# HOWFAR and HOWNEAR:<br>Geometry Modeling for<br>Monte Carlo Particle Transport

Alex F Bielajew

Draft Version: August 14, 1995

PIRS-0341

Ionizing Radiation Standards
Institute for National Measurement Standards
National Research Council of Canada
Ottawa, Canada
K1A 0R6
Tel: 613-993-2197
Fax: 613-952-9865
email: alex@irs.phy.nrc.ca

### Abstract

This report is motivated by a paucity of literature describing the elementary mathematics involved in the modeling of geometries for ray-tracing applications such as particle transport in Monte Carlo calculations. The general problem of solving for the intersection point of a straight line with an arbitrary quadric surface is developed (HOWFAR) as well as the problem of finding the closest distance of a point to the boundary of an arbitrary quadric surface (HOWNEAR). A general strategy for boundary-crossing logic is presented which circumvents ambiguities associated with numerical precision and end-of-step directional uncertainties (which can arise, for example, as a result of a multiple-scattering deflection angle being applied at the end of the step). The specific examples of surfaces given are planes, circular cylinders, spheres and circular cones with arbitrary orientation and position. Care is taken to develop the mathematical equations so that they can be computed with numerical accuracy and a discussion on the influence of machine precision on the accuracy of results is given.

# 1 Introduction

In particle ray tracing for applications such as the Monte Carlo simulation of particles being transported through media, it is required to compute the distance along a particle's direction at which a surface describing the geometry is intersected. In our convention, we call this quantity HOWFAR. Part of this report is devoted to the calculation of HOWFAR for arbitrary surfaces. Examples are provided for several quadric surfaces, in particular, for planes (which are really *linear* surfaces), spheres, circular cylinders and circular cones.

Another useful quantity that is required for certain variance reduction techniques [1] and the PRESTA electron algorithm [2, 3, 4], an accurate electron transport algorithm adapted for the EGS4 Monte Carlo code [5], is the quantity called HOWNEAR, the minimum distance (irrespective of direction) to any surface from the location of the initial particle position. Part of this report is devoted to the calculation of HOWNEAR for arbitrary quadric surfaces and for the example surfaces mentioned above. The general solution for the arbitrary quadric surface appears to be new.

A section is devoted to the general problem of boundary crossing and how the geometry model impacts upon it, with specific discussion relating to the algorithms of the EGS4 code.

The remainder of this report is devoted to the issue of accuracy performance of the algorithms developed herein and a comparison of one alternate boundary crossing scheme.

# 2 Boundary crossing

Boundary crossing with the EGS4 system is provided by a well-specified interface. On entering the subroutine HOWFAR, EGS4 provides the region number, position and direction of the particle and USTEP, the straight-line distance along the particle's direction that EGS4 proposes to transport the particle in the absence of geometry[1]. USTEP is generally determined by the physics of transport in infinite media plus any other step-size constraints that the user may specify, for example, ESTEPE, or SMAX[2]. The HOWFAR routine provides EGS4 with only two required pieces of information: 1) the new region number that the particle will go into if it crosses a boundary and, 2) the new, shorter value for USTEP if this occurs. There is one optional parameter, IDISC, that the user may set non-zero, if the user wants EGS4 to terminate the particle history. This is usually employed to signal to EGS4 that the particle has left the region of interest and need not be transported anymore.

This simple design allows EGS4 to concern itself only with the problem of transport of particles in infinite media and it only needs to know the composition of the medium that

[1] In principle, it is up to the user to demand any information from EGS4. However, this list of inputs represents the minimum required.

[2] ESTEPE and SMAX are parameters that users may specify to control electron step-size. For more information, consult Rogers [6].

particles are being transported in. Thus, when the new region number is communicated to EGS4 via HOWFAR, EGS4 merely has to check in its look-up tables whether or not the medium has changed and then takes appropriate action. This decoupling of physics and geometry is one of EGS4's most powerful features and permits arbitrary flexibility in specifying the geometry. Users are responsible for providing the geometry and it can be simple or as complicated as imagination will allow.

However, there is one drawback. In an "ideal" computer where floating point numbers could be specified to absolute accuracy, there would be no ambiguity[3]. Is the particle on a surface or not? Is it exactly on the surface or has truncation caused an "undershoot" or round-up caused an overshoot? The problem arises entirely from the finite precision of floating point numbers in computers. There are three possibilities one must consider:

**undershoot** The new value of USTEP does not quite reach the surface but EGS4 is made to assume that the surface has been reached.

**exact** Numerically, the particle is exactly on the surface.

**overshoot** The new value of USTEP is slightly overestimated so that the surface is actually crossed.

All of these possibilities occur with varying frequency during the course of a Monte Carlo calculation. In fact, it is correct to say that if you run a geometry code through enough examples with stochastic selection of input parameters, then everything that can happen *will* happen. Therefore, it is necessary to write geometry coding that is robust enough to handle all of these possibilities and also to be aware of the error handling that EGS4 attempts so that coding efforts will neither be redundant nor in conflict with EGS4. It has been suggested that all geometry be coded in double or extended precision to avoid these kinds of ambiguities. However, one must realise that double or extended precision does not mean absolute precision. Higher precision reduces the size of the undershoot or overshoot but does nothing to cure undershoot or overshoot ambiguities. A geometry code that survives using single precision arithmetic will work at higher precision providing that the coding does not make some intrinsic assumptions on precision or scale. The converse is not true. The routines developed for this report will work for both single and higher precision and the regions of validity for use with single precision is investigated.

EGS4 attempts to fix up ambiguities in the following fashion[4]: If the USTEP returned is less than or equal to zero, the region number is set to the new region number returned by HOWFAR (making the assumption that the user has set the region number to that region where the

[3] It is possible to recode geometry transport in integer arithmetic, thereby avoiding ambiguities. This "quantisation" of space approach has its drawbacks and further discussion would take us out of the scope of this report.

[4] The default behaviour is described above. The user has the ability to define another scheme by use of a "macro" substitution.

particle position implies that it is), then the medium number is changed (if needed) and the outer transport loop is restarted. Thus, the user is given the additional responsibility of initiating the error correction by providing the correct region number if a particle is "lost" and by signaling EGS4 that this happened by returning a USTEP that is less than or equal to zero.

This error recovery is completely general assuming that the user's geometry provides zero or slightly negative USTEP's and that the user's geometry takes advantage of the knowledge of where the particle is *supposed* to be. To illustrate this, imagine for the sake of argument that we are in region number 1 bounded by two infinite parallel planes, P1 and P2, and that HOWFAR is called with a USTEP large enough to escape region 1. All the different possibilities are depicted in figure 1. In order to save time, imagine that the user has coded HOWFAR to check planes that the particle is directed at. Therefore, only the distance to P2 is calculated (assumed to be a forward solution), USTEP is shortened, and the next region is region number 2 bounded by the parallel planes P2 and P3. Control returns to EGS4 and the transport step is effected and for electrons, the multiple scattering angle is sampled and the electron deflected. Now EGS4 thinks the particle is in region 2. HOWFAR is called. If the particle is directed at P3 and P2 is never checked, then there will be no ambiguity irrespective of an overshoot, undershoot, or exact transport to the surface on the previous step. A difficulty can arise for electrons if multiple scattering deflects the particle back towards P2. In the case of an overshoot on the previous step there is no problem as USTEP will just be set to a small number and a small transport step will take place on the next step. However, in the case of an undershoot or exact transport to the surface, a difficulty does arise. USTEP will be set to zero or some small negative number and HOWFAR will assume that the particle will re-enter region number 1. Thus, the elementary fix-up effected by EGS4 works! It is incumbent upon the user to generate geometry code that handles these ambiguities and make use of EGS4's error handling of the situation.

The following strategy will be adopted: The HOWFAR routines are expected to know whether or not a particle is inside or outside a surface irrespective of the numerics implied by the position of the particle with respect to the surface. If the geometry routine detects that a particle is not in the region where it thinks it is (presumably by virtue of an undershoot), then it is assumed that this is due to numerical inaccuracy. If the assumptions of where a particle is and the numerical calculation of its position conflict the following strategy is employed: If the particle is still headed in the direction of the surface, then this surface (at least the smallest solution to it) is ignored, as if the particle had penetrated the surface. If the particle is headed away from the surface a zero solution is provided and the surface is assumed to be crossed again. This idea has been published previously [7] although similar logic has been employed since 1978 [8] with the release of the EGS3 code [9].

