

Manuscript version: Author's Accepted Manuscript

The version presented in WRAP is the author's accepted manuscript and may differ from the published version or Version of Record.

Persistent WRAP URL:

<http://wrap.warwick.ac.uk/161356>

How to cite:

Please refer to published version for the most recent bibliographic citation information. If a published version is known of, the repository item page linked to above, will contain details on accessing it.

Copyright and reuse:

The Warwick Research Archive Portal (WRAP) makes this work by researchers of the University of Warwick available open access under the following conditions.

© 2022 Elsevier. Licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International <http://creativecommons.org/licenses/by-nc-nd/4.0/>.



Publisher's statement:

Please refer to the repository item page, publisher's statement section, for further information.

For more information, please contact the WRAP Team at: wrap@warwick.ac.uk.

Integration over discrete closed surfaces using the Method of Fundamental Solutions

Duncan A. Lockerby

^aSchool of Engineering, University of Warwick, Coventry, CV4 7AL, UK

Abstract

The Method of Fundamental Solutions (MFS) is an established technique for solving linear partial differential equations. In this paper it is used for a new purpose: the approximation of integrals over closed surfaces from a finite set of known points and values. The MFS is used to fit an implicit surface through the surface points, where the implicit equation is chosen such that a surface integral is provided by summing the weights of the fit. From the divergence theorem, these surface integrals can be related to specific integrals over the enclosed volume. As a demonstration, we calculate the surface area, volume, centroid and radius of gyration, for three solid geometries: a sphere, a torus, and an ellipsoid. Very quick convergence to analytical results is shown. Local surface properties, such as the components of curvature, can also be obtained accurately. The drawbacks and advantages of the method are discussed, and the potential to calculate properties of constant-density rigid bodies (e.g. the moment of inertia tensor) and averages of incompressible flow fields (e.g. average flow velocity and strain rate) is highlighted.

Keywords: Method of Fundamental Solutions, MFS, numerical integration, surface integration

1. Introduction

The Method of Fundamental Solutions (MFS) is a simple, efficient, and accurate approach to solving linear partial differential equations. The popularity of the MFS has grown over the past few years, and the interested reader is referred to a recent and comprehensive review given by Cheng and Hong [1]. In simple terms, the method aims to satisfy a set of boundary conditions on a domain by linear superposition of a set of known fundamental solutions. If the boundary conditions are on a closed surface and the domain external to it (e.g. when calculating the evaporation from a drop [2], or the drag on a particle [3, 4, 5]), the origins of the fundamental solutions are located inside the enclosed volume; see Fig. 1.

In this paper, instead of seeking a solution to a particular physics problem outside of a closed surface, we use the MFS as a means to integrate a field (known only at points) over the closed surface itself. These surface integrals can be related to volume integrals, allowing accurate approximation of, for example, the enclosed volume, its centroid and the gyration tensor. Tankelevich *et al.* [6, 7] have employed the MFS for a related purpose: to construct implicit surfaces

Email address: d.lockerby@warwick.ac.uk (Duncan A. Lockerby)

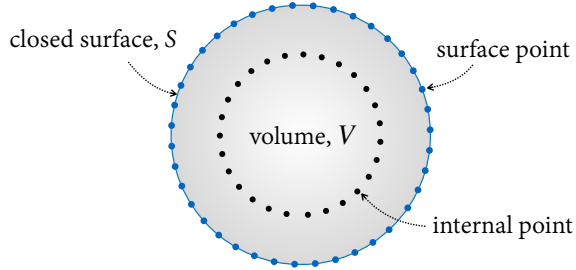


Figure 1: Arrangement of MFS points for solving boundary-value problems external to a closed surface; internal points are the centres/origins of fundamental solutions and their singularities; surface points are where boundary conditions are satisfied.

through surface points. However, the novelty of the current work lies in defining the implicit surface so that surface integrals can be obtained easily and accurately.

The standard approach to calculating integrals over discrete surfaces (i.e. a weighted sum of areas of a surface mesh) converges slowly with increasing resolution. High-order integration is much more involved; for example, combining “a stabilized least squares approximation, a blending procedure based on linear shape functions, and high-degree quadrature rules” [8]. The method proposed here, on the other hand, is both accurate and simple; a short MATLAB script is provided in Appendix A to illustrate the point.

This paper is structured as follows. In Sec. 2 the MFS methodology is introduced, in Sec. 3 the method is tested and benchmarked, and in Sec. 4 it is applied to the surface-area estimation of an arbitrarily-shaped particle. In Sec. 5 a brief summary and discussion is presented.

