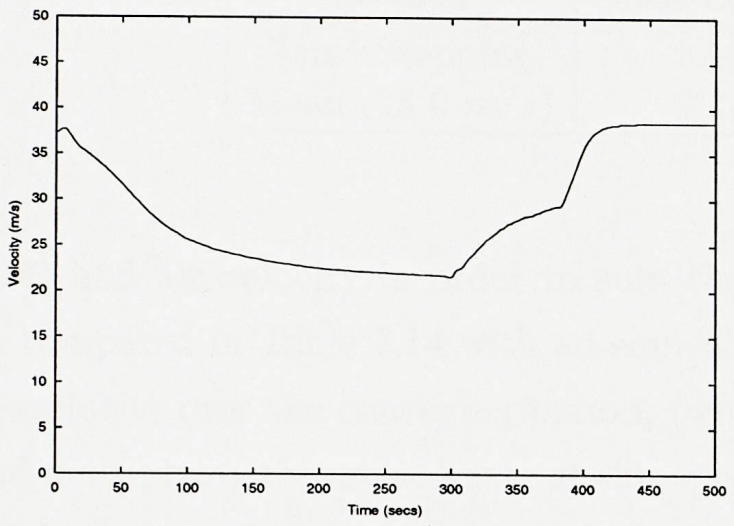


Figure 7.12: Receiving Hopper Mass

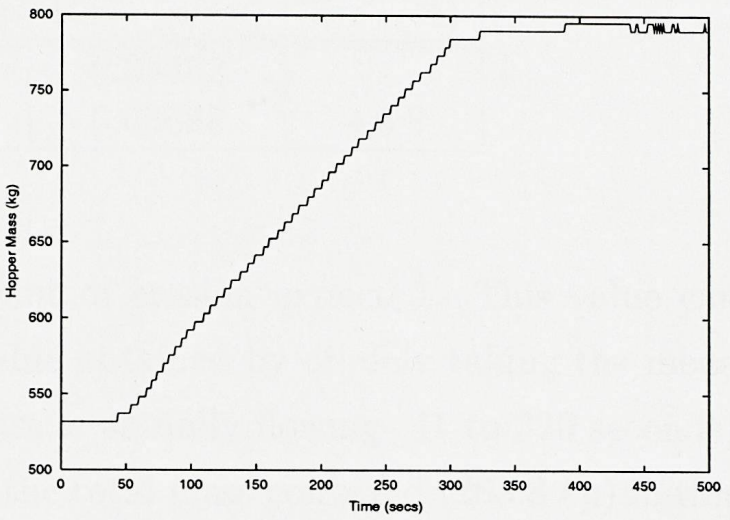


Figure 7.13: Erosion - Mild Steel, 30°

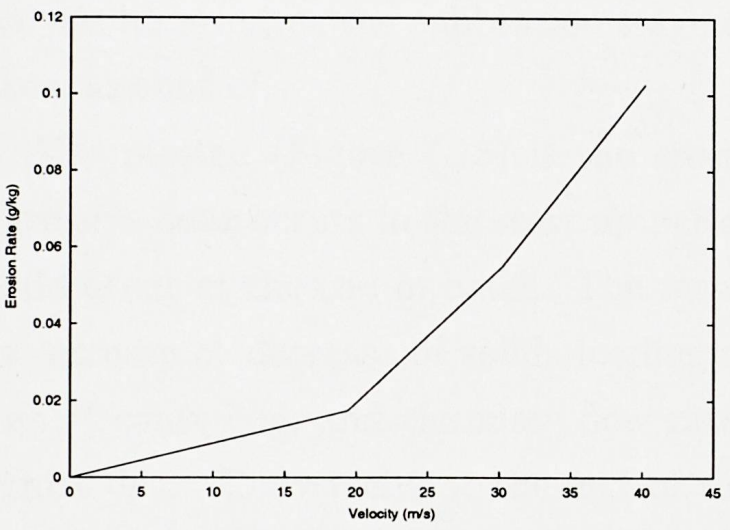


Figure 7.14: Mass Flow for Bend 2

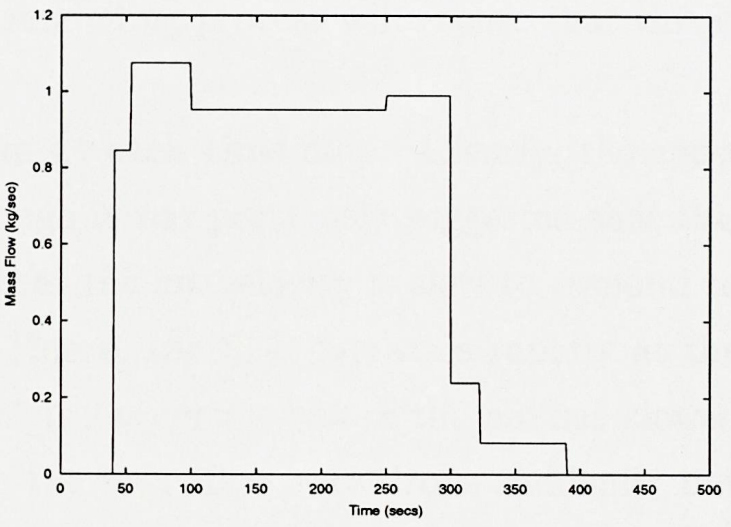


Figure 7.15: Erosion Rate for Bend 2

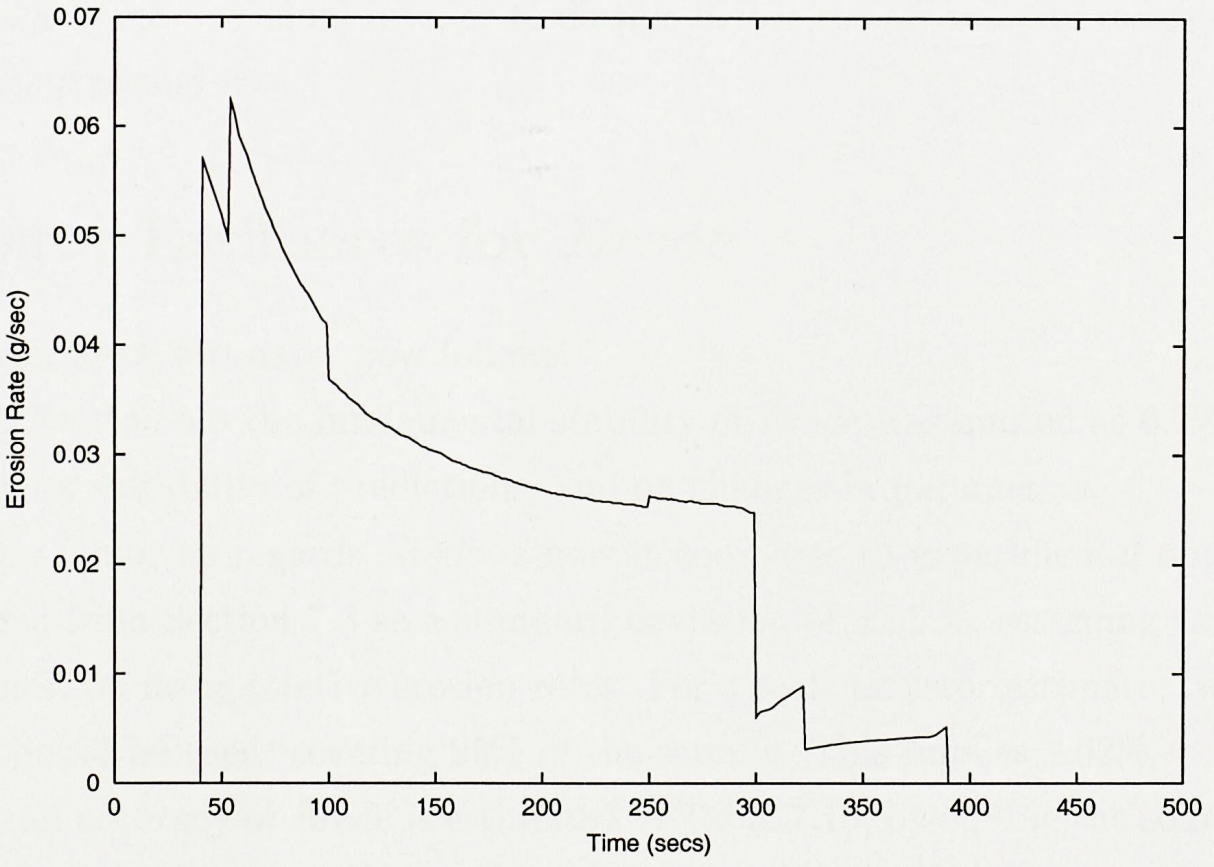


Table 7.14: Calculated Erosion Rates - Bend 2

Method	Mass Loss (g)	Rate (g/kg)	Err. (%)
Time Stepping	9.106	0.03452	-
Mean (25.0 m/s)	9.729	0.03688	+6.8

7.14) and air velocity in order to sum the amount of erosion expected. This value can be compared in Table 7.14 with an equivalent value obtained by crudely taking the mean air-velocity over the conveying period, (when mass is actually flowing- 41 to 320 seconds) and multiplying the erosion rate at this speed by the total mass conveyed (263.8 kg) in this batch. It is seen that the mean velocity prediction overestimates wear by 6.8 %, which is sufficiently close for present purposes. This error need not propagate into the overall error analysis however, as the calibration stage of the modelling process will ensure that this is taken account of.

Also plotted (Figure 7.15) is the erosion rate at each time step. Clearly, the most aggressive wear occurs in the start up period, though it was previously suggested that this would occur at the end of batch. The reason is that the air velocity is slow to respond to the increase or decrease of solids loading ratio. Hence, the SLR increases rapidly at the start of conveying, and the mass flow rate nears its maximum before the air has slowed a great deal. Conversely, at the end of the run, the mass flow rate drops suddenly, but the air does not accelerate quickly. This implies more material being conveyed at higher velocities at the start of the run, and a possible tactic to reduce wear might be to introduce the material at a more gradual rate, or to do this before the air velocity reaches maximum in the warm-up period.