The geometry code then carries on using the HOWFAR assumptions regarding the position of the particle. By way of example, imagine that an undershoot to a spherical surface has occurred. The computer logic assumes that the particle is inside the sphere but the numerics dictate that it is outside. Since HOWFAR knows where the particle *should* be, it may make use of

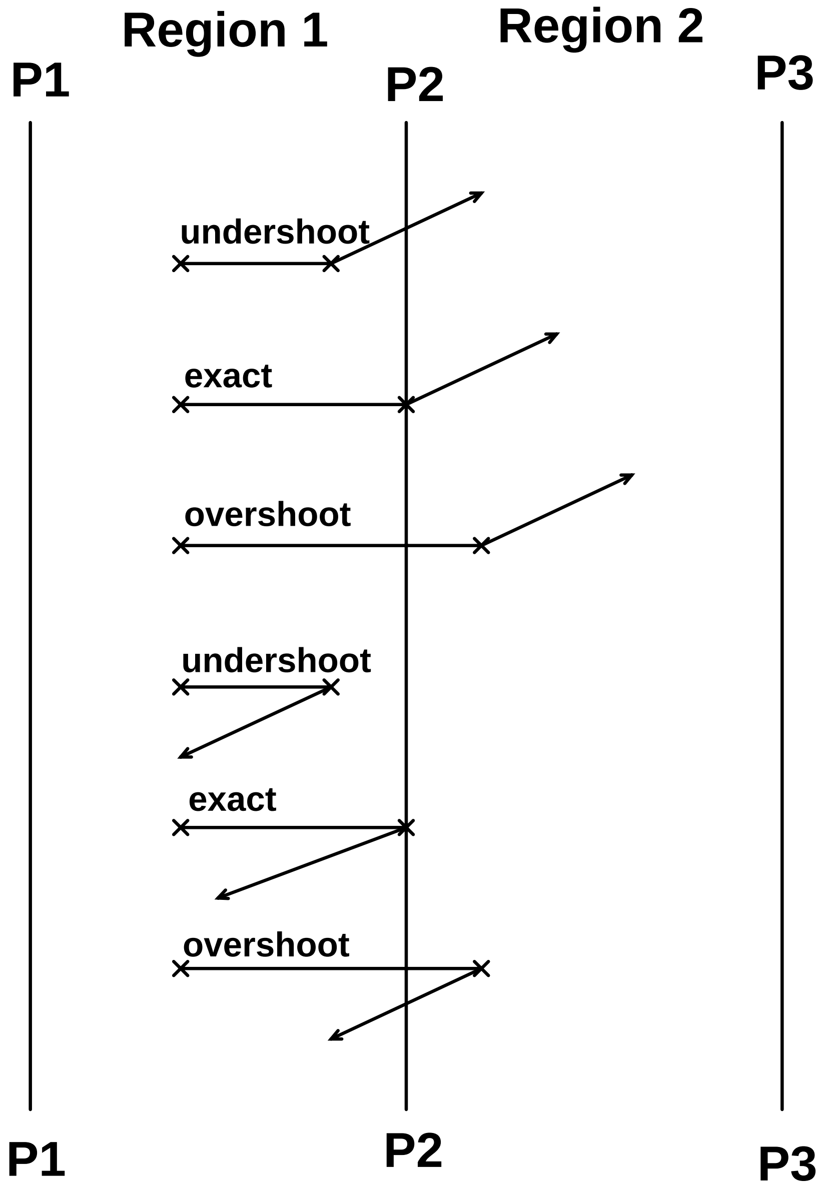

Figure 1: Boundary crossing of an electron across plane P2. There are six possibilities corresponding to undershoot, exact surface position, overshoot with either forward scatter or backscatter.

this information to obtain the unambiguous solution. If the particle is directed outside of the sphere, the geometry routine returns a zero USTEP and changes the region to that outside the sphere, thereby minimizing the effect of the inaccuracy. Similarly, if the particle is directed at the sphere, the USTEP solution is given to the far side of the sphere and a region changed signaled. Note that the distance from the present position of the particle to the *far* side of the sphere is computed, including the amount of undershoot. Thus, the undershoots do not accumulate and cause further problems for subsequent transport steps. This procedure is used for all surface types and the examples to follow give further clarification.

## 3 General solution for an arbitrary quadric

Borrowing from the notation of Olmsted [10], an arbitrary quadric surface in 3(x,y,z)-space[5] can be represented by:

$$f(\vec{x}) = \sum_{i,j=0}^{3} a_{ij} x_i x_j = 0. \qquad (1)$$

The $a_{ij}$'s are arbitrary constants and the 4-vector $x_i$ has components $(1, x, y, z)$. The zeroth component is unity by definition allowing a very compact representation and $a_{ij}$ is symmetric with respect to the interchange of $i$ and $j$, that is $a_{ij} = a_{ji}$. Equation 1 is very general and encompasses a wide variety of possibilities including solitary planes (*e.g.* only $a_{0i}$ non-zero), intersecting planes (*e.g.* only $a_{11}$ and $a_{22}$ non-zero), cylinders (circular, elliptical, parabolic and hyperbolic), spheres, spheroids and ellipsoids, cones (circular and elliptical), hyperboloids of one and two sheets and elliptic and hyperbolic paraboloids. These surfaces can be combined to make geometrical objects of arbitrary complexity and are extremely useful in Monte Carlo modeling of physical objects.

Despite having apparently 10 independent constants, eq. 1 represents only 18 independent surfaces (including the simple plane), unique after a translation and rotation to standard position. In fact the three cross terms ($a_{ij}$ for $i \neq j$ and $i, j \geq 1$) can be eliminated by rotation. The resultant equation then only involves terms like $x_i^2$ and $x_i$. In addition, providing that a given variable's quadratic constant is non-zero, the linear terms can be eliminated by a translation. The result is that there are only two generic forms:

$$f(\vec{x}) = \sum_{i=1}^{3} a_i x_i^2 + c = 0, \qquad (2)$$

and

$$f(\vec{x}) = \sum_{i=1}^{2} a_i x_i^2 + b x_3 = 0. \qquad (3)$$

Equations 2 and 3 describe only 10 distinct possibilities with real solutions.

[5] The only variance with Olmsted's notation is that the 4th component is labelled as the 0th component in this work.

1. **ellipsoids:** $a_1^2x_1^2 + a_2^2x_2^2 + a_3^2x_3^2 - c^2 = 0$.
2. **cones:** $a_1^2x_1^2 + a_2^2x_2^2 - a_3^2x_3^2 = 0$.
3. **cylinders:** $a_1^2x_1^2 + a_2^2x_2^2 - c^2 = 0$.
4. **hyperboloids of one sheet:** $a_1^2x_1^2 + a_2^2x_2^2 - a_3^2x_3^2 - c^2 = 0$.
5. **hyperboloids of two sheets:** $a_1^2x_1^2 + a_2^2x_2^2 - a_3^2x_3^2 + c^2 = 0$.
6. **elliptic paraboloids:** $a_1^2x_1^2 + a_2^2x_2^2 + a_3x_3 = 0$.
7. **hyperbolic paraboloids:** $a_1^2x_1^2 - a_2^2x_2^2 + a_3x_3 = 0$.
8. **hyperbolic cylinders:** $a_1^2x_1^2 - a_2^2x_2^2 + c^2 = 0$.
9. **parabolic cylinders:** $a_1^2x_1^2 + a_3x_3 = 0$.
10. **simple planes:** $a_3x_3 + c = 0$.

The first nine of these are shown[6] in fig. 2. (The magnitude of the above constants were all chosen to be unity for the purposes of display. Consequently, the first six of these surfaces shown exhibit at least one axis of rotational symmetry.) There are other imaginary surfaces (*e.g.* imaginary ellipsoids $a_1^2x_1^2 + a_2^2x_2^2 + a_3^2x_3^2 + c^2 = 0$) that we will not consider nor will we consider quadrics that can be made up of two independent planes in various orientations (*e.g.* intersection planes $a_1^2x_1^2 - a_2^2x_2^2 = 0$, parallel planes $a_1^2x_1^2 - c^2 = 0$, and coincident planes $a_1^2x_1^2 = 0$).

The HOWFAR solutions can be obtained using the constants specified in an arbitrary way. The solution for HOWNEAR is more involved and explicit analytic forms are given only for the special cases in this report. However, the canonical forms expressed by eqs. 2 and 3 are used later to discuss the general HOWNEAR solution but not to determine the HOWNEAR solutions explicitly.

For more information on the reduction to canonical form, the reader is encouraged to read Olmsted's book [10]. Olmsted also gives the classification of the surfaces and lists the entire set of 17 canonical quadric forms.

### 3.1 HOWFAR to an arbitrary quadric surface?

A geometric particle trajectory is represented by a line in 3-space and is most conveniently expressed in terms of a parametric equation:

$$\vec{x} = \vec{p} + \vec{u}s \tag{4}$$

[6] These figures were produced using a programme called QUADPLOT [11] that was developed in association with this study. QUADPLOT also displays HOWFAR and HOWNEAR solutions and includes an algorithm for quadric surface classification from arbitrary user-input parameters.

