SIAM J. SCI. COMPUT.
Vol. 43, No. 5, pp. A3244–A3268

EFFICIENT SCALING AND MOVING TECHNIQUES FOR SPECTRAL METHODS IN UNBOUNDED DOMAINS*

MINGTAO XIA†, SIHONG SHAO‡, AND TOM CHOU§

Abstract. When using Laguerre and Hermite spectral methods to numerically solve PDEs in unbounded domains, the number of collocation points assigned inside the region of interest is often insufficient, particularly when the region is expanded or translated in order to safely capture the unknown solution. Simply increasing the number of collocation points cannot ensure a fast convergence to spectral accuracy. In this paper, we propose a scaling technique and a moving technique to adaptively cluster enough collocation points in a region of interest in order to achieve fast spectral convergence. Our scaling algorithm employs an indicator in the frequency domain that both is used to determine when scaling is needed and informs the tuning of a scaling factor to redistribute collocation points in order to adapt to the diffusive behavior of the solution. Our moving technique adopts an exterior-error indicator and moves the collocation points to capture the translation. Both frequency and exterior-error indicators are defined using only the numerical solutions. We apply our methods to a number of different models, including diffusive and moving Fermi–Dirac distributions and nonlinear Dirac solitary waves, and demonstrate recovery of spectral convergence for time-dependent simulations. A performance comparison in solving a linear parabolic problem shows that our frequency scaling algorithm outperforms the existing scaling approaches. We also show our frequency scaling technique is able to track the blowup of average cell sizes in a model for cell proliferation. In addition to the Laguerre and Hermite basis functions with exponential decay at infinity, we also successfully apply the frequency-dependent scaling technique into rational basis functions with algebraic decay at infinity.

Key words. spectral method, unbounded domain, scaling, moving mesh, Laguerre function, Hermite function, rational basis function, blowup

AMS subject classifications. 65M70, 65F35, 65M50, 33C45, 41A05

DOI. 10.1137/20M1347711

1. Introduction.

Many scientific models described by PDEs with divergent solutions are set in unbounded domains. For example, in many models of cellular proliferation, a "blowup" in which the average size of a population of cells becomes uncontrolled and diverges over many generations of growth is possible [3]. The conditions under which blowup occurs is difficult to determine analytically [1] but has been explored numerically [25]. However, numerically tracking "blowup" behavior over long times is extremely difficult, as it requires solving the problem in a truly unbounded domain to capture the diverging mean size. There are many other problems in which it is desirable to find a numerical solution in an unbounded domain, including the stability of solitary waves arising from the nonlinear Dirac equation [14, 6], diffusion

*Submitted to the journal's Methods and Algorithms for Scientific Computing section June 23, 2020; accepted for publication (in revised form) June 2, 2021; published electronically September 20, 2021.

https://doi.org/10.1137/20M1347711

Funding: The work of the first and third authors was supported by the National Science Foundation through grant DMS-1814364 and by the Army Research Office through grant W911NF-18-1-0345. The work of the second author was supported by the National Natural Science Foundation of China through grant 11822102 and by the Beijing Academy of Artificial Intelligence (BAAI) and the computational resources provided by the High-performance Computing Platform of Peking University.

†Department of Mathematics, UCLA, Los Angeles, CA 90095-1555 USA, and LMAM and School of Mathematical Sciences, Peking University, Beijing 100871, China (xiamingtao97@g.ucla.edu).

‡Corresponding author. HEDPS, Center for Applied Physics and Technology, LMAM and School of Mathematical Sciences, Peking University, Beijing 100871, China (sihong@math.pku.edu.cn).

§Department of Mathematics, UCLA, Los Angeles, CA 90095-1555 USA (tomchou@ucla.edu).

in a parabolic system [10], and solving a fractional PDE which allows a solution with algebraic decay at infinity [28, 23].

Considerable progress has recently been made using spectral methods for solving PDEs in unbounded domains [17]. Among existing spectral methods, the direct approach that is typically used is based on orthogonal basis functions defined on infinite intervals, e.g., the Hermite and Laguerre spectral methods [5, 7, 23]. Rational basis functions such as the modified mapped Gegenbauer functions (MMGFs) [22] are of recently interest and can also be used. It has been demonstrated that the performance of these spectral methods can be greatly improved when a proper coordinate scaling is used [21, 17]. However, it is not clear how to systematically perform the scaling, especially when transient behavior arises. A Hermite spectral method with time-dependent scaling has been proposed for parabolic problems by introducing a time-dependent scaling factor $\beta(t)$ to meet the coercive condition [10]. Nonetheless, the form of $\beta(t)$ and related parameters are chosen based on specified knowledge of parabolic models and thus cannot be easily generalized to other problems.

Motivated by the success of adaptive methods in bounded domains [13, 20, 9], we propose two indicators to adaptively allocate a sufficient number of collocation points to represent the unknown solution in the region of interest. The first indicator, designed for matching the diffusion of unknown solutions, extracts the frequency-space information of intermediate numerical solutions and isolates its high frequency components. This frequency indicator not only provides a lower bound for the interpolation error but also measures the decay of the derivatives of the reference solution as $|x| \to +\infty$. By tuning a scaling factor in our proposed scaling technique, the frequency indicator can be maintained at a low level. However, the translation of unknown solutions may also amplify the frequency indicator and thus may result in larger errors for excessive scaling. To accommodate this scenario, a second, exterior-error indicator is used to calculate an upper bound for the error in the exterior domain, allowing one to capture translation via moving collocation points. Accordingly, for problems that may involve both translation and diffusion in unbounded domains, the above two indicators are combined in a "first moving then scaling" approach. Numerical experiments demonstrate their ability to recover a faster spectral convergence for time-dependent solutions.

The remainder of this paper is organized as follows. Section 2 introduces the frequency indicator, connects it to the approximation error, and proposes the frequency-dependent scaling technique for diffusion. Section 3 proposes the exterior-error-dependent moving technique for translating problems. We then combine, in section 4, the above two approaches to solve time-dependent problems involving both diffusion and translation. Section 5 compares the frequency-dependent scaling with a time-dependent scaling proposed in [10] for solving parabolic systems. Section 6 generalizes the scaling technique to MMGFs which characterize algebraic decay at infinity. Section 7 analyzes the efficiency of both scaling and moving techniques and discusses their dependence on parameters. In section 8, we apply the frequency-dependent scaling method to a PDE model describing structured cell populations to track blowup behavior. Finally, we summarize our approaches and make concluding remarks in section 9.

2. Frequency-dependent scaling. We formulate a scaling technique by extracting frequency domain information on the evolution of numerical solutions, the pseudocode of which is presented in Algorithm 2.1. Our discussion utilizes the generalized Laguerre functions of degree ℓ

$$\hat{\mathcal{L}}_\ell^{(\alpha,\beta)}(x) = \mathcal{L}_\ell^{(\alpha)}(\beta x)e^{-\frac{\beta}{2}x}, \quad \beta > 0, \tag{2.1}$$

which are mutually orthogonal on the half-line $\Lambda := (0, +\infty)$ with the weight function $\hat{\omega}_\alpha(x) = x^\alpha, \alpha > -1$. Here $\mathcal{L}_\ell^{(\alpha)}(x)$ denote the usual Laguerre polynomials [7], and $\hat{\mathcal{L}}_\ell^{(\alpha,\beta)}(x)$ reduce to the usual Laguerre functions when $\beta = 1$. In this work, we regard β to be the *scaling factor* and seek a time-dependent spectral approximation of $u(x,t)$ on Λ. Henceforth, for notational simplicity, the t-dependence will usually be omitted.

For any $u \in L^2_{\hat{\omega}_\alpha}(\Lambda)$, the spectral approximation using the interpolation operator $\mathcal{I}_{N,\alpha,\beta}$ is

$$u(x) \approx U_N^{(\alpha,\beta)}(x) = \mathcal{I}_{N,\alpha,\beta}u = \sum_{\ell=0}^{N} u_\ell^{(\alpha,\beta)} \hat{\mathcal{L}}_\ell^{(\alpha,\beta)}(x), \tag{2.2}$$

where the coefficients $u_\ell^{(\alpha,\beta)}$ can be computed by using, e.g., the Laguerre–Gauss collocation points $x_j^{(\alpha,\beta)}$,

$$u_\ell^{(\alpha,\beta)} = \frac{1}{\gamma_\ell^{(\alpha,\beta)}} \sum_{j=0}^{N} \hat{\mathcal{L}}_\ell^{(\alpha,\beta)}\left(x_j^{(\alpha,\beta)}\right) u\left(x_j^{(\alpha,\beta)}\right) \hat{w}_j^{(\alpha,\beta)}, \quad \ell = 0, 1, \ldots, N, \tag{2.3}$$

where N is the expansion order (i.e., $N+1$ collocation points or $N+1$ basis functions), $\gamma_\ell^{(\alpha,\beta)} = (\hat{\mathcal{L}}_\ell^{(\alpha,\beta)}, \hat{\mathcal{L}}_\ell^{(\alpha,\beta)})_{\hat{\omega}_\alpha}$ is the $L^2_{\hat{\omega}_\alpha}$ inner product, $\hat{w}_j^{(\alpha,\beta)}$ denotes the corresponding weight for collocation point $x_j^{(\alpha,\beta)}$, and

$$u\left(x_j^{(\alpha,\beta)}\right) = U_N^{(\alpha,\beta)}\left(x_j^{(\alpha,\beta)}\right) = \mathcal{I}_{N,\alpha,\beta}u\left(x_j^{(\alpha,\beta)}\right), \quad j = 0, 1, \ldots, N. \tag{2.4}$$

When the scaling factor is updated from β to $\tilde{\beta}$, the collocation points, weights, and $L^2_{\hat{\omega}_\alpha}$-norms are updated according to

$$x_j^{(\alpha,\tilde{\beta})} = \frac{\beta}{\tilde{\beta}} x_j^{(\alpha,\beta)}, \quad \hat{w}_j^{(\alpha,\tilde{\beta})} = \frac{\beta^{\alpha+1}}{\tilde{\beta}^{\alpha+1}} \hat{w}_j^{(\alpha,\beta)}, \quad \gamma_\ell^{(\alpha,\tilde{\beta})} = \frac{\beta^{\alpha+1}}{\tilde{\beta}^{\alpha+1}} \gamma_\ell^{(\alpha,\beta)}. \tag{2.5}$$

The expansion coefficients $u_\ell^{(\alpha,\tilde{\beta})}$ can then be estimated through (2.3) where we may use the approximation (2.2): $u(x_j^{(\alpha,\tilde{\beta})}) \approx U_N^{(\alpha,\beta)}(x_j^{(\alpha,\tilde{\beta})})$. This procedure constitutes the SCALE subroutine in lines 9 and 17 of Algorithm 2.1.