7.5 Error Estimates for *Erode*

A summary of error estimates now follows.

Recall in Section 5.8 the fundamental stability of *Erode* was quoted as 6.7%. This was based on the repeatability of predictions, and on changes in parameters.

The other factor as regards *Erode* is how it compares to experimental data, and this can be quoted from Section 7.3 as a standard deviation of ±31 %, assuming calibration to the rig in question using relative erosion rates. For a realistic error estimate, two standard deviations should be used, covering 95% of the sample. This implies ±62%.

The overall accuracy of *Erode* is estimated in Table 7.15, by adding the sources of error

Table 7.15: Error Estimates for *Desktop Erode*

Factor	Error (%)
Stability of *Erode*	±6.7
c.f. Experimental	±62
Total *Erode*	±63

in quadrature.

7.6 Chapter Summary

Some experimental pneumatic conveyor test rig results are supplied in this chapter, sourced from project partners. Bend weights, wear penetration rates and bend lives are all scaled to account for the reuse of material in the conveyor, and its subsequent degradation and loss of erosivity. Burst points are shown where achieved, and calibration of particle initial transverse velocities is described. The blowtank method of conveying is investigated as regards the amount of wear experienced during the start-up and blowdown stages of the cycle.

The major finding in this chapter and this project was the discrepancy found between erosion rates in an erosion tester and rates of mass lost in a full size pneumatic conveyor. This was a surprising finding unmentioned elsewhere in literature, but nevertheless suggestions have been made as to the possible reasons. After inclusion of a factor to account for the much greater wear seen in the erosion tester (4.08) scaled definitive lifetime predictions from *Erode* with error estimates were provided.

Chapter 8

A Predictive Tool for Engineers

The mathematical modelling work described in previous chapters was carried out as a part of an *EPSRC Bend Wear Project*, the ultimate aim of which was to produce some simple desktop predictive software for non–mathematical modellers. This software would enable design or maintenance engineers to go about their work with more knowledge of how the bends in their pneumatic conveyor would last.

In this chapter the design of the software is discussed, and techniques for scaling predictions made with the virtual rig *Erode* are recounted.

8.1 Defining Needs

At an early stage in the project it was identified that there were two areas where a predictive tool could be used: design and maintenance.

Anyone maintaining an existing conveying rig would have far less flexibility in their options with respect to changing their system to reduce bend wear. Presumably the throughput of their system, and pipe bore used would be fixed. Also, of those variables that could be changed, the range of variation might be very limited.

A system designer will have much more power to consider and reduce bend wear, but this issue may not be the prime concern, and might perhaps be sacrificed to a reliable throughput rate. From the point of view of a predictive tool however, a reasonable lifetime prediction should be obtainable for the widest range of conveying circumstances possible, and for the greatest number of variables.

Variables accounted for in this tool are: conveying velocity; R/d ratio; pipe wall thickness; mass flow rate; pipe diameter; pipe material; conveyed material and bend orientation. Only a limited number of predictions were made in the project, so the next section de-

scribes how these were scaled to cover most of the pipes and conveying conditions that are used in industry.

Assuming good predictions, a tool which successfully covers all the variables described above should have wide applicability in industry.

8.2 Scaling Predictions

8.2.1 Velocity

The velocity range allowed is 5 to 40 m/s, which is an extension of the limited experimental data set in order to cover most realistic conveying conditions. The validation range in the experimental bends is 22.12 to 36.03 m/s. Data from erosion testers was obtained at 20, 30 and 40 m/s.

In the predictive tool, linear interpolations are made between two of six values in the velocity range recounted above.

8.2.2 R/d Ratio

The R/d ratio validation range was 6 to 14.15, but the range allowed for the user is 0.8 to 15.0. This should cover most pipe bends used, and although the extremes are not validated, the geometry of the bend is one of the strengths of the virtual rig. The resulting volume of material in the pipe is accounted for precisely using a toroidal geometry, so varying R/d is thoroughly modelled.

In the predictive tool, linear interpolations are made in the R/d range recounted above.

8.2.3 Pipe-Wall Thickness

Pipe wall thickness validations range from 2.39 to 16.3 mm, and the allowed range is 0.01 to 1200 mm. Virtual tests showed a linear dependence on wall thickness up to wall thicknesses of 30% of the pipe bore (Section 5.2). No experimental validation of this is available.

In the predictive tool, a linear dependence of bend life is assumed for wall thickness, using one prediction from the matrix, and also assuming zero bend life for zero wall thickness.

Table 8.1: Pipe Diameter and Bend Life: R/d of 14.15

Diameter (m)	*Erode* Life (hrs)	Assumed Quadratic (hrs)	Error (%)
0.053	2.380	2.380	0
0.100	8.190	8.473	3.46
0.200	34.823	33.891	-2.68
0.400	140.898	135.564	-3.79
0.800	735.222	542.257	-26.24

8.2.4 Mass-Flow Rate

Validation mass flow rates were at 0.93 and 3.67 *tonnes per hour*. The allowed range for users in the tool is 0.01 to 1000 *tonnes per hour*, as the assumption of linear dependence on this variable has been made throughout this work. The model unfortunately does not account for particle-particle interaction in any way. *Burnett's* work [Bur96] suggests that any effect on erosion rate is not major at lean phase flow conveying densities.

8.2.5 Pipe Diameter

Pipe diameter was validated only at 53 *mm* but can range from 3 to 2000 *mm* in the tool. These extrapolations are made using a quadratic dependence of life on diameter, as this reflects the increase in amount of material in the bend wall. This was confirmed by virtual predictions shown in Table 8.1

For very large pipe diameters the profile of the particle impacts begins to change, as the air has much more time to have influence. This is reflected in Table 8.1 as an increasing deviance from the virtual prediction. To overcome this, two different predictive matrices are used in the desktop tool: one for small–scale and one for large–scale pipes, the cut–off coming at 400 *mm*.

8.2.6 Materials

Pipe material and conveyed material are currently the most limited variables in the end product, as for each combination of these erosion tests are necessary. There is a facility however whereby a single erosion test at impact speed of 30 *m/s* and at an impact angle of 30^o can be employed to scale an existing prediction using olivine sand and one of the four default pipe materials. If a new pipe material is encountered, and it falls broadly into one of the present families of materials (rolled steel, cast metal, ceramic, or rock)

Table 8.2: Bend Orientation-Large Scale

Bend Orientation	*Erode* Life (hrs)	Deviance from Horiz. Plane (%)
Horiz. Plane	1243.86	-
Horiz. to Up	1493.45	+20.1
Horiz. to Down	1225.56	-1.5
Up to Horiz.	1366.74	+9.9
Down to Horiz.	1314.64	+5.7

Table 8.3: Bend Orientation-Small Scale

Bend Orientation	*Erode* Life (hrs)	Deviance from Horiz. Plane (%)
Horiz. Plane	4.870	-
Horiz. to Up	4.745	-2.6
Horiz. to Down	4.695	-3.6
Up to Horiz.	4.854	-0.3
Down to Horiz.	4.504	-7.5

this scaling feature should provide a reasonable estimate, though error bounds cannot be provided. *The Wolfson Centre* can provide particle characterisation erosion data if required (*wolfsonenquiries@gre.ac.uk*)

8.2.7 Bend Orientation

Shown in Tables 8.2 and 8.3 are data which represent variations in lifetime according to bend orientation under gravity. Given that the particle tracks are affected by gravity, the impact intensities are also likely to vary, hence variable lifetimes.

The first point is clearly that the orientation has a much greater effect on larger scales. This can be attributed to the fact that particles are travelling through a bend for much longer when it has a larger diameter, and given that the time frequency of wall collisions will be less, gravity has more time to take effect. Pulverised Fuel particles were used for these predictions, of diameter $75\mu m$.

These bend lives have not been validated. Although different bend orientations were incorporated into the laboratory rig at *The Wolfson Centre*, they were at different points in the rig, and the air speed through them was consequently different, due to the expansion of the air as it traverses the pipework. As a result, the lives of correspondingly oriented bends cannot be directly compared.

No clear pattern emerges from these predictions. It should be borne in mind that the estimated stability of *Erode* predictions is given as ±6.7% in Section 5.8, and only three of the above predictions exceed this. Nevertheless, this is the only data available on bend orientation, and it has been incorporated into the end product by using simple scaling factors, assuming the shifts seen above are consistent. For example, for a small scale bend, an *Up to Horizontal* bend is assumed to last 0.970 times the lifetime of the *Horizontal to Horizontal* bend, and for a large scale bend, a *Horizontal to Up* bend is assumed to last 1.201 times the life of a *Horizontal to Horizontal* bend.

8.3 Interface Design

Given the mechanics of the desktop software, the means of communicating with the user was the next issue. This was required to be as simple and direct as possible, and also to be a *Microsoft* compatible product, to allow the majority of commercial PCs to run it.

A tool based on dialog boxes was considered suitable, and the first version was produced in *Visual C++*. This used several dialogs, and outputted the resulting lifetime prediction also in a dialog. Feedback from collaborators suggested that being able to print the screen showing all conditions provided to supply the prediction would be more useful.

The next version used the integrated development environment in *Excel 97*, whereby *Visual Basic* dialogs were used for data input and selections, but *Excel* spreadsheets were used for the output. Advantages here were that the screen could easily be printed showing all data, as there is more room on a spreadsheet. Also, other sheets were incorporated into the product to increase the functionality: a plotting facility existed whereby the user could drill a hole through the lifetime matrix for a particular range of a variable, and automatically generate a graph with axes labelled; the data used for this was also dumped onto a sheet, such that a different plotting method could be used if desired; a default sheet allowed the user to change the wake up values in the drop-down combo boxes on the input forms; and hidden sheets facilitated the storage of the data matrices from which all predictions were extracted.