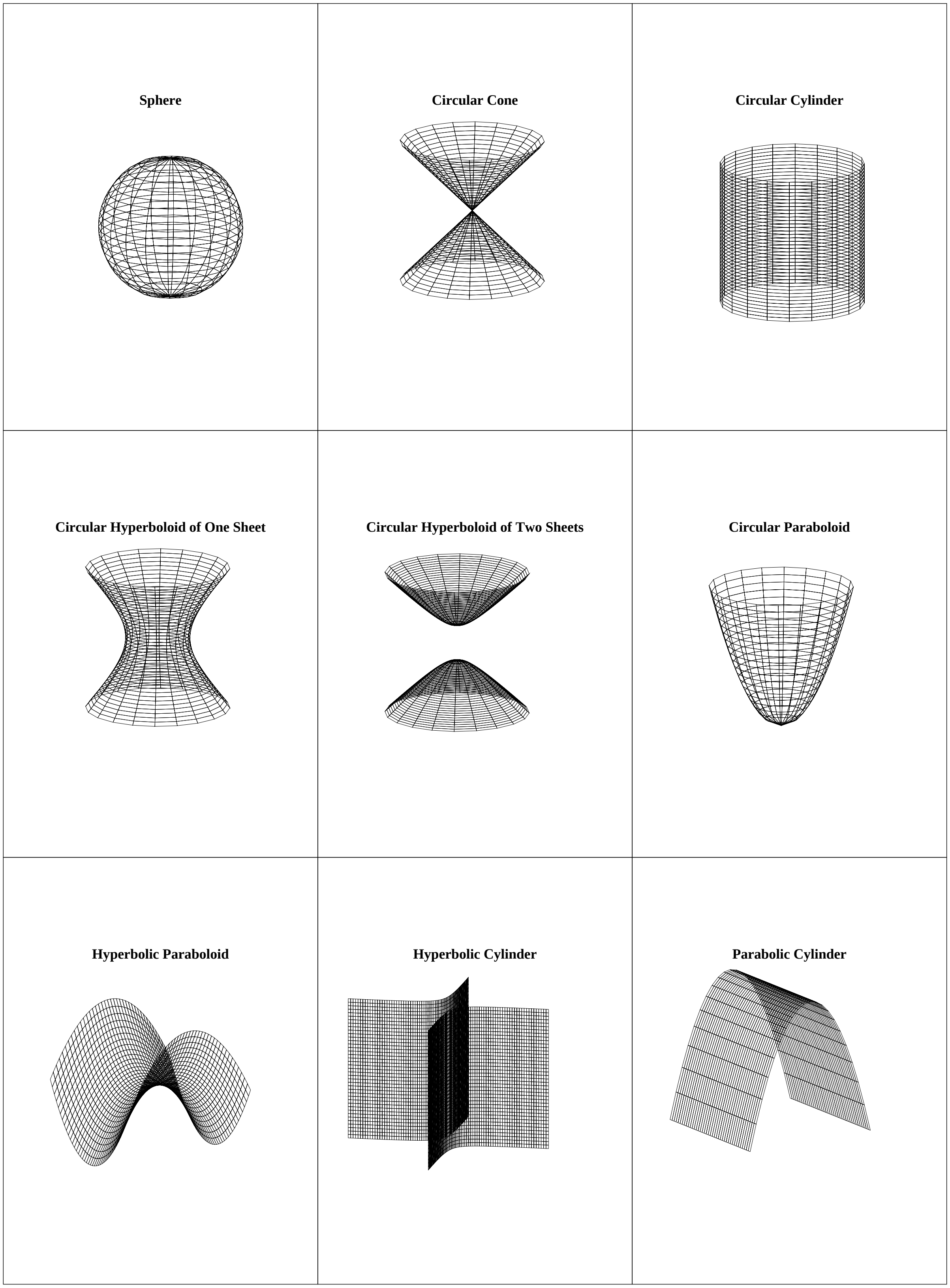

Figure 2: The nine real non-planar quadric surfaces.

where $\vec{x}$ is the vector notation for $(x, y, z)$: the position along the line, $\vec{u}$ is the vector notation for the direction of the line $(u, v, w)$ and $s$ is the geometric path-length from the starting position of the particle, $\vec{p} = (p_x, p_y, p_z)$. The components $(u, v, w)$ are the direction cosines of the line along the $(x, y, z)$-axes and by convention are normalised so that $|\vec{u}| = \sqrt{u^2 + v^2 + w^2} = 1$. This normalisation permits an identification of $s$ as the distance along the line from the original position $\vec{p}$. A positive value of $s$ expresses a distance along the direction that the particle is going (forward trajectory) and a negative value is associated with a distance that the particle came from (backward trajectory). Thus, negative solutions found for $s$ below are rejected.

In Monte Carlo particle transport calculations as well as ray-tracing algorithms a common problem is to find the distance a particle has to travel in order to intersect a surface. This is done by substituting for $\vec{x}$ from eq. 4 in eq. 1 to give:

$$s^2 \left( \sum_{i,j=0}^{3} a_{ij} u_i u_j \right) + 2s \left( \sum_{i,j=0}^{3} a_{ij} p_i u_j \right) + \left( \sum_{i,j=0}^{3} a_{ij} p_i p_j \right) = 0, \tag{5}$$

where we have adopted the convention that $u_0 = 0$ and $p_0 = 1$. This is a quadratic equation in $s$ of the form $A(\vec{u})s^2 + 2B(\vec{u}, \vec{p})s + C(\vec{p}) = 0$ where $A(\vec{u}) = \sum_{i,j=0}^{3} a_{ij} u_i u_j$, $B(\vec{u}, \vec{p}) = \sum_{i,j=0}^{3} a_{ij} p_i u_j$ and $C(\vec{p}) = \sum_{i,j=0}^{3} a_{ij} p_i p_j$.

### 3.1.1 Interpretation of the quadratic constants

The constant $C(\vec{p})$ is identically zero when $\vec{p}$ is on the surface. When $\vec{p}$ is not on the surface, the sign of $C(\vec{p})$ can be interrogated to see if the particle is inside or outside. There is some arbitrariness in the definition of what is "inside" or "outside". A sphere with radius $R$, for example, has the form $\vec{p}^2 - R^2 = 0$ and in this case $C(\vec{p}) > 0$ when $|\vec{p}| > R$. So, for this example, $C(\vec{p})$ is positive when $\vec{p}$ is outside and negative when inside. However, the same sphere is defined by $R^2 - \vec{p}^2 = 0$ giving opposite interpretation for the signs of $C(\vec{p})$ for points inside and outside. It is best to adopt a constant interpretation and be aware that two points on opposite sides of the surface in the sense that a line joining them intersects the surface only once, have different signs.

For planes and most of the other surfaces, "inside" and "outside" are arbitrary since multiplying eq. 2 or eq. 3 by a minus sign leaves the surface intact. However, there is one natural interpretation provided by the calculation of the normal to the surface, $\nabla f(\vec{p})$, where $\vec{p}$ is a point on the surface, *i.e.* $C(\vec{p}) = 0$. In the way they were defined, $\nabla f(\vec{p})$ points to the "outside" region which can be defined as follows: If more than one line can be drawn through a point such that the surface is not intersected in either the forward or backward direction, then this point is on the outside. If at most only one such line exists, then the point is on the "inside". This defines the inside and outside in a unique and natural way (inside a sphere, for example). There are three exceptions to this rule, the simple plane, the hyperboloid of one sheet and the hyperbolic paraboloid. In their standard forms given below eq. 3, the outside or inside of a plane is completely arbitrary, the outside of a hyperbolic paraboloid contains the positive $x_3$-axis and the inside of the hyperboloid of one sheet contains the $x_3$-axis which also seems to be a "natural" choice.

The constant $B(\vec{u},\vec{p})$ is related to the inner product of the particle's direction $\vec{u}$ with the normal to the surface at a point $\vec{p}$ when $\vec{p}$ is on the surface. Specifically, $B(\vec{u},\vec{p}) = \sum_{i,j=0}^{3} a_{ij} p_i u_j = \frac{1}{2}\vec{u}\cdot\nabla f(\vec{p})$. When $\vec{p}$ is on the surface $\nabla f(\vec{p})$ is its normal there. This can be exploited to decide to which side of a surface a particle is going if it happens to be on the surface and is pointed in some direction. Imagine a particle on the surface at point $\vec{p}$ with some direction $\vec{u}$ and consider an infinitesimal step $\epsilon$. The sign of $C(\vec{p}+\vec{u}\epsilon) = 2\epsilon B(\vec{u},\vec{p}) + O(\epsilon^2)$ will have the sign of $B(\vec{u},\vec{p})$. If $B(\vec{u},\vec{p}) = 0$ for $\vec{p}$ on the surface, it means that the particle is moving in the tangent plane to the surface at that point.

When a particle is on the surface, the constant $A(\vec{u})$ can be related to the curvature of the surface. It can be shown[7] that the radius of curvature at the point $\vec{p}$ on the surface in the plane containing the normal to the surface there, $\nabla f(\vec{p})$ and the direction of the particle on the surface, $\vec{u}$, is given by $|\nabla f(\vec{p})|/|A(\vec{u})|$. There is one case among the surfaces we consider where both $|\nabla f(\vec{p})|$ and $|A(\vec{u})|$ vanish simultaneously and that is of a point on the vertex of a cone. In this anomalous case we can take the radius of curvature to be zero.

$A(\vec{u})$ vanishes when the particle is travelling parallel to a "ruled line" of the surface, whether on the surface or not. A ruled line is a line that lies entirely on the surface. Quadrics with one or more vanishing quadratic constants (one of the $a_i$'s in eq. 2 or 3) always possess ruled lines, as do planes, cones, hyperboloids of one sheet and hyperbolic paraboloids. The constant $A(\vec{u})$ can also vanish for a particle having a trajectory that is parallel to an asymptote of a hyperboloid, or pointed at the "nose" of a paraboloid or in the plane perpendicular to it.

$A(\vec{u})$ can be used to decide where a particle is in relation to a surface in the case that $B(\vec{u},\vec{p})$ and $C(\vec{p})$ vanish, that is when the particle is on the surface and in the plane tangent to it at that point. In this case an infinitesimal transport $C(\vec{p}+\vec{u}\epsilon) = A(\vec{u})\epsilon^2$ will have the sign of $A(\vec{u})$. So, if ($A(\vec{u}) > 0$, $B(\vec{u},\vec{p}) = 0$, $C(\vec{p}) = 0$) the particle is headed outside, if ($A(\vec{u}) < 0$, $B(\vec{u},\vec{p}) = 0$, $C(\vec{p}) = 0$) the particle is headed inside, and if ($A(\vec{u}) = 0$, $B(\vec{u},\vec{p}) = 0$, $C(\vec{p}) = 0$) the particle is on the surface and directed along a ruling and there is no intercept in this case.

For planar surfaces ($A(\vec{u}) = 0$ always in this case) there is always a solution for $s$ unless the particle's trajectory is exactly parallel to the plane. If the solution for $s$ is negative, it is rejected since it not a forward solution. (Solutions that go back in time are usually not interesting except for adjoint problems.) If the solution for $s$ is positive, then it represents a

[7] The way to do this is consider a particle at point $\vec{p}$ on the surface with an initial direction $\vec{u}$ tangent to the surface and moving in the plane defined by the normal orthogonal vectors $\vec{u}$ and $\nabla f(\vec{p})/|\nabla f(\vec{p})|$. The trajectory of the particle is then described by $f(\vec{p} + \vec{u}s_u + (\nabla f(\vec{p})/|\nabla f(\vec{p})|)s_n) = 0$ where $s_u$ and $s_n$ are projections of the particle's position vector on the $\vec{u}$ and $\nabla f(\vec{p})/|\nabla f(\vec{p})|$ axes, respectively. This yields the equation of a conic. The radius of curvature is then obtained by the standard equation for motion in a plane, $R_{\rm c} = \{[1 + ({\rm d}s_n/{\rm d}s_u)^2]^{3/2}\}/|{\rm d}^2 s_n/{\rm d}s_u^2|$.

solution along the forward trajectory of the particle.