To implement the scaling technique, one needs to determine when to apply it and how to choose a new scaling factor $\tilde{\beta}$ such that spectral accuracy can be kept for a prescribed expansion of order N. To this end, we propose a *frequency indicator* acting on the numerical solution $U_N^{(\alpha,\beta)}$:

$$\mathcal{F}\left(U_N^{(\alpha,\beta)}\right) = \left(\frac{\sum_{\ell=N-M+1}^{N} \gamma_\ell^{(\alpha,\beta)} \left(u_\ell^{(\alpha,\beta)}\right)^2}{\sum_{\ell=0}^{N} \gamma_\ell^{(\alpha,\beta)} \left(u_\ell^{(\alpha,\beta)}\right)^2} \right)^{\frac{1}{2}}, \tag{2.6}$$

which measures the contribution of the M highest-frequency components to the $L^2_{\hat{\omega}_\alpha}$-norm of $U_N^{(\alpha,\beta)}$. The subroutine FREQUENCY_INDICATOR in lines 3, 6, 10, and 18 of Algorithm 2.1 calculates this contribution in which we choose $M = [\frac{N}{3}]$ in view of the often-used $\frac{2}{3}$-rule [8, 12].

Algorithm 2.1. Pseudocode of spectral methods with frequency-dependent scaling.

1: Initialize N, $\nu > 1$, $q < 1$, Δt, T, α, β, $U_N^{(\alpha,\beta)}(0)$, $\underline{\beta}$
2: $t \leftarrow 0$
3: $f_0 \leftarrow \text{FREQUENCY\_INDICATOR}(U_N^{(\alpha,\beta)}(t))$
4: **while** $t < T$ **do**
5: $\quad U_N^{(\alpha,\beta)}(t+\Delta t) \leftarrow \text{EVOLVE}(U_N^{(\alpha,\beta)}(t), \Delta t)$
6: $\quad f \leftarrow \text{FREQUENCY\_INDICATOR}(U_N^{(\alpha,\beta)}(t+\Delta t))$
7: $\quad$ **if** $f > \nu f_0$ **then**
8: $\quad\quad \tilde{\beta} \leftarrow q\beta$
9: $\quad\quad U_N^{(\alpha,\tilde{\beta})} \leftarrow \text{SCALE}(U_N^{(\alpha,\beta)}(t+\Delta t), \tilde{\beta})$
10: $\quad\quad \tilde{f} \leftarrow \text{FREQUENCY\_INDICATOR}(U_N^{(\alpha,\tilde{\beta})})$
11: $\quad\quad$ **while** $\tilde{f} \le f$ and $\tilde{\beta} \ge \underline{\beta}$ **do**
12: $\quad\quad\quad \beta \leftarrow \tilde{\beta}$
13: $\quad\quad\quad U_N^{(\alpha,\beta)}(t+\Delta t) \leftarrow U_N^{(\alpha,\tilde{\beta})}$
14: $\quad\quad\quad f_0 \leftarrow \tilde{f}$
15: $\quad\quad\quad f \leftarrow \tilde{f}$
16: $\quad\quad\quad \tilde{\beta} \leftarrow q\beta$
17: $\quad\quad\quad U_N^{(\alpha,\tilde{\beta})} \leftarrow \text{SCALE}(U_N^{(\alpha,\beta)}(t+\Delta t), \tilde{\beta})$
18: $\quad\quad\quad \tilde{f} \leftarrow \text{FREQUENCY\_INDICATOR}(U_N^{(\alpha,\tilde{\beta})})$
19: $\quad\quad$ **end while**
20: $\quad$ **end if**
21: $\quad t \leftarrow t + \Delta t$
22: **end while**

If the frequency indicator $\mathcal{F}(U_N^{(\alpha,\beta)})$ increases over time, the contribution of high frequency components to the numerical solution increases, indicating that the numerical solution is decaying more slowly in x and that we need to adjust the scaling factor to enlarge the computational domain $[x_0^{(\alpha,\beta)}, x_N^{(\alpha,\beta)}]$ demarcated by the smallest and largest collocation point positions. In line 7 of Algorithm 2.1, νf_0 is the threshold at some time t. If the value of the frequency indicator of the current numerical solution $f > \nu f_0$, then we consider scaling. The parameter ν is usually chosen to be slightly larger than 1 to prevent the frequency indicator becoming too large without invoking scaling.

However, the **if** condition is only a necessary condition. Only after we enter the **while** loop in line 11 will we perform scaling, which aims to ensure that the frequency indicator $\mathcal{F}(U_N^{(\alpha,\beta)})$ will not increase after scaling. Actually, this **while** loop tries to minimize $\mathcal{F}(U_N^{(\alpha,\beta)})$ by geometrically shrinking the scaling factor β (q in line 16 is the common ratio) to ensure sufficient scaling since $\mathcal{F}(U_N^{(\alpha,\beta)})$ is a lower bound for the numerical error, as shown in (7.2). A more continuous adjustment is preferred by setting q to be slightly less than 1, which may also prevent overshrinking of the scaling factor within one single time step. Henceforth, we will choose $q = 0.95$ and $\nu = 1/q$. Moreover, at the initial time $t = 0$, we also ensure the frequency indicator is small enough by choosing a suitable initial scaling factor.

In this work, the generalized Laguerre functions with $\alpha = 0$ are used and the relative $L^2_{\hat{\omega}_\alpha}$-error

$$\text{Error} = \frac{\|U_N^{(\alpha,\beta)} - u\|_{\hat{\omega}_\alpha}}{\|u\|_{\hat{\omega}_\alpha}} \tag{2.7}$$

is used to measure the quality of the spectral approximation $U_N^{(\alpha,\beta)}(x)$ to the reference solution $u(x)$. We always use the most updated scaling factor to calculate the above error.

Example 1. We use the spreading Fermi–Dirac distribution

$$u(x,t) = \frac{1}{1 + \mathrm{e}^{\frac{x-5}{2+t}}} \tag{2.8}$$

to test the performance of the scaling algorithm, Algorithm 2.1. It can be readily verified that the reference solution $u(x,t)$ expands over time as shown in Figure 1(a). The proposed frequency-dependent scaling with $N = 40$ effectively maintains the relative error under 10^{-10} up until time $t = 10$, whereas the error for the corresponding unscaled solution rapidly grows to over 10^{-4} (see Figure 1(b)). We also plot, as $u(x,t)$ evolves, the history of the scaling factor β and frequency indicator $\mathcal{F}(U_N^{(\alpha,\beta)})$ in Figures 1(c) and 1(d), respectively. It is clear that the frequency indicator increases for the unscaled solution as time evolves and that time-dependent scaling is required to preserve the accuracy. The proposed frequency-dependent scaling technique detects the error and shrinks the scaling factor in order to enlarge the computational domain in accordance with the expansion of the reference solution. The spectral convergence as a function of the expansion order N can also be recovered by Algorithm 2.1. The errors at the final time, for the scaled and unscaled approaches, are displayed in Figure 1(e). The final scaling factors at $t = 10$ are $0.3213, 0.3560, 0.3747, 0.3945, 0.3945$ for $N = 25, 30, 35, 40, 45$, respectively, having all decreased from the common initial scaling factor of 2.5. Figures 1(e), (f) show very similar and expected behavior of the

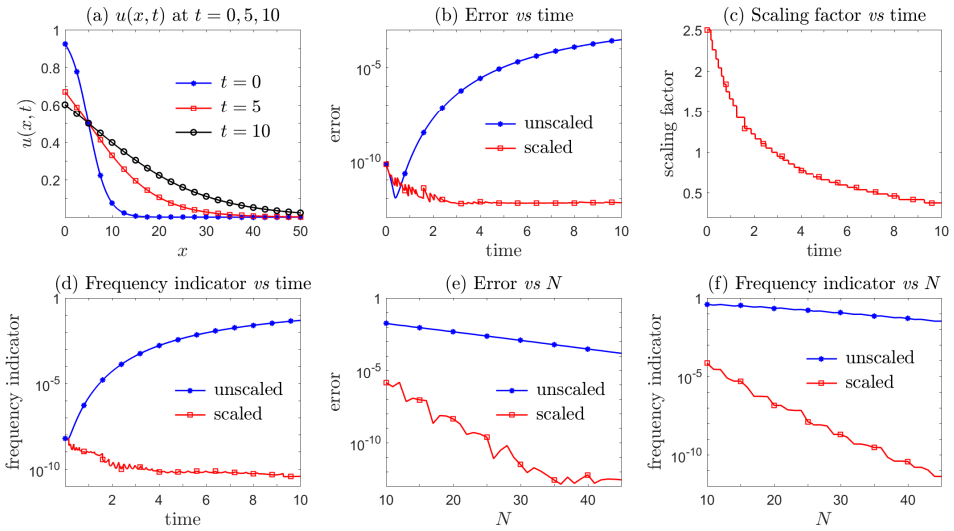

FIG. 1. *Numerical approximation to the diffusive Fermi–Dirac distribution* $u(x,t)$ *given by* (2.8). *The scaling algorithm, Algoritm* 2.1, *produces much more accurate solutions and recovers a faster spectral convergence with respect to the expansion order* N. *As we expected, the frequency indicator defined in* (2.6) *shows a similar behavior to the error defined in* (2.7) *against either time or* N. *The data in last two plots are measured at* $t = 10$.

frequency indicator and error as a function of N. Since the error and the frequency indicators behave similarly across time (see Figures 1(b, d)), we also expect them to behave similarly with N. These similarities suggest a possible connection between the error and the frequency indicator.