A fully–functioning version incorporating dummy predictions was sent out to the project's collaborators in June of 2001, along with a brief questionnaire. The feedback was mostly positive, and some additions were made as a result: more space was supplied on the result sheet to insert project names, sites, bend location and number etc.; another sheet was incorporated to store relative erosivities of materials; also some final adjustments to the variable ranges were made. There was disappointment, shared by the collaborators, that

no erosion data for pulverised fuel (finely crushed coal) would be incorporated into the tool in its first incarnation. This was due to time constraints and the dangers of using highly inflammable p.f. with erosion testers.

Given the three-year timescale of the entire project, the tool was quickly amended to a finished state. Some final user testing could then be performed.

The finished tool is called *Desktop Erode*, and is available from the University of Greenwich. Contact *Dr. Mayur Patel* at the *Centre for Numerical Modelling and Process Analysis* for more details.

Also, any feedback from users is most welcome. Please send your comments or suggestions likewise. Numerical comparisons of *Desktop Erode* predictions with real conveyors are especially welcome.

8.4 Chapter Summary

In this chapter is described the conception, planning and implementation of a simple desktop bendwear predictive tool.

Methods for interpolating and scaling predictions from fixed initial matrices of data are described and justified wherever possible, either from experimental results or from virtual predictions. The ranges of each input variable considered was decided upon by consultation with potential industrial users.

A version of the tool loaded with dummy data was tested by industrial project collaborators, and some adjustments were made. On the receipt of final mathematical modelling results, definitive predictions were inserted into the internal matrices. The current tool (Version 1.0) though as yet untested in the field is to be made available through The University of Greenwich.

Chapter 9

Summary

A mathematical model has been produced to predict the life of bends in pneumatic conveyors, subject to erosive wear. This model contributed a novel technique of particle projection through a toroidal geometry to optimise computation, and employed *Monte Carlo* techniques.

Particle impact data was obtained using CFD models, and filtered into a form whereby particle impacts could be randomly generated in a realistic fashion.

The geometry of the pipe bend in question was modelled by custom software *Erode*, and many thousands of particles were then generated until a hole was worn right through a tile-grid, representing the pipe wall.

All rebound angle, restitution and erosion data was sourced from experiments, carried out as a part of this project. Use of a scale-factor with the erosion rate allowed lifetime predictions to be made using under a million particles, with around 20 minutes of computation on a DEC Alpha 878 (433 MHz).

9.1 Conclusions

The erosion model *Erode* works on simple geometrical and physical principles, and as would be expected, produces repeatable results when used sensibly, to a quoted stability of $\pm 6.7\%$.

An unanticipated problem with the model was the dependence of particle impacts (and hence bend life) on the initialized transverse velocity components of the particles, input at the start of the CFD particle-tracking stage. Little data was found to help here, as most LDA–type work is aimed at measuring particle speed along the main axis of travel. It was therefore decided to calibrate this factor, and the result suggested the standard deviation

of the ratio of transverse components to main components should be 8.37%.

A surprise arose out of the decision to weigh the test rig bends. Calculating the mean mass loss per kg of material conveyed, this was around a factor of 4.08 lower than when measured in two types of erosion tester. This kind of discrepancy has not been reported elsewhere as far as is known. The mismatch is perhaps the most significant finding to emerge from the experimental work, but for the purposes of prediction, this factor has simply been incorporated into *Erode*, such that the matrix of predictions in *Desktop Erode* are calibrated to *Tong Deng's* rig.

It may be that each conveyor relates to erosion tester data uniquely, but this can only be established by further results. As long as each conveyor relates consistently to erosion tests, then the predictions in *Desktop Erode* will have much value once scaled to a particular conveyor. Explanatory notes to this effect are present in the tool.

Having calibrated the tool to *Deng's* rig, the accuracy over a sample of six predictions and validation points, can be quoted as ±62%. The overall accuracy of the predictions can be quoted as ±63% accurate.

Again, it must be stressed that this is true for the conveyor to which the predictions have been calibrated. One other data point, from a different conveyor with a different mass–flow rate, did not fit this calibration, and was distant by a factor of around four. Speculating, if a user finds his burst points occurring consistently above or below the predictions, it is suggested that a recalibration could be required.

9.2 Research Objectives

Here the questions asked in Section 1.2 are revisited, recounting the contributions made by this work.

How predictable are bend lives?

Replacing bend 2 at around the halfway point of *Deng's* experimental conveying suggests that bend life, in what were apparently *exactly the same conveying conditions*, can vary by 30% nevertheless. This allows for a small difference in pipe wall thickness, and sets a lower limit for the accuracy of any predictive technique. The overall accuracy of the predictions made by this work should therefore be seen in this context: a variation of ±30% of the quoted ±63% accuracy could be inherent with the conveying system used.

What is the best way to model bend wear?

It is clearly beneficial if using CFD in an erosion model, to separate the erosion modelling from the CFD code. One good reason for this is that commercial codes cannot be accessed everywhere that there might be a need to; another is that an erosion model tied in computational time to a CFD and particle tracking simulation is likely to be a very slow erosion model. A downside of the separation process however, is that much time needs to be expending in programming a custom erosion model for the geometry in question. The idea of code to handle a general geometry is most attractive, but was well beyond the scope of the present project.

Empirical erosion data was used in this work, but this requires time, labour, and access to an erosion tester.

Does surface evolution change the rate of wear significantly?

In this particular application with pipe bends, it is not a critical factor, but data shows (Figure 6.16) that ignoring the fact in the modelling process could introduce something like 15% of lifetime error for a long–radius bend, and less for shorter radii. Different processes using components with sharper corners could potentially exhibit greater sensitivity to the evolving impact angle trials performed in this work.

9.3 Future Work

The CFD particle tracking employed in this work was undoubtedly useful, but considering that dense particles travel mainly under inertia, particle shape and collisions would be very relevant to predicted tracks. In the CFD model used here, these are some of the weakest areas covered, and expectedly so, as they are so computationally intensive. As computational power increases, this erosion model may be revisited with more sophisticated particle impact data.

Recall that the calibration process was initiated to take account of initial particle velocities. If the computational power existed to track large samples of these particles down a long straight of pipe before the bend, and if particle shape and collisions were able to be accounted for, then the particle initial velocities would be far less relevant to the impacts.

Also in this work, an interesting geometrical idea was presented whereby particles could be tracked through a toroidal geometry using an approximation technique, such that the full equations of a line meeting a plane did not have to be solved. This appears most useful, and used within its limits can save considerable computation. Perhaps a geometrician will want to document this technique further in the future.

With the discovery of a mismatch in erosion rates, of around a factor of four between *Deng's* rig and erosion tests, a rich vein of work for the experimentalist is opened up. Hopefully others will now build full–scale test rigs, as this seems the only way to assess whether or not this factor is consistent or varying. *Desktop Erode* has a facility built–in in anticipation of further experimental data being obtained. The present investigative team are planning tests involving measurements from full scale rigs to establish the following: particle velocity as a proportion of air–speed, as solids loading ratio varies; and wear rate in a real bend, again, as the solids loading ratio changes.

Hopefully users of conveyors (often not the suppliers of the equipment) will also take more seriously the logging of conveying conditions, possibly weighing their bends, as this kind of information could be very powerful at a later date.

Undoubtedly other materials will be characterised on erosion testers, and these erosivities can then be employed in the desktop tool.

When users of the desktop tool start to get a feel for its performance over perhaps a year or more of use, the feedback from them will ultimately dictate further work on this project, or similar ones.

Appendix A

Toroidal Volumes

This section covers the details of how the volume of a torus can be calculated, and hence that of an individual tile, as used in this model.

A.1 Volume of a Torus

The volume is calculated using the Volume of Revolution technique, adding up the volume of a series of infinitesimally thin discs, or in our case, rings. First split the equation of a circle (Fig. A.1) into its two solutions for y:

$$x^2 + (y - R_o)^2 = a^2 \tag{A.1}$$

$$\Rightarrow \quad f(x) = R_o + \sqrt{a^2 - x^2} \tag{A.2}$$

$$\text{and} \quad g(x) = R_o - \sqrt{a^2 - x^2} \tag{A.3}$$

The volume created by the revolution of $f(x)$ about the x-axis minus the volume created by the revolution of $g(x)$ will result in the volume of a torus. So the integral

$$\int_{-a}^{+a} \pi y^2 \, dx \tag{A.4}$$

$$\text{here becomes} \quad \int_{-a}^{+a} \pi (f^2(x) - g^2(x)) \, dx \tag{A.5}$$

$$= \pi \int_{-a}^{+a} 4R_o \sqrt{a^2 - x^2} \, dx \tag{A.6}$$

with fortunately, many terms cancelling.