For the non-planar surfaces, the equation for $s$ is quadratic. In general, when $B(\vec{u},\vec{p})^2 - A(\vec{u})C(\vec{p}) < 0$, there are no solutions to the quadratic equation, which means that the particle's trajectory misses the surface. If the surface in question is one of the seven with the intuitive inside-outside interpretation, then one might guess that one does not have to test for the positiveness of $B(\vec{u},\vec{p})^2 - A(\vec{u})C(\vec{p})$ when the particle is inside, thereby saving computer time. However, there are conditions where the limited numerical precision causes $B(\vec{u},\vec{p})^2 - A(\vec{u})C(\vec{p})$ to be negative even when the particle is inside one of the natural surfaces. In this case it can be shown that the particle's trajectory is very close to that of being along a ruling and very close to the surface ($A(\vec{u}) \approx 0$, $B(\vec{u},\vec{p}) \approx 0$, $C(\vec{p}) \approx 0$). It is consistent, therefore, to assume that there is no solution in this case even if the trajectory is not exactly on the surface or exactly tangent to it. In this case the particle is assumed to travel along the ruling until it hits another surface in the problem or an interaction effects a change of direction whereupon a decision can be made whether the particle is headed inside or outside the surface.

Employing the error recovery strategy of the previous section, a general algorithm for an arbitrary quadric surface may be sketched:

```
IF $B^2 - AC < 0$ Particle does not intersect the surface.
ELSEIF HOWFAR thinks the particle is outside
    IF $B \geq 0$
        IF $A \geq 0$ No solution.
        ELSE $s = -(B + \sqrt{B^2 - AC}/A)$.
    ELSE $s = \max(0, C/[\sqrt{B^2 - AC} - B])$.
ELSE HOWFAR thinks the particle is inside
    IF $B \leq 0$
        IF $A > 0$ $s = (\sqrt{B^2 - AC} - B)/A$.
        ELSE No solution.
    ELSE $s = \max(0, -C/[\sqrt{B^2 - AC} + B])$.
```

All quadric surfaces are special cases that can be solved by this algorithm. Only the constants $A, B, C$ need to be specified for any case. Indeed, this algorithm will work for planes as well but the simplicity of planes motivates the construction of a more efficient algorithm specific to planes only.

### 3.2 HOWNEAR to an arbitrary quadric surface?

The problem of finding the minimum distance from a given point in 3-space (starting point of a particle trajectory) to any point on a quadratic surface (HOWNEAR) is tantamount to finding the minimum perpendicular distance to the surface. This is because *it can be shown* that the extrema of the distances to any continuous surface (a quadratic surface is a special case of a generalised continuous surface) from a given point lies along the normal to the surface at that surface point. Therefore, it remains to decide whether or not the extremum is a maximum or a minimum.

For this discussion it is assumed that the quadric surface is in one of its generic forms expressed by eqs. 2 or 3. The distance squared, $d^2$ from the point $\vec{p}$ to the surface is given by:

$$d^2 = \sum_{i=0}^{3}(x_i - p_i)^2, \tag{6}$$

subject to the constraint expressed by eqs. 2 or 3. This can be solved by the method of Lagrange multipliers.

Examining the case where the surface is of the non-parabolic type, taking derivatives of eqs. 6 and 2 gives the condition for extrema:

$$x_i = \frac{p_i}{1+\lambda a_i}, \quad i = 1, 2, 3. \tag{7}$$

where $\lambda$ is the Lagrange multiplier. Inserting this into eqs. 2 yields a solution to the Lagrange multiplier:

$$\sum_{i=1}^{3} a_i \left(\frac{p_i}{1+\lambda a_i}\right)^2 + c = 0. \tag{8}$$

Depending upon the values of the constants $a_i$ and $c$, several cases can be enumerated.

**Case I** $\underline{c \neq 0,\ a_i \neq 0,\ a_i\text{'s all distinct}}$ These are ellipsoids, and elliptic hyperboloids of one and two sheets. This yields a 6th order solution for $\lambda$.

**Case II** $\underline{c \neq 0,\ a_i \neq 0,\ \text{two } a_i\text{'s are the same}}$ This describes a surface with azimuthal symmetry about one axis and includes spheroids and circular hyperboloids of one and two sheets. This solution is 4th order.

**Case III** $\underline{c \neq 0,\ a_i \neq 0,\ \text{all } a_i \text{ are the same}}$ This is a sphere. The solution is 2nd order.

**Case IV** $\underline{c \neq 0,\ \text{one } a_i = 0,\ \text{the others are different}}$ This is an elliptic or hyperbolic cylinder. The solution is 4th order.

**Case V** $\underline{c \neq 0,\ \text{one } a_i = 0,\ \text{the others are the same}}$ This is an circular cylinder. The solution is 2nd order.

**Case VI** $\underline{c \neq 0, \text{ two } a_i\text{'s are zero}}$ This represents parallel planes. The solution is 2nd order.

**Case VII** $\underline{c = 0,\ a_i \neq 0,\ a_i\text{'s all distinct}}$ This is an elliptic cone. The solution is 4th order.

**Case VIII** $\underline{c = 0,\ a_i \neq 0, \text{ two } a_i\text{'s are the same}}$ This is a circular cone. The solution is 2nd order.

**Case IX** $\underline{c = 0, \text{ one } a_i = 0,\ a_i\text{'s all distinct}}$ This case corresponds to two intersecting planes. There are two solutions.

**Case X** $\underline{c = 0, \text{ two } a_i\text{'s are zero}}$ This case corresponds to a single plane. There is one solution.

In the case that the surface is of the parabolic type, taking the derivatives gives:

$$x_i = \frac{p_i}{1 + \lambda a_i}, \quad i = 1, 2; \quad x_3 = p_3 - \frac{\lambda b}{2}. \tag{9}$$

Inserting this into eqs. 3 yields a solution to the Lagrange multiplier:

$$\sum_{i=1}^{2} a_i \left( \frac{p_i}{1 + \lambda a_i} \right)^2 + b \left( p_3 - \frac{\lambda b}{2} \right) = 0. \tag{10}$$

Depending upon the values of the constants $a_i$, several cases can be enumerated.

**Case I** $\underline{a_i \neq 0,\ a_1 \neq a_2}$ This is an elliptic or hyperbolic paraboloid and yields a 5th order solution for $\lambda$.

**Case II** $\underline{a_1 = a_2 \neq 0}$ This describes a paraboloid with azimuthal symmetry about one axis. This solution is 3rd order.

**Case III** $\underline{a_1 = 0,\ a_2 \neq 0}$ This describes a parabolic cylinder. The solution is 3rd order.

**Case IV** $\underline{a_1 = a_2 = 0}$ This describes a solitary plane. There is only one solution.

In the most difficult cases one must resort to finding roots to 5th or 6th-order polynomials. It has been proven that there is no analytic technique for accomplishing this and one must resort to numerical methods. Analytic techniques exist for 4th-order and lower. It ought to be remarked that some of the solutions need not be real but in the case of surfaces of the hyperbolic type expressed by eq. 2 there must be at least two real solutions and in the case of parabolic type expressed by eq. 3, there must be at least one. A numerical search routine must be able to search for all the roots, and take the minimum among the real ones. In general, this is not a simple task and may be time consuming. For the purpose of this report we now consider only 2nd-order and less and this restricts us to planes, spheres, circular cylinders and circular cones.

# 4 Solutions for simple surfaces

## 4.1 Planes

The general equation for a plane of arbitrary orientation is:

$$\vec{n} \cdot (\vec{x} - \vec{P}) = 0 \tag{11}$$

where $\vec{n}$ is the unit normal of the plane and $\vec{P}$ is any point on the surface of the plane. Note the use of the inner product, $\vec{n} \cdot \vec{x} \equiv n_x x + n_y y + n_z z$.

### 4.1.1 `HOWFAR`?

Inserting eq. 4 into eq. 11 and solving for $s$ gives:

$$s = -\frac{\vec{n} \cdot (\vec{p} - \vec{P})}{\vec{n} \cdot \vec{u}} \tag{12}$$

We remark that there is no solution ($s = \infty$) when the particle direction is perpendicular to the normal of the plan ($\vec{n} \cdot \vec{u} = 0$). This is the solution of a particle travelling parallel to a plane and never hitting it. Only positive solutions for $s$ are acceptable and this depends upon whether or not the particle is travelling towards the plane.

Adopting the convention that a particle is considered to be outside the plane of it is on the side that the unit normal, $\vec{n}$, is pointing, we enumerate the possibilities:

**Case I** $\underline{\vec{n} \cdot \vec{u} = 0}$

Trajectory is parallel to the plane, no solution

**Case II** $\vec{n} \cdot (\vec{p} - \vec{P}) \geq 0$ and `HOWFAR` assumes outside

1. If $\vec{n} \cdot \vec{u} < 0$, $s = -\vec{n} \cdot (\vec{p} - \vec{P})/\vec{n} \cdot \vec{u}$
2. Elseif $\vec{n} \cdot \vec{u} > 0$, no solution

**Case III** $\vec{n} \cdot (\vec{p} - \vec{P}) \leq 0$ and `HOWFAR` assumes inside

1. If $\vec{n} \cdot \vec{u} > 0$, $s = -\vec{n} \cdot (\vec{p} - \vec{P})/\vec{n} \cdot \vec{u}$
2. Elseif $\vec{n} \cdot \vec{u} < 0$, no solution

**Case IV** $\vec{n} \cdot (\vec{p} - \vec{P}) < 0$ but `HOWFAR` assumes outside

1. If $\vec{n} \cdot \vec{u} < 0$, $s = 0$ effecting a region change
2. Elseif $\vec{n} \cdot \vec{u} > 0$, no solution

**Case V** $\vec{n} \cdot (\vec{p} - \vec{P}) > 0$ but HOWFAR assumes inside

1. If $\vec{n} \cdot \vec{u} > 0$, $s = 0$ effecting a region change
2. Elseif $\vec{n} \cdot \vec{u} < 0$, no solution

The case of a parallel trajectory is handled by **Case I**. The two "normal" conditions in **Case II** and **Case III** handle the eventuality where the particle is exactly on the plane, $\vec{n} \cdot (\vec{p} - \vec{P}) = 0$. **Case IV** and **Case V** handle the anomalies. In the case of an undershoot and backscatter out of the region where the HOWFAR thinks the particle is, then a zero distance is returned so that EGS may switch to the correct medium. If the case of an undershoot and forward scatter, no solution is given allowing other surface in the geometry to determine the intersection. Note that no correction is made for the undershoot distance and this will be included in the next transport step. Therefore, numerical inaccuracies are not allowed to accumulate.