3. Exterior-error-dependent moving. Dynamics in unbounded domains can be much richer than the simple diffusive behavior successfully captured by our frequency-dependent scaling. Other physical mechanisms may induce, for example, translation (Examples 2 and 3) and emerging oscillations (Example 4). A purely scaling approach fails in these cases.

In this section, we develop an exterior-error-dependent moving method that will be able to resolve a solution's decay in an undetermined exterior domain $\Lambda_{\mathrm{e}} := (x_L, +\infty)$. Algorithm 3.1 presents the pseudocode of our exterior-error-dependent moving technique. In the algorithm, we first need to determine the time-dependent left-end point x_L. Next, we move the spectral basis accordingly so that the spectral approximation for an unknown function $u(x)$ in Λ_{e} (denoted by $U^{(\alpha,\beta)}_{N,x_L}(x)$) maintains accuracy. To implement this procedure, we adopt an *exterior-error indicator*:

$$\mathcal{E}(U^{(\alpha,\beta)}_{N,x_L}, x_R) = \frac{\|\partial_x U^{(\alpha,\beta)}_{N,x_L} \cdot \mathbb{I}_{(x_R,+\infty)}\|_{\hat{\omega}_\alpha}}{\|\partial_x U^{(\alpha,\beta)}_{N,x_L} \cdot \mathbb{I}_{(x_L,+\infty)}\|_{\hat{\omega}_\alpha}}, \tag{3.1}$$

which measures the proportion of the norm $\|\partial_x U^{(\alpha,\beta)}_{N,x_L} \cdot \mathbb{I}_{(x_L,+\infty)}\|_{\hat{\omega}_\alpha}$ inside a prescribed unbounded domain $(x_R, +\infty)$.

The subroutine EXTERIOR_ERROR_INDICATOR in lines 5, 8, and 13 of Algorithm 3.1 calculates $\mathcal{E}(U^{(\alpha,\beta)}_{N,x_L}, x_R)$. Here, following the often-used $\frac{2}{3}$-rule [8, 12], we choose $x_R = x^{(\alpha,\beta)}_{[\frac{N+2}{3}]}$ from the collocation points $x^{(\alpha,\beta)}_j (j = 0, 1, \ldots, N)$ in the exterior domain Λ_{e}.

Algorithm 3.1. Pseudocode of spectral methods with exterior-error-dependent moving.

1: Initialize N, Δt, T, α, β, $U^{(\alpha,\beta)}_{N,0}(0)$, $\mu > 1$, $d_{\max} > \delta > 0$
2: $t \leftarrow 0$
3: $x_L \leftarrow 0$
4: $x_R \leftarrow x^{(\alpha,\beta)}_{[\frac{N+2}{3}]}$
5: $e_0 \leftarrow$ EXTERIOR_ERROR_INDICATOR$(U^{(\alpha,\beta)}_{N,x_L}(0), x_R)$
6: **while** $t < T$ **do**
7: $\quad U^{(\alpha,\beta)}_{N,x_L}(t + \Delta t) \leftarrow$ EVOLVE$(U^{(\alpha,\beta)}_{N,x_L}(t), \Delta t)$
8: $\quad e \leftarrow$ EXTERIOR_ERROR_INDICATOR$(U^{(\alpha,\beta)}_{N,x_L}(t + \Delta t), x_R)$
9: $\quad$ **if** $e > \mu e_0$ **then**
10: $\quad\quad (d_0, U^{(\alpha,\beta)}_{N,x_L+d_0}) \leftarrow$ MOVE$(U^{(\alpha,\beta)}_{N,x_L}(t + \Delta t), \delta, d_{\max}, \mu e_0)$
11: $\quad\quad x_L \leftarrow x_L + d_0$
12: $\quad\quad x_R \leftarrow x_R + d_0$
13: $\quad\quad e_0 \leftarrow$ EXTERIOR_ERROR_INDICATOR$(U^{(\alpha,\beta)}_{N,x_L}(t + \Delta t), x_R)$
14: $\quad$ **end if**
15: $\quad t \leftarrow t + \Delta t$
16: **end while**

Intuitively, if $u(x)$ moves rightward in time, such as the moving Fermi–Dirac distribution in Example 2, the spectral approximation at large distances may deteriorate and the exterior-error indicator $\mathcal{E}(U^{(\alpha,\beta)}_{N,x_L})$ will increase. Consequently, the moving mechanism is triggered in line 9 of Algorithm 3.1 and completed by updating the left-end point $x_L = x_L + d_0$ in line 11. Thus, the starting point of the spectral approximation also moves rightward with time to capture the translation.

The displacement $d_0 = \min\{n\delta, d_{\max}\}$ is determined by the MOVE subroutine in line 10, where n is the smallest integer satisfying $\mathcal{E}(U^{(\alpha,\beta)}_{N,x_L}, x_R + n\delta) < \mu e_0$, δ is the minimum displacement, $d_{\max}$ is the maximum displacement, and μ represents the threshold of the increase in the exterior-error indicator that we can tolerate. In practice, $d_{\max}$ should be based on a prior knowledge of the maximum translation speed of the function $u(x)$. We usually choose $\mu \gtrsim 1$ to prevent the exterior-error indicator from becoming too large without invoking moving. The MOVE subroutine also generates $U^{(\alpha,\beta)}_{N,x_L+d_0}$ from $U^{(\alpha,\beta)}_{N,x_L}$.

Example 2. In this example, we consider the moving Fermi–Dirac distribution

$$u(x,t) = \frac{1}{1+\mathrm{e}^{\frac{x-5t}{2}}}, \tag{3.2}$$

which travels to the right at a speed of 5 without any shape change (see Figure 2(a)). The scaling algorithm, Algorithm 2.1, equipped with the same parameters that worked well for the diffusive Fermi–Dirac distribution in Example 1, fails to capture the translation. In fact, the errors of the scaled solutions are larger than those of unscaled ones as shown in Figure 2(b). It seems that the decrease of the scaling factor (black curve with asterisks in Figure 2(c)) cannot compensate for the increase in the frequency indicator (black curve with asterisks in Figure 2(d)). In other words, the scaling algorithm, Algorithm 2.1, mistakes translation as diffusion and performs excessive scaling. In contrast, the exterior-error-dependent moving algorithm, Algorithm 3.1,

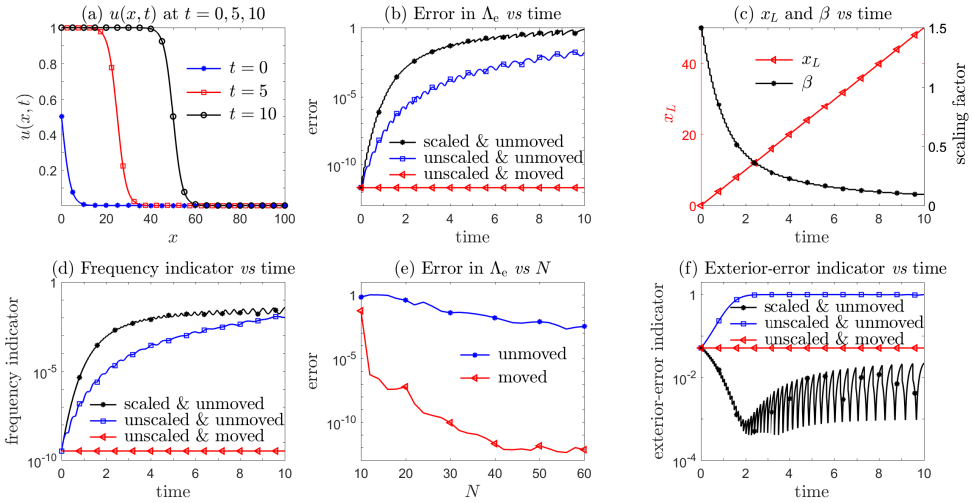

FIG. 2. *Numerical approximation to the moving Fermi–Dirac distribution* $u(x,t)$ *given by* (3.2). *The moving algorithm, Algorithm* 3.1*, produces much more accurate solutions and recovers a faster spectral convergence with respect to the expansion order* N *in the exterior domain* $\Lambda_e = (x_L, +\infty)$*, whereas a pure scaling fails to capture this translation. The data in* (e) *are measured at* $t = 10$.

with $\delta = 0.004$, $d_{\max} = 0.04$, and $\mu = 1.005$ succeeds in producing a much more accurate approximation to the moving Fermi–Dirac distribution given by (3.2) in the exterior domain Λ_e, with errors kept under 10^{-11} up to time $t = 10$ (red curve with left-pointing triangles in Figure 2(b)). The moving technique recovers a faster spectral convergence with respect to the expansion order N as shown in Figure 2(e).