2. Methodology

Our starting point is a known set of points on the closed surface, $\mathbf{r}_{\text{sur}}^i$, where i denotes the i^{th} ‘surface point’ of N ; see Fig. 1. For now, we will also assume that for each surface point a corresponding outward-facing normal is also known, \mathbf{n}^i (if not, we discuss how surface normals can be calculated in Sections 2.4 and 3.2)

The objective of this work is to construct an implicit surface through these surface points, from which we can evaluate surface integrals. To do this we require another set of points, but *inside* the enclosed volume, $\mathbf{r}_{\text{int}}^j$; where j denotes the j^{th} ‘internal point’ of M (also see Fig. 1). In the MFS literature these internal points are often referred to as ‘source nodes’ and considerable work has been done on deciding how they should be optimally located [9, 10]; however, for the problems considered here, a simple approach is adopted, which is described later.

2.1. Introduction to the MFS: a physical example

For the purposes of exposition, we will suppose we are tackling a specific problem with MFS: finding the temperature and heat-flux distribution around some solid object in free space. For this we will use fundamental solutions to Laplace’s equation:

$$\nabla^2 T = 0, \quad (1)$$

where T is temperature. Specifically, the fundamental solution we will adopt is a solution to

$$\nabla^2 T_f = -g \delta(\mathbf{r}), \quad (2)$$

where δ is the Dirac delta function, g is the ‘weight’ of the fundamental solution (located at the origin), and \mathbf{r} is the position vector. The well-known fundamental solution is simply

$$T_f(\mathbf{r}) = \frac{g}{4\pi\|\mathbf{r}\|}, \quad (3)$$

where $\|\cdot\|$ denotes the Euclidean norm.

In the MFS, the full temperature field is constructed using a linear superposition of these fundamental solutions, each centred at its own internal point and each with its own weight:

$$T(\mathbf{r}) = \frac{1}{4\pi} \sum_{j=1}^M \frac{g^j}{\|\mathbf{r} - \mathbf{r}_{\text{int}}^j\|}, \quad (4)$$

which is a solution to

$$\nabla^2 T = - \sum_{j=1}^M g^j \delta(\mathbf{r} - \mathbf{r}_{\text{int}}^j), \quad (5)$$

and satisfies the far-field condition $T \rightarrow 0$ as $\|\mathbf{r}\| \rightarrow \infty$. Note, Eq. (4) is a solution to Laplace’s equation at every position other than internal points. The heat flux can also be expressed explicitly by taking the gradient of Eq. (4):

$$\mathbf{q}(\mathbf{r}) = -\kappa \nabla T = \frac{\kappa}{4\pi} \sum_{j=1}^M \frac{g^j (\mathbf{r} - \mathbf{r}_{\text{int}}^j)}{\|\mathbf{r} - \mathbf{r}_{\text{int}}^j\|^3}. \quad (6)$$

If we evaluate Eq. (4) and (6) at each of the surface points ($\mathbf{r}_{\text{sur}}^i$), and take the surface-normal component of the heat-flux, we get

$$T^i = \frac{1}{4\pi} \sum_{j=1}^M \frac{g^j}{|\mathbf{r}_{\text{sur}}^i - \mathbf{r}_{\text{int}}^j|}, \quad (7a)$$

$$q_n^i = \frac{\kappa}{4\pi} \sum_{j=1}^M \frac{g^j (\mathbf{r}_{\text{sur}}^i - \mathbf{r}_{\text{int}}^j) \cdot \mathbf{n}^i}{\|\mathbf{r}_{\text{sur}}^i - \mathbf{r}_{\text{int}}^j\|^3}. \quad (7b)$$

where $T^i = T(\mathbf{r}_{\text{sur}}^i)$ and $q_n^i = \mathbf{q}(\mathbf{r}_{\text{sur}}^i) \cdot \mathbf{n}^i$. These can be rewritten in matrix form:

$$\mathbf{T} = \mathbf{A} \cdot \mathbf{g}, \quad (8a)$$

$$\mathbf{q}_n = \mathbf{B} \cdot \mathbf{g}, \quad (8b)$$

where $\mathbf{T} = T^i$, $\mathbf{q}_n = q_n^i$, $\mathbf{g} = g^j$, $\mathbf{A} = 1/(4\pi\|\mathbf{r}_{\text{sur}}^i - \mathbf{r}_{\text{int}}^j\|)$ and $\mathbf{B} = \kappa(\mathbf{r}_{\text{sur}}^i - \mathbf{r}_{\text{int}}^j) \cdot \mathbf{n}^i / (4\pi\|\mathbf{r}_{\text{sur}}^i - \mathbf{r}_{\text{int}}^j\|^3)$.