### 4.1.2 HOWNEAR?

Using the method of Lagrange multipliers discussed earlier, $\vec{x}$ has the solution:

$$\vec{x} = \vec{p} - \frac{\lambda \vec{n}}{2}. \tag{13}$$

Substituting this into equation for the plane, eq. 12 solves for the Lagrange multiplier,

$$\lambda = 2\vec{n} \cdot (\vec{p} - \vec{P}), \tag{14}$$

and yields a solution for the closest distance:

$$d = |\vec{n} \cdot (\vec{p} - \vec{P})|. \tag{15}$$

## 4.2 Spheres

The general equation for a sphere is:

$$(\vec{x} - \vec{X})^2 - R^2 = (x - X)^2 + (y - Y)^2 + (z - Z)^2 - R^2 = 0 \tag{16}$$

where $\vec{X} \equiv (X, Y, Z)$ is the location of the center of the sphere and $R$ is its radius.

### 4.2.1 HOWFAR?

Substituting the equation for particle trajectory, eq. 4, into the above yields a quadratic equation of the form $As^2 + 2Bs + C = 0$, where the quadratic constants, $A$, $B$ and $C$, are:

$$\begin{aligned} A &= 1 \\ B &= \vec{u}\cdot(\vec{p}-\vec{X}) \\ &= u(p_x-X)+v(p_y-Y)+w(p_z-Z) \\ C &= (\vec{p}-\vec{X})^2-R^2 \\ &= (p_x-X)^2+(p_y-Y)^2+(p_z-Z)^2-R^2 \end{aligned}$$

These constants may be employed in the general quadric surface algorithm for HOWFAR described previously.

### 4.2.2 HOWNEAR?

Using the method of Lagrange multipliers, $\vec{x}$ has the solution:

$$\vec{x} = \frac{\vec{p}+\lambda\vec{P}}{1+\lambda}. \tag{17}$$

Substituting this into equation for the sphere, eq. 16 solves for the Lagrange multiplier in terms of its two roots,

$$\lambda = \frac{-R \pm |\vec{p}-\vec{P}|}{R}, \tag{18}$$

and yields a solution for the closest distance:

$$d = |R \pm |\vec{p}-\vec{P}||_{\rm min} = |R - |\vec{p}-\vec{P}||. \tag{19}$$

Equivalently, if $\vec{p}$ is outside the sphere,

$$d = |\vec{p}-\vec{P}| - R, \tag{20}$$

and if $\vec{p}$ is inside the sphere,

$$d = R - |\vec{p}-\vec{P}|. \tag{21}$$

The other solution for d corresponds to the other extremum, the distance to the far side of the sphere.

## 4.3 Circular Cylinders

The general equation for a circular cylinder is:

$$(\vec{x}-\vec{P})^2 - [(\vec{x}-\vec{P})\cdot\vec{U}]^2 - R^2 = 0 \tag{22}$$

where $\vec{P} \equiv (P_x, P_y, P_z)$ is any fixed point on the axis of the cylinder, $\vec{U}$ is the direction vector of the axis of the cylinder, and $R$ is its radius.

### 4.3.1 HOWFAR?

Substituting the equation for particle trajectory, eq. 4, into the above yields a quadratic equation of the form $As^2 + 2Bs + C = 0$, where the quadratic constants, $A$, $B$ and $C$, are:

$$\begin{aligned} A &= 1 - (\vec{u} \cdot \vec{U})^2 \\ B &= \vec{u} \cdot \{(\vec{p} - \vec{P}) - \vec{U}[(\vec{p} - \vec{P}) \cdot \vec{U}]\} \\ C &= (\vec{p} - \vec{P})^2 - [(\vec{p} - \vec{P}) \cdot \vec{U}]^2 - R^2 \end{aligned}$$

These constants may be employed in the general quadric surface algorithm for HOWFAR described previously.

### 4.3.2 HOWNEAR?

Using the method of Lagrange multipliers, the component of $\vec{x}$ parallel to $\vec{U}$ has the solution:

$$(\vec{x} - \vec{P}) - \vec{U}[\vec{U} \cdot (\vec{x} - \vec{P})] = \frac{(\vec{p} - \vec{P}) - \vec{U}[\vec{U} \cdot (\vec{p} - \vec{P})]}{1 + \lambda}. \qquad (23)$$

Substituting this into equation for the cylinder, eq. 22 solves for the Lagrange multiplier in terms of its two roots,

$$\lambda = \frac{-R \pm \sqrt{(\vec{p} - \vec{P})^2 - [\vec{U} \cdot (\vec{p} - \vec{P})]^2}}{R}, \qquad (24)$$

and yields a solution for the closest distance:

$$d = \left| R \pm \sqrt{(\vec{p} - \vec{P})^2 - [\vec{U} \cdot (\vec{p} - \vec{P})]^2} \right|_{\rm min} = \left| R - \sqrt{(\vec{p} - \vec{P})^2 - [\vec{U} \cdot (\vec{p} - \vec{P})]^2} \right|. \qquad (25)$$

Equivalently, if $\vec{p}$ is outside the cylinder,

$$d = \sqrt{(\vec{p} - \vec{P})^2 - [\vec{U} \cdot (\vec{p} - \vec{P})]^2} - R, \qquad (26)$$

and if $\vec{p}$ is inside the cylinder,

$$d = R - \sqrt{(\vec{p} - \vec{P})^2 - [\vec{U} \cdot (\vec{p} - \vec{P})]^2}. \qquad (27)$$

The other solution for d corresponds to the other extremum, the distance to the far side of the cylinder.

## 4.4 Circular Cones

In standard quadric form, the general equation for a cone is:

$$\cos^2\Theta\{(\vec{x}-\vec{P})-\vec{U}[(\vec{x}-\vec{P})\cdot\vec{U}]\}^2-\sin^2\Theta[(\vec{x}-\vec{P})\cdot\vec{U}]^2=0 \tag{28}$$

This form, depicted in fig. 2, is actually two cones on the same axis situated point-to-point. To avoid ambiguities, we adopt the convention that $0<\Theta<\pi/2$ and use $\vec{U}$ to orient the cone. (The special case, $\Theta=\pi/2$, corresponds to the quadric surface for coincident planes, while the special case, $\Theta=0$ corresponds to a zero-radius cylinder.) Both cones are to be regarded as valid surfaces for which HOWFAR and HOWNEAR are to be calculated. If an application requires only one cone, then it will be assumed that the other "reflection" cone has been eliminated through the use of another surface that isolates only one of the cones.

### 4.4.1 HOWFAR?

Substituting the equation for particle trajectory, eq. 4, into the above yields a quadratic equation of the form $As^2+2Bs+C=0$, where the quadratic constants, $A$, $B$ and $C$, are:

$$\begin{aligned}
A &= \cos^2\Theta[\vec{u}-\vec{U}(\vec{u}\cdot\vec{U})]^2-\sin^2\Theta(\vec{u}\cdot\vec{U})^2\\
B &= \cos^2\Theta\vec{u}\cdot\{(\vec{p}-\vec{P})-\vec{U}[(\vec{p}-\vec{P})\cdot\vec{U}]-\sin^2\Theta\vec{U}[(\vec{p}-\vec{P})\cdot\vec{U}]\}\\
C &= \cos^2\Theta\{(\vec{p}-\vec{P})-\vec{U}[(\vec{p}-\vec{P})\cdot\vec{U}]\}^2-\sin^2\Theta[(\vec{p}-\vec{P})\cdot\vec{U}]^2
\end{aligned}$$

These constants may be employed in the general quadric surface algorithm for HOWFAR described previously.