During the moving process, the exterior-error indicator $\mathcal{E}(U^{(\alpha,\beta)}_{N,x_L}, x_R)$ is well controlled (red curve with left-pointing triangles in Figure 2(f)) and the left-end point of the exterior domain closely tracks the uniform linear motion (red curve with left-pointing triangles in Figure 2(c)). The exterior-error indicator monotonically increases for the unscaled and unmoved solutions (blue curve with squares in Figure 2(f)) and oscillates rapidly for the scaled and unmoved solutions (black curve with asterisks in Figure 2(f)). Moreover, the similarity between the relative error and the frequency indicator as a function of time is again confirmed by comparing Figure 2(d) to Figure 2(b), thus providing strong evidence for the effectiveness of using the frequency indicator (2.6). Spectral convergence in N is clearly observed for the moving spectral method in Figure 2(e), while the error decays slowly with N for the unmoved spectral method.

Example 3. Another class of dynamical systems are described by solitons or solitary waves in which nonlinearities and dispersion balance each other. While solitons have been well studied, there has been recent interest in nonlinear Dirac solitary waves as they emerge naturally in many physical systems [6]. Stability of the nonlinear Dirac solitary waves on the whole line and its connection to the multihump structure is a challenging topic of research [14, 27, 2]. In this example, we approximate a right-moving two-hump solitary wave, the explicit form of which is given in [15] with $v = 0.25$, $\lambda = 0.5$, $m = 1$, $x_0 = -1.5$, and $\Lambda = 0.1$. The reference solutions are plotted in Figure 3(a).

Numerical results are displayed in Figure 3, where we set $\delta = 0.004$, $d_{\max} = 0.012$, $\mu = 1.005$. It can be readily observed there that the exterior-error-dependent moving

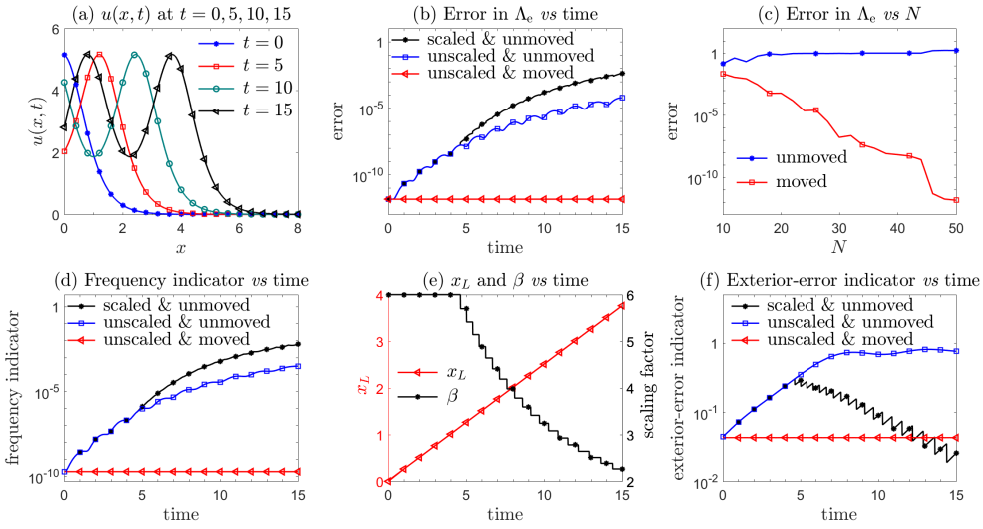

FIG. 3. *Approximating a two-hump nonlinear Dirac solitary wave. The moving algorithm, Algorithm* 3.1*, produces much more accurate solutions and recovers a faster spectral convergence with respect to the expansion order N in the exterior domain $\Lambda_e = (x_L, +\infty)$, whereas a pure scaling approach fails to capture this translation. The data in* (c) *are measured at $t = 15$.*

algorithm, Algorithm 3.1, produces much more accurate solutions with errors kept under 10^{-11} until the final time $t = 15$ (red curve with left-pointing triangles in Figure 3(b)). The moving algorithm also recovers a faster spectral convergence with respect to the expansion order N (see Figure 3(c)). The scaling-only algorithm, Algorithm 2.1, fails to maintain the accuracy (black curve with asterisks in Figure 3(b)). The similarity between the relative error and the frequency indicator is again confirmed by comparing Figure 3(d) to Figure 3(b).

In Examples 2 and 3, the exterior-error indicator (3.1) efficiently guides us in finding an x_L such that the moved spectral approximation retains accuracy in the resulting exterior domain. The accuracy arises from the fact that the exterior-error indicator is related to the upper bound of the error for asymptotically large x. If we assume a large indicator $\mathcal{E}(U^{(\alpha,\beta)}_{N,x_L}, x_R) > \mu$ with $\mu \in (0,1)$, then

$$\begin{aligned}\mathcal{E}(U^{(\alpha,\beta)}_{N,x_L}, x_R) > \mu &\Rightarrow \|\partial_x U^{(\alpha,\beta)}_{N,x_L} \cdot \mathbb{I}_{[x_R,+\infty)}\|_{\hat{\omega}_\alpha} > \mu \|\partial_x U^{(\alpha,\beta)}_{N,x_L} \cdot \mathbb{I}_{[x_L,+\infty)}\|_{\hat{\omega}_\alpha} \\ &\Rightarrow \|\partial_x U^{(\alpha,\beta)}_{N,x_L} \cdot \mathbb{I}_{[x_R,+\infty)}\|_{\hat{\omega}_{\alpha+1}} > \mu \|\partial_x U^{(\alpha,\beta)}_{N,x_L} \cdot \mathbb{I}_{[x_L,+\infty)}\|_{\hat{\omega}_{\alpha+1}},\end{aligned}$$

and since $\|\partial_x U^{(\alpha,\beta)}_{N,x_L} \cdot \mathbb{I}_{(x_L,+\infty)}\|_{\hat{\omega}_{\alpha+1}}$ are related to the upper bound of the interpolation error $\|(\mathcal{I}_{N,\alpha,\beta}u - u)\mathbb{I}_{(x_L,+\infty)}\|_{\hat{\omega}_\alpha}$ [16], a larger exterior-error indicator informs a worsening approximation in the exterior domain (x_R, ∞) in terms of the approximation in the whole domain (x_L, ∞). The solution in the interior domain $\Lambda_{\mathrm{i}} := (0, x_L]$ is not approximated by the basis functions used to approximate the solution in the exterior domain. Obstacles to designing moving mesh methods in unbounded domains include the construction of an *interior numerical solution* and its consistent coupling with the *exterior spectral approximation*. More on these issues will be illustrated in Example 4.

Example 4. Let us approximate the following function in Λ:

$$u(x,t) = \begin{cases} \cos(x - 10t), & x \le 10t, \\ \mathrm{e}^{-(x-10t)^2}, & x > 10t, \end{cases} \tag{3.3}$$

which represents a wave with period 2π traveling to the right with speed 10 and exponentially decaying at infinity. The reference solution $u(x, 10)$ is plotted by the green curve with circles in Figure 4(a), which coincides with the red curve with left-pointing triangles that approximates u seperately in Λ_i and Λ_e using different basis funcitons. As shown by the blue curve with squares in Figure 4(a), applying a Laguerre spectral approximation with $N = 30$ and $\beta = 5$ in Λ fails to accurately approximate $u(x,t)$. This failure arises because more oscillations emerge from $x = 0$ and translate to $+\infty$ as time evolves. Specifically, at $t = 10$, the reference solution $u(x,t)$ possesses 32 extrema while any Laguerre spectral approximation (2.2) with $N = 30$ can have at most 30 extrema, implying that the approximation is doomed to fail since all oscillations cannot be captured. Simply increasing the number of basis functions does little to help, even with different scaling factors as shown in Figure 4(b). The ineffectiveness of increasing N is mainly due to the presence of oscillatory components with significantly different frequencies in each of the two different domains. As shown by the black curves with asterisks in Figures 4(a), (c), (d), the scaling technique is also doomed to fail because it totally neglects this scale difference and only adjusts the scaling factor to redistribute collocation points.

We propose a divide-and-conquer strategy to address Example 4, which can be implemented by applying two subroutines within each time step. The first step is to

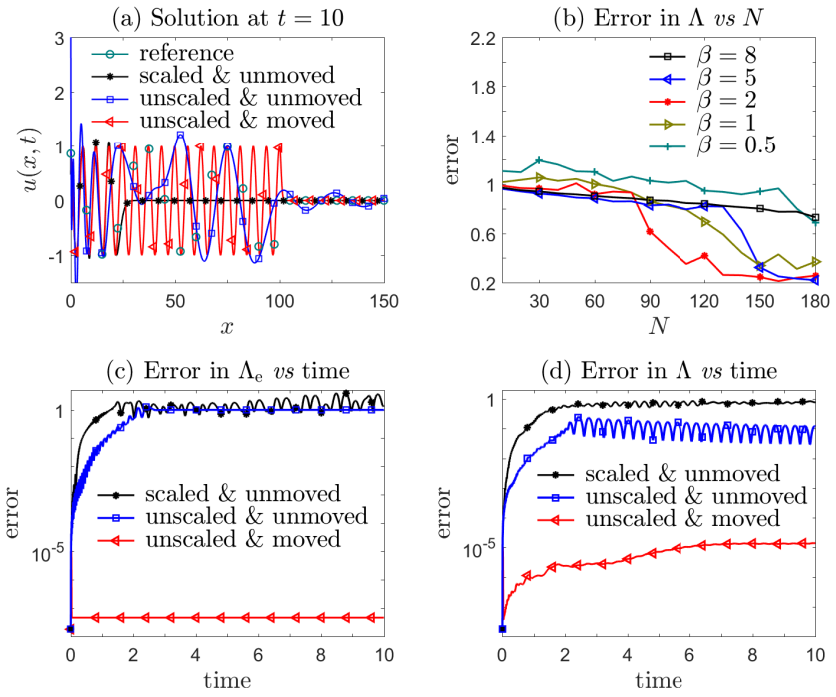

FIG. 4. *Oscillations emanate from the left but the moving algorithm, Algorithm* 3.1*, generates accurate solutions in the exterior domain* Λ_e*, with relative errors under* 10^{-7} *up to* $t = 10$ *with* $N = 30$ *(red curve with left-pointing triangles in* (c)*). By further coupling with a spectral approximation using* 80 *Chebyshev polynomials in the interior domain* Λ_i*, we generate the whole solution with total relative error, up until* $t = 10$*, under* 2×10^{-5}*, as shown by the red curves with left-pointing triangles in* (a) *and* (d)*. The data in* (b) *are measured at* $t = 10$*.*

use the exterior-error-dependent moving algorithm, Algorithm 3.1, to determine the exterior spectral approximation for the exponential decay component of the reference solution. The second step is to introduce a new spectral approximation in the remaining bounded interior domain Λ_i for the left-side oscillating component. The full numerical solution in the half-line Λ is constructed from concatenating the solution in the exterior domain Λ_e to the one in the interior domain Λ_i.