Now putting $x = a \sin p$, $\frac{dx}{dp} = a \cos p$, with $p = +\frac{\pi}{2} \Rightarrow x = +a$, and $p = -\frac{\pi}{2} \Rightarrow x = -a$, gives

$$\pi \int_{-\frac{\pi}{2}}^{+\frac{\pi}{2}} 4R_o a^2 \cos^2 p \, dp \tag{A.7}$$

$$= \pi \int_{-\frac{\pi}{2}}^{+\frac{\pi}{2}} 2R_o a^2 (1 + \cos 2p) \, dp \tag{A.8}$$

$$= 2\pi R_o a^2 [p + \frac{1}{2} \sin 2p]_{-\frac{\pi}{2}}^{+\frac{\pi}{2}} \tag{A.9}$$

$$= 2\pi^2 R_o a^2 \tag{A.10}$$

A.2 Volume of a Toroidal Tile

Before this torus can be sliced up into tiles using symmetry, notice that the volume of the inner half of the torus is less than that of the outer half. Using the simple function $h(x) = R_o$, calculate the volume of each half separately:

$$V_{out} = \int_{-a}^{+a} \pi y^2 \, dx \tag{A.11}$$

$$= \int_{-a}^{+a} \pi (f^2(x) - R_o^2) \, dx \tag{A.12}$$

$$= \pi \int_{-a}^{+a} 2R_o \sqrt{a^2 - x^2} \, dx + \pi \int_{-a}^{+a} a^2 - x^2 \, dx \tag{A.13}$$

and using the previous result for the left hand term,

$$V_{out} = \pi^2 R_o a^2 + \pi [a^2 x - \frac{1}{3} x^3]_{-a}^{+a} \tag{A.14}$$

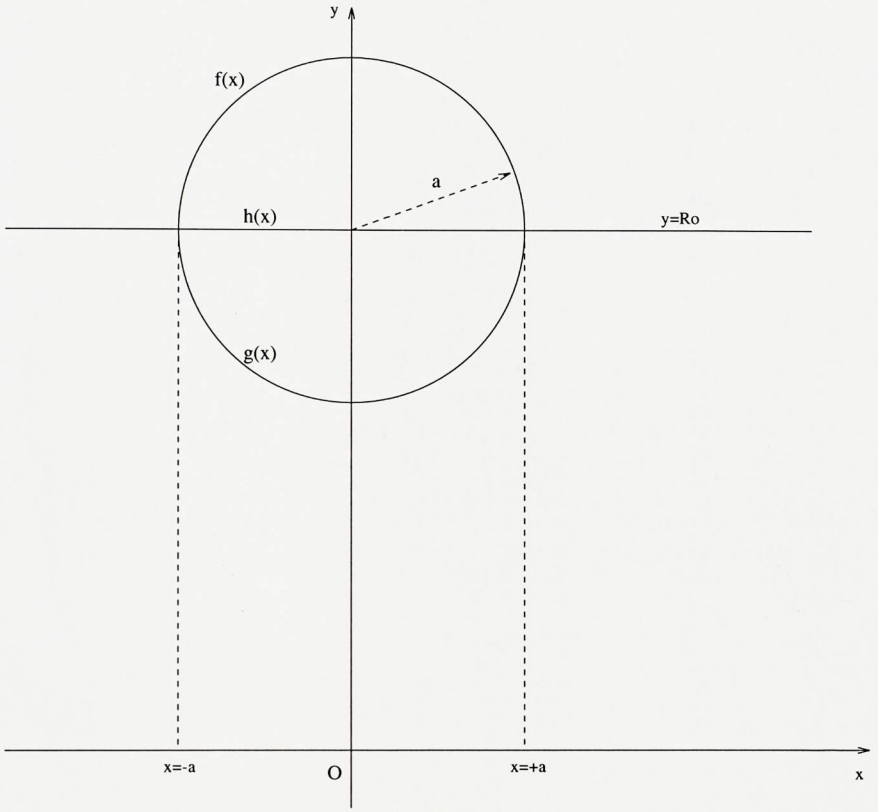
$$= \pi^2 R_o a^2 + \frac{4}{3} \pi a^3 \tag{A.15}$$

which is half of the torus volume plus that of a sphere of radius a.
Now for the inner half,

$$V_{in} = \int_{-a}^{+a} \pi y^2 \, dx \tag{A.16}$$

$$= \int_{-a}^{+a} \pi (R_o^2 - g^2(x)) \, dx \tag{A.17}$$

Figure A.1: Section through a Torus



$$= \pi \int_{-a}^{+a} 2R_o\sqrt{a^2 - x^2}\, dx - \pi \int_{-a}^{+a} a^2 - x^2\, dx \tag{A.18}$$

which differs from Equation (A.13) by only a sign, and hence will give

$$V_{in} = \pi^2 R_o a^2 - \frac{4}{3}\pi a^3 \tag{A.19}$$

Note that $V_{out} + V_{in} = 2\pi^2 R_o a^2$, which matches the initial result.

Now taking the radial position and thickness of a tile, the volumes of two outer torii (V_{out} as opposed to V_{in}) can be subtracted from each other in order to obtain the volume of a shell. This can then be divided up as a fraction of 2π in the ϕ direction, and as a

fraction of π in the θ direction, in order to achieve the volume of an individual tile.

Appendix B

Line Approximation

This section will demonstrate the technique of approximating a straight line in toroidal coordinates, and will give an idea of the magnitude of the errors involved.

B.1 Technique

These are the equations of the line approximation in toroidal geometry:

$$r\cos\theta + R_o = \frac{D}{\cos(\phi - \beta)} \tag{B.1}$$

$$r\cos(\theta - \alpha) = d \tag{B.2}$$

Recall that the first of these is not an approximation, but represents the line in question projected onto the ϕ plane. The equation is formed as in polar coordinates, by using the perpendicular from the origin to the point of closest approach of the line (distance D, angle β) and forming a triangle with the general point on the line. The adjacent side of this triangle is fixed at D, and the equation simply says *cosine = adjacent / hypotenuse*, where the hypotenuse is of length $R_o + r\cos\theta$.

To obtain the second equation, the same technique is used, but projecting onto the (r, θ) plane instead. Note that there is no single (r, θ) plane, but one at every value of ϕ, and the approximation is therefore made by assuming that the resulting projection of points onto this plane results in a straight line. The actual result of this is shown in Figures B.1 and B.2, the first for a line that does not cross the pipe bore in the plane of ϕ, and the second for a line that does. There are other special cases if the Cartesian equation of the line enjoys either of x, y or $z = k$, but these are tweaked by this model into the general

Figure B.1: Straight Line Projected Into Chosen (r, θ) Plane

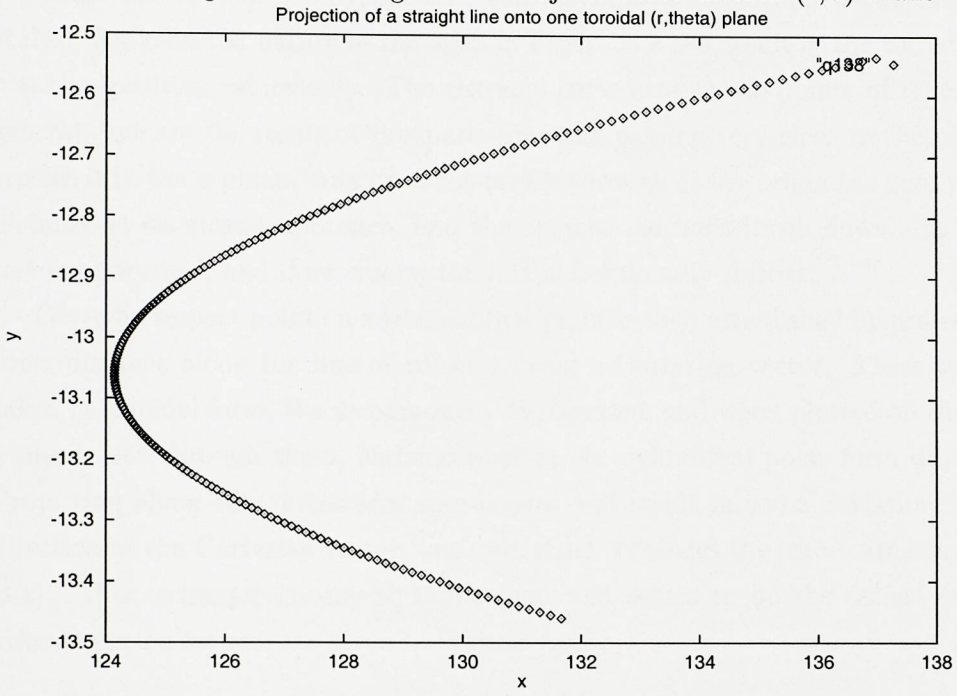
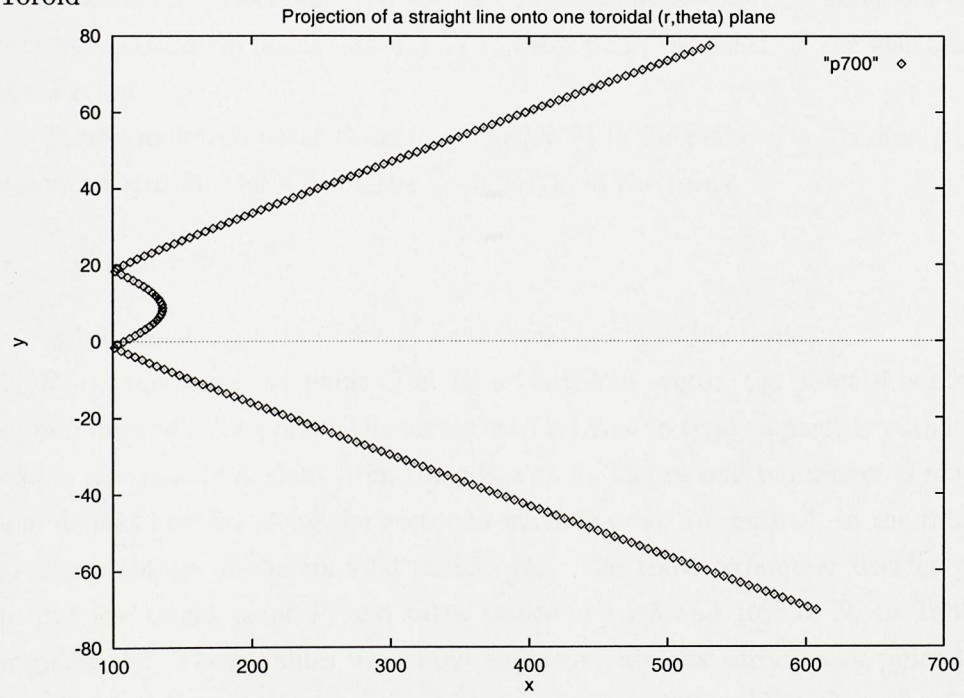


Figure B.2: Straight Line Projected Into Chosen (r, θ) Plane: Crosses Circle Origin of Toroid



form.

These curves look like hyperbolae, with asymptotic behaviour clearly visible on large scales. The reflected nature of the apex in Figure B.2 is a result of the toroidal coordinate r taking positive values only. The extreme curvature at the points of reflection are not general, but are the result of this particular line passing very close to the pipe bore as it crosses it in the ϕ plane. Any line that passes through either origin has zero perpendicular distance at its closest approach, and the current methods break down. To avoid this, a test is performed, and if necessary, the line is fractionally shifted.