### 4.4.2 HOWNEAR?

Using the method of Lagrange multipliers, $\vec{x}$ has the following solution in terms of components perpendicular and parallel to $\vec{U}$:

$$(\vec{x}-\vec{P})-\vec{U}[\vec{U}\cdot(\vec{x}-\vec{P})]=\frac{(\vec{p}-\vec{P})-\vec{U}[\vec{U}\cdot(\vec{p}-\vec{P})]}{1+\lambda}. \tag{29}$$

and

$$[\vec{U}\cdot(\vec{x}-\vec{P})]=\frac{[\vec{U}\cdot(\vec{p}-\vec{P})]}{1-\lambda\tan^2\Theta}. \tag{30}$$

Substituting this into equation for the cone, eq. 28 solves for the Lagrange multiplier in terms of its two roots,

$$\lambda=\frac{-\tan\Theta|\vec{U}\cdot(\vec{p}-\vec{P})|\pm\sqrt{(\vec{p}-\vec{P})^2-[\vec{U}\cdot(\vec{p}-\vec{P})]^2}}{-\tan\Theta|\vec{U}\cdot(\vec{p}-\vec{P})|\pm\tan^2\Theta\sqrt{(\vec{p}-\vec{P})^2-[\vec{U}\cdot(\vec{p}-\vec{P})]^2}}, \tag{31}$$

where the ± sign in the numerator are coupled, *i.e.* one solution for $\lambda$ has both positive and the other both negative. This yields a solution for the closest distance:

$$d = \left| \cos\Theta \sqrt{(\vec{p}-\vec{P})^2 - [\vec{U}\cdot(\vec{p}-\vec{P})]^2} \pm \vec{U}\cdot(\vec{p}-\vec{P}) \sin\Theta \right|_{\rm min} . \qquad (32)$$

# 5 Performance of the geometry routines

To test the performance of the geometry routines a test code was written to simulate particle transport from an arbitrary point to a surface. The "photon" model used always retained the initial direction vector while the "electron" model picked a random direction after each non-zero transport. Record was kept of the initial and subsequent "undershoots", "overshoots" and exact hits. Particles were tracked until they escaped to infinity. Also tallied was the average and maximum `HOWNEAR` after transport to the surface and the code could be run in either single or double precision. The numerical calculations performed in this section were done on a Silicon Graphics Indy workstation with a MIPS R4400 CPU Processor (chip revision 5.0) and a MIPS 4010 Floating Point Chip (chip revision 0.0) using IRIX 5.2 and Fortran 77 Version 4.0.1. The numerical calculations should be similar but not necessarily identical to those done with other computers, processors, operating systems or Fortran compilers. For each trial, $10^5$ iterations were performed. Thus, any scatter in the data us related to the characteristics or "graininess" of the numerical representation rather than a reflection of poor statistics that could be made better by running more iterations.

## 5.1 Single *vs.* double precision

For quadric surfaces one can expect numerical problems when the distance from the surface is very large in comparison with a measure of the radius of curvature of the surface in the vicinity of the intersection. Using the sphere as an example, the quadratic parameter is $C = (\vec{p}-\vec{X})^2 - R^2$ and thus when $R^2$, the radius squared of the sphere, becomes much smaller than the distance squared $(\vec{p}-\vec{X})^2$, the mathematics may not be able to resolve the sphere. For single precision arithmetic this threshold is about $10^{-7}$ while for double precision about $10^{-16}$. Thus, if one is using single precision arithmetic, one can expect difficulties in the neighbourhood of $|\vec{p}-\vec{X}|/R > 3000$, depending on the architecture. This may be important for some applications.

To illustrate this point, we consider the spherical geometry where a sphere of unit radius is centered at the origin. In fig. 3 is shown the intersection with the surface with the initial and subsequent intersection points on the sphere with rotations provided to render the drawings in two dimensions. The "electron" model was employed. Particles from distances $1000R$, $2000R$, $4096R$, $4097R$, were directed at the sphere from the left. The initial undershoots are depicted as triangles pointed to the left, initial overshoots as triangles pointed to the right and exact hits for initial or subsequent hits as circles, which also serve to delineate

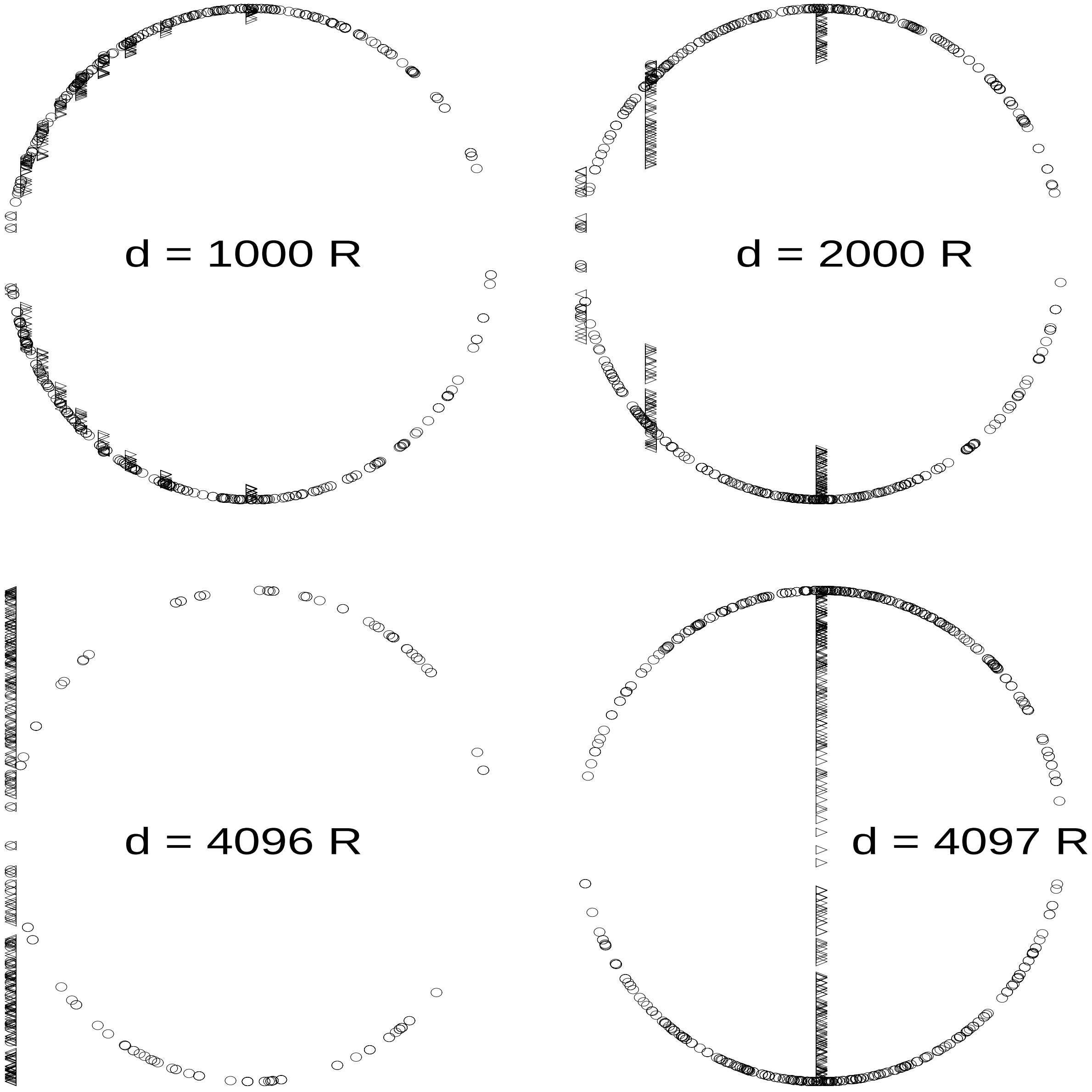

Figure 3: Particles from distances $1000R$, $2000R$, $4096R$, $4097R$, were directed at the sphere from the left. The initial undershoots are depicted as triangles pointed to the left, initial overshoots as triangles pointed to the right and exact hits for initial or subsequent hits as circles. The calculations were done in single precision arithmetic.

the sphere. The calculations were done in single precision arithmetic. The effects of single precision on this configuration are just barely discernible at $d = 1000R$. “Quantisation” of the initial undershoots and overshoots is evident by $d = 2000R$. By $d \approx 4000R$ the mathematics completely loses resolution of the sphere. For very large $d$ all the initial “hits” are placed on the plane normal to the vector from the center of the sphere to the source. Note the large difference between the $d = 4096R$ and the $d = 4097R$, an artefact produced by finite-precision mathematics. A double precision calculation only exhibits this behaviour at about $10^8R$.

A more quantitative demonstration is shown in fig. 4 where the mean and average `HOWNEAR` for this example are given for the initial “hit” on the sphere. The single and double precision calculations are contrasted. If one considered a 1% maximum undershoot or overshoot acceptable, then the single precision calculation is acceptable so long as the source of particles is within about $300R$ of the sphere. The double precision calculation performs much better but one has to ask whether the double precision overhead which can be significant on some 32-bit machines is worth it, particularly considering that the geometry routines are used intensively during typical applications. On 64-bit architectures, where double precision comes “for free”, one ought to employ double precision universally.

However, the use of higher precision does not eliminate nor change the characteristics of the undershoots or overshoots. This can be seen in fig. 5 which depicts the undershoot, overshoot and exact hit frequency *vs.* initial distance from the center of the sphere for initial hits and subsequent hits employing the “electron” model. Both single precision and double precision calculations are shown for both the initial hits and subsequent hits after direction randomisation. For the computer and software architecture tested, when the particle incident from within the sphere both single and double precision initial undershoot frequency is about 20%, the initial overshoot frequency is about 10% and the exact hits comprise the approximately 70%. When the particle incident from outside of the sphere both single and double precision initial undershoot and overshoot frequencies are about 50%. As the initial point gets farther from the sphere the single precision overshoot and undershoot results diverge in the vicinity of the numerical instability described previously. The subsequent hit data is relatively flat, showing little “history” of the initial overshoot or undershoot.

# 6 Alternate boundary crossing schemes

An alternative boundary-crossing scheme can be constructed by adding a small bit of extra transport to guarantee surface penetration. Combinatorial geometry packages that do not keep track of a particle’s assumed position with respect to a surface require such a scheme to attempt to eliminate problematic undershoots. A similar scheme is the concept of “fuzzy” surfaces that “shrink” by a small amount when a particle is directed at it and “swell” after penetration.