Figure 4(c) plots the error in the exterior domain against time and shows that the errors of the moved solution with $N = 30$, $\delta = 0.008$, $d_{\max} = 0.08$, and $\mu = 1.001$ are kept under 10^{-7} up to time $t = 10$ (red curve with left-pointing triangles), confirming that the Laguerre spectral approximation is accurate in the exterior domain. In fact, the numerical values of x_L obtained by the moving algorithm, Algorithm 3.1, are consistent with the expected value of $10t$ as shown in (3.3). Coupling the exterior solution with a spectral approximation using 80 Chebyshev polynomials in the interior domain, we find a combined numerical solution with total relative error under 2×10^{-5} up to $t = 10$ (red curves with left-pointing triangles in Figures 4(a), (d)) using $111 = 31 + 80$ total basis functions. By contrast, Figure 4(b) shows that the errors for direct refinement using $N = 180$ are larger than 0.2.

It must be pointed out that when solving PDEs in unbounded domains, we may need information about the solution in the exterior domain to construct the interior numerical solution. Further discussion on this point can be found in Example 6.

4. Spectral methods incorporating both scaling and moving. For problems that involve both translation and diffusion in unbounded domains, we need to

incorporate both the moving and scaling procedures. Since the scaling algorithm, Algorithm 2.1, may mistake translation for diffusion and trigger an inappropriate scaling as shown in Examples 2 and 3, we propose a "first moving then scaling" algorithm. The associated pseudocode is described in Algorithm 4.1. A direct application of Algorithm 4.1 to Example 1 recovers exactly the same results as Algorithm 2.1 since the moving procedure is not invoked. When Algorithm 4.1 is applied to Examples 2 and 3, it gives the same results as Algorithm 3.1 since the scaling mechanism is not triggered. That is, the combined moving-scaling algorithm, Algorithm 4.1, can deal with both translation-only and diffusion-only problems since it can distinguish translation from diffusion.

Algorithm 4.1. Pseudocode of spectral methods with both scaling and moving.

Initialize N, $\nu > 1$, $q < 1$, Δt, T, α, β, $U_N^{(\alpha,\beta)}(0)$, $\underline{\beta}$, $\mu > 1$, $d_{\max} > \delta > 0$, $x_R(0) = x^{(\alpha,\beta)}_{[\frac{N+2}{3}]}$
$t, x_L \leftarrow 0$
$x_R \leftarrow x^{(\alpha,\beta)}_{[\frac{N+2}{3}]}$
$f_0 \leftarrow \text{FREQUENCY\_INDICATOR}(U^{(\alpha,\beta)}_{N,x_L}(x,t))$
$e_0 \leftarrow \text{EXTERIOR\_ERROR\_INDICATOR}(U^{(\alpha,\beta)}_{N,x_L}(0), x_R)$
while $t < T$ **do**
    $x_R \leftarrow x^{(\alpha,\beta)}_{[\frac{N+2}{3}]}$
    $U^{(\alpha,\beta)}_{N,x_L}(x, t+\Delta t) \leftarrow \text{EVOLVE}(U^{(\alpha,\beta)}_{N,x_L}(x,t)), \Delta t)$
    $e \leftarrow \text{EXTERIOR\_ERROR\_INDICATOR}(U^{(\alpha,\beta)}_{N,x_L}(x, t+\Delta t), x_R)$
    if $e > \mu e_0$ **then**
        $(d_0, U^{(\alpha,\beta)}_{N,x_L+d_0}) \leftarrow \text{MOVE}(U^{(\alpha,\beta)}_{N,x_L}(x, t+\Delta t), \delta, d_{\max}, \mu e_0)$
        $x_L \leftarrow x_L + d_0$
        $e_0 \leftarrow \text{EXTERIOR\_ERROR\_INDICATOR}(U^{(\alpha,\beta)}_{N,x_L}(x, t+\Delta t), x_R)$
    end if
    $f \leftarrow \text{FREQUENCY\_INDICATOR}(U^{(\alpha,\beta)}_{N,x_L}(x, t+\Delta t))$
    if $f > \nu f_0$ **then**
        $\tilde{\beta} \leftarrow q\beta$
        $U^{(\alpha,\tilde{\beta})}_{N,x_L} \leftarrow \text{SCALE}(U^{(\alpha,\beta)}_{N,x_L}(x, t+\Delta t), \tilde{\beta})$
        $\tilde{f} \leftarrow \text{FREQUENCY\_INDICATOR}(U^{(\alpha,\tilde{\beta})}_{N,x_L})$
        while $\tilde{f} \le f$ and $\tilde{\beta} \ge \underline{\beta}$ **do**
            $\beta \leftarrow \tilde{\beta}$
            $U^{(\alpha,\beta)}_{N,x_L}(x, t+\Delta t) \leftarrow U^{(\alpha,\tilde{\beta})}_{N,x_L}$
            $f_0 \leftarrow \tilde{f}$
            $f \leftarrow \tilde{f}$
            $\tilde{\beta} \leftarrow q\beta$
            $U^{(\alpha,\tilde{\beta})}_{N,x_L} \leftarrow \text{SCALE}(U^{(\alpha,\beta)}_{N,x_L}(x, t+\Delta t), \tilde{\beta})$
            $\tilde{f} \leftarrow \text{FREQUENCY\_INDICATOR}(U^{(\alpha,\tilde{\beta})}_{N,x_L})$
        end while
    end if
    $t \leftarrow t + \Delta t$
end while

Algorithm 4.1 can be extended to unbounded domains in multiple dimensions in a dimension-by-dimension manner by using the tensor product of one-dimensional basis functions. For example, consider the two-dimensional spectral approximation

$$U^{(\vec{\alpha},\vec{\beta})}_{N,x_L,y_L}(x,y) \coloneqq \sum_{\ell=0}^{N_x}\sum_{m=0}^{N_y} u^{(\vec{\alpha},\vec{\beta})}_{\ell,m}\hat{\mathcal{L}}^{(\alpha_x,\beta_x)}_{\ell}(x)\hat{\mathcal{L}}^{(\alpha_y,\beta_y)}_{m}(y) \tag{4.1}$$

in $\Lambda^x_{\mathrm{e}} \times \Lambda^y_{\mathrm{e}} := (x_L, +\infty) \times (y_L, +\infty)$ where $\vec{\alpha} = (\alpha_x, \alpha_y)$ and $\vec{\beta} = (\beta_x, \beta_y)$. We choose the exterior-error indicator in the x-dimension to be

$$\mathcal{E}_x\left(U^{(\vec{\alpha},\vec{\beta})}_{N,x_L,y_L}(x,y), x_R\right) \coloneqq \mathcal{E}\left(\tilde{U}^{(\alpha_x,\beta_x)}_{N,x_L}(x), x_R\right), \tag{4.2}$$

$$\tilde{U}^{(\alpha_x,\beta_x)}_{N,x_L}(x) \coloneqq \int_{\Lambda^y_{\mathrm{e}}} U^{(\vec{\alpha},\vec{\beta})}_{N,x_L,y_L}(x,y)\mathrm{d}y. \tag{4.3}$$

Similarly, $\mathcal{E}_y(U^{(\vec{\alpha},\vec{\beta})}_{N,x_L,y_L}(x,y), y_R)$ gives the exterior-error indicator in the y-dimension. Accordingly, we use $\mathcal{E}_x(U^{(\vec{\alpha},\vec{\beta})}_{N,x_L,y_L}, x_R)$ to judge the **if** statement in line 10 of Algorithm 4.1. If satisfied, then the MOVE subroutine in line 11 will move the solution in the x-direction via $x_L \to x_L + d^x_0$. Simultaneously, we use $\mathcal{E}_y(U^{(\vec{\alpha},\vec{\beta})}_{N,x_L,y_L}(x,y), y_R)$ to determine the shift in the y-direction.

To allow scaling in the x-direction, the corresponding frequency indicator can be defined as

$$\mathcal{F}_x\left(U^{(\vec{\alpha},\vec{\beta})}_{N,x_L,y_L}\right) \coloneqq \left(\frac{\displaystyle\sum_{\ell=N_x-M_x+1}^{N_x}\sum_{m=0}^{N_y}\gamma^{(\alpha_x,\beta_x)}_{\ell}\gamma^{(\alpha_y,\beta_y)}_{m}\left(u^{(\vec{\alpha},\vec{\beta})}_{\ell,m}\right)^2}{\displaystyle\sum_{\ell=0}^{N_x}\sum_{m=0}^{N_y}\gamma^{(\alpha_x,\beta_x)}_{\ell}\gamma^{(\alpha_y,\beta_y)}_{m}\left(u^{(\vec{\alpha},\vec{\beta})}_{\ell,m}\right)^2}\right)^{\frac{1}{2}}, \tag{4.4}$$

where $M_x = [\frac{N_x}{3}]$ and N_x, N_y are the expansion orders in the x- and y-directions, respectively. Similarly, we define $\mathcal{F}_y$ to be the frequency indicator in the y-direction. We first keep β_y fixed and use $\mathcal{F}_x$ to evaluate the **if** statement in line 16 for scaling. If scaling in the x-direction is needed, then the **while** loop in line 20 will update the scaling factor to $\tilde{\beta}_x$. Simultaneously, we fix β_x and use $\mathcal{F}_y$ to update the scaling factor in the y-direction to $\tilde{\beta}_y$. After that, the scaling factors for time $t + \Delta t$ are set to $\tilde{\beta}_x$ and $\tilde{\beta}_y$.