The weights of the fundamental solutions can now be found from boundary conditions (on either temperature or surface-normal heat flux) by the appropriate matrix inverse, i.e.,

$$\mathbf{g} = \mathbf{A}^{-1} \cdot \mathbf{T} \quad \text{or} \quad (9a)$$

$$\mathbf{g} = \mathbf{B}^{-1} \cdot \mathbf{q}_n, \quad (9b)$$

where, if $M < N$, the inverse is the Moore-Penrose pseudoinverse. Having determined the weights, g^j , the temperature and heat-flux in the medium external to the enclosed volume can be evaluated at any point using Eq. (4) and Eq. (6).

Now that we have introduced the MFS with a physical example, we will set $\kappa = 1$ for the rest of the paper, for neatness. However, for ease of description, we will continue to refer to the vector field \mathbf{q} as the ‘flux vector’ and $\mathbf{q} \cdot \mathbf{n}$ as the ‘surface flux’.

2.2. Calculating surface integrals

Here we will use the MFS to fit a smooth closed surface and field, S and $\phi(\mathbf{r})$, through known surface points and values, \mathbf{r}^i and $\phi(\mathbf{r}^i)$, in a way that readily provides the integral of the field over the surface, $\oint_S \phi(\mathbf{r}) dS$, from the weights of the fit.

First, we rewrite Equation (5) in terms of the flux vector \mathbf{q} (with $\kappa = 1$):

$$\nabla \cdot \mathbf{q} = \sum_{j=1}^M g^j \delta(\mathbf{r} - \mathbf{r}_{\text{int}}^j), \quad (10)$$

which for any closed surface containing the internal points, $\mathbf{r}_{\text{int}}^{1:M}$, can be written (from the divergence theorem) as follows:

$$\oint_S \mathbf{q} \cdot \mathbf{n} dS = \sum_{j=1}^M g^j. \quad (11)$$

We can implicitly define this closed surface, S , using the condition that the local surface flux is equal to the scalar field we wish to integrate (ϕ):

$$\phi(\mathbf{r}) - \mathbf{q}(\mathbf{r}) \cdot \mathbf{n}(\mathbf{r}) = 0, \quad \mathbf{r} \in S. \quad (12)$$

Note, this implicit equation is written in terms of the (outward-facing) normal of the surface it is used to define, and so the latter must be known or approximated (see Sections 2.4 and 3.2). On combining Eqs (11) and (12) we obtain an expression for the desired surface integral:

$$\oint_S \phi(\mathbf{r}) dS = \sum_{j=1}^M g^j, \quad (13)$$

where the weights g^j required for S to pass through the known surface points (or close to, for $M < N$) are found from the MFS; i.e. by evaluating Eq. (9b) with $\mathbf{q}_n = \phi(\mathbf{r}_{\text{sur}}^i)$.

The surface area of S can be obtained from Eq. (13) by simply setting $\phi(\mathbf{r}) = 1$. Appendix A contains a short MATLAB script that performs the entire procedure, given known internal points, surface points and normals.

2.3. Calculating volume integrals

Integrals over the volume enclosed by the surface can also be related to the surface integral in Eq. (13) using the divergence theorem. For example, in the calculation of:

- the volume:

$$V = \iiint_V dV = \frac{1}{3} \iiint_V (\nabla \cdot \mathbf{r}) dV = \frac{1}{3} \oint_S (\mathbf{r} \cdot \mathbf{n}) dS \quad (14)$$

which can be calculated using Eq. (13) with $\phi(\mathbf{r}) = \mathbf{r} \cdot \mathbf{n}/3$.

- the centroid (here given just for the x -component):

$$\bar{x} = \frac{1}{V} \iiint_V x dV = \frac{1}{4V} \iiint_V \nabla \cdot (x\mathbf{r}) dV = \frac{1}{4V} \oint_S (x\mathbf{r} \cdot \mathbf{n}) dS. \quad (15)$$

which can be calculated using Eq. (13) with $\phi(\mathbf{r}) = x\mathbf{r} \cdot \mathbf{n}/(4V)$. The same process can be repeated for \bar{y} and \bar{z} .

- the volume average of any solenoidal field \mathbf{v} (x -component):

$$\bar{v}_x = \mathbf{i}_x \cdot \frac{1}{V} \iiint_V \mathbf{v} dV = \mathbf{i}_x \cdot \frac{1}{V} \iiint_V \nabla \cdot (\mathbf{v}\mathbf{r}) dV = \frac{1}{V} \oint_S (\mathbf{v}_x \mathbf{r} \cdot \mathbf{n}) dS, \quad (16)$$

where \mathbf{i}_x is the unit vector in the x -direction. Equation (16) can be calculated by substituting $\phi(\mathbf{r}) = \mathbf{v}_x \mathbf{r} \cdot \mathbf{n}/V$ into Eq. (13). The same can be repeated for \bar{v}_y and \bar{v}_z . The volume average of a divergence-free tensor field, \mathbf{P} , can be calculated similarly, with $\phi_{\alpha\beta}(\mathbf{r}) = P_{\alpha\beta} \mathbf{r} \cdot \mathbf{n}/V$.

