INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 20, Number 3, Pages 437–458

doi: 10.4208/ijnam2023-1018

A FULLY IMPLICIT METHOD USING NODAL RADIAL BASIS FUNCTIONS TO SOLVE THE LINEAR ADVECTION EQUATION

P.-A. GOURDAIN, M. B. ADAMS, M. EVANS, H. R. HASSON, J. R. YOUNG, I. WEST-ABDALLAH

Abstract. Radial basis functions are typically used when discretization schemes require inhomogeneous node distributions. While spawning from a desire to interpolate functions on a random set of nodes, they have found successful applications in solving many types of differential equations. However, the weights of the interpolated solution, used in the linear superposition of basis functions to interpolate the solution, and the actual value of the solution are completely different. In fact, these weights mix the value of the solution with the geometrical location of the nodes used to discretize the equation. In this paper, we used nodal radial basis functions, which are interpolants of the impulse function at each node inside the domain. This transformation allows to solve a linear hyperbolic partial differential equation using series expansion rather than the explicit computation of a matrix inverse. This transformation effectively yields an implicit solver which only requires the multiplication of vectors with matrices. Because the solver requires neither matrix inverse nor matrix-matrix products, this approach is numerically more stable and reduces the error by at least two orders of magnitude, compared to solvers using radial basis functions directly. Further, boundary conditions are integrated directly inside the solver, at no extra cost. The method is locally conservative, keeping the error virtually constant throughout the computation.

Key words. Radial basis functions, implicit scheme, hyperbolic equations.

1. Introduction

Radial basis function interpolation is one of the few methods that can approximate across a d-dimensional space a function only defined on a randomly distributed set of n nodes $x_1, x_2, ..., x_n \in \mathbb{R}^d$ [17, 14]. While initially used for interpolation problems, this method can be used to define surfaces in multiple dimensions [3] or solve partial differential equations [18, 19, 6, 9]. One important characteristic of radial basis functions is their definition using the relative position of nodes, obtained from the Euclidian norm $\|\cdot\|_d$, rather than their absolute location in space. For $x \in \mathbb{R}^d$ we define the radial basis function (RBF) at every node x_j as $\Phi_{x_j}(x) = \phi(||x - x_j||_d)$, which we will also write as $\Phi(x - x_j)$. Typically, Φ is normalized, i.e. $\Phi(0) = 1$. New functions can be generated by scaling of the modal function ϕ by a factor α, giving the standard definition of the radial basis function Φ_{α,x_j} as

$$\Phi_{\alpha,x_j}(x) = \phi(||x - x_j||_d/\alpha),$$

which we will also write as $\Phi_\alpha(x - x_j)$. We use the width parameter α rather than the usual shape factor (i.e. $1/\alpha$) in this paper because we will compare the radial basis function spread to the domain size throughout this paper.

A continuous function f can be approximated on a finite set of nodes $U = \{x_1, x_2, ..., x_n\}$ using radial basis functions by computing a set of weights ω_j defined by

$$\forall x_i \in U, f(x_i) = \sum_j \omega_j \Phi_\alpha (x_i - x_j).$$

Received by the editors on February 9, 2022 and, accepted on January 26, 2023.
2000 *Mathematics Subject Classification.* 65D12, 65F05, 35F50, 35F46, 65D05, 35L65.

The weights ω_j can be found by solving the linear system

$$\begin{bmatrix} \Phi_\alpha(x_1 - x_1) & ... & \Phi_\alpha(x_1 - x_n) \\ ... & ... & ... \\ \Phi_\alpha(x_n - x_1) & ... & \Phi_\alpha(x_n - x_n) \end{bmatrix} \begin{bmatrix} \omega_1 \\ ... \\ \omega_n \end{bmatrix} = \begin{bmatrix} f(x_1) \\ ... \\ f(x_n) \end{bmatrix}$$

written in compact form as $[\Phi_\alpha][\omega] = [f]$. To solve this system, we need to find the inverse of the matrix $[\Phi_\alpha]$ and compute the weights ω_j using

$$[\omega] = [\Phi_\alpha]^{-1}[f]. \tag{1}$$

The radial basis function Φ is said to be definite positive when $[\Phi]$ is invertible, supposing U does not have any redundant nodes (i.e. $x_i{=}x_j$ while $i \neq j$). Once the weights are known, the function f can be interpolated between nodes using the function $\overline{f}$ defined by

$$\overline{f}(x) = \sum_j \omega_j \Phi_\alpha \left(x - x_j\right). \tag{2}$$

Since radial basis functions can interpolate any smooth function using a linear combination of differentiable functions, it quickly spawned differential equation solvers for elliptic [24], hyperbolic [23], parabolic [36] or shallow-water [11] equations using different approaches such as spectral [30] or backward substitution [27, 37] methods. Even differential equations with fractional operators [25, 22], curvilinear coordinates [33] or complex boundary conditions [20] can be solved using this technique. The ease in defining spacial and temporal derivatives is probably the main reason this method has found universal applications.

However, one major issue raised by Eq. (2) is evident. The interpolated function $\overline{f}$ is now defined in term of the weights ω_j. This becomes an issue when solving differential equations using radial basis functions. For instance, the value of the function might be required to compute the value of another function or match a set of boundary conditions. Further if we want to interpolate a new function g, then all the weights ω_j must be computed again.

In the rest of the paper, we first define a set of nodal radial basis functions (N-RBF) that interpolates the impulse function. These functions form an orthonormal basis for the inner product of an interpolated function on U. First, we summarize the basic properties of NRBF formed using RBF with compact support, then we present the theory behind our linear advection equation solver and compare it to standard solvers. Finally, we conclude by showing how NRBF can be trivially extended to solve the advection equation with a velocity which varies across the domain. It is important to note that the method is completely independent of the number of spatial dimensions by construction. As a result, we will not look at multidimensional cases in this paper. While we do not claim that the method will perform well in a larger number of dimensions, the solver proposed is clearly dimension agnostic.

2. Definition of nodal radial basis functions

The solution to avoid computing f from the w_j is relatively straightforward. Rather than using radial basis functions directly, which have well defined, yet poorly matched, values at the node points, we can used them to interpolate the impulse functions $\delta(x - x_i)$ first. Once these new functions Ψ_{x_i} are defined, interpolation is trivial since the weights for each interpolant Ψ_{x_i} is $f(x_i)$. To construct them, we

can rewrite Eq. (1) as

$$[\omega] = [\Phi_\alpha]^{-1} \sum_j f(x_j)[\delta_j]. \tag{3}$$

Here the vectors $[\delta_j] = [\delta(x_i - x_j)]^T$ are defined using the impulse function

$$\forall x, y \in \mathbb{R}, \delta(x-y) = \begin{cases} 1 \text{ if } x = y \\ 0 \text{ if } x \neq y \end{cases}$$

This decomposition allows to create a series of interpolants on U for each translated impulse function $\delta(x - x_j)$

$$\forall x_i, x_j \in U,\ \Psi_{\alpha,x_j}(x_i) = \delta(x_i - x_j). \tag{4}$$

They can be expressed in term of our radial basis functions as

$$\Psi_{\alpha,x_j}(x) = \sum_i \Omega_{\mathrm{ij}} \Phi_\alpha(x_i - x). \tag{5}$$

and their weights Ω_{ij} can be computed using Eq. (1)

$$[\Omega_j] = [\Phi_\alpha]^{-1}[\delta_j]$$

It is interesting to note that these weights simply are the elements of the matrix $[\Phi_\alpha]^{-1}$.

Theorem 1. The interpolant of f formed using Φ_{x_i}, i.e. $\overline{f}(x) = \sum_j \omega_j \Phi_\alpha(x_i - x_j)$, and the interpolant of f formed using Ψ_{x_i}, i.e. $\overline{\overline{f}}(x) = \sum_i f(x_i)\Psi_{\alpha,x_i}(x)$, are identical.

