

SAND2017-10597 J  
Continuously differentiable PIC shape functions for triangular meshes

D. C. Barnes<sup>a</sup>

*Sandia National Laboratory, P.O. Box 5800, Mail stop 0878, Albuquerque, NM 87185*

---

**Abstract**

A new class of continuously-differentiable shape functions is developed and applied to two-dimensional electrostatic PIC simulation on an unstructured simplex (triangle) mesh. It is shown that troublesome aliasing instabilities are avoided for cold plasma simulation in which the Debye length is as small as 0.01 cell sizes. These new shape functions satisfy all requirements for PIC particle shape. They are non-negative, have compact support, and partition unity. They are given explicitly by cubic expressions in the usual triangle logical (areal) coordinates. The shape functions are not finite elements because their structure depends on the topology of the mesh, in particular, the number of triangles neighboring each mesh vertex. Nevertheless, they may be useful as approximations to solution of other problems in which continuity of derivatives is required or desired.

---

---

\*Corresponding author

*Email address:* dbarnes@sandia.gov (D. C. Barnes<sup>a</sup>)

<sup>a</sup>Permanent address: Coronado Consulting, Lamy, NM 87540

## 1. Introduction

Plasma simulation using the Particle-in-cell (PIC) method is one of the most valuable tools for describing kinetic phenomena.[1][2] Some applications of PIC to two-dimensional (2D) geometries employ triangular elements.[3, 4, 5] The use of triangles allows almost complete flexibility in obtaining a mesh for domains with complex shapes and/or which contain internal structures of complex shape, and are unmatched in this flexibility by any alternative meshing of such domains.

In these previous approaches, the particle shape function, which provides interpolation between the particles and mesh in both directions is usually a linear “tent” function which is continuous but has discontinuous derivative(s). Thus, for example, if the electromagnetic field is assumed to be electrostatic only, the electrical potential will be continuous, but its derivative which gives the electric field used to advance particle velocities is not. In this way, particles receive piece-wise constant accelerations as they move within a triangular element and experience jumps in this acceleration as they cross element boundaries. These jumps lead to increased particle noise and are suspected to exacerbate the aliasing instabilities which limit application of PIC to cold, drifting plasmas (*e.g.* beam problems).

In fact, it has become evident in practice that a much superior algorithm results from the use of continuously-differentiable particle shape functions, and it is often the case that the optimum continuity is realized by the use of  $C_1$  elements, such as tensor-product, quadratic B-splines for regular rectangular meshes. That is to say, there appears empirically to be a great advantage to the use of quadratic splines for PIC shape functions, and the advantages of higher-order splines (with more continuous derivatives) is only of marginal advantage in usual applications. Recent work has shed more light on this observation by quantitative analysis of several methods.[6]

The reason for choosing linear,  $C_0$  elements for triangular mesh PIC is, of course, not because the advantages of  $C_1$  elements do not apply, but simply because such elements are not readily available. Only a few practical  $C_1$  interpolation schemes are known for triangular meshes,[7][8] and these do not actually improve the usual PIC algorithms because of the following limitation. These  $C_1$  schemes introduce additional quantities per triangle to describe the elements. For example, the relatively economical scheme of Ref. [7] requires 6 quantities per vertex to determine the shape functions. If such a scheme is applied to a given number of particles per element, there will then be fewer PIC markers per degree-of-freedom, increasing the noise by the square root of this multiplicity. In this way, the additional advantages accrued by the increased smoothness are largely or completely canceled by this increase in noise. In contrast, consider the application of the usual quadratic B-spline on a rectangular mesh. In this case, there is but a single degree-of-freedom per vertex, and the shape function extends over a larger portion of the mesh. In this case, there are actually more PIC markers per degree-of-freedom, so the advantages of smoothness are supplemented by the reduction of noise associated with this gain.

There has been significant related work on this subject, both within PIC applications and in related finite-element applications to continuum problems. Unstructured PIC has been extended to three dimensional domains with complicated boundary conditions using usual  $C_0$  elements [9][10], while other authors have employed direct numerical convolution of smooth particle shapes to move toward  $C_1$  elements.[11, 12] These applications might benefit from the reduced noise and improved dispersion properties of  $C_1$  elements given by analytic (polynomial) expressions, with obvious potential computational advantages. All of these PIC algorithms employ a “momentum conserving” algorithm in which forces are interpolated from mesh to particles using the same shape functions as used to deposit charge from particles to mesh, and hence are restricted in cell size to a small multiple of Debye length. [6]

There have also been a large body of work on higher-order polynomial elements within the general FEM framework. [see for example the nice summary in [13]] In the present work, we have borrowed heavily from some of the methods used in this work, including our pseudo-polar coordinates used later. Such approaches, when applied to unstructured meshes, are formally  $C_0$  but offer higher-order convergence

because of providing a better approximation to a smooth solution. Introduction of additional degrees of freedom limits the advantages of smoother fields, as already noted. This is avoided in the present approach which broadens the support of a single particle.

In this paper, a new class of  $C_1$  interpolation is developed and applied as shape functions to triangular PIC. The shape functions described here are perhaps the direct analog of the rectangular mesh quadratic B-spline. The development follows a simple observation, which is that the quadratic B-spline is obtained as a moving average of the linear B-spline, in which the window for averaging is a simple square wave (often referred to as “boxcar” averaging). It is shown how to generalize this to the triangular case, and how to resolve difficulties which arise because of the intrinsically unstructured nature of the mesh, which then presents a multiplicity of cases of different connectivity which must be resolved. This first paper is focused on the mechanics of obtaining the desired functions. Nevertheless, their application to a full nonlinear PIC simulation is illustrated by a single application and a cursory comparison with usual  $C_0$  elements. Such testing cannot provide a full verification and/or validation of the method, nor quantify many of its (low) collisional properties, but space does not permit a deeper study and presentation, which subjects is left for future investigation.

While our discussion is focused on the application of the resulting interpolation functions to PIC and the development accordingly described in the language of PIC, many other applications of these functions are possible. For example, the finite-element solution of higher-order partial differential equations is most conveniently done using elements with higher continuity. The  $C_1$  elements derived here are useful for solution of fourth-order elliptic problems, for example. Some additional applications are mentioned in the discussion section here.

The remainder of this paper is organized as follows. The following section describes the mathematics of the approach, while section 3 works out the details for all required cases. Section 4 gives examples of the resulting shape functions, and section 5 shows their prototype application to triangular PIC. The final section contains a discussion and conclusions and points toward generalization of this approach to three dimensions.

## 2. The moving window approach

### 2.1. 1D formulation

Consider first the case of a 1D non-uniform (trivially) rectangular grid. There is a “logical coordinate”  $\xi$  which assumes integer values at each mesh point and varies continuously between, so that a continuous mapping from logical (L-space) to Cartesian (C-space) coordinate  $x(\xi)$  exists. We derive a (the usual) set of  $C_1$  (functions of a single variable with continuous derivative) useful for interpolation on this mesh. For this, we will first consider the usual linear interpolation which provides a  $C_0$  interpolation and then modify this accordingly. Consider the usual “tent functions” which span the space of linear interpolation functions. The  $i$ -th such function is simply

$$X_i^0(\xi) = \begin{cases} \xi - i + 1, & i - 1 < \xi < i \\ i + 1 - \xi, & i < \xi < i + 1 \\ 0, & \text{elsewhere} \end{cases} \quad (1)$$

We then have a  $C_0$  interpolate in C-space of a nodal field  $\{\phi_i\}$  given parametrically by

$$\phi^0 \left[ \sum_i X_i^0(\xi) x_i \right] = \sum_i X_i^0(\xi) \phi_i \quad (2)$$

To obtain a similar  $C_1$  interpolate, we replace the basis functions  $\{X_i^0\}$  by a set with continuous derivative. This leads to some differences in the representation. In particular, the nodal values of the independent

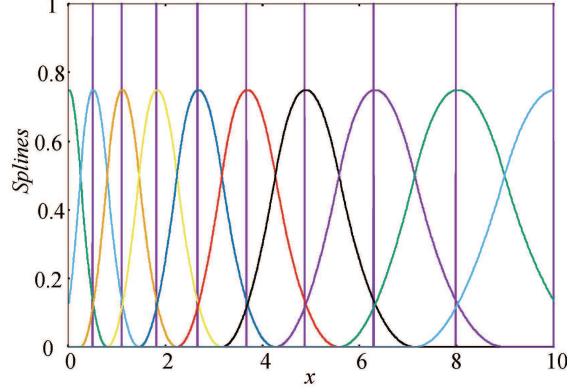


Figure 1: Quadratic splines on non-uniform 1D mesh. Mesh nodes are shown by vertical tick marks. Notice that splines are not maximum at nodes.

( $x$ ) and dependent ( $\phi$ ) variables are not the coefficients  $\{x_i, \phi_i\}$  because it is not possible to obtain a set of basis functions which are simultaneously nodal (non-zero only at a single node), positive, and  $C_1$ . In order to obtain a useful PIC method, we require positivity for our interpolation functions, so we are forced to adopt a non-nodal basis set.

We describe now an algorithm for obtaining the desired  $C_1$  set of basis functions from the lower continuity set  $\{X_i^0\}$  which algorithm may be generalized to higher dimensions and eventually to un-structured meshes. The method is to average the lower continuity set over the logical coordinate. Thus,

$$X_i^1(\xi) = \int_{\xi-1/2}^{\xi+1/2} d\xi' X_i^0(\xi') \quad (3)$$

It is easy to show that the functions  $\{X_i^1\}$  satisfy the following,  $\forall i$ :

- $X_i^1$  is non-negative
- $X_i^1$  has compact support and vanishes for  $|\xi - i| \geq 3/2$
- $X_i^1$  is continuous and has continuous derivative
- $X_i^1$  is an interpolation set;  $\sum_i X_i^1(\xi) = 1, \forall \xi$ .

In fact, it is clear that the  $X_i^1$  are just the quadratic B-splines with knots at 1/2-integer  $\xi$  values. We have thus described the usual PIC quadratic interpolation scheme, using the elements shown in Fig. 1.