- the x -component of the volume average of a gradient field $\nabla\psi$:

$$\overline{\nabla\psi}_x = \mathbf{i}_x \cdot \frac{1}{V} \iiint_V \nabla\psi dV = \mathbf{i}_x \cdot \frac{1}{V} \iiint_V \nabla \cdot (\psi\mathbf{I}) dV = \frac{1}{V} \oint_S (\psi\mathbf{i}_x \cdot \mathbf{n}) dS, \quad (17)$$

where \mathbf{I} is the identity tensor. Equation (17) can be calculated by substituting $\phi(\mathbf{r}) = n_x\psi/V$ into Eq. (13). The same can be repeated for $\overline{\nabla\psi}_y$ and $\overline{\nabla\psi}_z$. Similarly, the components of the volume average of a gradient tensor field, $(\nabla w)_{\alpha\beta}$, can be calculated with $\phi_{\alpha\beta}(\mathbf{r}) = n_\alpha w_\beta/V$.

- a diagonal component of the volume's gyration tensor about the origin:

$$\mathcal{R}_{zz} = \iiint_V (x^2 + y^2) dV = \frac{1}{5} \iiint_V \nabla \cdot ((x^2 + y^2)\mathbf{r}) dV = \frac{1}{5} \oint_S ((x^2 + y^2)\mathbf{r} \cdot \mathbf{n}) dS \quad (18)$$

which can be calculated using Eq. (13) with $\phi(\mathbf{r}) = \frac{1}{5}(x^2 + y^2)\mathbf{r} \cdot \mathbf{n}$.

- an off-diagonal component of the volume's gyration tensor about the origin:

$$\mathcal{R}_{xy} = \iiint_V (-xy) dV = -\frac{1}{5} \iiint_V \nabla \cdot (xy\mathbf{r}) dV = -\frac{1}{5} \oint_S (xy\mathbf{r} \cdot \mathbf{n}) dS \quad (19)$$

which can be calculated using Eq. (13) with $\phi(\mathbf{r}) = -\frac{1}{5}xy\mathbf{r} \cdot \mathbf{n}$.

2.4. Calculating surface normals and curvature

Adopting the approach of Tankelevich *et al.* [6, 7], we can define another implicit surface that runs through the surface points (approximately, if $M < N$) by substituting $T = 1$ into Eq. (4):

$$F(\mathbf{r}) = \frac{1}{4\pi} \sum_{j=1}^M \frac{g^j}{\|\mathbf{r} - \mathbf{r}_{\text{int}}^j\|} - 1 = 0. \quad (20)$$

where the weights are found by evaluating Eq. (9a) with $T^{1:N} = 1$. The explicit expressions

$$\nabla F(\mathbf{r}) = -\frac{1}{4\pi} \sum_{j=1}^M g^j \frac{(\mathbf{r} - \mathbf{r}_{\text{int}}^j)}{\|\mathbf{r} - \mathbf{r}_{\text{int}}^j\|^3} \quad (21)$$

and

$$\nabla \nabla F(\mathbf{r}) = \mathbf{H} = -\frac{1}{4\pi} \sum_{j=1}^M g^j \left[\frac{\mathbf{I}}{\|\mathbf{r} - \mathbf{r}_{\text{int}}^j\|^3} - \frac{3(\mathbf{r} - \mathbf{r}_{\text{int}}^j)(\mathbf{r} - \mathbf{r}_{\text{int}}^j)}{\|\mathbf{r} - \mathbf{r}_{\text{int}}^j\|^5} \right] \quad (22)$$

provide the *outward-facing* normal vector

$$\mathbf{n}(\mathbf{r}) = -\frac{\nabla F}{\|\nabla F\|} \quad (23)$$

and the Hessian matrix, $\mathbf{H}(\mathbf{r})$, from which local curvature can be calculated (see Section 4 of [11]):

$$K_M = \frac{\mathbf{n} \cdot \mathbf{H} \cdot \mathbf{n}}{2\|\nabla F\|}, \quad K_G = \frac{\mathbf{n} \cdot \mathbf{H}^* \cdot \mathbf{n}}{\|\nabla F\|^2}, \quad \text{and} \quad k_{1,2} = K_M \pm \sqrt{K_M^2 - K_G}, \quad (24)$$

where K_M , K_G , $k_{1,2}$ are the mean, Gaussian and principal curvatures, respectively. Note, in the evaluation of the mean curvature we have used $\text{tr}(\mathbf{H}) = \nabla^2 F = 0$.