Example 5. We will investigate the performance of Algorithm 4.1 in a two-dimensional unbounded domain by considering the function

$$u(x,y,t) = \cos\left(\frac{xy}{400}\right)\cdot\frac{1}{1+\mathrm{e}^{\frac{x-6t-2-t\cos(t)}{2+0.3t}}}\cdot\frac{1}{1+\mathrm{e}^{\frac{y-4t-2-t\sin(t)}{2+0.4t}}},\ \ x,y,t>0, \tag{4.5}$$

which displays both advective and diffusive behavior. This function exhibits oscillations in space from the factor $\cos(\frac{xy}{400})$, an exponential decay, and a translation to infinity with time-varying velocity $\vec{v} = (v_x, v_y) = (6 + \cos(t), 4 + \sin(t))$. The numerical results shown in Figure 5 are generated using a time step $\Delta t = 0.01$, the same parameters in the x- and y- directions, and $N_x = 40$, $\mu_x = 1.003$, $\delta_x = 0.005$, $d^x_{\max} = 0.1$.

As expected, only the combined scaling-moving algorithm, Algorithm 4.1, keeps the errors in the exterior domain under 10^{-11} (up to the final time $t = 4$), as shown by

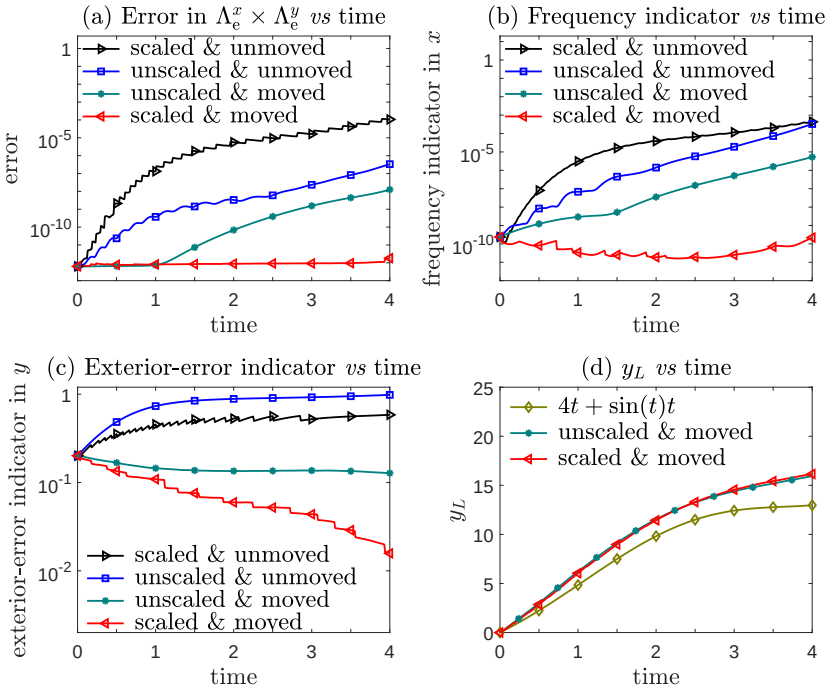

FIG. 5. *A two-dimensional oscillatory function with both translation and diffusion given by* (4.5). *Only the combined moving-scaling algorithm, Algorithm* 4.1, *produces accurate solutions in the exterior domain with errors kept under* 10^{-11} *up to* $t = 4$. *The need for combining moving and scaling is evident. For simplicity, we only used* $\mathcal{F}_x$ *(the frequency indicator in the* x*-direction),* $\mathcal{E}_y$ *(the exterior-error indicator in the* y*-direction), and* y_L *(the left end of* Λ_e^y*) as an example. The corresponding curves for* $\mathcal{F}_y$, $\mathcal{E}_x$, *and* x_L *are very similar and not shown. Here, we used* $N_x = N_y = 40$, *and the initial scaling factors:* $\beta_x = \beta_y = 2.5$.

the error curves in Figure 5(a). This accuracy is achieved because the corresponding frequency indicator and exterior-error indicator are controlled by our "first moving then scaling" techniques; see, Algorithm, e.g., $\mathcal{F}_x$ in Figure 5(b) and $\mathcal{E}_y$ in Figure 5(c).

Although the moving algorithm, Algorithm 3.1, may accurately capture the function near the left end of the exterior domain, the resulting exterior-error indicator does not stay low enough to preserve accuracy in the exterior domain $\Lambda_e^x \times \Lambda_e^y$, as shown by the green curves with asterisks in Figures 5(a), (c), (d). The moving algorithm neglects the diffusion and thus uses an improper (smaller) x_R and y_R. The right choice for these two variables depends on proper scaling for the diffusion, revealing why we need to update x_R in line 7 of Algorithm 4.1 after scaling. That is, the moving determines x_L while the scaling determines x_R, making it necessary to combine moving with scaling.

As we have mentioned in Example 4, numerically solving evolving PDEs in unbounded domains requires both the interior solution $U^{\text{interior}}_{x_L(t)}(x,t)$ in $\Lambda_i(t) = (0, x_L(t)]$ and the exterior solution $U^{(\alpha,\beta)}_{N,x_L(t)}(x,t)$ in $\Lambda_e(t)=(x_L(t), +\infty)$ after applying the divide-and-conquer strategy. When using the moving-scaling algorithm, Algorithm 4.1, to march the solution from t to $t + \Delta t$, if the moving mechanism is not triggered (i.e., x_L is unchanged), then the interior and exterior solutions can be updated individually in the normal way. If it is triggered, extra steps are needed to approximate the solution in the enlarged interior domain $\Lambda_i(t+\Delta t) = \Lambda_i(t) \cup (\Lambda_e(t) \setminus \Lambda_e(t+\Delta t))$ since $x_L(t + \Delta t) = x_L(t) + d_0$ after running line 12 of Algorithm 4.1.

In the next example, we will test the ability of Algorithm 4.1 to solve a one-dimensional PDE where we will use the intermediate (unmoved) exterior solution $U^{(\alpha,\beta)}_{N,x_L(t)}(x, t + \Delta t)$ (obtained immediately after running line 8) to interpolate the required function values in $\Lambda_{\rm i}(t + \Delta t) \setminus \Lambda_{\rm i}(t)$.

Example 6. We solve the following first order PDE:

$$\partial_t u(x,t) + \left(2 + \frac{x-2t}{2+t}\right)\partial_x u(x,t) = 0 \tag{4.6}$$

with initial data $u(x,0) = (1+\mathrm{e}^{\frac{x}{2}})^{-1}$ and Dirichlet boundary condition $u(0,t) = (1+\mathrm{e}^{\frac{-2t}{2+t}})^{-1}$. The analytical solution is a moving and diffusive Fermi–Dirac distribution, $u(x,t) = (1+\mathrm{e}^{\frac{x-2t}{2+t}})^{-1}$, which travels rightward to infinity at a speed of 2. A simple numerical scheme for evolving (4.6) is employed here for testing the performance of Algorithm 4.1 within the divide-and-conquer strategy.

Specifically, we adopt the Laguerre spectral approximation (2.2) in the exterior domain, the first order backward finite difference method in the interior domain, and the second order improved Euler scheme in time. We use a nonuniform mesh, e.g., 10 Gauss–Lobatto points, to avoid possible poor resolution in the tiny interior domain $0 < x_L < d_{\max}$ at short times. For $x_L \geq d_{\max}$, a uniform mesh with spacing $\Delta x = \delta = 0.02$ is used so new grid points in $\Lambda_{\rm i}(t + \Delta t) \setminus \Lambda_{\rm i}(t)$ can be easily added. The other parameters were set to $N = 40$, $\mu = 1.004$, $d_{\max} = 0.2$, and $\Delta t = 0.001$.

The results summarized in Figure 6 clearly show that, up to the final time $t = 5$, the proposed divide-and-conquer strategy maintains the errors in the whole domain $\Lambda = \Lambda_{\rm i} \cup \Lambda_{\rm e}$ under 2×10^{-4} (red curve with left-pointing triangles in Figure 6(a)). Algorithm 4.1 succeeds in capturing the translation, as shown by the red curve with left-pointing triangles in Figure 6(b), thus determining the exterior domain $\Lambda_{\rm e}$. Without this strategy, a straightforward use of the Laguerre spectral approximation in Λ leads to huge errors as indicated by the blue curve with right-pointing triangles in Figure 6(a).

Figure 6(c) shows that the frequency indicator is always kept under 3×10^{-10} as shown by the black curve with asterisks, a sufficiently small lower error bound for scaling, by continually shrinking the scaling factor shown as the black curve with asterisks in Figure 6(b). The exterior-error indicator is always maintained around 0.2 as shown by the red curve with left-pointing triangles in Figure 6(c), which implies the error in $(x_R, +\infty)$ divided by the error in $\Lambda_{\rm e}$ is almost unchanged, ensuring small errors at infinity. Figure 6(d) plots $|U(x,t) - u(x,t)|$ at different times ($U(x,t)$ and $u(x,t)$ denote the numerical and analytical solution, respectively). There is a clear divide near x_L arising from the different numerical treatments between the interior and exterior domains.