## 2.2. 2D formulation

To extend the previous discussion to the case of 2D rectangular meshes, we just consider the tensor product of 1D formulations for each of the Cartesian coordinates. Thus, there are now two logical coordinates  $\xi_x$  and  $\xi_y$ . We construct a  $C_0$  basis using the basis functions

$$Z_{ij}(\xi_x, \xi_y) = X_i^0(\xi_x) X_j^0(\xi_y) \quad (4)$$

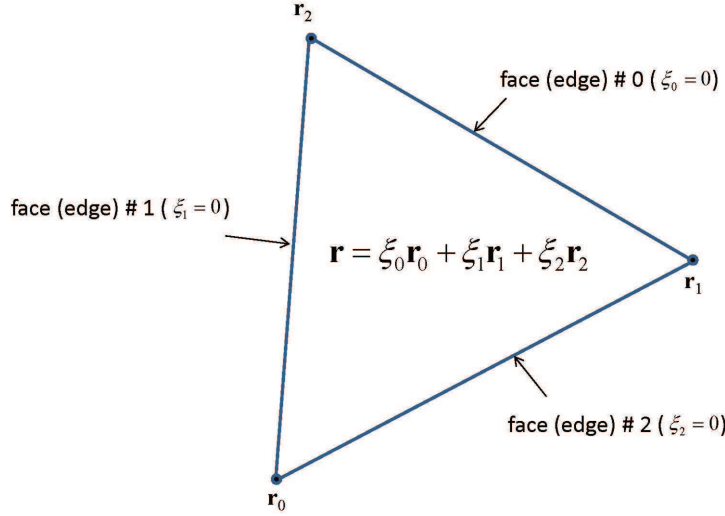


Figure 2: Logical coordinates for triangle

and then smooth this by averaging over an appropriate region in (2D) L-space.

To define this region in a way that will be subsequently useful for extension to the un-structured case, we introduce the concept of a metric on the logical space. For any two points  $(\xi_x, \xi_y)$  and  $(\xi'_x, \xi'_y)$ , we define the distance between these in terms of the  $L_\infty$  norm

$$D[(\xi_x, \xi_y), (\xi'_x, \xi'_y)] = \max \{ |\xi_x - \xi'_x|, |\xi_y - \xi'_y| \} \quad (5)$$

and then define the region of integration for averaging as  $D[(\xi_x, \xi_y), (\xi'_x, \xi'_y)] \leq 1/2$ . It is easy to verify that this again leads to the usual 2D PIC quadratic interpolation scheme. Notice that the neighborhood defined by  $D$  is topologically a “circle” with center at the field point  $(\xi_x, \xi_y)$ .

### 2.3. Triangular case

If we can find an appropriate metric for a general triangular mesh, we can attempt the same program discussed previously to obtain  $C_1$  elements for triangular PIC applications. It is natural to use the usual logical coordinates for triangular elements as shown in Fig. 2. With the PIC application in mind, we will subsequently refer to the mesh nodes as “vertices” and the interior of the elements as “cells”.

A natural generalization of the previous metric is to define the distance as

$$D_\Delta[(\xi_0, \xi_1, \xi_2), (\xi'_0, \xi'_1, \xi'_2)] = \max \{ |\xi_0 - \xi'_0|, |\xi_1 - \xi'_1|, |\xi_2 - \xi'_2| \} \quad (6)$$

The unstructured nature of the triangular mesh makes it necessary to interpret the definition of Eq. (6) as follows:

- Two points within the same cell have distance as defined literally by Eq. (6).
- This is then extended to neighboring cells by finding the geodesic. Consider all possible paths joining two points  $P$  and  $P'$  by traveling from  $P$  to a point on the boundary of its triangle  $t$ , then across the neighboring cell to another boundary point, and so on until reaching the boundary of  $t'$ , the triangle to which  $P'$  belongs, and finally joining this last boundary point with  $P'$ . The length of this path is then

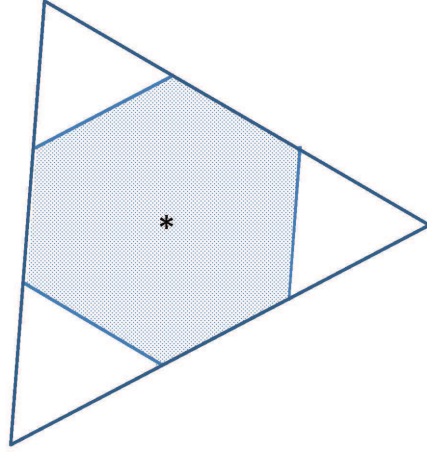


Figure 3: Neighborhood of triangle barycenter.

computed as the sum of the lengths of each of its segments, each of which share a common triangle, so that Eq. (6) can be used to evaluate these segment lengths. The (a) geodesic is then the path which minimizes this total travel distance between  $P$  and  $P'$  and the distance between  $P$  and  $P'$  is the length of this path.

It develops that we are interested in only a small set of cases, since we need compute an average over a small neighborhood of the mesh. We illustrate for the specific case of the neighborhood defined by  $D_{\Delta} \leq 1/3$ , which is the triangular analog of the 2D rectangular case. We denote this region of L-space as  $H_{r_v}(\ell)$ , where  $\ell$  is the L-space location of the field point [consisting apparently of a home cell index and a triad of logical coordinates  $(\xi_0, \xi_1, \xi_2)$ ], and  $r_v = 1/3$  is the “radius” of the neighborhood. For this choice, a field point at the exact center (barycenter) of a triangle has a  $H$  which is completely contained within the “home” triangle. This is illustrated in Fig. 3 where the neighborhood of size  $1/3$  of the barycenter point (marked with a “\*”) is shown. As is seen, this neighborhood is defined by the inequalities  $0 \leq \xi_i \leq 2/3, \forall i$ , and forms a hexagonal region about the barycenter  $\xi_0 = \xi_1 = \xi_2 = 1/3$ .

Because we restrict the radius of our averaging region to be  $\sim 1/3$ ,  $H_{r_v}(\ell)$  extends from any field point only over a small number of neighboring cells. For the purposes of this paper, the resulting topologies will simply be stated, as the arguments by which these are determined are complex and somewhat irrelevant to the main results.

To begin, there are two distinct cases.

- When the field point lies within the central hexagon shown in Fig. 3, the neighborhood may be represented as a rigid shift of the hexagon as shown in Fig. 4. This is referred to as the “C-reg” case.
- When the field point lies within one of the three triangles outside the central hexagon and close to a vertex, the neighborhood assumes a shape which may be very complicated, and which depends on the number of neighbors of the subject vertex. This is referred to as the “V-reg” case.

Both of these cases are considered in detail in the next section. In order to carry out the averages over the L-space region of interest, it is necessary to make specific the area measure on this space, which is somewhat arbitrary, since there is no real connection to Euclidean geometry here. The area element is chosen to be

$$dA = d\xi_0 d\xi_1 d\xi_2 \delta(1 - \xi_0 - \xi_1 - \xi_2) \quad (7)$$

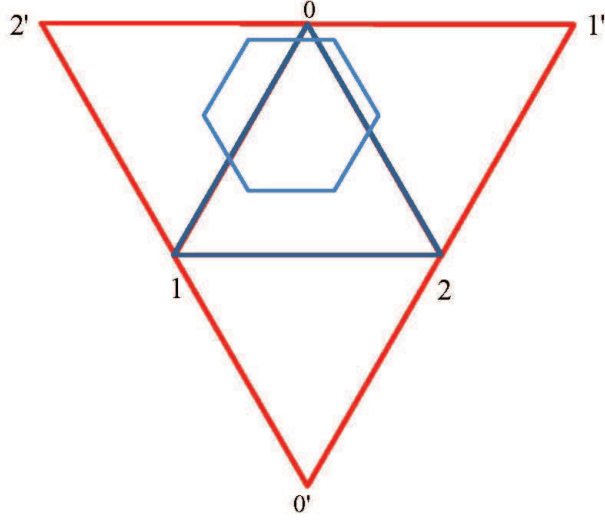


Figure 4: L-space representation of cell region case

This is the slant plane area element when the logical coordinates are represented as three space orthogonal coordinates.

Next, super-coordinates for the neighborhood of the home cell shown in Fig. 4 are introduced. These super-coordinates are just the logical coordinates of the home cell extended into the domain shown in Fig. 4, and thus ranging over the interval  $[-1, 1]$ . We denote these coordinates by upper-case Greek letters to distinguish from the local (to each cell) coordinates denoted by lower-case Greek letters. Thus, we have the mappings from super-coordinates  $(\Xi_0, \Xi_1, \Xi_2)$  to cell local logical coordinates

$$\begin{aligned}
 \Delta_{0,1,2}: \xi_0 &= \Xi_0, \xi_1 = \Xi_1, \xi_2 = \Xi_2 \\
 \Delta_{0,2',1}: \xi_0 &= 1 - \Xi_1, \xi_{2'} = -\Xi_2, \xi_1 = 1 - \Xi_0 \\
 \Delta_{1,0',2}: \xi_1 &= 1 - \Xi_2, \xi_{0'} = -\Xi_0, \xi_2 = 1 - \Xi_1 \\
 \Delta_{0,2,1'}: \xi_0 &= 1 - \Xi_2, \xi_2 = 1 - \Xi_0, \xi_{1'} = -\Xi_1
 \end{aligned} \tag{8}$$

### 3. The triangular elements

The  $C_1$  elements may now be obtained by the moving window method. Each such element will be associated with a vertex of the mesh, in a manner which will become clear soon. At any given field point, the elements associated with only a few nearby vertices will be non-zero. Explicit expressions for these will be obtained by direct integration or by alternative means.

#### 3.1. C-reg

In the C-reg case, required integrals can be computed analytically. The development is outlined here. The resulting formulas have been checked by using the Mathematica[14] symbolic manipulation software. It is convenient to differentiate six sub-cases, depending on the location of the field point within the central hexagon, as indicated in Fig. 5, because each of these cases has a distinct stencil of non-zero vertex elements. The six sub-regions are:

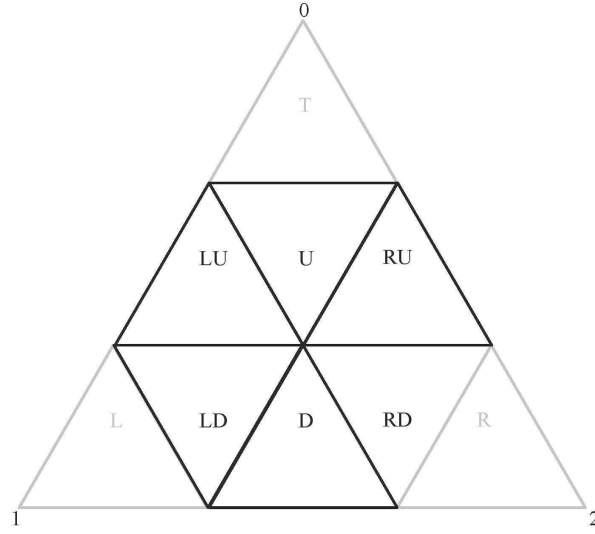


Figure 5: Division of the home cell

- Up (U)  $1/3 < \xi_0 < 2/3$  &  $0 < \xi_i < 1/3, i = 1, 2$
- Left-up (LU)  $1/3 < \xi_j < 2/3, j = 0, 1$  &  $0 < \xi_2 < 1/3$
- Left-down (LD)  $1/3 < \xi_1 < 2/3$  &  $0 < \xi_i < 1/3, i = 0, 2$
- Left (L)  $2/3 < \xi_1 < 1$
- Down (D)  $1/3 < \xi_j < 2/3, j = 1, 2$  &  $0 < \xi_0 < 1/3$
- Right-down (RD)  $1/3 < \xi_2 < 2/3$  &  $0 < \xi_i < 1/3, i = 0, 1$
- Right-up (RU)  $1/3 < \xi_2 < 2/3, 0, 2$  &  $0 < \xi_1 < 1/3$

Only two of the six C-reg sub-regions are distinct, with the remaining four determined from these by cyclic permutations of the vertices. For the LU sub-region, only vertices of the home cell (0, 1, 2), and the opposite vertex (2') of the neighbor opposite vertex 2 have non-zero elements.