Proof. We can write $\omega_j = \sum_i f(x_i)\Omega_{\mathrm{ij}}$ using Eq. (3), and Eq. (2) becomes

$$\bar{f}(x) = \sum_j \sum_i f(x_i)\Omega_{\mathrm{ij}}\Phi_\alpha(x - x_j),$$

The sum operators can be easily permuted to give

$$\bar{f}(x) = \sum_i f(x_i) \sum_j \Omega_{\mathrm{ij}}\Phi_\alpha(x - x_j).$$

Since $\Omega_{\mathrm{ij}} = \Omega_{\mathrm{ji}}$ since $\Phi_\alpha(x - x_j) = \Phi_\alpha(x_j - x)$ because $||x - x_j||_d = ||x_j - x||_d$, we can rewrite the above equation as

$$\overline{f}(x) = \sum_i f(x_i)\Psi_{\alpha,x_i}(x). \tag{6}$$

So $\overline{\overline{f}} \equiv \overline{f}$. □

While the functions Ψ_{α,x_i} are constructed using radial basis functions, they are fundamentally different. Figure 1 shows $\Psi_{\alpha,0}$ together with one of the radial basis functions used for its construction. In fact, it looks more similar to the barycentric form of rational interpolants [10]. Clearly, Eq. (2) and Eq. (6) are equivalent and the interpolants $\overline{f}$ and $\overline{\overline{f}}$ are mathematically identical. While Eq. (5) is used in the radial point interpolation collocation method (RPICM) [31] or pseudo-spectral methods [5, 7], it is important to note that we defined here the function Ψ_{α,x_j} to be the interpolant of the translated impulse function $\delta(x - x_j)$, a definition similar to the construction of cardinal functions [2]. However, there is no advantage in using Ψ_{α,x_j} as an interpolation method. The equivalence between Eq. (2) and Eq. (6)

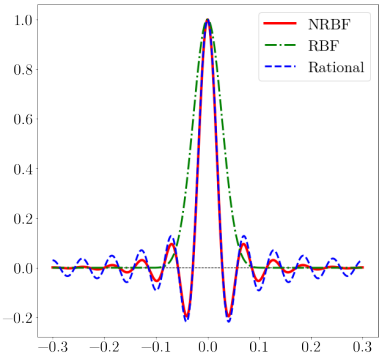


FIGURE 1. The nodal radial basis function (NRBF) with 0 as its main support node and the radial basis function (RBF) that was used to generate it. The rational basis function centered on the same node is also given. We show only a portion of the interval [-1,1] for clarity. Each interval nodes (besides 0) can be found where the nodal radial basis function crosses the x-axis.

shows that we get the exact same interpolant, yet we need to compute many radial basis functions at every node in Eq. (6), and then perform a linear system solve to form a single Ψ_{α,x_j}.

However, the functions Ψ_{α,x_j} can be used efficiently in solving differential equations, since the interpolating weights of a function f using Ψ_{α,x_j} are the values of the functions f at the nodes x_j. The main reason is that the functions Ψ_{α,x_j} are an orthonormal basis of $\overline{U}$, the space of interpolants on U, for the inner product defined as

$$< \overline{f}.\overline{g} >= \sum_{k=1}^{n} |\overline{f}\,(x_k)\,\overline{g}(x_k)|,$$

for any two interpolants $\overline{f}$ and $\overline{g}$ in $\overline{U}$.

Theorem 2. The set of functions $\{\Psi_{x_j}\}_{x_j \in U}$ forms an orthonormal basis for the inner product of interpolated functions $< . >$.

Proof. Using the definition of the nodal radial basis function given by Eq. (4)

$$\forall i,j \;\; < \Psi_{x_i}.\Psi_{x_j} >= \sum_{k=1}^{n} |\Psi_{x_i}(x_k)\Psi_{x_j}(x_k)| = \sum_{k=1}^{n} \delta_{ik}\delta_{jk}$$

So, $< \Psi_{x_i}.\Psi_{x_j} >= 0 \iff i \neq j$ and $< \Psi_{x_i}.\Psi_{x_j} >= 1 \iff i = j$. □

Based on the definition of the nodal radial function from Eq. (5), the spatial derivative $\partial_k \Psi_{\alpha,x_j}$ can be computed easily

$$\partial_k \Psi_{\alpha,x_j}(x) = \sum_i \Omega_{\mathrm{ij}} \partial_k \Phi_\alpha(x_i - x). \tag{7}$$

Reverting to the modal basis function at this stage, we now rewrite Eq. (7) as

$$\partial_k \Psi_{\alpha,x_j}(x) = \sum_i \Omega_{\mathrm{ij}} (d_r \phi)(||x_i - x||_d/\alpha) \partial_k ||x_i - x||_d/\alpha. \tag{8}$$

While Eq. (8) uses the elements of $[\Phi_\alpha]^{-1}$ we will show later that it is not necessary to compute the matrix inverse of $[\Phi_\alpha]$. Rather, the solver uses a Cholesky decomposition to compute the relevant spatial derivatives. The functions $\partial_k \Psi_{\alpha,x_j}(x)$ only depend on the node distribution and should be recomputed only when nodes change locations, or when nodes are added (or dropped).

$\Psi_{\alpha,x_j}(x)$ is nodal in the sense that it is defined using nodes, rather than modal, i.e. the scaled translation of the modal function ϕ used in Eq. (1). It is also radial in the sense that it is solely defined by a series of Euclidian distances. Since the family of functions defined by all the points inside the domain forms an orthogonal basis for this domain, we call the functions Ψ_{α,x_j} nodal radial basis functions (NRBF) in the remainder of this paper.

Another key advantage of nodal radial basis functions, over more conventional discretization schemes, boils down to computing discretized derivatives as a sum of exact derivatives, given by Eq. (8). Exact derivatives, as opposed to discretized derivatives, are always locally conservative i.e. the derivative operator ∂_q with respect to the variable q is such that for any smooth function g

$$\int_{\Omega_c} \partial_q g \mathrm{dv} = \int_{\partial\Omega_c} g n_q \mathrm{ds}, \tag{9}$$

where Ω_c is the cell of the discretized domain and $\partial\Omega_c$ is the cell boundary. This is the divergence theorem applied to a single direction.

Theorem 3. The derivative operator $\widetilde{\partial}_q$, used in Eq. (13) and defined as

$$\widetilde{\partial}_q f = \sum_i \overline{f}(x_i) \partial_q \Psi_{\alpha,x_i}$$

for any interpolant $\overline{f}$ is locally conservative.

Proof. The derivative can be written as

$$\widetilde{\partial}_q f = \sum_i f(x_i) \sum_j \Omega_{\mathrm{ij}} \partial_q \Phi_\alpha(x - x_j)$$

using Eq. (7). Since $\widetilde{\partial}_q$ is a finite sum of exact derivatives, it trivially verifies Eq. (9). □

3. Basic characteristics of nodal radial basis functions using compact radial basis functions

In this section, we focus on the one-dimensional case, mostly with homogeneously distributed nodes, to illustrate the properties of nodal radial basis functions in a simple framework. However, this work can be directly extended to multiple space dimensions and random node distributions.

The key advantage of nodal radial basis functions is $\forall x_i, x_j \, \Psi_{\alpha,x_j}(x_i) = \delta_{\mathrm{ij}}$ by construction, a property shared with Lagrange polynomials and rational interpolants [8]. Yet, they retain the multivariate interpolation capabilities on a set of randomly distributed nodes, a theme central to radial basis function interpolation. Unless otherwise stated, the modal radial basis functions used to form Ψ_{α,x_j} are compact Wendland functions [32] defined by

$$\psi_{p,0}(r) = (1-r)_+^p = \begin{cases} (1-r)^p & \text{for } 0 \le r \le 1 \\ 0 & \text{for } r > 1 \end{cases}$$

and

$$\psi_{p,q}(r) = \mathfrak{I}^q \psi_{p,0} \quad \text{for } 0 \leq r \leq 1,$$

where p and q are integers. The operator $\mathfrak{I}$ above is defined as $\mathfrak{I}f(r) = \int_r^\infty f(t)tdt$ for $0 \leq r$. Wendland functions are C^k and can be computed analytically, then lead to a strictly positive definite matrix $[\Phi_\alpha]$ in $\mathbb{R}^d$, where $d<p$ and $k=2q$.