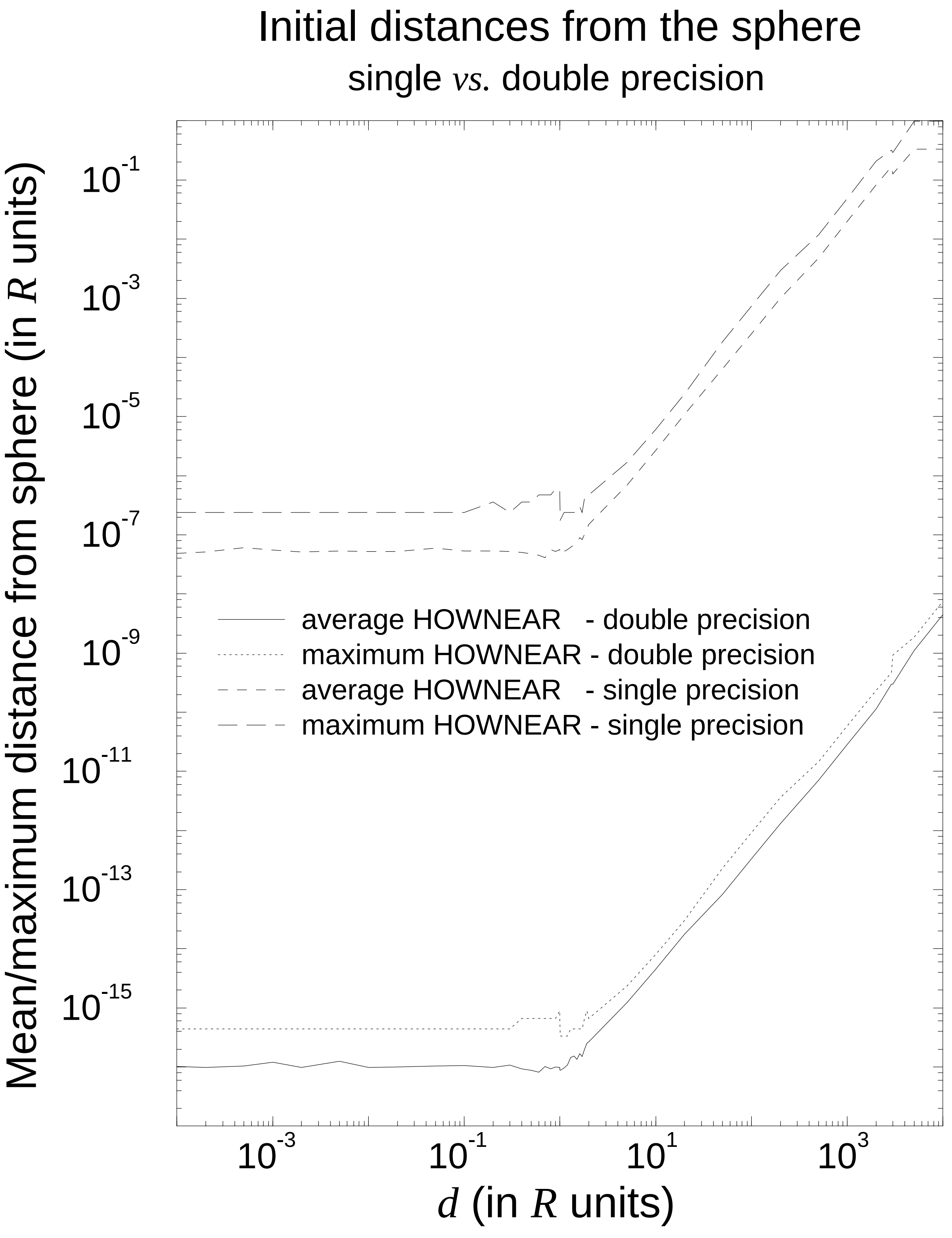

Figure 4: Mean and average HOWNEAR for this example are given for the initial "hit" on the sphere contrasting the performance of single and double precision arithmetic.

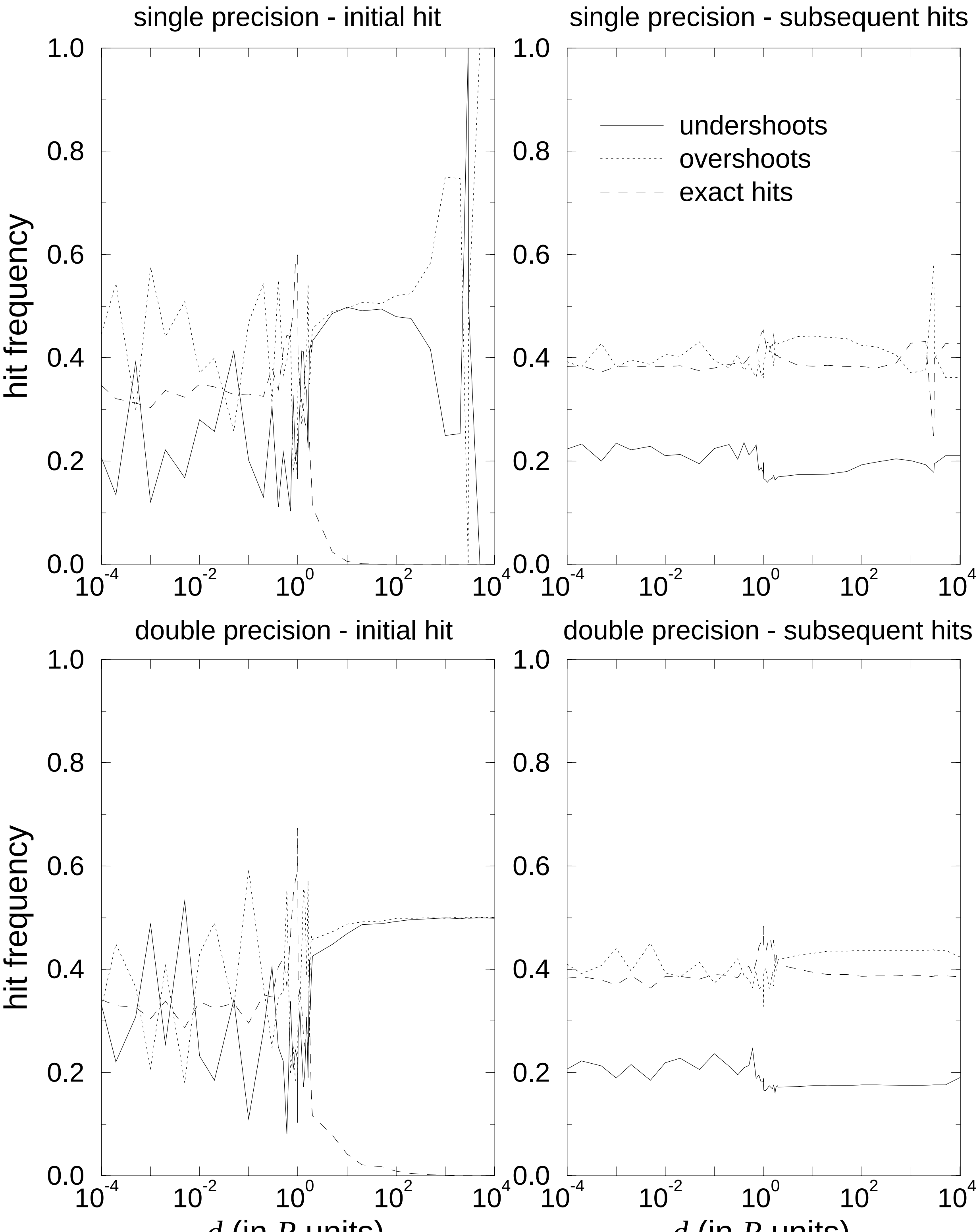

Figure 5: Undershoot, overshoot and exact hit frequency *vs.* initial distance from the center of the sphere for initial hits and subsequent hits employing the "electron" model. Single precision and double precision calculations are shown.

To study this, consider the effect of extra transport on the initial hits on a sphere. A single precision calculation is shown in fig. 6. In this case the "extra transport" was $\epsilon = 2.4\times 10^{-4}R$. In this case one notes that the overshoot frequency is 100% until the initial point is a distance of about $20R$. This threshold can be changed by choosing a different value for $\epsilon$. In a scheme where undershoots are problematic one notes that the price to be paid for having to invoke extra transport is poorer performance in `HOWNEAR` after the step to the surface, as indicated in the bottom half of fig. 6.

The double precision calculation is shown in fig. 7 which employed $\epsilon = 7.5 \times 10^{-9}R$. For this case the undershoot frequency has a threshold of about $300R$. `HOWNEAR` after transport is again much larger although values of about $10^{-8}R$ are acceptable for many calculations.

# 7 Conclusions

A particle tracking scheme that keeps track of the assumed position with respect to a surface was discussed. By employing this logic one can optimise the accuracy of the final position with respect to the surface. One does not need "fuzzy" boundaries or "extra" small transport distances to formulate a robust tracking algorithm.

Unless all transport is to take place in the vicinity of a given surface, it is best to use double precision.

A universal algorithm for all quadric surfaces was given. The `HOWFAR` intercept to any quadric surface can be calculated by determining the quadratic constants and then substituting them into the universal algorithm.

The `HOWNEAR` solution for an arbitrary quadric requires the determination of the zeros of a $5^{\rm th}$ or $6^{\rm th}$ polynomial. Symmetries reduce the order of the polynomial. Spheres, circular cones and circular cylinders are second order while a plane is first order. Analytic expressions were given for these surfaces.

# Acknowledgements

The author wishes to thank Keath Borg of the University of Victoria, Canada (Co-op student, Winter term 1995) for the work on QUADPLOT [11] which generated fig. 2 and Kenny Toh of the Intel Corporation for his rapid modifications to the MTV package that were required make this figure.