In case the field point lies within LU, the neighborhood extends only over 2 cells, the home cell and the neighbor opposite vertex #2 (Fig. 7) which is  $\Delta_{0,2',1}$ . The extended coordinates of the field point [Eq. (8)] are  $(\Xi_0, \Xi_1, \Xi_2) = (\xi_0, \xi_1, \xi_2)$ , and the neighborhood in the extended coordinate space is  $\xi_j - 1/3 < \Xi_j < \xi_j + 1/3, j = 1, 2, 3$ . The area of the neighborhood is

$$A = \int_{\xi_0-1/3}^{\xi_0+1/3} d\Xi_0 \int_{\xi_1-1/3}^{\xi_1+1/3} d\Xi_1 \int_{\xi_2-1/3}^{\xi_2+1/3} d\Xi_2 \delta(1 - \Xi_0 - \Xi_1 - \Xi_2) \quad (9)$$

In order that the  $\delta$  function's argument vanish, we require the argument to be negative at the upper limit and positive at the lower limit, or

$$1 - \Xi_0 - (\xi_2 + 1/3) < \Xi_1 < 1 - \Xi_0 - (\xi_2 - 1/3) \quad (10)$$

Using  $\xi_0 + \xi_1 + \xi_2 = 1$ , we see that Eq. (10) becomes

$$\xi_0 + \xi_1 - 1/3 - \Xi_0 < \Xi_1 < \xi_0 + \xi_1 + 1/3 - \Xi_0 \quad (11)$$



When  $\Xi_0 < \xi_0$ , the lower limit of Eq. (11) is more restrictive than that coordinate limit of Eq. (9) in the integral over  $\Xi_1$ . Similarly, when  $\Xi_0 > \xi_0$ , the upper limit of Eq. (11) is more restrictive than that coordinate limit of Eq. (9) in the integral over  $\Xi_1$ . Thus,

$$A = \int_{\xi_0-1/3}^{\xi_0} d\Xi_0 \int_{\xi_0+\xi_1-1/3-\Xi_0}^{\xi_1+1/3} d\Xi_1 + \int_{\xi_0}^{\xi_0+1/3} d\Xi_0 \int_{\xi_1-1/3}^{\xi_0+\xi_1+1/3-\Xi_0} d\Xi_1$$

$$= \frac{1}{3} \quad (12)$$

We have given the details here because this sort of integral is of a standard form which will recur in the sequel. Notice that the area of the neighborhood is independent of its position, as long as the entire neighborhood is within the integration region. This will be true for the entire C-reg.

With this area as normalization, we can then compute the  $C_1$  elements. Consider first vertex #0 (of Figs. 6 & 7). The  $C_0$  basis for this vertex is just  $\xi_0 = \Xi_0$  in the home triangle and  $\xi_0 = 1 - \Xi_1$  in  $\Delta_{0,2',1}$ . Thus, we find

$$AX_0^1 = \int_0^{\xi_2+1/3} d\Xi_2 \int_{\xi_1-1/3}^{\xi_1+1/3} d\Xi_1 \int_{\xi_0-1/3}^{\xi_0+1/3} d\Xi_0 \Xi_0 \delta(1 - \Xi_0 - \Xi_1 - \Xi_2)$$

$$+ \int_{\xi_2-1/3}^0 d\Xi_2 \int_{\xi_1-1/3}^{\xi_1+1/3} d\Xi_1 (1 - \Xi_1) \int_{\xi_0-1/3}^{\xi_0+1/3} d\Xi_0 \delta(1 - \Xi_0 - \Xi_1 - \Xi_2) \quad (13)$$

Carefully carrying out the indicated integrals and using  $A = 1/3$ , we find the final form

$$X_0^1 = \frac{25}{27} - \xi_1 - \frac{\xi_2}{2} - \xi_2^2 + \frac{\xi_2^3}{2} \quad (14)$$

In an exactly analagous manner, we obtain the shape functions for vertex 1, 2, and 2' giving the result:

**LU**

$$X_0^1 = \frac{25}{27} - \xi_1 - \frac{\xi_2}{2} - \xi_2^2 + \frac{\xi_2^3}{2}$$

$$X_1^1 = -\frac{2}{27} + \xi_1 + \frac{\xi_2}{2} - \xi_2^2 + \frac{\xi_2^3}{2}$$

$$X_2^1 = \frac{2}{27} + \frac{\xi_2}{2} + \xi_2^2 - \frac{\xi_2^3}{2}$$

$$X_{2'}^1 = \frac{2}{27} - \frac{\xi_2}{2} + \xi_2^2 - \frac{\xi_2^3}{2}$$

If the field point is located in the upper triangle of the central hexagon (U of Fig. 8) there is a contribution to the desired overlap integrals from the home cell, and from the two neighboring cells opposite to vertices #1 and #2. In complete analogy to the derivation of Eq. (13), we find

$$AX_0^1 = \int_0^{\xi_2+1/3} d\Xi_2 \int_0^{\xi_1+1/3} d\Xi_1 \int_{\xi_0-1/3}^{\xi_0+1/3} d\Xi_0 \Xi_0 \delta(1 - \Xi_0 - \Xi_1 - \Xi_2)$$

$$+ \int_{\xi_2-1/3}^0 d\Xi_2 \int_0^{\xi_1+1/3} d\Xi_1 (1 - \Xi_1) \int_{\xi_0-1/3}^{\xi_0+1/3} d\Xi_0 \delta(1 - \Xi_0 - \Xi_1 - \Xi_2)$$

$$+ \int_0^{\xi_2+1/3} d\Xi_2 \int_{\xi_1-1/3}^0 d\Xi_1 (1 - \Xi_2) \int_{\xi_0-1/3}^{\xi_0+1/3} d\Xi_0 \delta(1 - \Xi_0 - \Xi_1 - \Xi_2) \quad (15)$$

Carrying out the algebra as earlier, we find the final form

$$X_0^1 = \frac{23}{27} - \frac{\xi_1}{2} - \frac{\xi_2}{2} - \xi_1^2 - \xi_2^2 + \frac{\xi_1^3}{2} + \frac{\xi_2^3}{2} \quad (16)$$

There is no modification to the element associated with vertex #2', while the element associated with vertex #1' is exactly the same with the interchange  $1 \longleftrightarrow 2$ . Vertex #1 is treated in analogy with the earlier calculation for the LU region and vertex #2 is the same as vertex #1 with the interchange  $1 \longleftrightarrow 2$ .

Summarizing, the non-zero  $C_1$  elements are:

**U**

$$\begin{aligned} X_0^1 &= \frac{23}{27} - \frac{\xi_1}{2} - \frac{\xi_2}{2} - \xi_1^2 - \xi_2^2 + \frac{\xi_1^3}{2} + \frac{\xi_2^3}{2} \\ X_1^1 &= \frac{\xi_1}{2} + \frac{\xi_2}{2} + \xi_1^2 - \xi_2^2 - \frac{\xi_1^3}{2} + \frac{\xi_2^3}{2} \\ X_2^1 &= \frac{\xi_1}{2} + \frac{\xi_2}{2} - \xi_1^2 + \xi_2^2 + \frac{\xi_1^3}{2} - \frac{\xi_2^3}{2} \\ X_{1'}^1 &= \frac{2}{27} - \frac{\xi_1}{2} + \xi_1^2 - \frac{\xi_1^3}{2} \\ X_{2'}^1 &= \frac{2}{27} - \frac{\xi_2}{2} + \xi_2^2 - \frac{\xi_2^3}{2} \end{aligned}$$

The 4 remaining sub-regions of the cell region of Fig. 7 have elements which can be obtained by cyclic permutations of the logical coordinates. These are

**LD**

$$\begin{aligned} X_0^1 &= \frac{\xi_0}{2} + \frac{\xi_2}{2} + \xi_0^2 - \xi_1^2 - \frac{\xi_0^3}{2} + \frac{\xi_1^3}{2} \\ X_1^1 &= \frac{23}{27} - \frac{\xi_0}{2} - \frac{\xi_2}{2} - \xi_0^2 - \xi_2^2 + \frac{\xi_0^3}{2} + \frac{\xi_2^3}{2} \\ X_2^1 &= \frac{\xi_0}{2} + \frac{\xi_2}{2} - \xi_0^2 + \xi_1^2 + \frac{\xi_0^3}{2} - \frac{\xi_1^3}{2} \\ X_{0'}^1 &= \frac{2}{27} - \frac{\xi_0}{2} + \xi_0^2 - \frac{\xi_0^3}{2} \\ X_{2'}^1 &= \frac{2}{27} - \frac{\xi_2}{2} + \xi_2^2 - \frac{\xi_2^3}{2} \end{aligned}$$

**D**

$$\begin{aligned} X_0^1 &= \frac{2}{27} + \frac{\xi_0}{2} + \xi_0^2 - \frac{\xi_0^3}{2} \\ X_1^1 &= \frac{25}{27} - \xi_2 - \frac{\xi_0}{2} - \xi_0^2 + \frac{\xi_0^3}{2} \\ X_2^1 &= -\frac{2}{27} + \xi_2 + \frac{\xi_0}{2} - \xi_0^2 + \frac{\xi_0^3}{2} \\ X_{0'}^1 &= \frac{2}{27} - \frac{\xi_0}{2} + \xi_0^2 - \frac{\xi_0^3}{2} \end{aligned}$$