3. Numerical verification and benchmarking

Here we test the method proposed above for calculating integral and local properties of a solid sphere, torus and ellipsoid; for which there are known analytical solutions. The sphere and ellipsoid can be implicitly defined by

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} - 1 = 0, \quad (25)$$

where we have chosen $a = b = c = 1$, for the sphere, and $a = 2$, $b = 1.5$ and $c = 1$ for the triaxial ellipsoid. The torus is defined by

$$(x^2 + y^2 + z^2 + a^2 - b^2)^2 - 4a^2(x^2 + y^2) = 0, \quad (26)$$

where we have chosen $a = 3$ and $b = 1$.

3.1. Integral calculations

The points on the surface of each object are distributed evenly, found using the MATLAB code (DistMesh) written by Persson and Strang [12]. The points are shown in Fig. 2 (visualised using a triangular mesh) for the different resolutions used.

There are not fixed rules on the location of the internal points (see [9, 10] for work on the subject), but for the problems discussed here a surface-normal projection into the volume typically works well, i.e.: $\mathbf{r}_{\text{int}}^i = \mathbf{r}_{\text{sur}}^i - |\alpha^i| \mathbf{n}^i$, where α^i can reflect the scale of the geometry locally. For a unit sphere, a constant $\alpha = 0.1$ – 0.9 is useable, providing greater accuracy as α increases. A drawback to larger α is numerical condition, which can be greatly improved by removing a small fraction of internal points (such that $M < N$). However, for consistency and simplicity, for

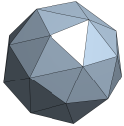
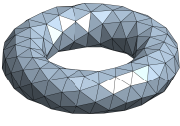
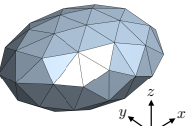
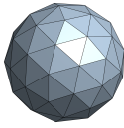
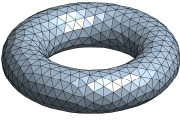
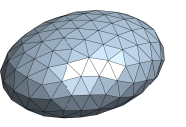
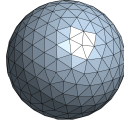
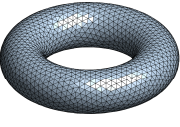
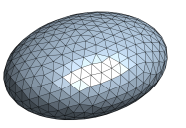
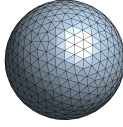
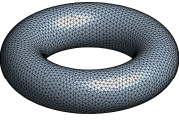
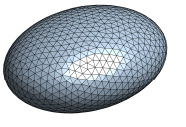
Resolution (Key)	Sphere points	Torus points	Ellipsoid points
Coarse (C)			
Medium (M)			
Fine (F)			
Extra fine (XF)			

Figure 2: Surface points at different resolution on a sphere, torus and ellipsoid; triangular meshes are for visualisation. Points generated using DistMesh [12].

Table 1: MFS predictions of integral properties of the solid unit sphere.

Res.	S	V	$ \bar{x} $	R_{zz}
C (24)	12.470797593	4.1569325309	0.0000001878	1.6627740398
M (54)	12.562339287	4.1874464291	0.0000000317	1.6750031141
F (222)	12.566370216	4.1887900721	0.0000000000	1.6755160112
XF (480)	12.566370614	4.1887902047	0.0000000000	1.6755160819
Analytical	12.566370614	4.1887902048	0.0000000000	1.6755160819

all results in this paper, $M=N$, and in Sec. 3, $\alpha = 0.5$. It is assumed, for now, that the normals are known.

Tables 1-3 show MFS predictions for surface area (S), volume (V), the x -component of the centroid (\bar{x}), and the radius of gyration about the z -axis (R_{zz}) for each of the three objects. The convergence to the respective analytical result is very fast in all cases. Appendix A provides the MATLAB script used to generate these results.

Table 4 compares the MFS prediction of surface area to that obtained by summing the facets of the triangular mesh. As expected, the triangular mesh prediction also converges to the exact result, but extremely slowly by comparison. It is worth noting that the analytical result for the surface area of the triaxial ellipsoid is non-trivial, involving the evaluation of incomplete elliptic integrals.

Table 2: MFS predictions of integral properties of the solid torus ($a = 3, b = 1$).

Res. (N)	S	V	$ \bar{x} $	R_{zz}
C (192)	114.97468933	58.531995322	0.0025537993	568.04497226
M (568)	118.29956624	59.249680824	0.0000893860	577.45564344
F (1952)	118.43495971	59.217818016	0.0000001851	577.37264276
XF (4296)	118.43525172	59.217627580	0.0000000025	577.37186209
Analytical	118.43525281	59.217626407	0.0000000000	577.37185746

Table 3: MFS predictions of integral properties of the solid ellipsoid ($a = 2, b = 1.5, c = 1$).