Figure 1 shows the difference between the nodal radial basis function $\psi_{\alpha=7\varepsilon_0,x=0}$ and the modal radial basis function $\phi_{\alpha=7\varepsilon_0}$ that was used to generate it on the interval $[-1, 1]$, with ε_0 the distance between two consecutive nodes. A rational basis function [10] is shown for comparison. While the nodal and rational basis functions are similar near the main support node, the nodal radial basis function quickly drops to zero away from this node. To this extent, the nodal radial basis functions resemble more compact functions like interpolets [4]

However, some radial basis functions are not ideal candidates to build nodal radial basis functions. For instance, Gaussian functions, that are infinitely smooth can trigger Runge's phenomenon [13] in the interpolation of the impulse function for width parameters relatively small, as shown in Figure 2. There is no Runge's phenomenon for Wendland functions, even for large width parameters. Since this work focuses on solving a partial differential equation, getting large oscillations near the boundary is problematic for two reasons. First, large oscillations are usually caused by ill-conditioned matrices, a well-known problem when working with radial basis functions, that will limit the precision of the interpolation or the differential equation solver. Second, these solvers are sensitive to boundary condition errors and, using functions that are widely oscillatory there would clearly be problematic. The number of sideband oscillations surrounding the main support

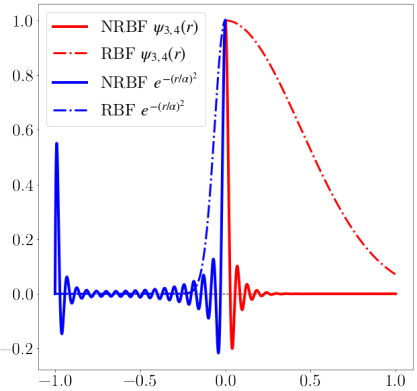
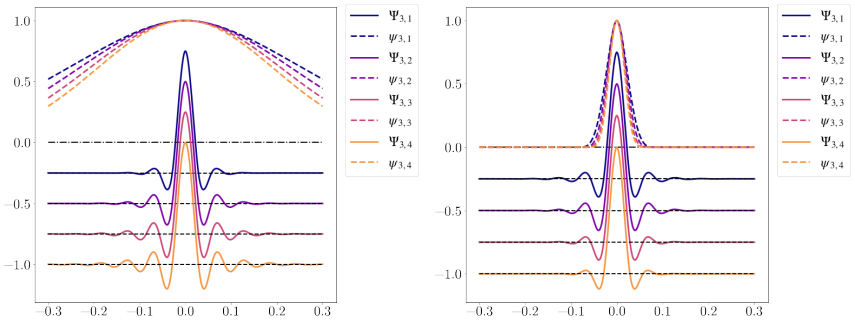


FIGURE 2. Left: The Gaussian nodal radial basis function is wide enough to trigger Runge's phenomenon in the nodal radial basis function at the edge of the domain. Right: Wendland functions (here $\psi_{3,4}$) yield nodal radial basis functions with no boundary instabilities.

node can be controlled by the degree of smoothness and the width parameters of the radial basis function. Figure 3-a shows that an increase in the smoothness of radial basis functions (using Wendland functions $\psi_{3,1}$ through $\psi_{3,4}$ leads to larger sideband oscillations, a direct consequence of a weaker exponential decay of scaling coefficients. However, this trend reverses if the width parameter is too small (as seen in Figure 3-b).



(a) (b)

FIGURE 3. Nodal radial basis functions $\Psi_{d,k}$ for Wendland functions $\psi_{d,k}$ of different degrees of smoothness k for width parameters a) $\alpha = 35\varepsilon_0$ and b) $\alpha = 3\varepsilon_0$. Here ε_0 is the smallest distance between neighboring nodes in U. The interval was truncated for clarity. The nodal functions are plotted using a solid line. A vertical offset was added to disentangle the oscillations of the function. The corresponding Wendland functions are plotted using dashed lines and were not given any offset. Note here the nodal function subscripts identify the Wendland functions rather than the geometrical scaling and translating parameters as it is done in the text.

This observation leads to the question of compactness. A priori, nodal radial basis functions are not compact. Even if $[\Phi_\alpha]$ is a banded matrix (i.e. non-zero elements are close to the diagonal), $[\Phi_\alpha]^{-1}$ is not necessarily banded and could even be dense. Figure 4 shows the scaling coefficients for several nodal radial basis functions with the same support node (0 in this case case). The logarithmic scale clearly shows the exponential decay of the coefficient away from the main support node. In some cases, the exponential decay is not constant, and it depends on the smoothness and width parameter. In fact, after the initial decay, the function rebounds. This rebound gets pushed further out as the width parameter increases (see Figure 4-a and b). At this point, boundary effects become dominant, leading to a weaker exponential decay. We notice that these trends tend to disappear as the function smoothness increases (see Figure 4-d). Yet, it is relatively easy to truncate the functions generated by radial basis functions with low smoothness. Truncation is simply enforced by dropping all scaling coefficients that are o(1). As the smoothness and width parameter increase, more coefficients should be retained. On small domains, like the one used in this section, truncation is not possible since there is no coefficient o(1) for $\psi_{3,4}$ when width parameters are larger than 15 nodes, as shown in Figure 4-d.

Figure 3 also shows that, after some initial variations, the shape of the nodal radial basis functions remains virtually the same as the width parameter increases. This is a clear departure from radial basis functions, where the width parameter clearly impacts the shape of the function. As the width parameter grows (and the shape parameters goes to 0), radial basis functions become remarkably flat, leading to better interpolants at the cost of ill-conditioned linear systems [29]. The resulting trade-off between interpolation accuracy and numerical stability is difficult

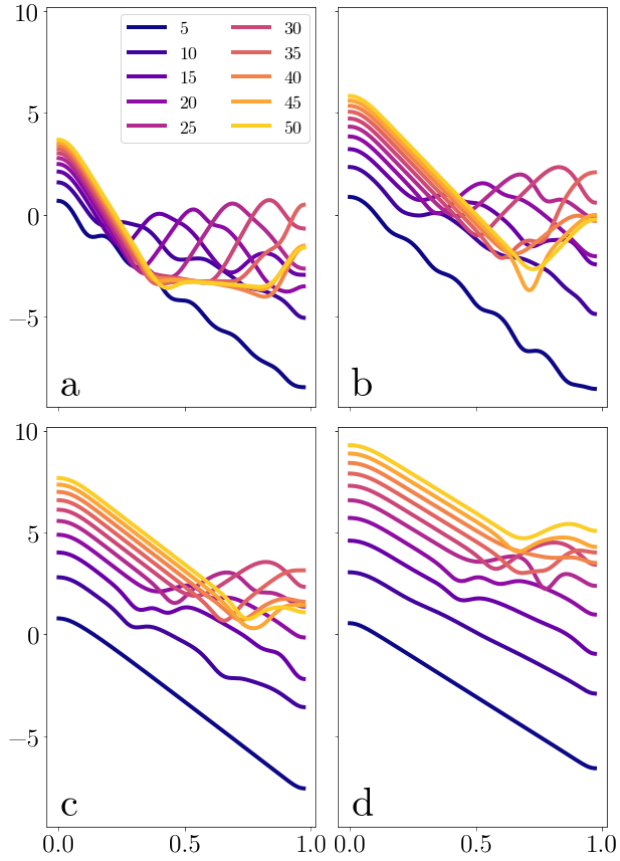


FIGURE 4. The $\log_{10}$ value of the scaling coefficients used to form nodal radial basis function centered on 0 and using a) $\psi_{3,1}$, b)$\psi_{3,2}$, c)$\psi_{3,3}$ and d) $\psi_{3,4}$ for different width parameters, given in number of nodes. Each coefficient is associated with a node location.

to quantify, even if qualitative arguments can be inferred from a wide range of studies (e.g. Ref. [8]). Nodal radial basis functions are less dependent on the width parameter than radial basis functions. Figure 5-a to c shows that when changing the width parameter by 10%, the maximum difference between nodal radial basis functions stays below 10^{-5}, for width parameters spanning 50 nodes. Under the same conditions, Figure 5-d to e shows the maximum change between consecutive radial basis functions is much larger than 10^{-2}. While not shown on the figure, this trend is also valid for basis functions centered on the domain boundaries. The shape of the nodal radial basis functions there differs from the shape of the functions in the domain interior, as shown in Figure 6. However, their construction is identical to the other functions and does not require special treatment. The NRBF method presented here is not bound to Wendland functions. However, the decays of the coefficients is crucial to suppress the Gibbs phenomenon seen on Fig. 2 and they should be studied carefully before using other types of radial basis functions [12], especially if they are infinitely smooth.

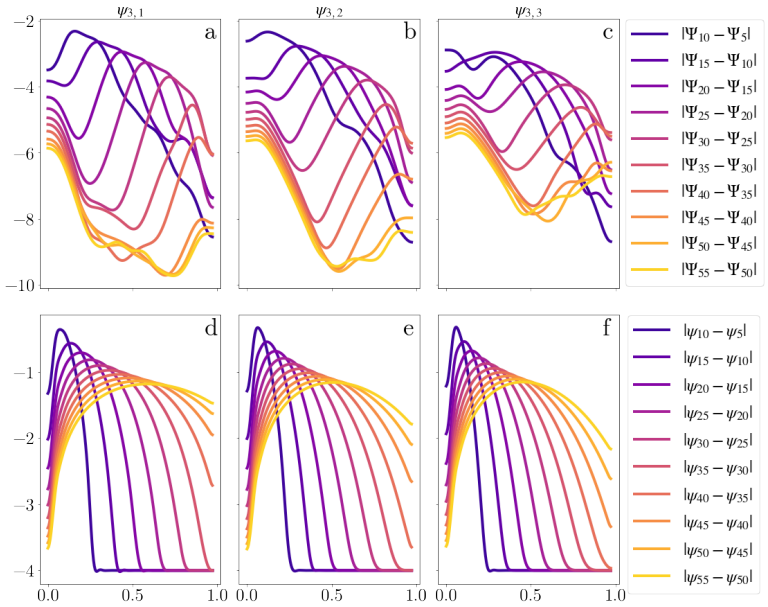


FIGURE 5. The $\log_{10}$ difference between nodal radial basis functions using a) $\psi_{3,1}$, b) $\psi_{3,2}$ and c) $\psi_{3,3}$ with consecutive width parameters (given in number of nodes). The difference was computed halfway between nodes since difference at the nodes is 0 by construction. The $\log_{10}$ difference between two consecutive radial basis functions c) $\psi_{3,1}$, b) $\psi_{3,2}$ and d) $\psi_{3,3}$, using 10^{-4} cut-off.