Given an impact point on a tile, another point is then established by projecting forward some distance along the line of velocity using a Cartesian vector. These two points are taken in toroidal form, the ϕ component disregarded, and when plotted on the (r, θ) plane, a line is put through them, and recorded in the cylindrical polar form described above. Projecting along this in toroidal coordinates will result in some deviation from the true direction of the Cartesian vector, but over short distances the errors are small (e.g. Table B.1). This technique is easy to implement, and seems to be the easiest (and probably fastest) option for reasons given in Section 4.4.5.

B.2 Error Testing

Four Tables (B.1–B.4) are given that describe attempts to project along our toroidal geometry straight line approximation, and to hit a point obtained by the equivalent Cartesian translation.

Taking an initial point O at the point $(R, \frac{\pi}{4})$ in the plane of ϕ, the first parameter *start* describes how far this is from the circle origin of the torus

$$R = (1 + \text{start})R_o \tag{B.3}$$

where recall, R_o is the radius of the circle origin of the torus.

Projecting from the point O along a Cartesian vector, the point A is obtained as our second interpolation point. The vector used is taken to typify a particle path in a pipe bend, and is not greatly deviant from the plane of ϕ. The second parameter is now introduced, and defines how far along the vector to move in order to reach A. In the tables it is given as a percentage of the toroidal radius, R_o. The third parameter decides how far away to put the target point P, and takes values of 1,2,5 and 10% of R_o in Tables (B.1–B.4) respectively. These values were used to encompass the variation of pipes that are used in industrial pneumatic conveyors, where a long–radius mild steel bend might have wall

thickness of 0.5% of the bend radius. If at the other extreme wall thicknesses are going beyond 10%, it would be a simple matter to use the first line approximated over a fraction of the thickness, and then to update the line approximation, using the same Cartesian velocity vector, but from the most recent position.

The interpolation points *(O,A)* are projected onto the (r, θ) plane in toroidal coordinates, and the line approximation is constructed as previously described. The target point P is obtained and converted into toroidal coordinates (r_p, θ_p, ϕ_p). The solver now projects along the toroidal line approximation until it hits each of the above constants, and in each case, the point obtained is converted back into Cartesian coordinates. If these points are described as *R,S,* and *T*, the moduli of the vectors $\vec{RP}$, $\vec{SP}$ and $\vec{TP}$ are found, and the largest of these recorded. What appear in the tables are the percentage sizes of these largest moduli against the modulus of the step taken, *OP*.

Describing the errors in this way is more appropriate than quoting angles, because when the vector in question is nearly parallel to a line of ϕ or θ, the tiniest difference in angle might result in a deviation of kilometres. A rough guide however would be that the percentages given divided by one hundred would approximate the tangent of the deviation, so that 1% implies 0.57^o, and 3% implies 1.72^o.

B.3 Data Analysis

As expected, projecting further away results in greater errors. Note also the tiny errors around 10^{-11} or smaller. These coincide with the interpolation points used to define the line approximation, and once passed, the errors begin to grow again.

The dependence upon Interpolation Step (IS) shows that the best way to project a fixed distance, is to use an interpolation step of a similar size. Using progressively larger or smaller steps worsens the error rate.

The dependence upon Start Position (SP) is harder to explain, as for a fixed IS error increases with SP. As the SPs in question were receding from the torus origin, it might have been expected that because they were therefore interpolating a straighter part of the curves shown in Figures B.1 and B.2, accuracy would have increased, whereas the opposite was observed.

However, recall that only the largest error was recorded out of three possibilities; consistently the θ_p error. Only one line was investigated but undoubtedly each set of errors is line–specific. In this case the other two errors were diminishing as SP increased, but because the vector was reasonably close to parallel to a line of constant θ in the coordinate

Table B.1: % Error: Vector Projection along 1% of Bend Radius

Interpolation Step (%)	Start Position (%)			
	1.0	2.0	5.0	10.0
0.01	.0885	.125	.208	.284
0.1	.0804	.114	.189	.258
0.5	.0445	.0630	.105	.143
1.0	9e-11	7e-11	3e-11	6e-11
5.0	.339	.482	.803	1.10
10.0	.725	1.03	1.73	2.39

Table B.2: % Error: Vector Projection along 2% of Bend Radius

Interpolation Step (%)	Start Position (%)			
	1.0	2.0	5.0	10.0
0.01	.228	.298	.451	.591
0.1	.217	.284	.431	.564
0.5	.171	.224	.339	.444
1.0	.113	.148	.225	.295
5.0	.326	.428	.651	.857
10.0	.827	1.08	1.66	2.20

system, this error dominated throughout by factors of 2 to 8.

Note that in the model, because our points are always at least a pipe bore away from the pipe centre ($r = 0$) the part of the projection with highest curvature should be avoided.

B.4 Example: Using *Erode*

Shown in Figures B.3 and B.4 is an example of a line tracked through a virtual pipe-wall using the approximation technique inside *Erode*. In order to maximize the chances of seeing the curvature, this vector was manipulated to strike the pipe wall at an extremely low angle. Using a pipe wall thickness of 2.4mm, the plotted track is 4cm long.

Playing with the scale in Figure B.3, the curvature is plain to see. Out of interest, when the u component of velocity had its sign swapped, the mirror image of this track was obtained. Plotted on axes with equal scales, the line is revealed to be practically straight. The deviation that occurred was estimated to be 0.35 mm in 4 cm, or around 0.5^o.

Table B.3: % Error: Vector Projection along 5% of Bend Radius

Interpolation Step (%)	Start Position (%)			
	1.0	2.0	5.0	10.0
0.01	.947	1.08	1.37	1.63
0.1	.929	1.06	1.35	1.60
0.5	.850	.972	1.23	1.46
1.0	.752	.861	1.09	1.30
5.0	1e-11	5e-12	5e-12	6e-13
10.0	.863	.992	1.27	1.52

Table B.4: % Error: Vector Projection along 10% of Bend Radius

Interpolation Step (%)	Start Position (%)			
	1.0	2.0	5.0	10.0
0.01	3.03	3.18	3.47	3.70
0.1	3.00	3.15	3.44	3.67
0.5	2.88	3.01	3.29	3.51
1.0	2.71	2.84	3.10	3.31
5.0	1.46	1.53	1.67	1.79
10.0	5e-12	2e-12	1e-13	1e-12

Figure B.3: False Scale

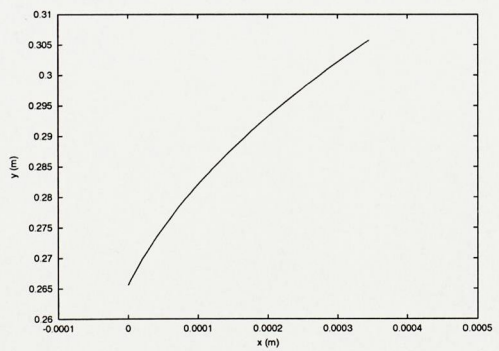
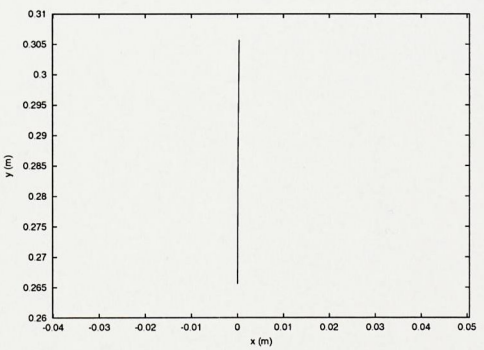


Figure B.4: True Scale



B.5 Summary

The data in this appendix is far from exhaustive, but it gives some insight into the behaviour of the approximation as made. The errors look acceptable, especially given that all positions and velocities are randomly generated in any case. It is not clear if one particular bias results from this approximation, or whether the errors are uniformly spread. This is an interesting field for further research, though its application currently looks specific to the toroidal geomety used. Any technique that can speed up or simplify mathematical modelling will only increase in value as the field expands in line with the availability of processing power.

Appendix C

Construction of a Cartesian Plane

C.1 Equation of a Plane

Given a point in space $\vec{A}(a_1, a_2, a_3)$, and a vector passing through it $\vec{N}(n_1, n_2, n_3)$, a unique plane containing $\vec{A}$ can be defined perpendicular to the vector, which can then be considered the *normal* to the plane.

Take a general point $\vec{P}(x, y, z)$ in this plane, and defining the vector $\vec{AP} = (x - a_1, y - a_2, z - a_3)$, the scalar product of this vector and the normal must always be zero, as these two are always perpendicular.

Hence

$$\vec{N}.\vec{AP} = 0 \tag{C.1}$$

$$(n_1, n_2, n_3).(x - a_1, y - a_2, z - a_3) = 0 \tag{C.2}$$

$$n_1 x + n_2 y + n_3 z = d \tag{C.3}$$

where $d = a_1 n_1 + a_2 n_2 + a_3 n_3$. Equation C.3 is the Cartesian equation of a plane.

Appendix D

Experimental Data Set

This is a summary of the experimental data incorporated into *Erode*. The data used was taken from more extensive work carried out by *Dr. David Allsopp* and *Professor Ian Hutchings* at The University of Cambridge, as a part of this *EPSRC*–funded project. The data shown here was used to produce realistic erosion, rebound and restitution algorithms for *Erode*.

D.1 Erosion Rates

Figures D.1 to D.4 show erosion data as gathered by a gas–blast type of tester. Each material was investigated at three velocities and three angles of impact.

D.2 Rebounds

Figures D.5 and D.6 show the distribution of speeds and angles from the horizontal for the incoming plume used in the rebound experiments. The data was gathered using multiflash photography, and samples of around a hundred particles were used for each distribution.