**RD**

$$\begin{aligned}
X_0^1 &= \frac{\xi_0}{2} + \frac{\xi_1}{2} + \xi_0^2 - \xi_1^2 - \frac{\xi_0^3}{2} + \frac{\xi_1^3}{2} \\
X_1^1 &= \frac{\xi_0}{2} + \frac{\xi_1}{2} - \xi_0^2 + \xi_1^2 + \frac{\xi_0^3}{2} - \frac{\xi_1^3}{2} \\
X_2^1 &= \frac{23}{27} - \frac{\xi_0}{2} - \frac{\xi_1}{2} - \xi_0^2 - \xi_1^2 + \frac{\xi_0^3}{2} + \frac{\xi_1^3}{2} \\
X_{0'}^1 &= \frac{2}{27} - \frac{\xi_0}{2} + \xi_0^2 - \frac{\xi_0^3}{2} \\
X_{1'}^1 &= \frac{2}{27} - \frac{\xi_1}{2} + \xi_1^2 - \frac{\xi_1^3}{2}
\end{aligned}$$

**RU**

$$\begin{aligned}
X_0^1 &= -\frac{2}{27} + \xi_0 + \frac{\xi_1}{2} - \xi_1^2 + \frac{\xi_1^3}{2} \\
X_1^1 &= \frac{2}{27} + \frac{\xi_1}{2} + \xi_1^2 - \frac{\xi_1^3}{2} \\
X_2^1 &= \frac{25}{27} - \xi_0 - \frac{\xi_1}{2} - \xi_1^2 + \frac{\xi_1^3}{2} \\
X_{1'}^1 &= \frac{2}{27} - \frac{\xi_1}{2} + \xi_1^2 - \frac{\xi_1^3}{2}
\end{aligned}$$

### 3.2. *V-reg*

In principle, it is possible to extend the method of the previous discussion to the V-reg. There are however, grave difficulties in this approach. First of all, the hexagon of Fig. 4 may extend into regions of L-space which do not correspond to any cell, in case there are less than 6 cells in the vertex region. At the same time, it may extend into cells which are “images” of actual cells, which are covered more than once by the mapping from L-space. For example, if there are 3 cells at a given vertex, the hexagonal neighborhood becomes distorted as shown in Fig. 6. In this case, the neighborhood is obtained by picturing the vertex as having 6 triangle neighbors and then identifying the three not shown with the three shown in the Fig. The dotted line portion of the hexagonal neighborhood results from this identification, and the red triangle is an additional region of the neighborhood.

In addition to the complex shape of the neighborhood, there are additional difficulties here. The area of the neighborhood is no longer constant, nor does shifting the field point correspond to a rigid shift of the neighborhood, considerably complicating the calculations of the average-over-neighborhood  $C_1$  functions.

For various reasons, we seek a more straightforward method to represent the  $C_1$  functions in the vertex region. An alternative approach is to notice that the vertex region consists of a polygon of three or more sides which is subdivided into triangular elements, 1/3 of each neighboring cell. Further, the functions and their gradients are known on the edges of these triangles opposite the vertex of interest and are continuous all around the perimeter formed by joining these into the polygonal shape. Suppose then that we interpolate these functions directly into the vertex region using a consistent interpolation, which assures continuity of the functions and their derivatives. This is sufficient to assure that the interpolate differs from the brute force average described previously by a very small amount (3rd order in the mesh spacing). Such an interpolate is then useful for PIC applications and results will be indistinguishable from the actual average-over-neighborhood functions.

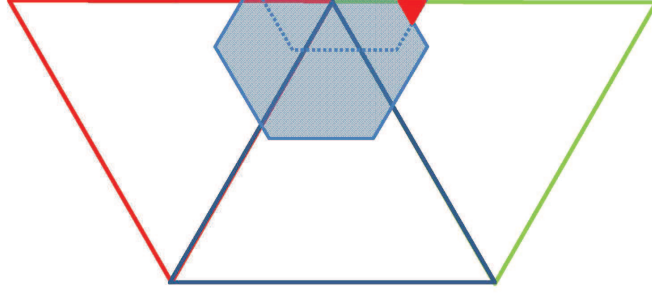


Figure 6: Logical space representation of the vertex region when only 3 cells intersect at the vertex.

We can obtain such an interpolation rule by considering each ray extending from  $V$  to an arbitrary point on the perimeter (Fig. 10), and finding formula for the value and directional derivative along the ray at each end, and then using Hermite cubic interpolation in the single coordinate along the ray.

We have thus reduced the problem to finding the value of the functions and their derivatives at the central point  $V$  (the vertex of interest). In contrast to the complexly-shaped neighborhood at field points intermediate between the cell region and  $V$ , the neighborhood assumes a very simple shape for field points close to  $V$ . In this locale of  $L$ -space, the neighborhood over which averages are to be computed is simply the region pictured in Fig. 7, the  $1/3$  of all cells neighboring  $V$  which is closest to  $V$ . It is then straightforward to compute the required average, leading to

$$\langle Q \rangle_V = \frac{7}{9}Q_V + \frac{2}{9N_v} \sum_{i=1}^{N_v} Q_i \quad (17)$$

where  $N_v$  is the number of cells meeting at vertex  $V$ , and  $\{Q_i\}$  the values at the vertices neighboring  $V$ . Obviously, the value of the vertex shape functions at  $V$  are  $7/9$  for vertex  $V$  and  $2/9N_v$  for each of the neighbors of  $V$ .

Computing the gradient of a given field at  $V$  by performing a virtual displacement of the field point shows that our prescription needs a small modification. In order to obtain a consistent interpolation, we allow the radius  $r_v$  of the neighborhood to vary with the position of the field point in such a way that the volume of the averaging region remains constant. This will assure that the gradient of  $X_V^1$  vanish at  $V$  and allow a consistent interpolation up to what will be the pole of our coordinates. Now, suppose that the field point of interest lies within a particular triangle neighboring  $V$ , the “home” cell (shown in green in Fig. 7) and is infinitesimally near  $V$ . The neighborhood of integration is then nearly the polygon of Fig. 10, formed by the  $N_v$  triangles which consist of points within a distance  $r_v$  of  $V$ , but the boundary of  $H$  is shifted slightly from this and the average is changed as a result.

For clarity, we will omit the details of calculating these shifts for all possible cases, as these details depend on the topology of the  $V$ -region (namely on  $N_v$ ), but roughly speaking, the shifts are rigid normal to the boundary of integration and are given for the home cell and its two direct (edge) neighbors by rigidly shifting the boundary normally by the virtual displacement. For other triangles not edge neighbors of the home cell, the situation can become more complicated, because the distance associated with a path through several neighboring cells does not correspond to a rigid shift of the boundary. In these cells, the normal shift of the boundary will be either zero or the negative of the virtual displacement.

To complete our calculation of the gradient at  $V$ , we introduce pseudo-polar coordinates in the logical space near  $V$ . For this, we introduce an angle-like coordinate in the home cell as shown in Fig. 8, where  $\xi$  varies from 0 to 1 for points adjacent to edge #2 (pointing toward vertex #1) to edge #1, with  $\xi$  constant

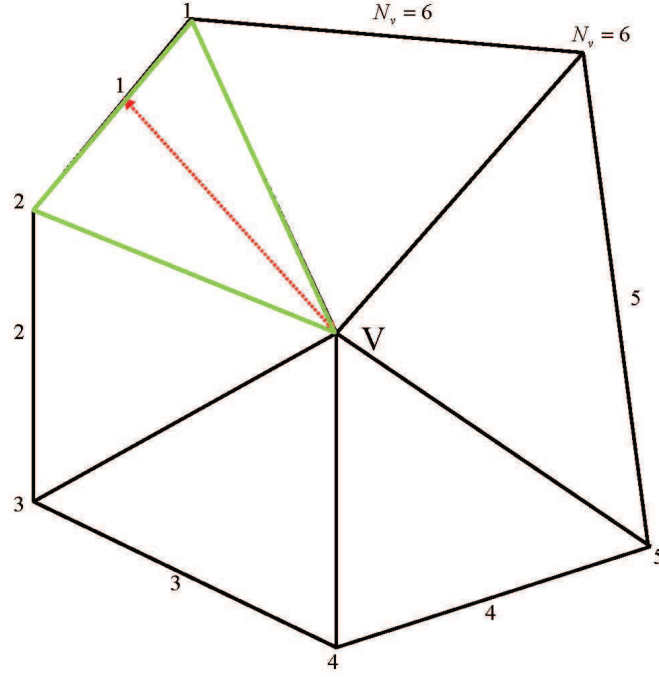


Figure 7: Vertex region

Edge	Displacement	Modified Displacement
1	1	$1/2$
2	$\xi$	$\xi - 1/2$
3	0	$-1/2$
4	$1 - \xi$	$1/2 - \xi$

Table 1: Edge displacements for  $N_v = 4$

along a ray from V. The remaining radial-like coordinate is  $\rho = 1 - \xi_0$ .

If we label the edge of  $H$  in the home cell as 1 and continue lexically around the boundary in order up to  $N_v$ , we can work out the virtual displacement of all edges in terms of a displacement along the ray  $\xi$  by a change of  $\rho = \delta$ . For example, the result for  $N_v = 4$  “normalized” to the radial displacement  $\delta$  are summarized in Table 1.

The condition that the area of  $H = 1/3$  gives the radius at V

$$r_v = \sqrt{\frac{2}{3N_v}} \quad (18)$$

and the value of the average given by Eq. (17) is modified to

$$\langle Q \rangle_V = \left(1 - \frac{2r_v}{3}\right) Q_V + \frac{2r_v}{3N_v} \sum_{i=1}^{N_v} Q_i \quad (19)$$

so that the value of the vertex shape functions at V are  $1 - 2r_v/3$  for vertex V and  $2r_v/3N_v$  for each of the neighbors of V.

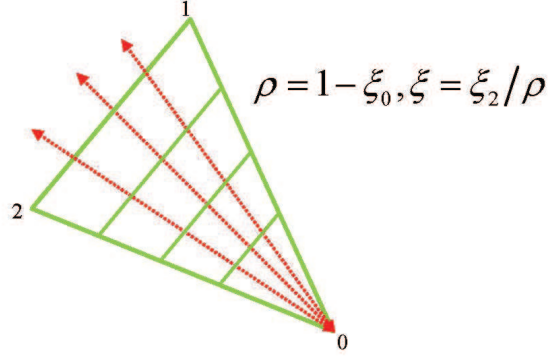


Figure 8: Pseudo-polar coordinates for home cell. Green lines are constant  $\rho$ , while red arrows are constant  $\xi$ .