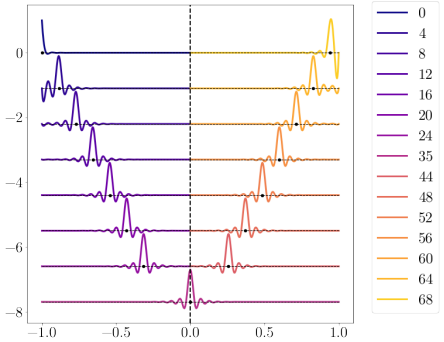


FIGURE 6. Nodal radial basis functions for different support nodes, indicated by a black dot). The total number of nodes is 71. The effect of the boundary, while dramatic, does not really depend on the width parameters for parameters larger than 50%. A bias was added to each function to improve the clarity of the plot.

4. The linear advection equation solved using Euler's backward method

Now that we have explored the basic properties of nodal radial basis functions, we would like to solve the following partial differential equation in multiple dimensions

$$\frac{\partial \rho}{\partial t} + \overrightarrow{\nabla}.(\rho \overrightarrow{u}) = S \tag{10}$$

on the discretized domain U. This equation is hyperbolic and describes mass conservation, where ρ is a mass density, $\overrightarrow{u}$ is the velocity and S is the source term. The backward Euler time discretization scheme is given by

$$\rho_t + \overrightarrow{\nabla}.(\rho_t \overrightarrow{u})\,\Delta t = \rho_{t-\Delta t} + S_t \Delta t.$$

It is convenient to drop the subscript term t and rewrite the equation as

$$\rho + \overrightarrow{\nabla}.(\rho \overrightarrow{u})\,\Delta t = \rho_{-\Delta t} + S\Delta t = G \tag{11}$$

where $\rho_{-\Delta t}$ is the solution at the previous time step.

4.1. Nodal radial basis function solver. Using the nodal radial basis functions defined earlier, we can interpolate the mass density ρ, the mass density flux $\rho\overrightarrow{u}$ in Eq. (11) as

$$\rho = \sum_j \rho_j \Psi_{\alpha,x_j},$$

and

$$\rho \overrightarrow{u} = \sum_j \rho_j \overrightarrow{u}_j \Psi_{\alpha,x_j}, \tag{12}$$

as well as

$$G = \sum_j G_j \Psi_{\alpha,x_j}.$$

Notice that we make no assumptions regarding the velocity distribution. It could be space and time dependent at this point. Using NRBFs in Eq. (11), we get

$$\sum_j \rho_j \Psi_{\alpha,x_j} + \sum_j \sum_k \rho_j u_{k,j} \partial_k \Psi_{\alpha,x_j} \Delta t = \sum_j G_j \Psi_{\alpha,x_j}, \tag{13}$$

where $u_{k,j}$ is the k^{th} component of the vector $\overrightarrow{u}$ and ∂_k is the derivative along the direction k. So, we get $\forall x_i \in U\ \sum_j \rho_j \Psi_{\alpha,x_j}(x_i) + \sum_j \rho_j \sum_k u_{k,j} \partial_k \Psi_{\alpha,x_j}(x_i)\Delta t = \sum_j G_j \Psi_{\alpha,x_j}(x_i)$. Using Eq. (4) we now have

$$\forall x_i \in U,\ \rho_i + \sum_j \rho_j \sum_k u_{k,j} \partial_k \Psi_{\alpha,x_j}(x_i)\Delta t = G_i. \tag{14}$$

Eq. (14) is valid for any number of spatial dimensions, is completely agnostic of the node distribution and can be written in matrix form

$$(I - \Delta t A)[\rho] = [G].$$

Here I is the identity matrix and the elements of A are

$$A_{\text{ij}} = -\sum_k u_{k,j} \partial_k \Psi_{\alpha,x_j}(x_i).$$

The solution to this system of equations is found by inverting the matrix $(I - \Delta t A)$.

$$[\rho] = (I - \Delta t A)^{-1}[G]. \tag{15}$$

Theorem 4. We can approximate the solution $[\rho]$ with $[\rho^*]$ using only matrix-vector products

$$[\rho^*] = \sum_{k=0}^{N} A[G_k]\Delta t^k, \tag{16}$$

where

$$\forall k > 0, [G_k] = A[G_{k-1}] \text{ and } A[G_0] = [G].$$

Proof. Since $(1-x)^{-1} = \sum_{k=0}^{\infty} x^k$, $(I-\Delta t A)^{-1}$ can be approximated by a truncated series for Δt small enough, and we get

$$(I - \Delta t A)^{-1} = \sum_{k=0}^{N} \Delta t^k A^k, \tag{17}$$

where $A^0 = I$. As a result, one can find the approximate solution ρ^* directly using

$$[\rho^*] = \sum_{k=0}^{N} A^k [G] \Delta t^k. \tag{18}$$

We also use the product of the matrix A by a vector v as $A(Av)$ rather than $(AA)v$ since it is more efficient to compute when the number of nodes is large. □

However, taking time steps small enough to warrant the approximation leading to Eq. (18) is not realistic in practice. However, if the velocity $\overrightarrow{u}$ and the source term S vary on a time scale ΔT, which is large compared to our choice of Δt, then the algorithm can "step over" the slow temporal change in source and density. Using Eq. (15) as our induction relation, we can start from a given solution at $t - \Delta T$ where ΔT is defined as $\Delta T = P\Delta t$. At this point, we can substantially reduce Δt while increasing P in such a way that the computational time step ΔT keeps the method stable. This method is implicit, and time stepping is not limited by the Courant–Friedrichs–Lewy (CLF) condition [21]. However, this method is still limited by a Nyquist-Shannon condition, as taking a time step ΔT much larger than the physical time evolution of the solution cannot capture the actual evolution of the solution.

Theorem 5. *We can approximate the solution at t using a large time step $\Delta T = P\Delta t$*

$$[\rho] = (I - \Delta t A)^{-P} [G_{-P\Delta t}] \tag{19}$$

where $[G_{-P\Delta t}] = [\rho_{-P\Delta t}] + \sum_{k=0}^{P-1} (I - \Delta t A)^k [S_{-P\Delta t}] \Delta t$.

Proof. Starting from Eq. (15) we get

$$[\rho] = (I - \Delta t A)^{-1} [\rho_{-\Delta t}] + (I - \Delta t A)^{-1} [S] \Delta t.$$

Since we suppose S constant during this interval of time we now have

$$[\rho] = (I - \Delta t A)^{-2} ([\rho_{-2\Delta t}] + [S] \Delta t + (I - \Delta t A) [S] \Delta t).$$

We find Eq. (19) by induction. □

Using $\sum_{k=0}^{P-1} (1-x)^k = \sum_{k=0}^{P-1} (-1)^k \binom{P}{k+1} x^k$ we get

$$[G_{-\Delta T}] = [\rho_{-\Delta T}] - \sum_{k=0}^{P-1} \left(-\frac{1}{P}\right)^{k+1} \binom{P}{k+1} A^k [S_{-\Delta T}] \Delta T^{k+1}.$$

Besides A^k and ΔT^{k+1}, $\left(\frac{1}{P}\right)^{k+1} \binom{P}{k+1}$ also decays quickly with k and we can truncate the finite series given above to its M first terms as

$$\left[G^*_{-\Delta T}\right] = \left[\rho^*_{-\Delta T}\right] - \sum_{k=0}^{M} \left(-\frac{1}{P}\right)^{k+1} \binom{P}{k+1} A^k [S_{-\Delta T}] \Delta T^{k+1}.$$