Figure D.7 shows a rebound angle distribution for each of the four target materials used: Alumina, Basalt, Mild Steel, and Ni–Hard. Note that the target was orientated to the incoming plume at 30^o, so that the modal rebound angle for an incoming angle of 30^o is around 10^o; that is, 30^o must be subracted from the measured and plotted angle to obtain the true angle of rebound. Clearly some particles then show a small, negative rebound angle, but this can be attributed to the later influence of gravity.

Figure D.8 shows distributions for the speeds of these same particles, for the four different target materials.

Figure D.1: Mild Steel at 20,30 and 40m/s

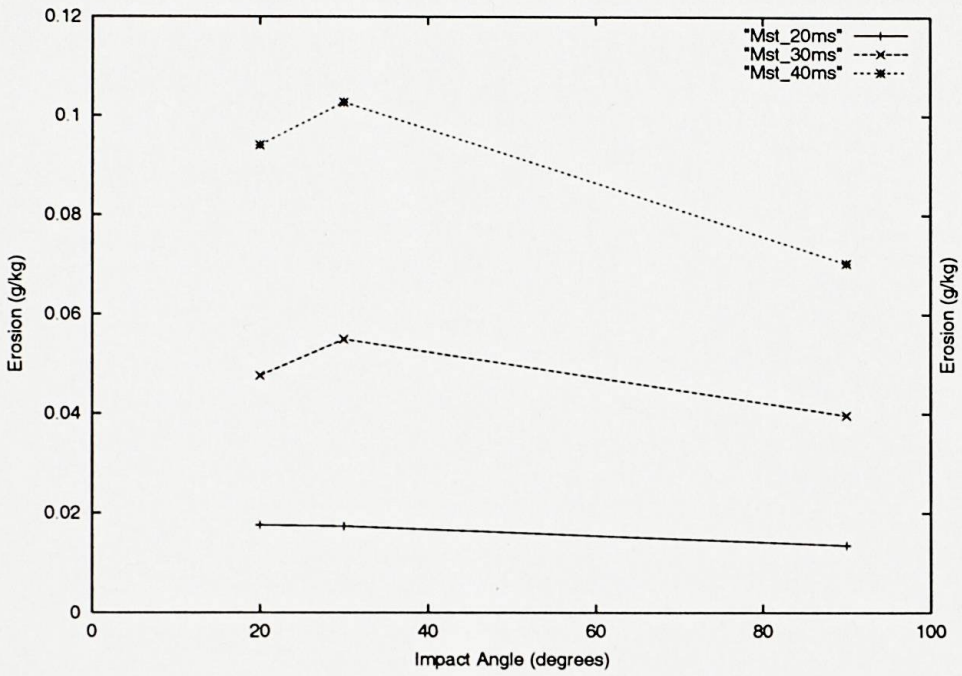


Figure D.2: Ni-Hard at 20,30 and 40m/s

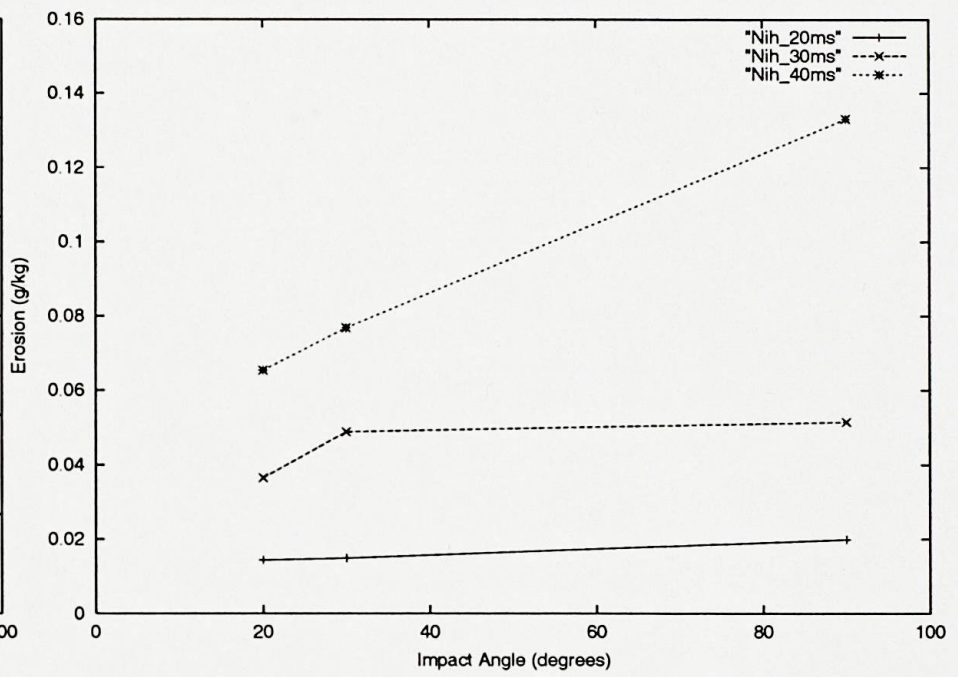


Figure D.3: Basalt at 20,30 and 40m/s

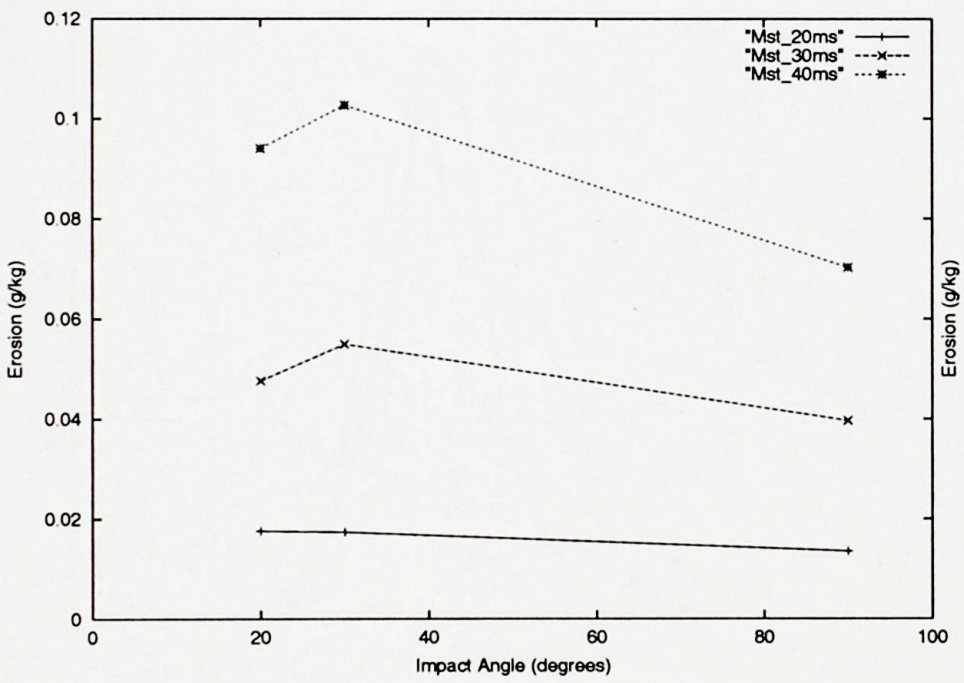


Figure D.4: Alumina at 20,30 and 40m/s

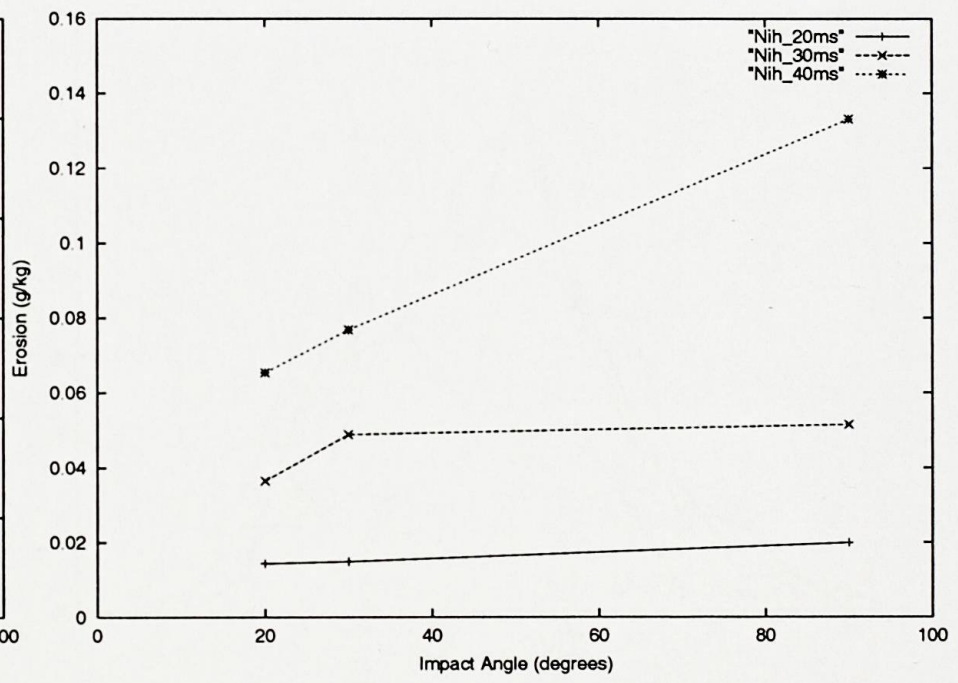


Figure D.5: Incoming Plume: Angles

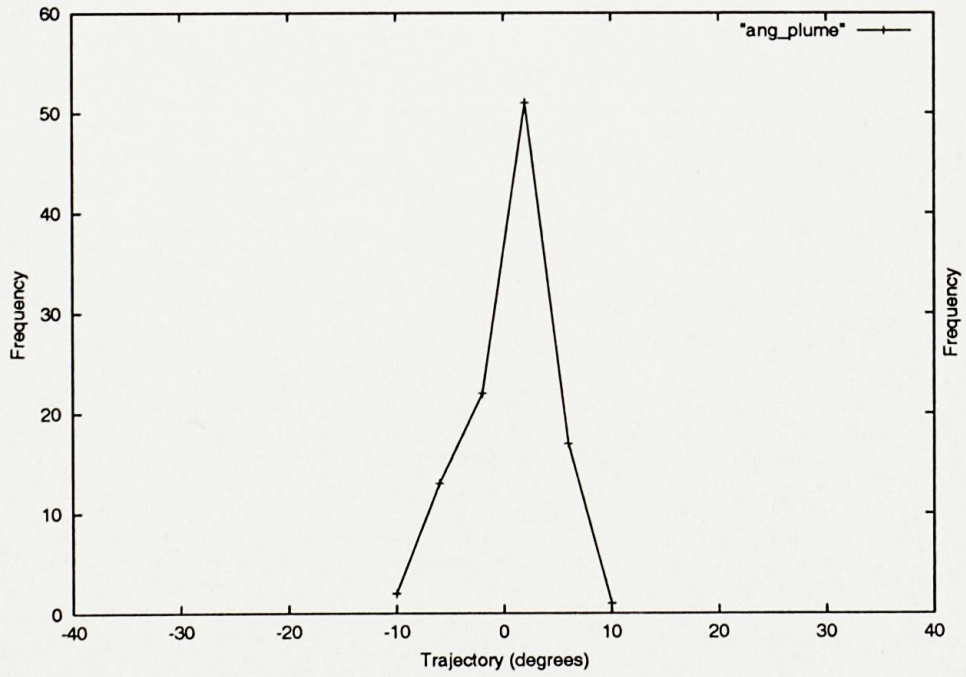


Figure D.6: Incoming Plume: Speeds

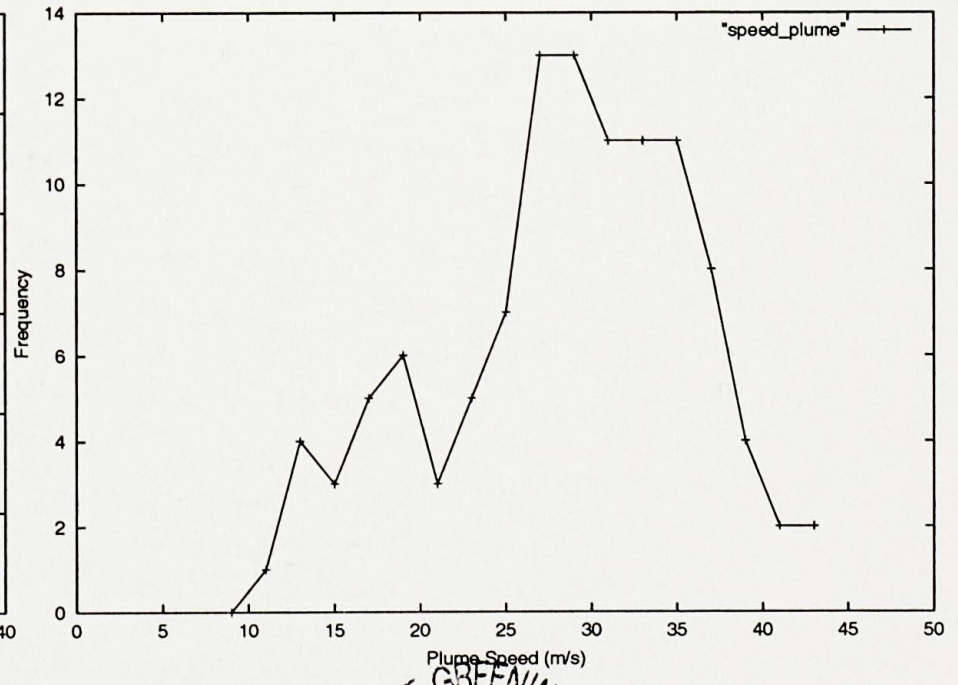


Figure D.7: Rebounds: Angles

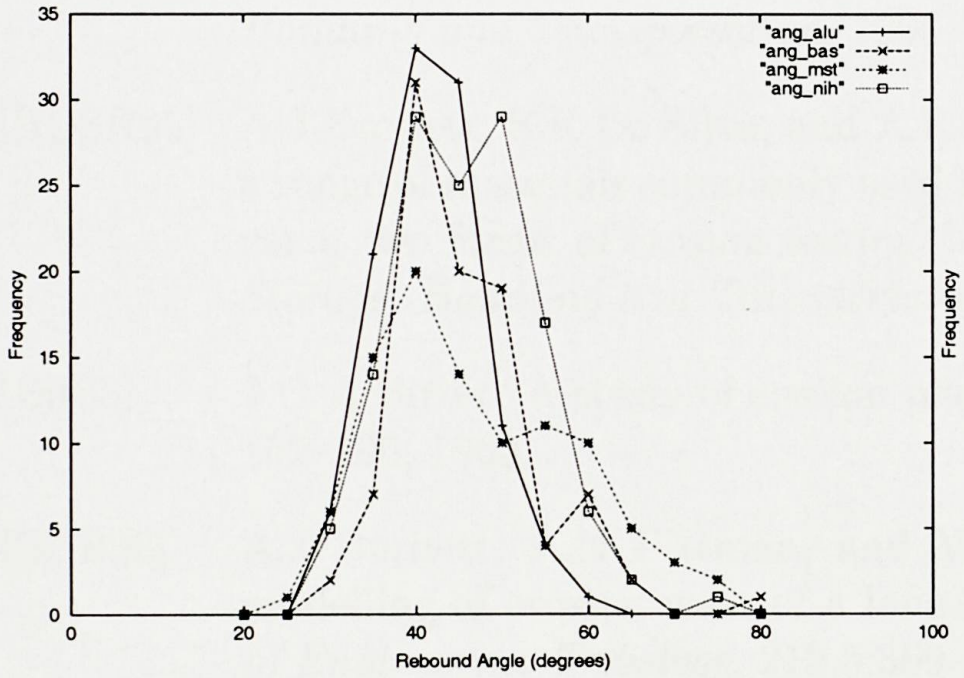
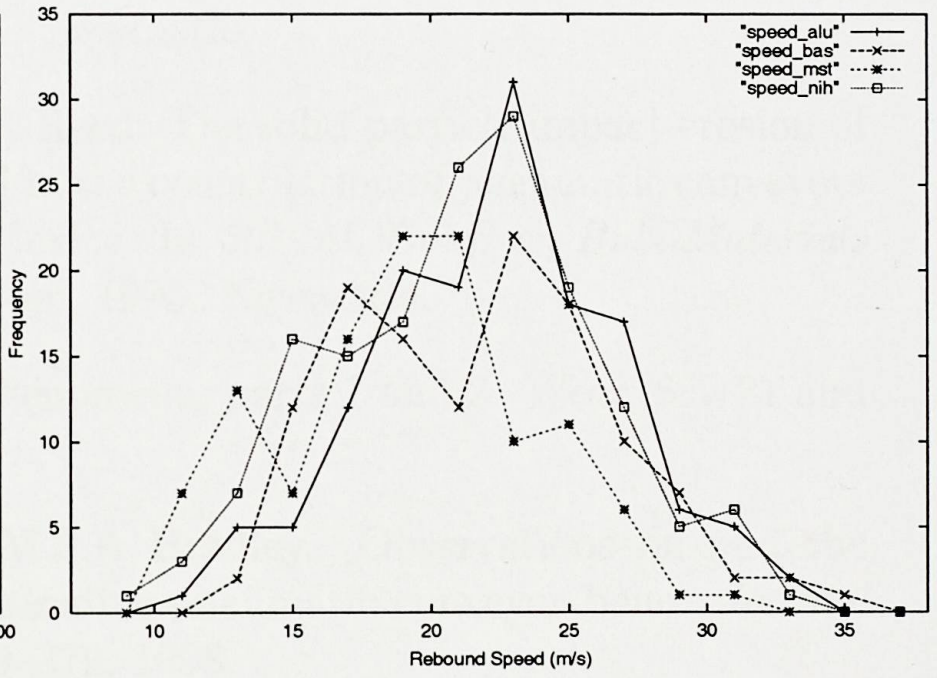


Figure D.8: Rebounds: Speeds



Bibliography

[AH83] D.R Andrews and N Horsfield. Particle collisions in the vicinity of an eroding surface. *J.Phys. D:Appl.Phys.*, 16:525–538, 1983.

[Bah91] S Bahadur. The structure of erosive wear models. *Tribological Modeling for Mechanical Designers*, ASTM STP 1105, 1991.