We can now complete the calculation of the gradient of the  $C_1$  functions which enter the V-region calculation. Those of the function associated with V vanishes by construction. The remaining vertices labeled  $1, 2, \dots, N_v$  beginning with 1 and 2 of the home element and proceeding counter-clockwise around V are given by Tables 2 – 6. Because the virtual displacement is along a ray away from V, these derivatives correspond to  $\partial_\rho|_\xi = -\partial_{\xi_0}|_\xi = -d_0 + (1 - \xi)d_1 + \xi d_2$ , where we have introduced the notation  $d_i$  to indicate the formal partial derivative with respect to  $\xi_i$ . Since these are not independent of one another (being required to sum to unity) and since expressions can be transformed (using this property), we use this unconventional notation to remind us of these constraints.

$i$	$N_v \partial_\rho X_i^1$
0	0
1	$2/3 - \xi$
2	$\xi - 1/3$
3	$-1/3$

Table 2:  $\rho$  derivatives of basis functions for  $N_v = 3$

The remainder of the calculation is straightforward. One takes the expression for the shape functions in the U region and forms these and their normal ( $\rho$ ) derivative into a polynomial in  $\xi$  along the boundary  $\rho = 1/3$  of the V-region's home cell. The derivative at V is given by  $A_0 + A_1 \xi$ . The value of the interpolate is known at V from Eq. (19). This allows the construction of the interpolation function according to

$i$	$N_v \partial_\rho X_i^1$
0	0
1	$1 - \xi$
2	$\xi$
3	$\xi - 1$
4	$-\xi$

Table 3:  $\rho$  derivatives of basis functions for  $N_v = 4$

$$\begin{aligned}
 X(\rho, \xi) = & 27\rho^2(1 - 2\rho)X(2/3, \xi) + \rho^2(3 - 9\rho)\partial_\rho X(2/3, \xi) \\
 & + 9(2/3 - \rho)^2(1 + 6\rho)X_V - \rho(1 - 3\rho)^2(A_0 + A_1\xi)
 \end{aligned} \tag{20}$$

We have written  $X_V$  for the  $\xi$ -independent value of the function at V.

For convenience, we express all these resulting functions in terms of the field point logical coordinates  $(\xi_1, \xi_2)$ , and a remarkable result is that all denominators involving  $\rho = 1 - \xi_0 = \xi_1 + \xi_2$  (which occur up to the third power!) cancel, leaving finally simple polynomials as the shape functions in the V-region. It is convenient to separate the resulting functions into an  $N_v$ -independent part and an  $N_v$ -dependent part. The former is associated with data at  $\rho = 1/3$  and is

$i$	$N_v \partial_\rho X_i^1$
0	0
1	$8/5 - \xi$
2	$\xi + 3/5$
3	$2\xi - 7/5$
4	$-7/5$
5	$-2\xi + 3/5$

Table 4:  $\rho$  derivatives of basis functions for  $N_v = 5$

$$\begin{aligned}
X_V^{NNV} &= 1 - 8\xi_1^2 - 8\xi_2^2 - 14\xi_1\xi_2 + 13\xi_1^3 + 13\xi_2^3 + \frac{75}{2}\xi_1^2\xi_2 + \frac{75}{2}\xi_1\xi_2^2 \\
&\quad - \frac{2r_v}{3}(1 + 6\xi_1 + 6\xi_2)(1 - 3\xi_1 - 3\xi_2)^2 \\
X_1^{NNV} &= 4\xi_1^2 + 2\xi_2^2 + 6\xi_1\xi_2 - 5\xi_1^3 - 4\xi_2^3 - \frac{27}{2}\xi_1^2\xi_2 - \frac{27}{2}\xi_1\xi_2^2 \\
X_2^{NNV} &= 2\xi_1^2 + 4\xi_2^2 + 6\xi_1\xi_2 - 4\xi_1^3 - 5\xi_2^3 - \frac{27}{2}\xi_1^2\xi_2 - \frac{27}{2}\xi_1\xi_2^2 \\
X_3^{NNV} &= \xi_2 \left( \xi_1 + 2\xi_2 - 3\xi_1^2 - 4\xi_2^2 - \frac{15}{2}\xi_1\xi_2 \right) \\
X_{N_v}^{NNV} &= \xi_1 \left( 2\xi_1 + \xi_2 - 4\xi_1^2 - 3\xi_2^2 - \frac{15}{2}\xi_1\xi_2 \right)
\end{aligned} \tag{21}$$

The case  $N_v = 3$  is special in that the last members of Eq. (21) should be summed and the result associated with the single vertex #3.

The  $N_v$ -dependent part is associated with data at  $\rho = 0$  and is

$$\begin{aligned}
X^{NV} &= [A_0\xi_1 + (A_0 + A_1)\xi_2](1 - 3\xi_1 - 3\xi_2)^2 \\
&\quad + \frac{2r_v}{3N_v}(1 - 3\xi_1 - 3\xi_2)^2(1 + 6\xi_1 + 6\xi_2)
\end{aligned} \tag{22}$$

where the coefficients are given in Tables 2 – 6 and are summarized in Table 7.

In most applications, the topology of the mesh is fixed for all time. In this case, these coefficients can be evaluated at the initial problem time and stored per vertex, allowing a speed-up of evaluation of the required elements.

$i$	$N_v \partial_\rho X_i^1$
0	0
1	$2 - \xi$
2	$1 + \xi$
3	$2\xi - 1$
4	$\xi - 2$
5	$-1 - \xi$
6	$1 - 2\xi$

Table 5:  $\rho$  derivatives of basis functions for  $N_v = 6$

$i$	$N_v \partial_\rho X_i^1$
0	0
1	$2 - \xi + 2(N_v - 6)/N_v$
2	$1 + \xi + 2(N_v - 6)/N_v$
3	$2\xi - 1 + 2(N_v - 6)/N_v$
4	$\xi - 2 + 2(N_v - 6)/N_v$
$5 - N_v - 2$	$-2 + 2(N_v - 6)/N_v$
$N_v - 1$	$-1 - \xi + 2(N_v - 6)/N_v$
$N_v$	$1 - 2\xi + 2(N_v - 6)/N_v$

Table 6:  $\rho$  derivatives of basis functions for  $N_v > 6$

$N_v$	0	1	2	3	4	$5 - N_v - 2$	$N_v - 1$	$N_v$
3	$\frac{0}{0}$	$\frac{-1/3}{1}$	$\frac{2/3}{-1}$	$\frac{-1/3}{0}$	—	—	—	—
4	$\frac{0}{0}$	$\frac{1}{-1}$	$\frac{1}{0}$	$\frac{-1}{1}$	$\frac{-1}{0}$	—	—	—
5	$\frac{0}{0}$	$\frac{8/5}{-1}$	$\frac{3/5}{1}$	$\frac{-7/5}{2}$	$\frac{-7/5}{0}$	$\frac{3/5}{-2}$	—	—
6	$\frac{0}{0}$	$\frac{2}{-1}$	$\frac{1}{1}$	$\frac{-2}{2}$	$\frac{-2}{1}$	$\frac{-1}{-1}$	$\frac{1}{-2}$	—
>6	$\frac{0}{0}$	$\frac{4-12/N_v}{-1}$	$\frac{3-12/N_v}{1}$	$\frac{1-12/N_v}{2}$	$\frac{-12/N_v}{1}$	$\frac{-12/N_v}{0}$	$\frac{1-12/N_v}{-1}$	$\frac{3-12/N_v}{-2}$

Table 7: Coefficients of  $N_v$ -dependent part of shape functions. The rows correspond to different values of  $N_v$ . The columns correspond to vertex number around V. For each case,  $A_0$  is written as the numerator and  $A_1$  as the denominator. All coefficients are divided by  $N_v$ .

#### 4. PIC application

The elements derived here are used as shape functions for charge deposition and for interpolation of the electrostatic potential in a two-dimensional (2D) PIC code. Azimuthal symmetry is assumed, so that all quantities are assumed functions of the C-space cylindrical coordinates  $r, z$ . The resulting SUSIE code (for Sandia UnStructured Ion-Electron), following previous work, uses time-implicit differencing and also incorporates a digital filter to remove mesh-scale modes. These details are described in the Appendices, while here our focus is placed on the spatial differencing only.

Particles are advanced using leap-frog time differencing with position and velocity in 3D Cartesian, using a standard transformation

$$\begin{aligned}
\mathbf{v}_p^{n+1/2} &= \mathbf{v}_p^{n-1/2} - \frac{q\Delta t}{m} \nabla \phi^n(\mathbf{x}_p^n) \\
\mathbf{x}_p^{n+1} &= \mathbf{x}_p^n + \Delta t \mathbf{v}_p^{n+1/2}
\end{aligned} \tag{23}$$

The gradient of the electrostatic potential  $\phi$  is evaluated using the  $C_1$  elements as shape functions

$$\nabla \phi(\mathbf{x}_p^n) = \nabla \sum_i \phi_i X_i^1[\xi(\mathbf{x}_p^n)] \tag{24}$$

This, in turn, requires the location of the particle in L-space and the evaluation of the gradient of the logical coordinates. Recall that the C-space position is given by

$$\mathbf{r}(\xi) = \sum_i \mathbf{r}_i X_i^1(\xi) \tag{25}$$



so that the L-space particle location is found by Newton's method inverting  $\mathbf{r}(\xi_p) = (\sqrt{x_p^2 + y_p^2}, z_p)$ . The Newton update is

$$\xi_p \leftarrow \xi_p - \left[ \mathbf{r}(\xi_p) - (\sqrt{x_p^2 + y_p^2}, z_p) \right] \bullet \nabla \xi(\xi_p) \quad (26)$$

where the gradient of the logical coordinates is evaluated from Eq. (25). This step requires a special treatment because there are three L-space coordinates but only two C-space coordinates. The resolution is to note that the three L-space coordinates are constrained to sum to unity.

The formal derivatives of the shape functions are directly computed from the expressions of the previous section. Then, we have the system of differential forms

$$\begin{aligned} dr &= \sum_{j=1}^3 \left( \sum_i r_i d_j X_i^1 \right) d\xi_j \\ dz &= \sum_{j=1}^3 \left( \sum_i z_i d_j X_i^1 \right) d\xi_j \\ 0 &= \sum_{j=1}^3 d\xi_j \end{aligned} \quad (27)$$

which can be inverted to give the solution

$$d\xi_j = (\partial_r \xi_j) dr + (\partial_z \xi_j) dz \quad (28)$$

which is the desired  $\nabla \xi$  in component form.

The gradient of the shape functions then follows directly from the chain rule.