5. Performance comparison in solving parabolic PDEs. We now apply the frequency-dependent scaling algorithm, Algorithm 2.1, to solve

$$\partial_t u(x,t) - \partial_{xx} u(x,t) = f(x,t) \tag{5.1}$$

in $\mathbb{R} \times \Lambda$, and compare our results with those obtained with the time-dependent scaling method developed in [10]. First, we need to generalize our scaling approach from Λ to $\mathbb{R}$ by using the scaled Hermite functions: $\hat{\mathcal{H}}^{(\beta)}_\ell(x) = \mathcal{H}_\ell(\beta x)\mathrm{e}^{-(\beta x)^2/2}$ with $\mathcal{H}_\ell$ being the Hermite polynomials [16]. Similarly, we use β to denote the *scaling factor* and the frequency indicator defined in (2.6).

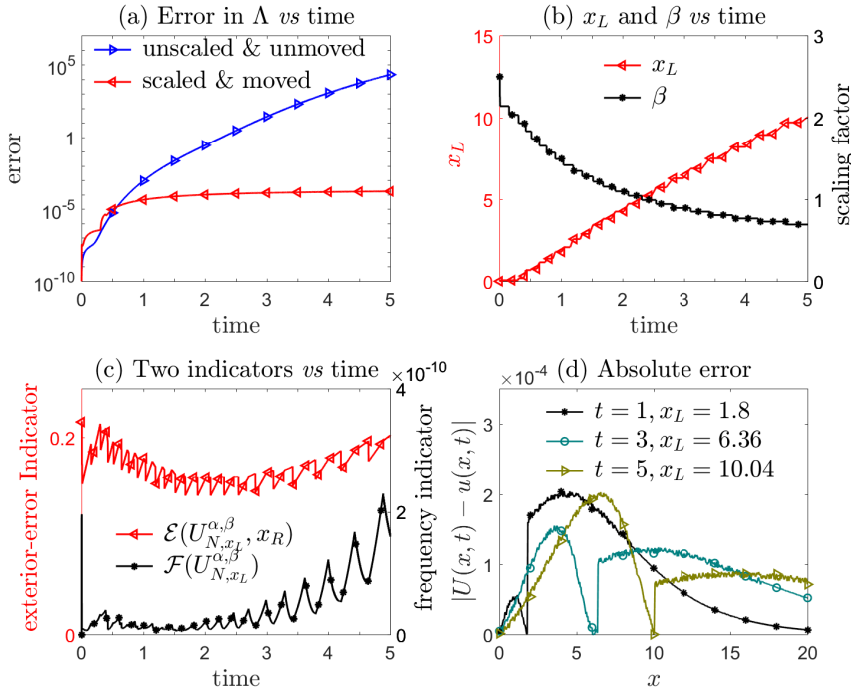

FIG. 6. *Numerical results obtained by the moving-scaling algorithm, Algorithm* 4.1*, for the one-dimensional problem in* (4.6)*. The proposed divide-and-conquer strategy maintains the errors in the whole domain* $\Lambda = \Lambda_i \cup \Lambda_e$ *under* 2×10^{-4} *until the final time* $t = 5$ *where the exterior domain* Λ_e *is determined by the "first moving then scaling" technique built into Algorithm* 4.1*. We adopt the Laguerre spectral approximation* (2.2) *with* $N = 40$ *in the exterior domain* $\Lambda_e = (x_L, +\infty)$*, the first order backward finite difference method with spacing* $\Delta x = 0.02$ *in the interior domain* $\Lambda_i = (0, x_L]$*, and the second order improved Euler time marching scheme with* $\Delta t = 0.001$*. The last plot displays the absolute difference between the numerical solution* $U(x,t)$ *and the analytical one* $u(x,t)$ *at different times.*

A standard Galerkin Hermite spectral method is used to find a solution $U_N^{(\beta)} = \sum_{\ell=0}^{N} u_\ell^{(\beta)} \hat{\mathcal{H}}_\ell^{(\beta)}(x)$ in $V_N^{(\beta)} = \text{span}\{\hat{\mathcal{H}}_0^{(\beta)}(x), \ldots, \hat{\mathcal{H}}_N^{(\beta)}(x)\}$ satisfying the initial condition and

$$\left(\partial_t U_N^{(\beta)}, v\right) + \left(\partial_x U_N^{(\beta)}, \partial_x v\right) = (f, v) \quad \forall\, v \in V_N^{(\beta)}, \tag{5.2}$$

where $(\cdot,\cdot)$ is the conventional inner product in $L^2(\mathbb{R})$ space. The Galerkin discretization (5.2) is stable in the sense that

$$\left(\partial_x U_N^{(\beta)}, \partial_x U_N^{(\beta)}\right) = \sum_{\ell=0}^{N+1} \frac{\ell+1}{2} \left(u_\ell^{(\beta)}\right)^2 - \sum_{\ell=0}^{N-2} \sqrt{(\ell+1)(\ell+2)}\, u_\ell^{(\beta)} u_{\ell+2}^{(\beta)} \tag{5.3}$$

is strictly positive and can be controlled by $(N+1)\|U_N^{(\beta)}\|_2^2 = (N+1)\sum_{\ell=0}^{N}(u_\ell^{(\beta)})^2$. By contrast, a time-dependent scaling factor

$$\beta(t) = \frac{1}{2\sqrt{\delta_0(\delta t + 1)}} \tag{5.4}$$

was taken in [10] to fix the instability of the Petrov–Galerkin discretization by tuning the parameters δ_0 and δ.

Example 7. We apply the frequency-dependent scaling algorithm, Algorithm 2.1, to Example 6.1 in [10]. In order to facilitate comparison, we also adopt the same second order accurate Crank–Nicholson scheme to march (5.2) and the same errors E_N and $E_{N,\infty}$ to measure the accuracy. Table 1 presents the numerical errors with different time steps and expansion orders where the second order accuracy in time and the spectral convergence in space are clearly demonstrated. Table 2 compares the errors E_N without scaling to those obtained using the scaling algorithm, Algorithm 2.1, and the time-dependent scaling method in [10] on the same mesh. Both scaling methods produce much more accurate numerical results but the proposed frequency-dependent scaling keeps the errors around or below 10^{-7}, outperforming the time-dependent scaling of [10].

The scaling factor adjusted adaptively by the frequency indicator (2.6) takes on the value $\beta = 0.5357$ at $t = 1$ for all choices of time steps shown in Table 2, whereas the time-dependent scaling factor in [10] decreases to $\beta = 0.3536$ at $t = 1$ (equation (5.4)). The smaller scaling factor arises from the stability requirement $\beta'(t) + 2\beta^3(t) \leq 0$, an initial value of 0.5, and using $\delta_0 = \delta = 1$ in (5.4) [10] and prevents the error from decreasing when the time step is refined from 1/4000 to 1/16000 (see the third column of Table 2). There is no accuracy improvement without scaling when the time step is decreased as shown in the second column of Table 2, where a scaling factor is fixed to $\beta = 0.85$. Regardless of what time step is used in the unscaled method, the error E_N experiences a sudden increase across $t \in [0.3, 0.7]$, rising from below 10^{-6} to about 10^{-4}, as it fails to capture the diffusion. A similar observation was shown in Table 6.1 of [10].

6. Application to rational basis functions. Apart from the Laguerre and Hermite functions with exponential decay at infinity, rational basis functions which characterize algebraic decay at infinity are increasingly of research interest [22].

TABLE 1

Numerical results for the parabolic problem in (5.1)*: Errors associated with the frequency-dependent scaling algorithm* 2.1 *at* $t = 1$ *with different time step and expansion order* N*.*

Time step	N	$E_N(1)$	Order	$E_{N,\infty}(1)$	Order
10^{-1}	25	2.500e-04		2.182e-04	
10^{-2}		2.499e-07	2.000	2.227e-06	1.991
10^{-3}		2.500e-09	2.000	2.227e-08	2.000
10^{-4}		2.555e-10	1.991	2.350e-10	1.977
1/40000	10	2.203e-04		1.619e-04	
	15	2.189e-07	$N^{-16.85}$	4.335e-08	$N^{-20.29}$
	20	1.353e-09	$N^{-17.68}$	8.880e-09	$N^{-13.52}$
	25	4.840e-11	$N^{-14.93}$	6.183e-11	$N^{-11.94}$

TABLE 2

Numerical results for the parabolic problem in (5.1)*: Comparison of the errors at* $t = 1$ *with* $N = 20$*.*

Time step	No scaling	Time-dependent scaling in [10]	Frequency-dependent scaling in Algorithm 2.1
1/250	3.969e-04	2.598e-06	3.998e-07
1/1000	3.910e-04	1.189e-06	2.503e-08
1/4000	3.390e-04	1.117e-06	2.085e-09
1/16000	3.390e-04	1.117e-06	1.381e-09

In this section we generalize our scaling technique to solve a fractional heat equation, the solution of which displays algebraic decay at infinity. We shall use MMGFs [22]: $R_n^{(\lambda,\beta)}(x) = (1+(\beta x)^2)^{-(\lambda+1)/2} C_n^\lambda(\beta x/\sqrt{1+(\beta x)^2})$ with C_n^λ being the Gegenbauer polynomial of order n, which behaves like $(\mathrm{sign}(x))^n \frac{(2\lambda)_n}{n!}(1+(\beta x)^2)^{-(1+\lambda)/2}$ as $|x| \to +\infty$. We still use β to be the scaling factor and define the frequency indicator for the spectral decomposition $U_N^{(\beta)} = \sum_{i=0}^N u_i^{(\beta)} R_i^{(\lambda,\beta)}(x)$ in the same way as (2.6).