[BBB95] A.J Burnett, M.S Bingley, and M.S.A Bradley. An investigation of the low velocity/low concentration solid particle erosion of a structural mild steel using a centripetal erosion tester. In *5th Int. Conf. on Bulk Materials Storage, Handling and Transportation*, 1995. Newcastle.

[BDSR95] A.J Burnett, S.R De Silva, and A.R Reed. The solid particle impact erosion of a range of materials commonly used in the construction of pneumatic conveyors using two forms of erosion testing device. In *5th Int. Conf. on Bulk Materials Storage, Handling and Transportation*, 1995. Newcastle.

[Bit63] J.G.A Bitter. A study of erosion phenomena, parts 1 and 2. *Wear*, 6:5–21 and 169–190, 1963.

[BPB98] A.J Burnett, A.N Pittman, and M.S.A Bradley. Observations on and the modelling of erosion wear of a long-radius pneumatic conveyor bend. *Journal of Engineering Tribology*, 212,5:369–379, 1998.

[Bur96] A.J.B Burnett. *The Use of Laboratory Erosion Tests for the Prediction of Wear in Pneumatic Conveyor Bends.* PhD thesis, University of Greenwich, 1996.

[BWBB95] I Bridle, S.R Woodhead, A.J Burnett, and R.N Barnes. A review of techniques for the investigation of particle degradation in pneumatic conveying systems. In *5th Int. Conf. on Bulk Materials Storage, Handling and Transportation*, 1995. Newcastle.

[CH83] A.K Cousens and I.M Hutchings. Influence of erodent particle shape on the erosion of mild steel. In *Proc. 6th Int. Conf. on Erosion by Liquid and Solid Impact*, 1983.