Charge deposition to the vertices is done with the same shape functions

$$\rho_i = \frac{1}{V_i} \sum_p X_i^1(\mathbf{x}_p) \quad (29)$$

where the vertex volumes  $\{V_i\}$  are obtained by the method of Verboncouer[15] using the shape functions.

The method outlined here is energy conserving in the limit of  $\Delta t \rightarrow 0$  because of the choice of the same shape functions for interpolation of the charge and electrostatic potential.

It is useful, before considering the PIC application, to visualize the shape functions. One of them is shown in Fig. 9 for a test mesh of a few triangles. The methods of this section were used to tabulate values of this shape function on a uniform Cartesian mesh, from which the contour plot was generated. Horizontal line-outs, shown in Fig. 10, are similar to the quadratic B-splines used on regular, rectangular meshes.

SUSIE has been applied to the spherical expansion of a cool plasma (Debye length  $\ll$  system size) sourced at an inner radius and allowed to freely expand outward. The quality of the solution is monitored by several figures of merit:

- Spherical symmetry should be maintained
- Quasi-neutrality should hold over the entire plasma domain
- Ions, which are injected cold (and supersonically) should remain a cold beam

The solution is also compared with an analytic two-temperature calculation which neglects collisions entirely.

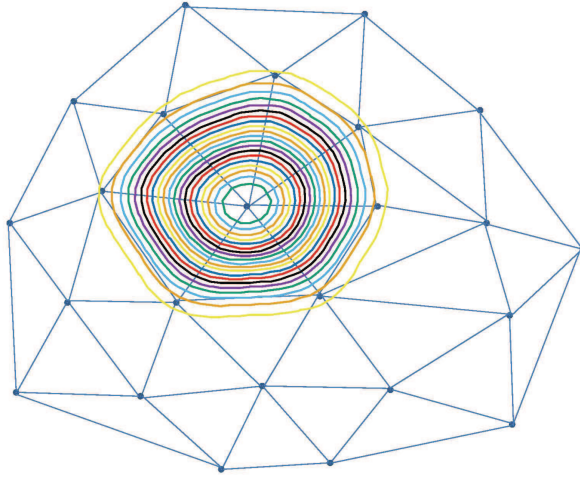


Figure 9: Test mesh with contours of single shape function.

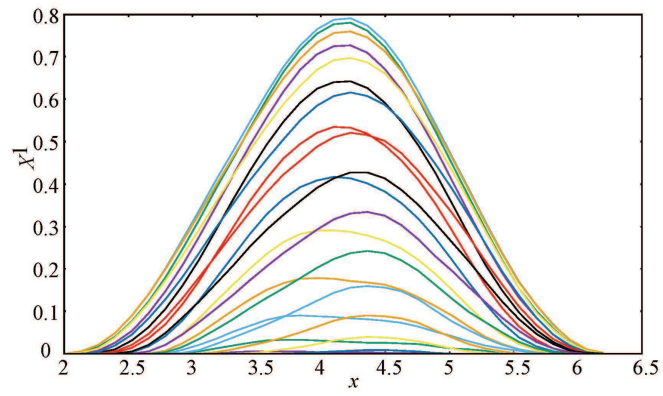


Figure 10: Line-outs of shape function of Fig. 9.

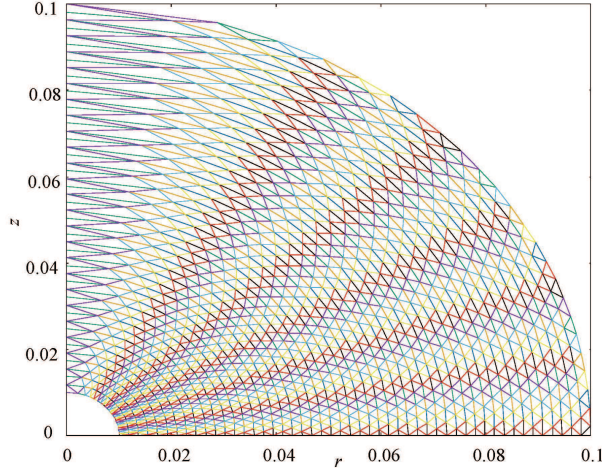


Figure 11: Triangle mesh used for spherical expansion.

The mesh used is shown in Fig. 11. The inner radius is 1 cm (0.01 m). The outer radius is 10 cm (0.1 m). The triangle mesh is constructed with uniform divisions of the spherical radius (50) and uniform divisions of the cosine of the polar angle (25). This tessellation results in approximately equal number of particles per cell for this expansion problem. Symmetry is imposed at both the vertical axis ( $r = 0$ ) and the horizontal axis ( $z = 0$ ). Ghost triangles are constructed by reflection about the four boundaries, so that the shape functions can be applied without modification near the boundaries.

A spherical expansion was simulated by uniformly injecting cold ions at the ion acoustic speed ( $10^4$  m/s) at such a rate as to maintain a density of  $2 \times 10^{17} \text{ m}^{-3}$ . Electrons were uniformly injected at the inner boundary from a one-sided Maxwellian with  $T_e = 1 \text{ eV}$  to maintain overall neutrality. A potential difference of 1.75 V, with the inner boundary as the anode, was determined to be that required to avoid a sheath at the outer boundary when the steady state was reached. The time step was allowed to change slowly (so that the leap-frog centering was not materially disturbed) until the streaming parameter reached a specified value of 0.3. The streaming parameter was defined as the average number of cells moved by an electron in a time step, with this number measured as the number of cells traversed when locating an average electron. The calculation reached a steady state in a physical time of  $9 \mu\text{s}$ . In this steady state, the time step reached approximately  $6 \times 10^{-10} \text{ s}$ , which corresponded to  $\omega_{pe}\Delta t = 15.$ , while the Debye length at the inner boundary corresponded to  $\Delta/\lambda_D \approx 100$ .

The quality of the solution is shown by an examination of Figs. 12 – 15. Figure 12 shows contours of the natural logarithm of electron number density. The total variation is approximately 150 times, as may be seen in Fig. 13, where the polar-angle-averaged electron and ion densities are compared. As can be seen, both quasi-neutrality and spherical symmetry are excellent.

The ion phase space is summarized in Fig. 14, where the kinetic energy of individual ions is plotted against their spherical radius. The ion temperature in the beam frame is very cold, of order 0.1 eV, nor is there evidence of collective modes associated with small Debye length finite grid instabilities. The ion energy vs. spherical radius compares favorably with the model of App. B. There is an offset of 0.5 eV added to the energy of Fig. 14 which is the injection kinetic ion energy.

As a direct comparison, the same calculation was carried-out with the conventional, piece-wise linear  $C_0$  shape functions. In order to run the calculation beyond a few time steps, it was necessary to reduce the time step (and streaming limit) by a factor of 6. Even then, strong aliasing instabilities appeared. Reducing

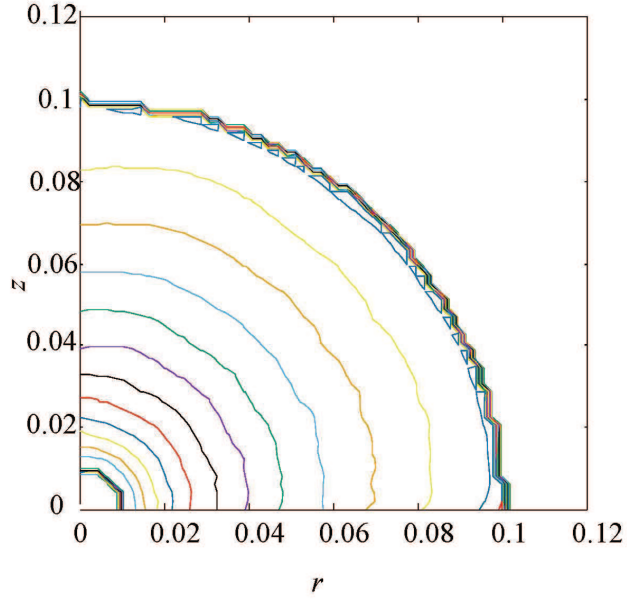


Figure 12: Contours of logarithm of electron number density. Colors are assigned arbitrarily, with each contour representing a density ratio of 1.54. Absolute density values are better displayed by Fig. 13.

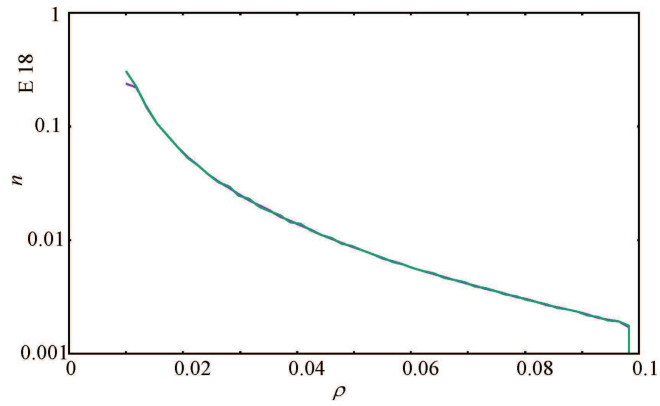


Figure 13: Polar-angle-averaged (logarithmic) electron and ion density vs. spherical radius.

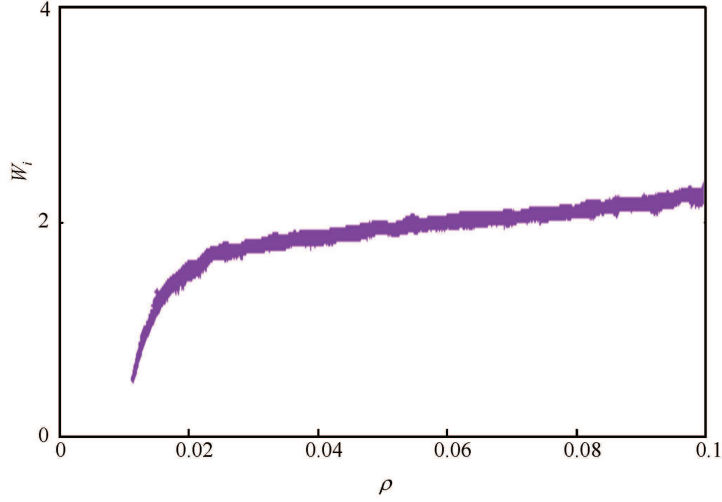


Figure 14: Ion phase space for spherical expansion problem.

the time step by an overall factor of 10 was required to reduce these to a manageable level. The resulting ion phase space is shown in Fig. 15, which may be compared directly with Fig. 14 with the conclusion that beam heating is still considerably increased for this comparison case. The reduction of time step resulted in a significant increase in run time. Using an OpenMP implementation with (up to) 32 hyperthreads on a modest, dedicated Xeon machine gave a wall clock run time of 91.1 hrs for the  $C_0$  calculation, compared with 15.3 hrs for the  $C_1$  calculation. While additional optimization of the  $C_0$  code is likely possible (*e.g.* some nonlinear location is not required) it is clear that use of the new  $C_1$  shape functions is computationally advantageous for quasi-neutral problems of the sort considered here.