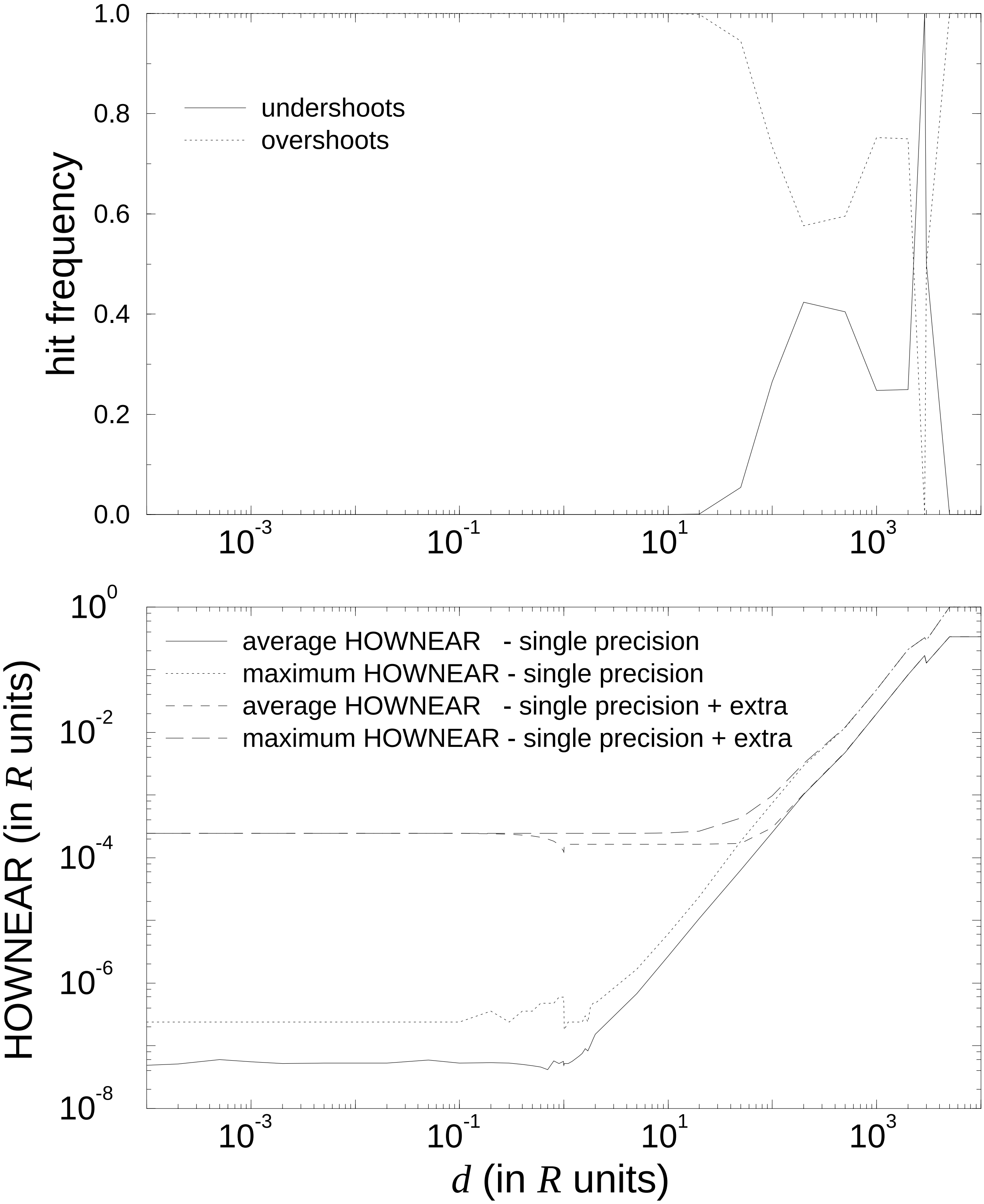

Figure 6: Undershoot, overshoot and exact hit frequency *vs.* initial distance from the center of the sphere for initial hits obtained by added a small quantity to the transport step in a single precision calculation. Also shown are the average and maximum `HOWNEAR`'s with and without a small extra transport of $2.4 \times 10^{-4} R$.

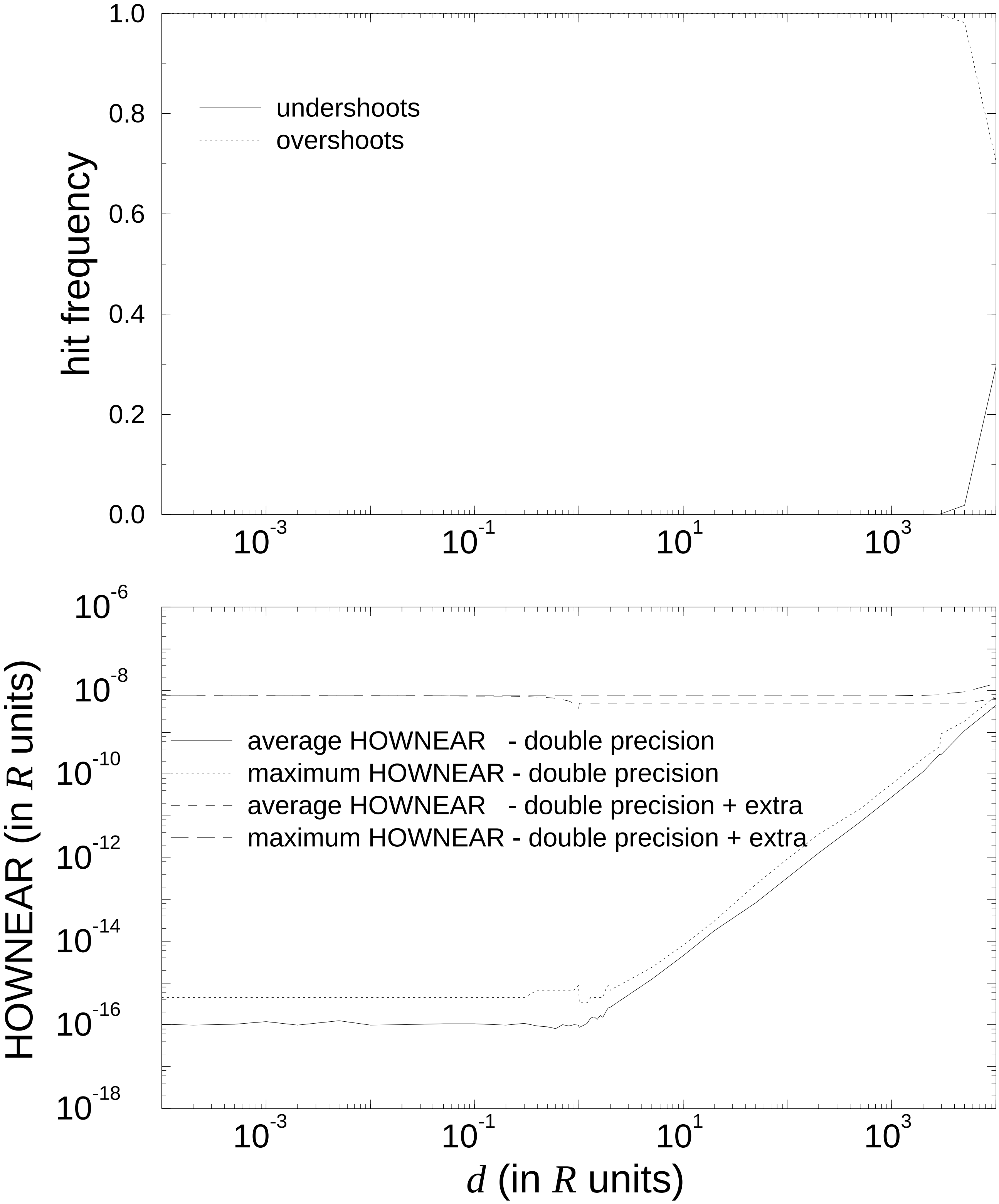

Figure 7: Undershoot, overshoot and exact hit frequency *vs.* initial distance from the center of the sphere for initial hits obtained by added a small quantity to the transport step in a double precision calculation. Also shown are the average and maximum `HOWNEAR`'s with and without a small extra transport of $7.5 \times 10^{-9} R$.

# References

[1] A.F. Bielajew and D.W.O. Rogers, "Variance-Reduction Techniques," in "Monte Carlo Transport of Electrons and Photons Below 50 MeV", eds. T.M. Jenkins, W.R. Nelson, A. Rindi, A.E. Nahum and D.W.O. Rogers, (Plenum Press) 407 – 419 (1989).

[2] A.F. Bielajew and D.W.O. Rogers, "PRESTA: The Parameter Reduced Electron-Step Transport Algorithm for Electron Monte Carlo Transport," National Research Council of Canada Report PIRS-0042 (1986).

[3] A.F. Bielajew and D.W.O. Rogers, "PRESTA: The Parameter Reduced Electron-Step Transport Algorithm for Electron Monte Carlo Transport," Nuclear Instruments and Methods **B18** 165 – 181 (1987).

[4] A.F. Bielajew and D.W.O. Rogers, "Electron Step-Size Artefacts and PRESTA," in "Monte Carlo Transport of Electrons and Photons Below 50 MeV", eds. T.M. Jenkins, W.R. Nelson, A. Rindi, A.E. Nahum and D.W.O. Rogers, (Plenum Press) 115 – 137 (1989).

[5] W.R. Nelson, H. Hirayama and D.W.O. Rogers, "The EGS4 Code System," Stanford Linear Accelerator Center Report SLAC-265 (Stanford Calif) (1985).

[6] D.W.O. Rogers, "Low energy electron transport with EGS," Nucl. Inst. Meth. **227** 535 – 548 (1984).

[7] P.A. Aarnio, "Particle tracking across boundaries in simulation of high energy hadronic and electromagnetic cascades," Physica Scripta **T33** 147 – 151 (1990).

[8] W.R. Nelson, "private communication," (conversation with A.F. Bielajew) (1995).

[9] R.L. Ford and W.R. Nelson, "The EGS code system – Version 3," Stanford Linear Accelerator Center Report SLAC-210 (1978).

[10] J.M.H. Olmsted, "Solid Analytic Geometry," (Appleton-Century-Crofts Inc, New York) (1947).

[11] K.R. Borg and A.F. Bielajew, "QUADPLOT: A programme to plot quadric surfaces," National Research Council of Canada Report PIRS-0491 (1995).