[CSS77] C.T Crowe, M.P Sharma, and D.E Stock. The particle-source-in cell (PSI-CELL) model for gas-droplet flows. *J. of Fluids Engineering*, 99,2:325–332, 1977.

[CW85] C.P Chen and P.E Wood. A turbulence closure model for dilute gas-particle flows. *The Canadian Journal of Chemical Engineering*, 63:349–360, 1985.

[DBB01] T Deng, M.S Bingley, and M.S.A Bradley. Influence of particle dynamics on erosion test conditions within the centrifugal accelerator type erosion tester. *Wear*, 249,12:1059–1069, 2001.

[Den01] T Deng. *The Influence of Particle Dynamics on Erosion Caused by Solid Particles within Laboratory Erosion Testers.* PhD thesis, University of Greenwich, 2001.

[Eas98] W.J Easson. DTI Coal R&D Programme - Quarterly Progress Report, Oct. 1998.

[EGR78] A.G Evans, M.E Gulden, and M Rosenblatt. Impact damage in brittle materials in the elastic-plastic response regime. In *Proc. R. Soc.(London)A*, volume 361, pages 343–365, 1978.

[Ell93] R.R.R Ellermaa. Erosion prediction of pure metals and carbon steels. *Wear*, 162-164:1114–1122, 1993.

[FA96] M Founti and Klipfel A. Numerical simulation of pneumatic transport of pulverised lignite and induced erosive wear in the distribution ducts from the mills to the furnace burners in large power plants. In *Erosive Processes, ASME FED*, volume 236, pages 717–724, 1996.

[FA01] A Flodin and S Andersson. A simplified model for wear prediction in helical gears. *Wear*, 249:285–292, 2001.

[Fin60] I Finnie. Erosion of surfaces by solid particles. *Wear*, 3:87–103, 1960.

[Fin65] I Finnie. Formation of surface ripples. *Wear*, 8:60–69, 1965.

[Fin72] I Finnie. Some observations on the erosion of ductile metals. *Wear*, 19:81–90, 1972.

[FTH98] A Forder, M Thew, and D Harrison. A numerical investigation of solid particle erosion experienced within oilfield control valves. *Wear*, 216:184–193, 1998.

[GI83] A.D Gosman and E Ioannides. Aspects of computer simulation of liquid–fueled combustors. *Journal of Energy*, 7(6):482–490, 1983.

[GM91] E Galea and NC Markatos. The mathematical modelling and computer simulation of fire development in aircraft. *Int. J. of Heat and Mass Transfer*, 34(1):181–197, 1991.

[Has88] M Hashish. An improved model of erosion by solid particles. In *Proc. 7th Int. Conf. on Erosion by Liquid and Solid Impact*, 1988. Paper 66.

[HKS00] A Huser, O Kvernvold, and T Søntvedt. Erosive wear in choke valves. In *Valve World 2000 Conference and Expo.*, 2000.

[HL89] I.M Hutchings and A.V Levy. Thermal effects in the erosion of ductile metals. *Wear*, 131:105–121, 1989.

[HN68] F.H Harlow and P.I Nakayama. Transport of turbulence energy decay rate. *Los Alamos Science Lab.*, LA-3854, 1968.

[HP00] R Hanson and M.K Patel. Development of a model to predict the life of pneumatic conveyor bends subject to erosive wear. In *Proc. ASME 2000 Fluids Eng. Div. Summer Meeting*, Boston, Massuchusetts, 2000.

[Hum90] J.A.C Humphrey. Fundamentals of fluid motion in erosion by solid particle impact. *Int. J. Heat and Fluid Flow*, 11(3), 1990.

[IAoHER] International Association of Hydraulic Engineering and Research. *www.iahr.org/newsweb/conferences/bangkok2000.htm.*

[KFP95] I Kleis, F Frishman, and T Pappel. Method and installation for studying hard–particle kinematics at impact. In *OST-95 Symposium on Machine Design: Proceedings*, pages 15–22, 1995. Oulu, Finland, May 18-19.

[Lev95] Alan V Levy. *Solid Particle Erosion and Erosion–Corrosion of Materials.* ASM International, 1995.

[LM98] A Levy and D.J Mason. The effect of a bend on the particle cross-section concentration and segregation in pneumatic conveying systems. *Powder Technology*, pages 95–103, 1998.

[Mas89] B.S Massey. *Mechanics of Fluids.* Chapman & Hall, 1989. Sixth edition.

[MGHP94] R Mawhinney, E Galea, N Hoffman, and M Patel. A critical comparison of a PHOENICS based fire field model with experimental compartment fire data. *J. of Fire Protection Engineering*, 6(4):137–152, 1994.

[MSM99] M Manickam, M.P Schwarz, and M.J McIntosh. CFD analysis of erosion of bifurcation duct walls. In *2nd Int. Conf. on CFD in the Minerals and Process Industries, CSIRO, Melbourne, Australia*, pages 243–248, 1999.

[NEL] NEL. *www.nelflow.co.uk/PTP/Malcolm.htm.*

[NG68] J.H Neilson and A Gilchrist. Erosion by a stream of solid particles. *Wear*, 11:111–122, 1968.

[NS95] J.R Nicholls and D.J Stephenson. Monte Carlo modelling of erosive processes. *Wear*, 186:64–77, 1995.

[OP93] B Oesterle and A Petitjean. Simulation of particle-to-particle interactions in gas-solid flows. *Int. J. Multiphase Flow*, 19:199–211, 1993.

[Öqv01] M Öqvist. Numerical simulations of mild wear using updated geometry with different step size approaches. *Wear*, 249:6–11, 2001.

[Pat80] S.V Patankar. *Numerical Heat Transfer and Fluid Flow.* McGraw Hill, New York, 1980.

[PC94] K.A Pericleous and K.S Chan. The sea method for free surface problems with heat transfer and change of phase. *Numerical Methods in Multiphase Flows*, 185:227–236, 1994. ASME FEDSM.

[PC97] A.V Potapov and C.S Campbell. The two mechanisms of particle impact breakage and the velocity effect. *Powder Technology*, 93:13–21, 1997.

[PG84] Robert.H Perry and Don.W Green. *Perry's Chemical Engineers Handbook.* MacGraw–Hill, 1984. 7th Edition.

[PPS75] S.V Patankar, V.S Pratap, and D.B Spalding. Prediction of turbulent flow in curved pipes. *J. Fluid Mech.*, 67:583–595, 1975.

[PS72] S.V Patankar and D.B Spalding. A calculation procedure for heat, mass and momentum transfer in three–dimensional parabolic flows. *Int. J. Heat Mass Transfer*, 15:1787, 1972.

[PSST99] G.I Parslow, D.J Stephenson, J.E Strutt, and S Tetlow. Investigation of solid particle erosion in components of complex geometry. *Wear*, 233:737–745, 1999.

[Roe87] P.L Roe. *Error Estimates for Cell-Vertex Solutions of the Compressible Euler Equations.* ICASE Rep No. 87-6, 1987. NASA Langley Research Center.

[SB84] W.Y Soh and S.A Berger. Laminar entrance flow in a curved pipe. *J.Fluid Mech.*, 148:109–135, 1984.

[SF66] G.L Sheldon and I Finnie. The mechanism of material removal in the erosive cutting of brittle materials. *J. Eng. Ind. (Trans. ASME, Ser. B),*, 88:393–400, 1966.

[Sha92] C.T Shaw. *Using Computational Fluid Dynamics.* Prentice Hall, 1992.

[SHES81] S Sőderberg, S Hogmark, U Engman, and H Swahn. Erosion classification of materials using a centrifugal erosion tester. *Tribology Int'l.*, December:333–343, 1981.

[SMSR95] S.A Shirazi, B.S McLaury, J.R Shadley, and E.F Rybicki. Generalization of the API RP14E guideline for erosive services. *Journal of Petroleum Technology*, 47(8):693–698, 1995.

[SSY95] S Sato, A Shimizu, and T Yokomine. Numerical prediction of erosion for suspension duct flow. *Wear*, 186:203–209, 1995.

[Til73] G.P Tilly. A two stage mechanism of ductile erosion. *Wear*, 23:87–96, 1973.

[TSM91] Y Tsuji, N.Y Shen, and Y Morikawa. Lagrangian simulation of dilute gas-solid flows in a horizontal pipe. *Advanced Powder Technol.*, 2,1:63–81, 1991.

[VDR84] J.P Van Doormal and G.D Raithby. Enhancements of the simple method for predicting incompressible fluid flows. *Numerical Heat Transfer*, 7:147–163, 1984.

[VM95] H.K Versteeg and W Malalasekera. *An Introduction to Computational Fluid Dynamics.* Longman Scientific and Technical, Harlow, Essex, England, 1995.

[VSC98] V Venkatakrishnan, Manuel D. Salas, and Sukumar R. Chakravarthy. *Barriers and Challenges in Computational Fluid Dynamics.* Kluwer Academic, Dordrecht/Boston/London, 1998. ISBN 0-7923-4855-9.

[Woo92] S.R Woodhead. *The Measurement of Particle Velocity and Suspension Density in Pneumatic Coal Injection Systems.* PhD thesis, University of Greenwich, 1992.

[WPS+00] M.S Wallace, J.S Peters, T.J Scanlon, W.M Dempster, S McCulloch, and J.B Ogilvie. CFD-based erosion modelling of multi-orifice choke valves. In *Proc. ASME 2000 Fluids Eng. Div. Summer Meeting*, pages 1–13, Boston, Massuchusetts, June 2000.

[YC78] C.S Yust and R.S Crouse. Melting at particle impact sites during erosion of ceramics. *Wear*, 51:193–196, 1978.

[Yeu79] W.S Yeung. Erosion in a curved pipe. *Wear*, 55:91–106, 1979.

[ZM96] Y Zhong and K Minemura. Measurement of erosion due to particle impingement and numerical prediction of wear in pump casing. *Wear*, 199:36–44, 1996.