Res. (N)	S	V	$ \bar{x} $	R_{zz}
C (50)	27.422434712	12.467113152	0.0027413632	15.910670092
M (154)	27.876354954	12.565583874	0.0000648984	15.717781876
F (454)	27.886421384	12.566377863	0.0000027213	15.708023900
XF (1022)	27.886442683	12.566370172	0.0000011104	15.707961418
Analytical	27.886442474	12.566370614	0.0000000000	15.707963268

Table 4: Percentage error from analytical results for calculated surface area.

Res.	Sphere		Torus		Ellipsoid	
	MFS	Tri-mesh	MFS	Tri-mesh	MFS	Tri-mesh
C	0.76 %	12 %	2.9 %	2.3 %	1.7 %	6.7 %
M	0.032 %	5.6 %	0.11 %	0.79 %	0.036 %	2.2 %
F	3.2×10^{-6} %	1.4 %	2.5×10^{-4} %	0.23 %	7.6×10^{-5} %	0.76 %
XF	1.7×10^{-9} %	0.64 %	9.2×10^{-7} %	0.11 %	7.5×10^{-7} %	0.34 %

Table 5: Average angle between estimated and exact normal (in radians) for the ellipsoid.

Resolution	MFS	vertexNormal	Meyer <i>et al.</i> [13, 14]
C	0.0294	0.0681	0.0647
M	0.0086	0.0286	0.0246
F	0.0013	0.0109	0.0091
XF	0.0002	0.0052	0.0042

Table 6: Average (over the ellipsoid surface points) percentage error in calculated mean curvature from exact result.

Resolution	MFS	Meyer <i>et al.</i> [13, 14]
C	4.14 %	3.98 %
M	0.84 %	1.31 %
F	0.17 %	0.50 %
XF	0.03 %	0.30 %

3.2. Local calculations

For cases where the surface normals are not known, estimates can be obtained from a triangulation of the surface points; using, for example, MATLAB's `vertexNormal` function or the method proposed by Meyer *et al.* [13]. These estimates can be used to locate the internal points, allowing more accurate estimates (potentially) to be obtained using the MFS procedure described in Sec. 2.4.

Table 5 shows the average angle between the calculated and true surface normal for the ellipsoid (averaged over surface points), where a comparison is made between the MFS (Sec. 2.4), `vertexNormal` and Dastan's implementation [14] of Meyer *et al.*'s method [13].

For the finest mesh, the MFS normals are an order of magnitude more accurate, though more expensive to calculate. The same is true for the average error in mean curvature, which is shown in Table 6.

4. Application to an asymmetric particle

For a more challenging test, we now apply the MFS to the surface-area estimation of an arbitrarily-shaped particle; see Fig. 3. Unlike the geometries considered in Sec. 3, this particle is asymmetric and has significant variation in its surface curvature. The particle is defined with an arbitrarily-chosen implicit function, with surface points determined from DistMesh [12]; the volume of the particle is roughly equal to that of a unit sphere.

The variation in surface curvature make placement of the internal points less straightforward than in Sec. 3. Here, α^i is chosen such that each internal point is exactly 5% closer to its respective surface point than any other, which helps to maintain the condition of the matrix.

Table 7 compares the MFS estimates of surface area with the standard and simple approach of summing the face areas of a surface triangulation. No analytical solution for the surface area exists, but both methods converge to a similar value with increasing resolution. The rate of convergence for the MFS is far greater; it appears that an MFS estimate with 76 surface points is similarly accurate to a triangular-mesh estimate with 3282 surface points.

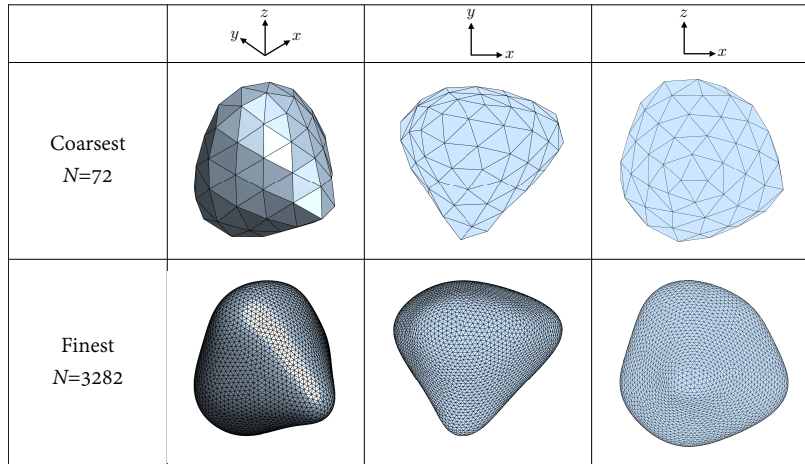


Figure 3: Surface points on an asymmetric particle. Points generated using DistMesh [12].