In the problem discussed in this paper, M is between 10 and 20, while P can be as large as 10^{10}. We can recast this series as a successive product of a matrix with a vector when necessary, as we did in Eq. (16). So far, we kept the source

term S for completion. We drop this term in the rest of the paper to simplify the discussion as it does not impact mathematical foundations of the method. Since $(1-x)^{-P} = \sum_{k=0}^{\infty} \binom{k+P-1}{P-1} x^k$, we can also write Eq. (19) as a truncated series where we keep the first N terms

$$[\rho^*] = \sum_{k=0}^{N} \left(\frac{1}{P}\right)^k \binom{k+P-1}{P-1} A^k \left[G_{-\Delta T}\right] \Delta T^k. \tag{20}$$

As for M, N can also be chosen between 10 and 20. Eq. (20) can be transformed into a successive product of a matrix by a vector, as was done in Eq. (16). It is important to note that we only need to know $\partial_k \Psi_{\alpha,x_j}(x_i)$ to solve this is problem. So, there is no need to compute the matrix inverse of $[\Phi_\alpha]$ to get the coefficients Ω_{ij}. Eq. 5 shows that we can compute $\partial_k \Psi_{\alpha,x_j}(x_i)$ by solving the linear system

$$[\Phi_\alpha][\partial_k \Psi_{\alpha,x_j}] = [\partial_k \Phi_\alpha]$$

using Cholesky's factorization, which does not require any explicit matrix inversion to solve Eq (33). So, the proposed solver uses neither matrix inversions nor matrix products in its final form. It is only based on matrix-vector multiplications, as other efficient solvers (e.g. Ref. [28]).

4.2. Radial basis function solver. We now focus on solving the same equation using radial basis functions and compared both methods. The mass density can be rewritten as

$$\rho = \sum_j \omega_{\rho,j} \Phi_{\alpha,x_j}. \tag{21}$$

If we suppose constant velocity across the whole domain, we have

$$\rho \overrightarrow{u} = \sum_j \omega_{\rho,j} \Phi_{\alpha,x_j} \overrightarrow{u}. \tag{22}$$

We will make this assumption in the remainder of this section. As we did before, we take G to be

$$G = \sum_j \omega_{G,j} \Phi_{\alpha,x_j}.$$

So, we get $\forall x_i \in U$

$$\sum_j \omega_{\rho,j} \Phi_{\alpha,x_j}(x_i) + \sum_j \omega_{\rho,j} \sum_k u_k \partial_k \Phi_{\alpha,x_j}(x_i) \Delta t = \sum_j \omega_{G,j} \Phi_{\alpha,x_j}(x_i). \tag{23}$$

We can rewrite this system in matrix form as

$$[\Phi_\alpha - \Delta t B][\omega_\rho] = [\Phi_\alpha][\omega_{G,-\Delta t}] \tag{24}$$

where B is the matrix derivative defined by

$$B_{\mathrm{ij}} = -\sum_k u_k \partial_k \Phi_{\alpha,x_j}(x_i).$$

If we want to use the larger time step $\Delta T = P \Delta t$, the procedure described previously applied to equation Eq. (24) gives

$$[\omega_\rho] = ([\Phi_\alpha - \Delta t B]^{-1}[\Phi_\alpha])^P [\omega_{G,-\Delta T}] \tag{25}$$

However, the inverse of the matrix $[\Phi_\alpha - \Delta tB]$ needs to be computed explicitly here, even if we use a series approximation to compute the matrix product $([\Phi_\alpha - \Delta tB]^{-1}[\Phi_\alpha])^P$. We can also rewrite Eq. (24) as

$$(I - \Delta tC)[\omega_\rho] = [\omega_G]$$

where $C = [\Phi_\alpha]^{-1}B$. In this case we get,

$$[\omega_\rho] = (I - \Delta tC)^{-P}[\omega_{\rho,-\Delta T}].$$

In this form, the system can be solved similarly to Eq. (19) and we can use the same procedure to obtain an equation like Eq. (20). However, the matrix inverse $[\Phi_\alpha]^{-1}$ needs to be computed explicitly, even when using a series approximation to compute $(I - \Delta tC)^{-1}$.

4.3. Comparison between the different solvers. Before embarking into a parameter scan to understand the limits of nodal radial basis functions as an implicit advection equation solver, we first would like to compare it to a solver using radial basis functions. Unless otherwise indicated, the radial basis function-based solvers (i.e. RBF and NRBF) used the Wendland function $\psi_{3,4}$, which guarantees strictly positive definiteness in up to three dimensions. We chose this function for its exceptional smoothness. Besides the RBF solver, we also compared the NRBF solver to a standard implicit solver, the second order centered implicit (CI) solver given by

$$\rho_i^{n+1} + \frac{u\Delta T}{12\Delta x}(-\rho_{i+2}^{n+1} + 8\rho_{i+1}^{n+1} - 8\rho_{i-1}^{n+1} + \rho_{i-2}^{n+1}) = \rho_i^n,$$

This equation can be turned into a matrix form leading to a form similar to Eq. (19). We also compared the method to the explicit Lax-Wendroff (LW) discretization scheme given by

$$\rho_i^{n+1} = \rho_i^n - \frac{\Delta T}{2\Delta x}u\left[\rho_{i+1}^n - \rho_{i-1}^n\right] + \frac{T^2}{2\Delta x^2}u^2\left[\rho_{i+1}^n - 2\rho_i^n + \rho_{i-1}^n\right] \tag{26}$$

Both solver are notoriously locally non-conservative, allowing to check our locally conservative NRBF approach. To obtain an absolute measure of the error, we compared all numerical solutions to a smooth solution of the advection equation Eq. (10) in one dimension with no source, namely

$$F_\sigma(x - x_0 - ut) = 1 + exp-\left[(x - x_0 - ut)/\sigma\right]^2,$$

traveling from the left to the right. We used the value of the solution F_σ as a Dirichlet boundary condition for the different methods, applied to three nodes to the left and right sides of the domain. Our initial condition also used the solution F_σ with its peak set on the left boundary.

Since Eq. (10) is dimensionless, we took the velocity u to be 1 and a domain [-2,2]. For this comparison, we discretized our domain with 501 nodes. The peak of F_σ starts at the left boundary node and travels to the right. As written, the function peak reaches the right boundary at t=4. The main reason for adding a constant to the function is to let the solution to relax to 1 after the Gaussian pulse traversed the domain, allowing to test the long-term stability of the different solvers for non-trivial solutions. While other methods can be used to solve this differential equation, we use here the same method across all three implicit schemes to compare all the three implicit solvers on the same footings.

Figure 7 shows the maximum error e_{max} between the true solution, given by Eq. (26), and the numerical solution, computed using different solvers. As expected, the Lax-Wendroff solver (dashed black line in Figure 7) comes last, with a linear

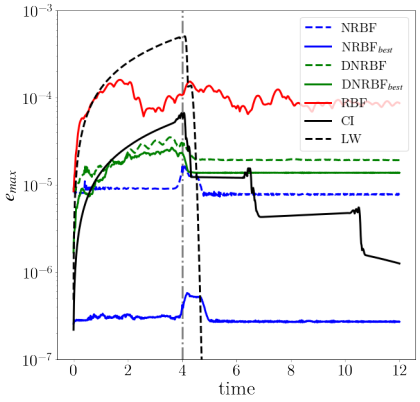


FIGURE 7. Comparison between the error of three different implicit methods using centered implicit finite differences (CI), radial basis functions (RBF), nodal radial basis functions using a series approximation for the matrix inversion (NRBF) or a direct solver (DNRBF). The nodal NRBF methods using a large width parameter are indicated using the subscript "best". The explicit Lax-Wendroff (LW) method is shown for reference. The vertical dot-dashed gray line corresponds to the time when the Gaussian peak reaches the right boundary of the domain.

increase of the error (seen as a logarithmic curve in the log plot of Figure 7). This solver is known to have large numerical viscosity. It is also subject to the CFL condition, forcing the time stepping to be much smaller than any implicit methods, requiring 4 times as many as steps and degrading the solution even further. The centered implicit solver (solid black line in Figure 3) does better than the explicit solver. But the error also increases linearly (also seen as a logarithmic curve in the log plot of Figure 7) throughout the computation, a sign that both solvers are not conservative.

The radial basis function solver (solid red in Figure 7) starts with an error similar to the Lax-Wendroff method but the error saturates rather than increasing linearly. While numerical fluctuations are visible throughout, they never turn unstable, keeping the error around 10^{-4}, even after the Gaussian pulse exited the domain. This error was obtained with a width parameter of 30 nodes (or 6% the total domain width). Larger parameters caused numerical instabilities. The NRBF solver, using the same width parameter as the RBF solver (dashed blue line in Figure 7), performs much better, with an error almost an order of magnitude smaller. Further, the error is virtually constant throughout the computation, an indication that the method is locally conservative, in the sense of Eq. (9). If it was not, the error would increase throughout the simulation. It is interesting to note that the quality of the solution comes mostly from computing the inverse of the matrix $(I-\Delta t A)^{-P}$ using a series approximation. When the matrix inverse is computed directly (D-NRBF, dashed green line in Figure 7), the error worsens noticeably. Note that both the DNRBF and the NRBF methods give the exact same answer when using $\psi_{3,3}$ instead of $\psi_{3,4}$, indicating that the poor results of the direct solvers truly come from roundoff errors in the inverse computed in Eq. (25). If we reduce the width parameter enough to limit numerical instabilities inside the RBF solver, then both solvers have the exact same error, independent of the matrix inversion method.