## 5. Summary and conclusion

A new set of continuously-differentiable ( $C_1$ ) particle shape functions for applications to unstructured, triangle mesh PIC have been developed. By the method of averaging the usual  $C_0$  shape functions over a neighborhood, with the neighborhood defined as a “circle” in logical space with an appropriate metric, explicit cubic forms in the usual logical coordinates are obtained. This method is not so useful for field points near a vertex, so a different interpolation strategy is adopted there, yielding again a cubic expression.

It is good to keep in mind that the approach of this work does not provide a *unique* solution to the problem of obtaining  $C_1$  shape functions useful for PIC and other possible applications on an unstructured simplex mesh. A particular strong point of the present approach is the introduction of a metric and area (volume) on the (logical) L-space. Working in L-space seems a natural solution to the issue of widely varying cell sizes which is common in applications, as it occurs in our spherical expansion problem because of spherical symmetry. Approaches which average in (configuration) C-space[12][11] probably need to vary the particles physical size to match the mesh in some sense. The conforming of the shape functions to the mesh geometry (Fig. 9) would be surprising if not engineered by this approach. On the other hand, there is nothing unique about choosing the  $L_\infty$  distance of Eq. (6) to define the averaging circle. In initial work toward 3-D generalizations, for example, it seems that the  $L_2$  distance may be more natural. This could be done in 2-D as well, giving a circle instead of the hexagon (in the proper rendering) in L-space for the

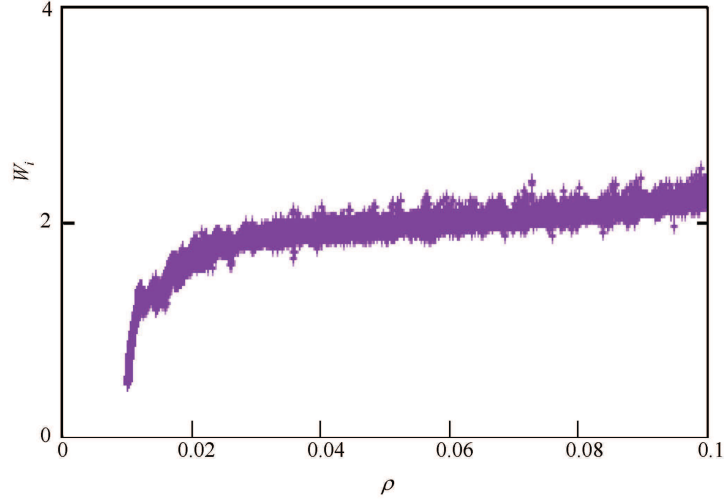


Figure 15: Ion phase space using  $C_0$  shape functions.

averaging. The present functions may not yet be optimal, but appear to offer large advantages for PIC on unstructured triangle meshes.

These shape functions may have general applicability to problems where a  $C_1$  approximation is required, but are shown here specifically to be extremely useful for PIC simulation of cold plasmas. Usually troublesome aliasing instabilities are avoided by use of these shape functions, combined with an energy-conserving (in limit of  $\Delta t \rightarrow 0$ ) differencing, time-implicit differencing and digital filtering to remove mesh-scale modes from the calculations. A case which is robustly unstable using usual  $C_0$  shape functions is shown to be stable for  $\Delta/\lambda_D \approx 100$  when the new  $C_1$  shape functions are applied.

Extension of these techniques to three-dimensions appears possible but introduces new difficulties associated with the richer mesh topology possible there. These extensions are beyond the present scope and will be considered in subsequent publications.

### Acknowledgment:

This work was supported by Sandia National Laboratory under purchase order #1581115. The author gratefully acknowledges the constant encouragement and many useful conversations with Tom Hughes of Sandia National Laboratory and many important suggestions from the Referees. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

### References:

- [1] C. Birdsall and A. Langdon, *Plasma Physics via Computer Simulation*. New York: McGraw-Hill, 1985.
- [2] R. Hockney and J. Eastwood, *Computer Simulation Using Particles*. Bristol, UK: Taylor & Francis, Inc, 1988.

- [3] M. Matsumoto and S. Kawata, “Tripic: Triangular-mesh particle-in-cell code,” *J. Comput. Phys.*, vol. 87, pp. 488–493, 1990.
- [4] D.-Y. Na, Y. A. Omelchenko, H. Moon, B.-H. V. Borges, and F. L. Teixeira, “Axisymmetric charge-conservative electromagnetic particle simulation algorithm on unstructured grids: Application to microwave vacuum electronic devices,” *J. Comput. Phys.*, vol. 346, pp. 295–317, 2017.
- [5] H. Moon, F. L. Teixeira, and Y. A. Omelchenko, “Exact charge-conserving scatter-gather algorithm for particle-in-cell simulations on unstructured grids: A geometric perspective,” *Computer Physics Communications*, vol. 194, pp. 43–53, 2015.
- [6] D. C. Barnes and L. Chacón, “Finite spatial-grid effects in the fully implicit, energy and charge conserving, electrostatic particle-in-cell algorithm,” *J. Comput. Phys.*, vol. To Appear, 2018.
- [7] S. Jardin, “A triangular finite element with first-derivative continuity applied to fusion MHD applications,” *J. Comput. Phys.*, vol. 200, pp. 133–152, 2004.
- [8] J. Argyris, I. Fried, and D. Scharpf, “The tuba family of plate elements for the matrix displacement method,” *Aero. J. Roy. Aero. Soc.*, vol. 72, pp. 701–709, 1968.
- [9] D. Han, P. Wang, X. He, T. Lin, and J. Wang, “A 3d immersed finite element method with non-homogeneous interface flux jump for applications in particle-in-cell simulations of plasma-lunar surface interactions,” *J. Comp. Phys.*, pp. 965–980, 2016.
- [10] N. A. Gatsonis and A. Spirkin, “A three-dimensional electrostatic particle-in-cell methodology on unstructured delaunay-voronoi grids,” *J. Comp. Phys.*, pp. 3742–3761, 2009.
- [11] G. Jacobs and J. Hesthaven, “High-order nodal discontinuous galerkin particle-in-cell method on unstructured grids,” *J. Comp. Phys.*, pp. 96–121, 2006.
- [12] E. M. Wolf, M. Causley, A. Christlie, and M. Bettencourt, “A particle-in-cell method for the simulation of plasmas based on an unconditionally stable field solver,” *J. Comp. Phys.*, pp. 342–372, 2016.
- [13] R. M. Kirby and G. E. Karniadakis, *Encyclopedia of Computational Mechanics*, ch. Volume 3: Fluids, pp. 61–90. John Wiley & Sons, Ltd., 2004.
- [14] Wolfram Research Inc., *Mathematica*. Champaign, Illinois: Wolfram Research Inc., version 11.1 ed., 2017.
- [15] J. P. Verboncoeur, “Symmetric spline weighting for charge and current density in particle simulation,” *J. Comput. Phys.*, vol. 174, no. 1, pp. 421 – 7, 2001.
- [16] J. Denavit, “Time-filtering particle simulations with  $\omega_{pe}\Delta t \gg 1$ ,” *J. Comput. Phys.*, vol. 42, no. 2, pp. 337 – 66, 1981.
- [17] R. J. Mason, “Implicit moment particle simulation of plasmas,” *J. Comput. Phys.*, vol. 41, no. 2, pp. 233 – 44, 1981.
- [18] J. Brackbill and D. Forslund, “An implicit method for electromagnetic plasma simulation in two dimensions,” *Journal of Computational Physics*, vol. 46, p. 271, 1982.
- [19] B. I. Cohen, A. B. Langdon, and A. Friedman, “Implicit time integration for plasma simulation,” *J. Comput. Phys.*, vol. 46, no. 1, pp. 15 – 38, 1982.

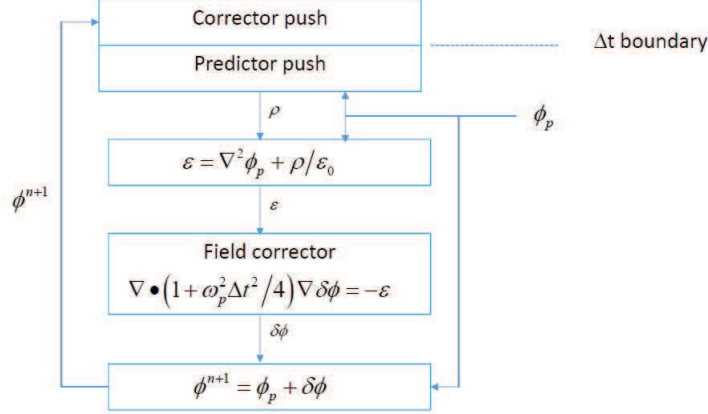


Figure A.1: Time-step loop for SUSIE.

## Appendices

### A. Implicit time differencing and spatial filter

Because analysis shows that absence of aliasing instabilities results from any scheme which is energy-conserving in the limit of zero time step, we adopt here the semi-implicit method of “classical” implicit PIC developed first shortly after 1980.[16][17][18][19] The flow of the time step is as shown in Fig. A1. A single corrector step is applied to the particle motion using a corrected time-advanced electric field. The correction to the time-advanced electric field, in turn, is computed using the simplest plasma response, included either by cold-fluid moment equations or equivalently by direct-implicit calculation of the response, with suitable compromises of the resulting expressions.

We first observe that the correct time centering for energy conservation is given by using the time- and space-filtered potential  $\bar{\phi} = F(\phi^{n-1} + \phi^n + \phi^{n+1})/4$ , where superscripts indicate time levels and where  $F$  is the spatial filter. For, suppose the filter is applied to the potential to obtain a smoother potential, which is then used to advance the particles. The energy-conserving particle push uses the gradient of the interpolated potential to accelerate the particles, Eq. (23) of the main text.