Table 7: Estimates of surface area for the asymmetric particle shown in Fig. 3.

Resolution (N)	MFS	Tri-Mesh
76	13.6278372	13.0113930
130	13.6409691	13.2611840
228	13.6417555	13.4217729
940	13.6426379	13.5881170
2092	13.6426331	13.6181171
3282	13.6426330	13.6270117

5. Discussion and Summary

Fitting implicit surfaces to surface points is established in the fields of computer graphics and medical imaging (e.g. [6, 7, 15, 16, 17, 18]). The novelty of this work lies in defining an implicit surface, using fundamental solutions, such that integrals over the implicit surface can be obtained easily and accurately.

Two key aspects of the method are 1) the implicit equation used to specify the surface is defined in terms of a vector field, \mathbf{q} , with a known volume-integrated divergence (known, because \mathbf{q} is the gradient of a superposition of fundamental solutions to Laplace’s equation, and thus the volume-integrated divergence of \mathbf{q} is equal to the sum of the fundamental-solution weights); 2) the implicit surface is defined such that the surface-normal component of \mathbf{q} is equal to some field ϕ , allowing us to apply the divergence theorem to relate the surface integral of ϕ to the sum of the fundamental-solution weights. The approximation arises in fitting this implicit surface through discrete points, but the numerical accuracy for the objects we have considered is extremely good.

It is possible to define another implicit surface (as proposed by Tankelevich *et al.* [6, 7]) to calculate local properties, such as surface normals and curvature (the former is required to calculate surface integrals). The accuracy of the MFS is an improvement on the existing methods considered, but comes at a greater computational cost. However, if the inverse is required for solving a physical problem (for example, in solving Laplace’s equation external to the object), these calculations come, effectively, for free. The same is true for the calculations of surface and volume integrals.

The greatest potential of the method might be in the calculation of properties needed for dynamics calculations (for constant density objects), e.g. the mass, the centre of mass, and the moment of inertia tensor, or for calculating volume integrals of solenoidal or gradient fields from discrete data at known containing surfaces (e.g. calculating average incompressible flow velocity or strain rate)

All the documented drawbacks of the MFS apply to this method. Primarily these relate to the unknown position of the internal points; though some pragmatic guidelines can be established, and the technique outlined above (a surface-normal projection into the surface) is generally effective, given a reasonable estimate of the normal. Because of the nature of the method, it is not possible to model non-smooth surfaces or very slender objects as easily or accurately, which applies equally to MFS in its conventional context.

Acknowledgments

The author is grateful to James Sprittles for helpful discussions and to the EPSRC for financially supporting the work (grants EP/N016602/1, EP/V01207X/1).

Data Availability

The MATLAB script and input files (positions of the surface points, volume points, and surface normals for the various geometries considered) are available to download from the Warwick Research Archive Portal (WRAP) at <http://wrap.warwick.ac.uk/161189/>.

Appendix A. MATLAB Script

This short MATLAB script loads surface points, internal points and surface normals, and calculates the properties presented in Tables 1-3.

```
1 rsur=load ('rsur.txt'); %load surface points (1:N,1:3)
2 rint=load ('rint.txt'); %load internal points (1:N,1:3)
3 n=load ('n.txt'); %load surface normals (1:N,1:3)
4
5 N=length(rsur);
6 B=zeros(N,N);
7 for i=1:N
8     for j=1:N
9         R=norm(rsur(i,:)-rint(j,:));
10        rn=(rsur(i,:)-rint(j,:))*n(i,:);
11        B(i,j)= (rsur(i,:)-rint(j,:))*n(i,:)/(4*pi*R^3); % construct B
12    end
13 end
14 INVB=inv(B); % calculate B-1
15
16 rn=sum(rsur' .* n)';
17 x=rsur(:,1);y=rsur(:,2);
18
19 phi_area(1:N,1)=1;
20 S=sum(INVB*phi_area) % calculate surface area S
21
22 phi_volume=rn/3;
23 V=sum(INVB*phi_volume) % calculate volume V
24
25 phi_centroid=x.*rn/(4*V);
26 centroid=sum(INVB*phi_centroid) % calculate x-centroid  $\bar{x}$ 
27
28 phi_Rzz=1/5*(x.^2+y.^2).*rn;
29 Rzz=sum(INVB*phi_Rzz) % calculate Rzz
```