While the maximum value of the width parameter is problem dependent for both the nodal and standard radial basis function solvers, we cannot avoid matrix inversions for the latter, as Eq. (25) shows. Since the former solver uses a Cholesky decomposition, round-off errors are bounded, which allows a width parameter twice as large. After that, the NRBF method also becomes crippled by round-off errors. While we show here the best case scenario, an increase in the width parameter does not yield necessarily a better solution when the inverse of the matrix $(I - \Delta t A)^{-P}$ is computed directly (DNRBF$_{best}$, solid green line in Figure 7). But there is a clear improvement when the series approximation is used (NRBF$_{best}$, solid blue line in Figure 7), with a reduction of the error by more than an order of magnitude compared to all other solvers.

5. Accuracy of the implicit nodal radial basis function solver

Nodal radial basis functions can solve the linear advection equation with greater precision compared to other standard methods. However, these functions are defined implicitly, and it would be difficult to determine the impact of the different parameters on the quality of the solution. Yet, we have just seen that time stepping, the smoothness of the radial basis functions and possibly the presence of a boundary can affect the solution. In this section we are now in a position to use the 'no boundary' condition [26, 16] as an open boundary condition at the right boundary. This condition was not used in the previous section since the explicit finite difference scheme needed boundary values at both boundaries and we wanted to treat all boundary conditions on the same footing.

Now, the impact of time stepping is often an issue for computational methods. While the method presented here is implicit and is not subject to the CFL condition, it can only support a time stepping ΔT that is three times the ΔT_{CFL} , the time step given by the CFL, when using $\psi_{3,4}$ and four times the ΔT_{CFL} when using $\psi_{3,3}$. This limit can be increased further by using more points outside the boundaries.

At present, it is not possible to define this limit precisely since the nodal radial basis functions are not explicitly formed. However, we can look in greater details at the sub-cycle timestep Δt. Figure 8 shows how the sub-cycling (i.e the ratio $\Delta T/\Delta t$) impacts the error for nodal radial basis functions formed by the smoothest Wendland functions used in this paper, namely $\psi_{3,4}$. When the sub-cycle ratio is small, the solver is not conservative, and the error builds up linearly. In this case, the average error of a simulation (solid line) is close to the maximum error of that simulation (dashed line), which is also the end error, while the minimum error (dotted line) is orders of magnitude smaller than the average error. If we were to plot this error with time it would behave like the error of the implicit centered method.

While the error steadily diminishes with larger ratios, the method is still not conservative until increasing the ratio does not yield better error. Once the error has settled, we see that there is very little difference between the minimum and maximum error, indicating that the method is now locally conservative and the average error stays close to the minimum error. When plotted against time in Figure 7, the error would be roughly flat throughout the simulation. For small width parameters, this error quickly becomes insensitive to the ratio $\Delta T/\Delta t$ as it gets dominated by interpolation error, controlled by the width parameter. It is remarkable that the method is still locally conservative despite the large error. As the width parameter becomes larger though, the kink in the error curve is pushed to higher values of $\Delta T/\Delta t$ and the error diminishes steadily.

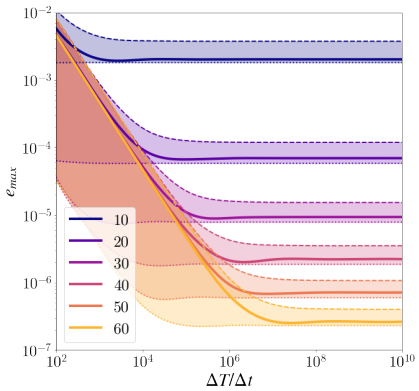


FIGURE 8. Error for different ratios $\Delta T/\Delta t$, computed for different width parameters, given in number of nodes. The solid lines give the average error throughout the best NRBF simulations for different width parameters, while the shaded region gives the error bracket, bounded by the minimum (dotted line) and maximum (dashed line) errors. Typically, when the average error is close to the minimum error, then the method is locally conservative. In this case the maximum error comes from the bump caused by the solution going through the boundary. When the average error is close to the maximum error, then the solver is not conservative and lead to a linear increase of the error, as seen in centered implicit or Lax-Wendroff methods. The size of the width parameter is given as a function of half the number of nodes spanned by the modal radial basis function. The domain has a total of 501 nodes.

As we saw in Figure 3, the width parameter and the smoothness greatly change the way the function couples the neighboring nodes to the main support node. Figure 9-a shows that their impact is dramatic. When the width parameter is small, the function smoothness does not impact the error at all and the weak coupling between nodes (narrow stencil) yield a relatively large error. As the width parameter becomes large and more nodes are coupled, the error strongly depends on the function smoothness and the width parameter. In the extreme case of $\psi_{3,4}$, increasing the width parameter six times reduces the error by four orders of magnitude. However, despite the large error, the method remains locally conservative, showing a small error variation throughout the simulation. We see that the error starts to saturate for even larger width parameters, since the shape of the nodal radial basis functions becomes independent of the shape parameter. As the coupling between nodes initially increases with large width parameters, the number of nodes that need to be added outside of the domain should also increase. This is necessary so boundary conditions can be imposed with a spatial order that is consistent with the spatial order of the method. Figure 9-b shows that the error does improve with more nodes located outside of the domain, but this change is weakly dependent of on the number of boundary nodes. Since we are looking at a particular solution of the partial differential equation, which is relatively smooth, and the error is mostly driven by the width parameter and the function smoothness, the increase in the number of nodes does not provide a better approximation of the solution, as shown in Figure 10-a. However, numerical instabilities start to become problematic if the

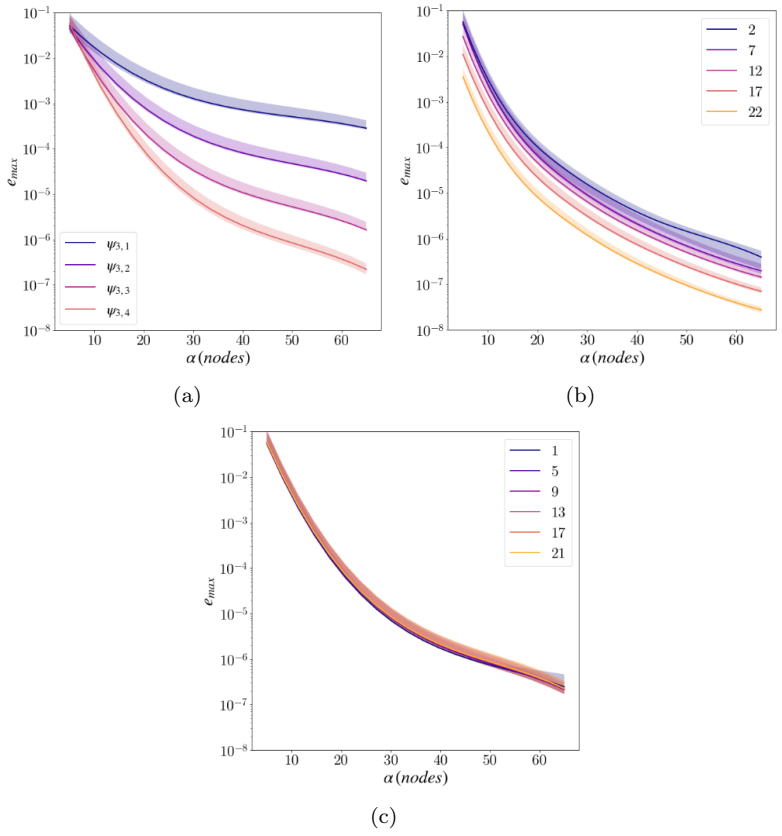


FIGURE 9. Error for different width parameters (given as number of nodes) for a) for the Wendland functions $\psi_{3,1}$, $\psi_{3,2}$, $\psi_{3,3}$, and $\psi_{3,4}$, b) a different number of boundary nodes and c) different Gaussian pulse widths σ, defined in Eq. (26), both for the Wendland function $\psi_{3,4}$. The solid lines give the average error throughout the best NRBF simulations, while the shaded region gives the error bracket. The domain has 501 nodes.

width parameter is too large, forcing the error to increase. Since we have an excellent approximation with fewer nodes, we can generate a random grid distribution with larger variations (up to +/-30% from the homogeneous node locations) to see the impact of the random distribution on the overall quality of the solution. Here we let the simulation runs until t=10 to verify that the solver reaches the steady state solution (1 according to Eq. (26)) and stays stable throughout the simulation. While the error increases noticeably, Figure 10-b shows that the error is bounded and remains reasonable even when nodes are displaced substantially.