To make the scheme energy-conserving, it is necessary to use an implicit time-averaged potential. Suppose that the potential corresponding to the density at time  $n$  is  $\phi^n$ , so that Poisson is satisfied (in difference form). The required potential for the advance of Eq. (23) is then

$$F\phi = F \frac{\phi^{n-1} + 2\phi^n + \phi^{n+1}}{4} \quad (\text{A.1})$$

To show energy conservation (for  $\Delta t \rightarrow 0$ ), take the dot product of the velocity advance part of Eq. (23) with the time-averaged velocity, and sum over particles to obtain



$$\begin{aligned}
\Delta K^n &\equiv \frac{m}{2} \sum_p \left( \mathbf{v}_p^{n+1/2} \right)^2 - \frac{m}{2} \sum_p \left( \mathbf{v}_p^{n+1/2} \right)^2 \\
&= -\frac{q}{2} \sum_i (F\phi)_i \sum_p (\mathbf{x}_p^{n+1} - \mathbf{x}_p^{n-1}) \bullet \nabla X_i^1(\mathbf{x}_p^n) \\
&\approx -\frac{q}{2} \sum_i (F\phi)_i \sum_p [X_i^1(\mathbf{x}_p^{n+1}) - X_i^1(\mathbf{x}_p^{n-1})] \\
&= -\sum_i (F\phi)_i \frac{Q_i^{n+1} - Q_i^{n-1}}{2}
\end{aligned} \tag{A.2}$$

where  $Q$  is the PIC-deposited charge.

Now, forgetting for the moment the filter, notice that the time-centering gives

$$\begin{aligned}
\frac{Q_i^{n+1} - Q_i^{n-1}}{2} &= Q^{n+1/2} - Q^{n-1/2} \\
&= -\epsilon_0 V L \phi^{n+1/2} + \epsilon_0 V L \phi^{n-1/2} \\
\phi &= \frac{\phi^{n+1/2} + \phi^{n-1/2}}{2}
\end{aligned} \tag{A.3}$$

where the intermediate time levels are defined as simple averages of the integer time levels, where  $L$  is the discrete Laplacian, and  $V$  is the node volume.

Because the discrete Laplacian is adjoint with respect to the vertex volume sum (discrete representation of the volume integral), combining the above gives the energy conservation

$$\Delta K^n = -\frac{\epsilon_0}{2} \sum_i V_i \left| \mathbf{G} \phi^{n+1/2} \right|^2 + \frac{\epsilon_0}{2} \sum_i V_i \left| \mathbf{G} \phi^{n-1/2} \right|^2 \tag{A.4}$$

where  $\mathbf{G}$  is the discrete gradient operator. Hence, time  $n + 1/2$  energy is conserved for small  $\Delta t$ .

To deal with inaccuracies introduced by non-time-centered boundary conditions, *etc.*, it is desirable to add a small and controlled amount of dissipation by changing the time-centering to a slightly time-advanced one. We introduce the additional parameter  $\theta$ , and use the forward-biased form

$$F\phi = \frac{(1 - \theta) \phi^{n-1} + 2\phi^n + (1 + \theta) \phi^{n+1}}{4} \tag{A.5}$$

A value of  $\theta$  between zero and 0.1 is sufficient for all problems considered to date. Notice that this corresponds to a forward bias of less than  $0.025\Delta t$ , with negligible effect on energy conservation.

The required relations hold with the filter if the discrete Poisson equation is replaced by one using a filtered charge, in which the filter is the adjoint of that used in Eq. (23). Thus, the discrete Poisson equation should be written as

$$-\epsilon_0 V L \phi = F^\dagger Q \tag{A.6}$$

One is encouraged to use a filter which is self-adjoint, so that the same filter is applied to the potential as is applied to the mesh charge. This is not always possible. For example, in the geometry of SUSIE, the symmetry boundaries break the self-adjoint property which otherwise would naturally be produced by the edge-based average. Hence, it is important to use the adjoint of the potential filter for the charge.

A straightforward way to achieve the filtering on a triangular mesh is to use an edge-based approach. In this case, the value of the filtered vertex field is obtained by adding and subtracting a multiple of the edge difference of the un-filtered field. The corresponding equation is

$$(FQ)_i = Q_i - \lambda \sum_{e \in i} (Q_i - Q_{i'})_e \quad (\text{A.7})$$

The band pass of the filter depends on the gain  $\lambda$ . For a triangular mesh, the choice  $\lambda = 1/12$  gives the equivalent (on average) of the two-dimensional binomial filter, which replaces each vertex quantity with 1/2 the un-filtered value plus 1/2 the average of un-filtered values of nearest neighbors. This filter will completely eliminate the “2D Nyquist” mode, which is one in which alternate vertices have alternating values (this is the “red-black” mode for rectangular meshes). This choice is not optimal for all applications, as often the flow is mostly one-dimensional and the most troublesome modes are those which alternate in this one direction, with no structure in the orthogonal direction.

The solution adopted for SUSIE is to increase the gain to over damp the 2D Nyquist mode and then apply the filter twice (*i.e.* twice to  $Q$  and twice to  $\phi$ ). The SUSIE implementation uses  $\lambda = 1/8$  so that the 2D mode is replaced by -1/2 its amplitude each filter pass, giving a total response suppression of 1/16. The 1D Nyquist mode has  $k^2$  smaller by 1/2 and is thus replaced by 1/2 its amplitude per filter pass, so is also suppressed by a total factor of 1/16. This choice seems to give good results for the spherical expansion problem and has not been troublesome for other applications.

## B. Spherical expansion analytic solution

Consider a spherical expansion of a cold ion beam with the electrons having two temperatures,  $T_r$  and  $T_\perp$ . The electron pressure tensor is

$$\overleftrightarrow{P}_e = nk_B \left[ T_r \hat{\mathbf{r}}\hat{\mathbf{r}} + T_\perp \left( \overleftrightarrow{1} - \hat{\mathbf{r}}\hat{\mathbf{r}} \right) \right] \quad (\text{B.1})$$

This produces the force density

$$-\nabla \bullet \overleftrightarrow{P}_e = - \left( \frac{1}{r^2} \partial_r r^2 nk_B T_r - \frac{2nk_B T_\perp}{r} \right) \hat{\mathbf{r}} \quad (\text{B.2})$$

Neglecting electron mass, this gives for the electrostatic potential  $\phi$

$$e \partial_r \phi = \frac{1}{r^2} \partial_r r^2 nk_B T_r - \frac{2nk_B T_\perp}{r} \quad (\text{B.3})$$

As an example, consider the case where electrons are completely collisionless and are injected at a source radius  $r_0$  with a isotropic temperature  $T_0$ . In this case, the radial temperature remains constant with expansion (because of conservation of energy) while the perpendicular temperature decreases as  $1/r^2$  (because of conservation of total angular momentum). In this case, Eq. (B.3) becomes

$$e \partial_r \phi = k_B T_0 \left( \partial_r \log r^2 n - \frac{2r_0^2}{r^3} \right) \quad (\text{B.4})$$

which can be integrated to

$$\frac{e\phi}{k_B T_0} = \frac{r_0^2}{r^2} - 1 + \log \frac{r^2 n}{r_0^2 n_0} \quad (\text{B.5})$$

assuming without loss of generality that  $\phi(r_0) = 0$ .

The ions, by assumption, flow outward with all having the same radial velocity, given by

$$u = \sqrt{u_0^2 - 2e\phi/M} \quad (\text{B.6})$$

and conservation of particles implies that  $nur^2$  is a constant.

Combining these, we find

$$\xi = \log \sqrt{1 + \frac{2\xi}{M_\#^2}} + 1 - \frac{r_0^2}{r^2} \quad (\text{B.7})$$

where the normalized potential  $\xi = -e\phi/k_B T_0$ , and  $M_\# = u_0/\sqrt{k_B T_0/M}$  is the ion Mach number at the injection radius.

One can solve Eq. (B.7) for the normalized potential at large radius for any  $M_\#$ , but physical solutions only exist for  $M_\# \geq 1$ , just as in the single temperature, isothermal case. Some solutions are shown in Figs. 1 and 2. Figure 1 shows the applied potential for large expansion as a function of the injection Mach number.

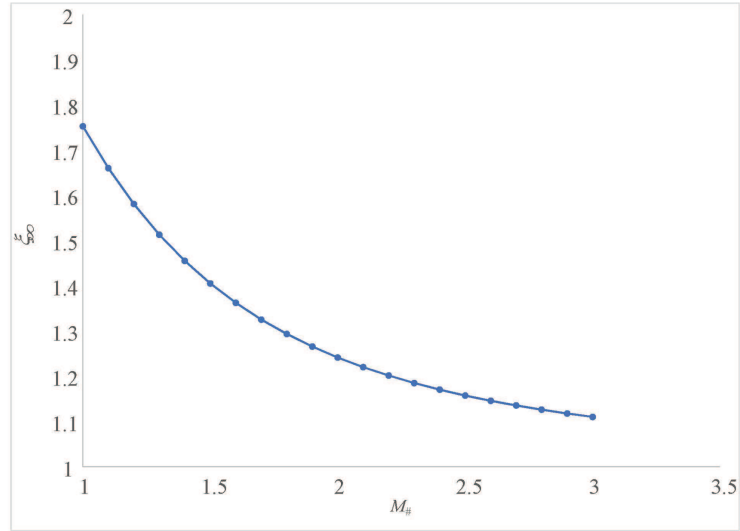


Figure B.1: Normalized total potential vs. injection Mach number.

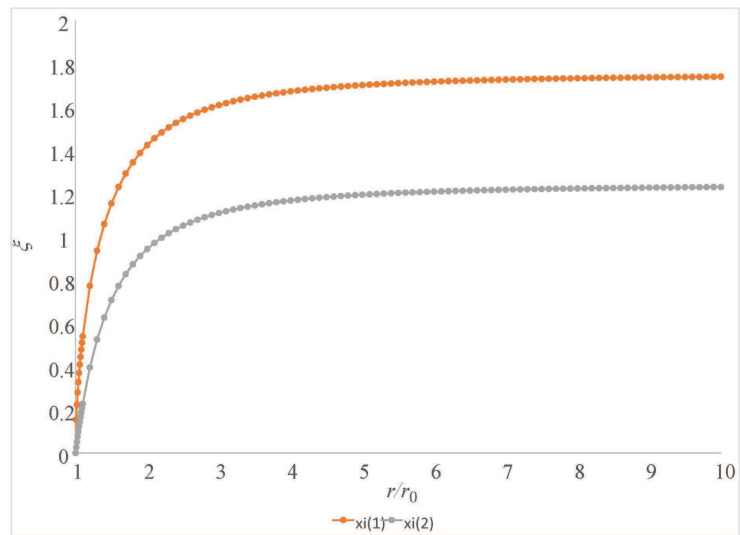


Figure B.2: Normalized potential vs. radius for various Mach numbers (1.0,2.0)