References

- [1] A. H. Cheng, Y. Hong, An overview of the method of fundamental solutions—Solvability, uniqueness, convergence, and stability, *Eng Anal Bound Elem* 120 (2020) 118–152. doi:10.1016/j.enganabound.2020.08.013.
- [2] A. S. Rana, S. Saini, S. Chakraborty, D. A. Lockerby, J. E. Sprittles, Efficient simulation of non-classical liquid–vapour phase-transition flows: a method of fundamental solutions, *J. Fluid Mech.* 919 (2021) A35. doi:10.1017/jfm.2021.405.
- [3] A. Karageorghis, D. Lesnic, The method of fundamental solutions for the Oseen steady-state viscous flow past obstacles of known or unknown shapes, *Numer Methods Partial Differential Eq.* 35 (6) (2019) 2103–2119. doi:10.1002/num.22404.
- [4] R. Claydon, A. Shrestha, A. S. Rana, J. E. Sprittles, D. A. Lockerby, Fundamental solutions to the regularised 13-moment equations: efficient computation of three-dimensional kinetic effects, *J. Fluid Mech.* 833 (2017) R4. doi:10.1017/jfm.2017.763.

- [5] D. A. Lockerby, B. Collyer, Fundamental solutions to moment equations for the simulation of microscale gas flows, *J. Fluid Mech.* 806 (2016) 413–436. doi:10.1017/jfm.2016.606.
- [6] R. Tankelevich, G. Fairweather, A. Karageorghis, Y.-S. Smyrlis, Potential field based geometric modelling using the method of fundamental solutions, *Int. J. Numer. Meth. Engng* 68 (12) (2006) 1257–1280. doi:10.1002/nme.1763.
- [7] R. Tankelevich, G. Fairweather, A. Karageorghis, Three-dimensional image reconstruction using the PF/MFS technique, *Eng Anal Bound Elem* 33 (12) (2009) 1403–1410. doi:10.1016/j.enganabound.2009.04.015.
- [8] N. Ray, D. Wang, X. Jiao, J. Glimm, High-Order Numerical Integration over Discrete Surfaces, *SIAM J. Numer. Anal.* 50 (6) (2012) 3061–3083. doi:10.1137/110857404.
- [9] A. Karageorghis, A Practical Algorithm for Determining the Optimal Pseudo-Boundary in the Method of Fundamental Solutions, *AAMM* 1 (4) (2009) 510–528. doi:10.4208/aamm.09-m0916.
- [10] C. S. Chen, A. Karageorghis, Y. Li, On choosing the location of the sources in the MFS, *Numer Algor* 72 (1) (2016) 107–130. doi:10.1007/s11075-015-0036-0.
- [11] R. Goldman, Curvature formulas for implicit curves and surfaces, *Computer Aided Geometric Design* 22 (7) (2005) 632–658. doi:10.1016/j.cagd.2005.06.005.
- [12] P.-O. Persson, G. Strang, A Simple Mesh Generator in MATLAB, *SIAM Rev.* 46 (2) (2004) 329–345. doi:10.1137/S0036144503429121.
- [13] M. Meyer, M. Desbrun, P. Schröder, A. H. Barr, Discrete Differential-Geometry Operators for Triangulated 2-Manifolds, in: *Visualization and Mathematics III*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 35–57. doi:10.1007/978-3-662-05105-4.
- [14] A. Dastan, Gaussian and mean curvatures calculation on a triangulated 3d surface, *MATLAB Central File Exchange* (2017). URL <https://www.mathworks.com/matlabcentral/fileexchange/61136-gaussian-and-mean-curvatures-calculation-on-a-triangulated-3d-surface>
- [15] C. S. Chen, L. Amuzu, K. Acheampong, H. Zhu, Improved geometric modeling using the method of fundamental solutions, *Eng Anal Bound Elem* 130 (2021) 49–57. doi:10.1016/j.enganabound.2021.04.025.
- [16] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, T. R. Evans, Reconstruction and representation of 3D objects with radial basis functions, in: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques - SIGGRAPH '01*, ACM Press, 2001, pp. 67–76. doi:10.1145/383259.383266.
- [17] B. Rodrigues de Araújo, J. Armando Pires Jorge, Adaptive polygonization of implicit surfaces, *Computers & Graphics* 29 (5) (2005) 686–696. doi:10.1016/j.cag.2005.08.027.
- [18] X.-Y. Liu, H. Wang, C. S. Chen, Q. Wang, X. Zhou, Y. Wang, Implicit surface reconstruction with radial basis functions via PDEs, *Engineering Analysis with Boundary Elements* 110 (2020) 95–103. doi:10.1016/j.enganabound.2019.09.021.