As we discussed earlier, nodal radial basis functions can be truncated. Figure 10-c shows the nodal radial basis functions are not conservative for extreme truncation. However, as we reduced sparsity, the method becomes locally conservative and remains such until the minimum sparsity has been reached. At this point, we simply stopped plotting the curve. Note that the sparsity measurement is not absolute. As we saw in Figure 10-a, the precision comes from the total number

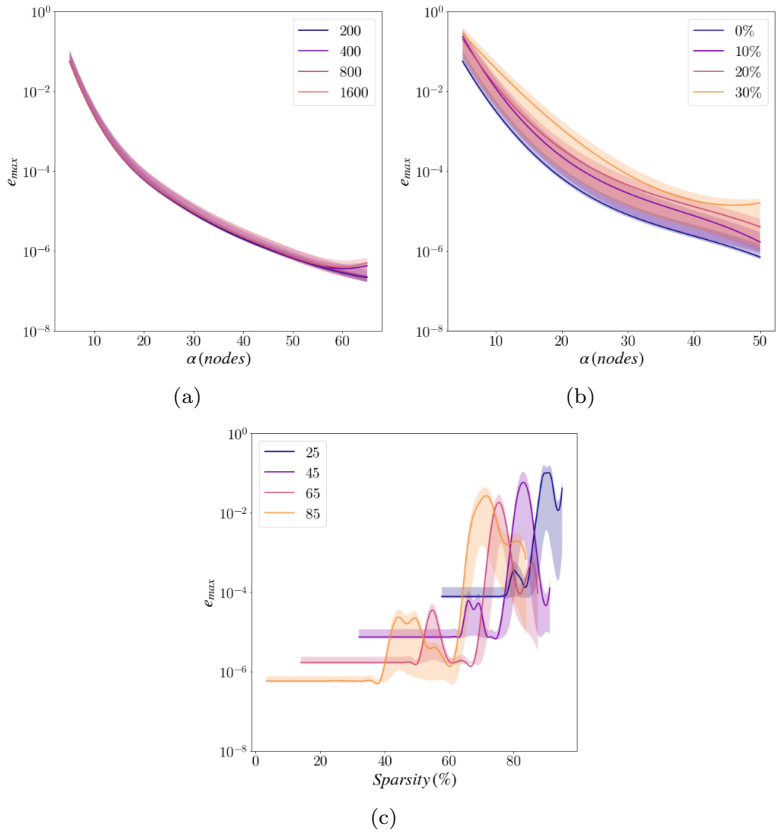


FIGURE 10. a) Using $\psi_{3,4}$ with 200, 400, 800 and 1600 nodes and b) with 200 nodes randomly distributed by 0%, 10%, 20% and 30% compared to the homogeneous internode distribution distance. c) Error caused by truncation of the nodal radial basis function for different width parameters (in number of nodes) for $\psi_{3,3}$. The curve stops where the matrix is sparse without truncation.

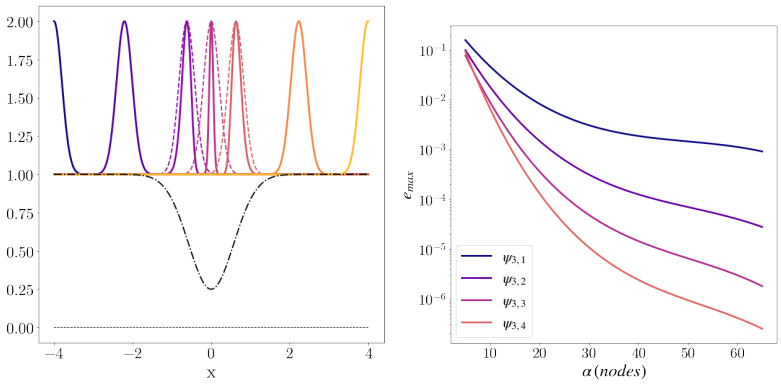
of nodes contained inside the nodal radial basis function, which is controlled by the width parameter, rather than the total number of points contained inside the domain. When solving real problems, the total number of points inside the domain will increase while the precision is still dictated by the width parameter and the function smoothness. So, the sparsity will increase drastically for a given precision.

6. The advection equation with non-homogeneous velocity.

As an example, we can solve Eq. (10) with the following velocity distribution

$$u_{\gamma,\sigma}(x - x_c) = 1 - \gamma exp - [(x - x_c)/\sigma]^2,$$

where we took x_c to be the domain center node, γ the damping gain and σ the velocity distribution width. We chose σ small enough so the velocity at both boundaries is 1 and the discretization can sample $u_{\gamma,\sigma}$ relatively well. We used here the domain [-4,4] to guarantee that the velocity distribution is virtually 1 at both boundaries. The velocity distribution used here is shown in Figure 11-a. The solution of Eq. (10) is compressed in regions where $u_{\gamma,\sigma}$ decreases and stretched in regions where



(a) (b)

FIGURE 11. a) The solution (solid line) to the differential equation as a function of time from boundary to boundary, together with the Gaussian pulse following a ballistic trajectory (dashed line) and the velocity distribution across the domain (dot-dashed line). b) The maximum error between the ballistic solution and the numerical solution at the right boundary for the Wendland functions $\psi_{3,1}$, $\psi_{3,2}$, $\psi_{3,3}$, and $\psi_{3,4}$ with different width parameters (given in number of nodes). The domain has 501 nodes. The left boundary has Dirichlet's conditions. The right boundary has a 'no boundary' condition.

$u_{\gamma,\sigma}$ increases. The Gaussian pulse of Eq. (26) was used as initial condition and as a Dirichlet condition of the left boundary throughout the simulation. The right boundary here is open (i.e. the 'no boundary' boundary condition). Figure 11-a shows the numerical solution as it moves from the left to the right.

If our Gaussian pulse was to pass through the domain with a ballistic trajectory, the location x_t of the peak would be given by

$$x(t) - x\left(t_0\right) = \int_{t_0}^{t} u_{\gamma,\sigma}\left(x(t') - x_c\right) \mathrm{d}t'.$$

Because the velocity distribution is symmetric with respect to x_c, the Gaussian pulse following the ballistic trajectory is also a solution of Eq. (10), but only in regions where $u_{\gamma,\sigma} = 1$. The ballistic pulse traversing the domain is also shown in Figure 11-a. As we can see, it is indistinguishable from the numerical solution at the periphery of the domain.

We compared the numerical solution against the ballistic pulse at t_{final}, when both peaks are located at the right boundary. Figure 11-b shows that the maximum error is comparable to the case with constant velocity, and shown in Figure 9-a. The other errors were also similar and were not included to the paper. There is no minimum and maximum error bracket here since we looked at the final simulation time t_{final} and not at the whole time series. We have not computed the error near x_c with this method.

7. Conclusion

This paper shows how radial basis functions can be used to construct implicitly a family of nodal radial basis functions on a discrete set U of nodes, which

are interpolant of the translated impulse function $\delta(x - x_j)$. Unlike radial basis functions, which are translated and scaled version of a single modal function, the nodal radial basis functions depend on the node distribution. These functions form an orthonormal basis on $\overline{U}$, the space of interpolant operating on U, leading to a simplified expression of the solver obtained when discretizing the linear advection equation. This solver can be extended trivially to the case where the velocity varies across the whole domain.

One advantage of the nodal radial basis function method over the radial basis function method is easily imposing boundary conditions. In general, boundary conditions are given in term of the solution (Dirichlet) or its derivatives (Neumann). Yet, the radial basis function solver computes the solution in term of weights rather than the actual solution values, using Eq. (23). So, at every time step, the solution has to be computed at the domain nodes using Eq. (21), the boundary conditions have to be applied, and then, the new weights have to be computed using Eq. (1).

One possible issue with nodal radial basis functions comes from its computational complexity. The Cholesky decomposition is $\mathrm{O}(\mathrm{N}^3)$, followed by N computations of the nodal radial basis function derivatives, each $\mathrm{O}(\mathrm{N}^2)$, to form the matrix A. So, we face another computation complexity that is $\mathrm{O}(\mathrm{N}^3)$, which is not present when using radial basis functions. Each time step is $\mathrm{O}(\mathrm{N}^2)$ after that. When the time evolution of the equation requires a number of time steps that is larger than N, the nodal radial basis function becomes more advantageous. Indeed, we would need to go back and forth between the actual value of the solution and its weight to impose boundary conditions using radial basis functions, a transformation requiring $\mathrm{O}(\mathrm{N}^2)$ operations.

The $\mathrm{O}(\mathrm{N}^3)$ dependence is problematic compared to the centered implicit and Lax-Wendroff methods. But there are also simple remedies to this ailment. Global nodal radial basis functions, as the one used in this paper, can be truncated easily, leading to a computational complexity that is O(N) where one node is connected to a limited set of neighboring nodes, as opposed to all the nodes in the set leading to an $\mathrm{O}(\mathrm{N}^2)$ dependance. We can also use a partition of unity approach [1], similar to the one used with radial basis functions [34], also leading to an O(N) scaling.

While the present work mostly used global nodal radial basis functions, it showed that truncating global nodal radial basis functions can lead to sparse matrix algebra. It can be extended relatively easily to partition-of-unity methods [15, 35], to reduce the computational complexity from $\mathrm{O}(\mathrm{N}^2)$ to a O(N). This work was done in relatively ideal conditions, staying away from solutions with sharp gradients naturally arising from hyperbolic PDEs, which we plan to investigate further using adaptive techniques. While adaptive meshing is not easy to implement, it clearly does not conflict with the methods presented herein and will be explored in future works.

Acknowledgements

One of us (PAG) would like to thank Carter Ball and Carlos Cabrera with setting up some simulations used in this paper. This research was supported in part by the NSF Awards PHY-1725178, PHY-1943939 and the Horton Fellowship from the laboratory for laser energetics.

References

[1] Babuška, I., and Melenk, J. M. The partition of unity method. International Journal for Numerical Methods in Engineering 40, 4 (1997), 727–758.

[2] Buhmann, M. Multivariate cardinal interpolation with radial-basis functions. Constructive Approximation 6, 3 (1990), 225–255.
[3] Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C., and Evans, T. R. Reconstruction and representation of 3d objects with radial basis functions. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques (2001), pp. 67–76.
[4] Deslauriers, G., and Dubuc, S. Symmetric iterative interpolation processes. In Constructive approximation. Springer, 1989, pp. 49–68.
[5] Fasshauer, G. RBF collocation methods as pseudospectral methods. WIT Transactions on Modelling and Simulation 39 (2005).
[6] Fasshauer, G. E. Solving differential equations with radial basis functions: multilevel methods and smoothing. Advances in Computational Mathematics 11, 2 (1999), 139–159.
[7] Fasshauer, G. E. Meshfree approximation methods with MATLAB, vol. 6. World Scientific, 2007.
[8] Fasshauer, G. E., and Zhang, J. G. On choosing optimal shape parameters for RBF approximation. Numerical Algorithms 45, 1 (2007), 345–368.
[9] Fasshauer, G. E., and Zhang, J. G. Preconditioning of radial basis function interpolation systems via accelerated iterated approximate moving least squares approximation. In Progress on meshless methods. Springer, 2009, pp. 57–75.
[10] Floater, M. S., and Hormann, K. Barycentric rational interpolation with no poles and high rates of approximation. Numerische Mathematik 107, 2 (2007), 315–331.
[11] Flyer, N., and Wright, G. B. A radial basis function method for the shallow water equations on a sphere. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 465, 2106 (2009), 1949–1976.
[12] Fornberg, B., Flyer, N., Hovde, S., and Piret, C. Locality properties of radial basis function expansion coefficients for equispaced interpolation. IMA Journal of Numerical Analysis 28, 1 (2008), 121–142.
[13] Fornberg, B., and Zuev, J. The Runge phenomenon and spatially variable shape parameters in rbf interpolation. Computers & Mathematics with Applications 54, 3 (2007), 379–398.
[14] Franke, R. Scattered data interpolation: tests of some methods. Mathematics of Computation 38, 157 (1982), 181–200.
[15] Griebel, M., and Schweitzer, M. A. A particle-partition of unity method for the solution of elliptic, parabolic, and hyperbolic PDEs. SIAM Journal on Scientific Computing 22, 3 (2000), 853–890.
[16] Griffiths, D. F. The no boundary condition outflow boundary condition. International Journal for Numerical Methods in Fluids 24, 4 (1997), 393–411.
[17] Hardy, R. L. Multiquadric equations of topography and other irregular surfaces. Journal of Geophysical Research 76, 8 (1971), 1905–1915.
[18] Kansa, E. Multiquadrics-A scattered data approximation scheme with applications to computational fluid-dynamics-I Surface approximations and partial derivative estimates. Computers and Mathematics with Applications 19, 8-9 (1990), 127 -145.
[19] Kansa, E. Multiquadrics-A scattered data approximation scheme with applications to computational fluid-dynamics-II Solutions to parabolic, hyperbolic and elliptic partial differential equations. Computers and Mathematics with Applications 19, 8-9 (1990), 147 -161.
[20] Karageorghis, A., Tappoura, D., and Chen, C. The Kansa RBF method with auxiliary boundary centres for fourth order boundary value problems. Mathematics and Computers in Simulation 181 (2021), 581–597.
[21] Lewy, H., Friedrichs, K., and Courant, R. Über die partiellen differenzengleichungen der mathematischen physik. Mathematische Annalen 100 (1928), 32–74.
[22] Lin, J., Bai, J., Reutskiy, S., and Lu, J. A novel RBF-based meshless method for solving time-fractional transport equations in 2d and 3d arbitrary domains. Engineering with Computers (2022), 1–18.
[23] Liu, X. Radial point collocation method (RPCM) for solving convection-diffusion problems. Journal of Zhejiang University-SCIENCE A 7, 6 (2006), 1061–1067.
[24] Liu, X., Liu, G., Tai, K., and Lam, K. Radial point interpolation collocation method (RPICM) for partial differential equations. Computers & Mathematics with Applications 50, 8-9 (2005), 1425–1442.
[25] Mirzaee, F., and Samadyar, N. Combination of finite difference method and meshless method based on radial basis functions to solve fractional stochastic advection–diffusion equations. Engineering with Computers 36, 4 (2020), 1673–1686.

[26] Papanastasiou, T. C., Malamataris, N., and Ellwood, K. A new outflow boundary condition. International Journal for Numerical Methods in Fluids 14, 5 (1992), 587–608.
[27] Reutskiy, S. Y. A method of particular solutions for multi-point boundary value problems. Applied Mathematics and Computation 243 (2014), 559–569.
[28] Saad, Y., and Schultz, M. H. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM Journal on Scientific and Statistical Computing 7, 3 (1986), 856–869.
[29] Schaback, R. Error estimates and condition numbers for radial basis function interpolation. Advances in Computational Mathematics 3, 3 (1995), 251–264.
[30] Shivanian, E. A new spectral meshless radial point interpolation (SMRPI) method: a well-behaved alternative to the meshless weak forms. Engineering Analysis with Boundary Elements 54 (2015), 1–12.
[31] Wang, J., and Liu, G. A point interpolation meshless method based on radial basis functions. International Journal for Numerical Methods in Engineering 54, 11 (2002), 1623–1648.
[32] Wendland, H. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. Advances in Computational Mathematics 4, 1 (1995), 389–396.
[33] Wright, G. B., Flyer, N., and Yuen, D. A hybrid radial basis function - pseudospectral method for thermal convection in a 3D spherical shell. Geochem. Geophys. Geosyst. 11 (2010), Q07003.
[34] Yokota, R., Barba, L. A., and Knepley, M. G. Petrbfa parallel o(n) algorithm for radial basis function interpolation with gaussians. Computer Methods in Applied Mechanics and Engineering 199, 25-28 (2010), 1793–1804.
[35] Yokota, R., Barba, L. A., and Knepley, M. G. Petrbfa parallel o(n) algorithm for radial basis function interpolation with gaussians. Computer Methods in Applied Mechanics and Engineering 199, 25-28 (2010), 1793–1804.
[36] Zamolo, R., and Nobile, E. Numerical solution of heat conduction problems by means of a meshless method with proper point distributions. In Proceedings of CHT-17 ICHMT International Symposium on Advances in Computational Heat Transfer (2017), Begel House Inc.
[37] Zhang, Y., Lin, J., Reutskiy, S., Sun, H., and Feng, W. The improved backward substitution method for the simulation of time-dependent nonlinear coupled burgers equations. Results in Physics 18 (2020), 103231.

Physics and Astronomy Department, University of Rochester, Rochester, New York, 14627, USA
E-mail: gourdain@pas.rochester.edu