Example 8. We numerically solve the fractional heat equation on $\mathbb{R}$ in [29, 28]:

$$u_t + (-\Delta)^s u = f(x,t), \quad s \in (0,1), \tag{6.1}$$

which admits an analytic solution $u(x,t) = ((\frac{x}{t+0.5})^2+1)^{-1/2}$ for an appropriate source function. Therefore, we choose $\lambda = 0$ in MMGFs, which decay at a rate of $(1+(\beta x)^2)^{-1/2}$, to match the decaying behavior of the analytic solution. Clearly, the solution is diffusive over time, requiring us to decrease the scaling factor β. Figure 7 gives the numerical results for $s = 0.1$, 0.2, and 0.8, where we have adopted the improved Euler scheme in EVOLVE of Algorithm 4.1, and set $\Delta t = 0.005$, $N = 20$, $q = \nu^{-1} = 0.95$, and $\beta_0 = 2$. In Figures 7(c), (f), (i), it is easily observed that the scaling factor β matches the intrinsic scaling of the analytic solution and decreases from 2 to about 0.6 over time, during which the errors are well maintained under 10^{-6} for all

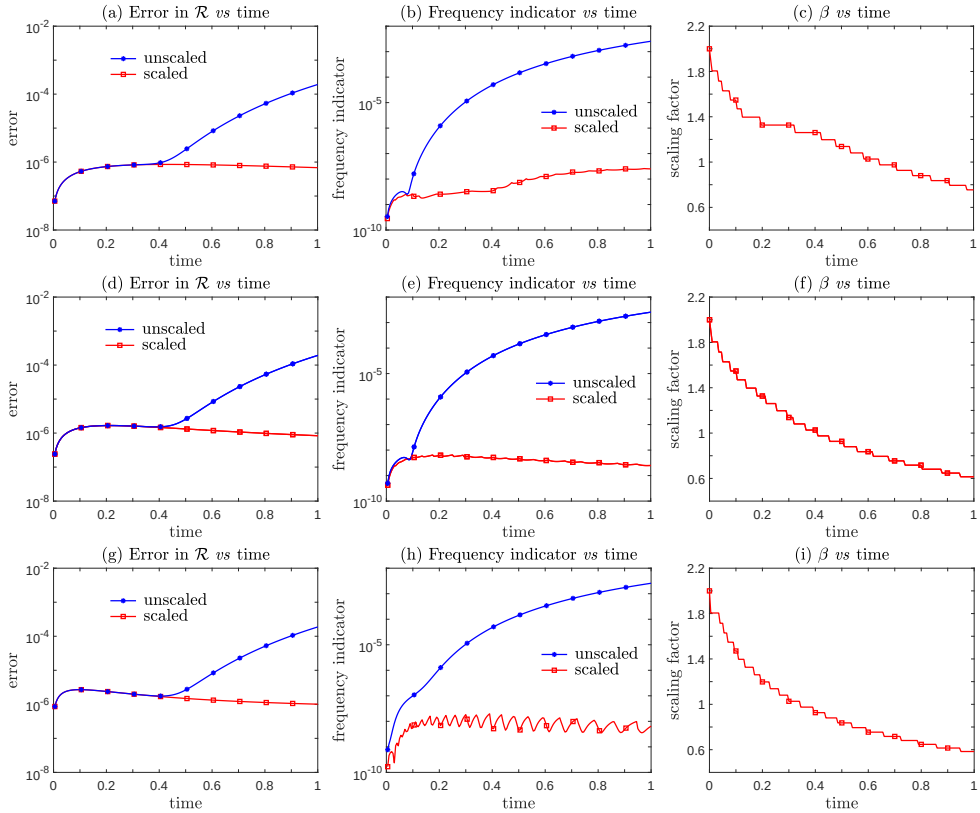

FIG. 7. *Errors, frequency indicators, and scaling factors obtained with MMGFs in solving* (6.1) *for* $s = 0.1$ *(the first row),* 0.5 *(the second row), and* 0.8 *(the third row). Both the error and the frequency indicator are well maintained owing to an approriate adjustment of the scaling factor.*

three fractional orders (see the red curves in Figures 7(a), (d), (g)). Failure to adjust β leads to a fast increasing frequency indicator (see the blue curves in Figures 7(b), (e), (h)) as well as a much larger error (see the blue curves in Figures 7(a), (d), (g)). Comparing the red curves in Figures 7(a), (d), (g) with ones in Figures 7(b), (e), (h), we have observed that the strong correlation of the error and the frequency indicator is again confirmed with these rational MMGF basis functions. Thus we can conclude that, no matter what s in (6.1) is, the frequency-dependent scaling is still effective in MMGFs as long as they are able to capture the decaying behavior of the unknown solution at infinity. For functions which display different algebraic decaying behavior at infinity, it should also work well after choosing an appropriate λ in MMGFs to match the algebraic decaying. In a word, the proposed frequency-dependent scaling technique can be successfully generalized to the rational basis functions.

7. Analysis of the scaling and moving techniques. We first illustrate the effectiveness of the frequency-dependent scaling in maintaining small errors. Second, we analyze how the moving technique controls the errors in the exterior domain. Finally, we use two examples to show how both techniques are sensitive to the parameters.

7.1. Numerical analysis. The success of the scaling algorithm, Algorithm 2.1, is rooted in the connection between the frequency indicator (2.6) and the evolution of the information embedded in the numerical solutions. Let $\hat{A}^r_{\alpha,\beta}(\Lambda)$ be the anisotropically weighted Sobolev space. For any integer $r \geq 0$, its seminorm and norm are $|u|_{\hat{A}^r_{\alpha,\beta}} = \|\hat{\partial}_x^r u\|_{\hat{\omega}_{\alpha+r}}$ and $\|u\|_{\hat{A}^r_{\alpha,\beta}} = \sqrt{\sum_{k=0}^r |u|^2_{\hat{A}^k_{\alpha,\beta}}}$, respectively, with $\hat{\partial}_x u = \partial_x u + \frac{\beta}{2} u$. For any $u \in \hat{A}^r_{\alpha-1,\beta}(\Lambda) \cap \hat{A}^r_{\alpha,\beta}(\Lambda)$ with integer $r \geq 1$, a direct corollary of Theorem 3.5 in [7] for estimating the interpolation error using the Laguerre functions is

$$\|\mathcal{I}_{N,\alpha,\beta} u - u\|_{\hat{\omega}_\alpha} \leq c(\beta N)^{\frac{1-r}{2}} \left(\beta^{-1} |u|_{\hat{A}^r_{\alpha-1,\beta}} + \left(1 + \beta^{-\frac{1}{2}}\right) (\ln N)^{\frac{1}{2}} |u|_{\hat{A}^r_{\alpha,\beta}}\right), \tag{7.1}$$

where c denotes a generic positive constant which does not depend on α, β, N, or any function. This error estimate plays a crucial role in the formal development and successful implementation of our scaling and moving techniques. For simplicity we only consider diffusive behavior without translation, set $x_L = 0$, and drop the subscript x_L.

There are two reasons for us to use the frequency indicator given in (2.6). Starting with $M = [\frac{n}{3}]$ and a sufficiently large expansion order N, we have

$$\begin{aligned} \frac{1}{2}\mathcal{F}(U_N^{(\alpha,\beta)}) &\approx \frac{1}{2} \frac{\|\mathcal{I}_{N,\alpha,\beta} u - \mathcal{I}_{N-M,\alpha,\beta} u\|_{\hat{\omega}_\alpha}}{\|\mathcal{I}_{N,\alpha,\beta} u\|_{\hat{\omega}_\alpha}} \\ &\leq \frac{1}{2} \frac{\|u - \mathcal{I}_{N,\alpha,\beta} u\|_{\hat{\omega}_\alpha} + \|u - \mathcal{I}_{N-M,\alpha,\beta} u\|_{\hat{\omega}_\alpha}}{\|\mathcal{I}_{N,\alpha,\beta} u\|_{\hat{\omega}_\alpha}} \leq \frac{\|u - \mathcal{I}_{N-M,\alpha,\beta} u\|_{\hat{\omega}_\alpha}}{\|\mathcal{I}_{N,\alpha,\beta} u\|_{\hat{\omega}_\alpha}}, \end{aligned} \tag{7.2}$$

which provides an estimate to the lower bound of $\|u - \mathcal{I}_{N-M,\alpha,\beta} u\|_{\hat{\omega}_\alpha}$. Minimizing $\mathcal{F}(U_N^{(\alpha,\beta)})$ in Algorithm 2.1 may reduce the lower bound of the interpolation error. Moreover, a straightforward application of the interpolation error estimator (7.1) to the two terms in the first numerator of (7.2) yields

$$\left(\sum_{\ell=N-M+1}^{N} \gamma_\ell^{(\alpha,\beta)} \left(u_\ell^{(\alpha,\beta)}\right)^2\right)^{1/2} \leq c_F (\beta N)^{\frac{1-r}{2}} \left(\beta^{-1} |u|_{\hat{A}^r_{\alpha-1,\beta}} + \left(1 + \beta^{-\frac{1}{2}}\right) (\ln N)^{\frac{1}{2}} |u|_{\hat{A}^r_{\alpha,\beta}}\right), \tag{7.3}$$

where the constant $c_F \equiv (1+2^{\frac{r-1}{2}})c$. Thus, we find

$$\mathcal{F}\left(U_N^{(\alpha,\beta)}\right) \le c_F(\beta N)^{\frac{1-r}{2}} \left(\beta^{-1}\frac{|u|_{\hat{A}^r_{\alpha-1,\beta}}}{\|U_N^{(\alpha,\beta)}\|_{\hat{\omega}_\alpha}} + \left(1+\beta^{-\frac{1}{2}}\right)(\ln N)^{\frac{1}{2}}\frac{|u|_{\hat{A}^r_{\alpha,\beta}}}{\|U_N^{(\alpha,\beta)}\|_{\hat{\omega}_\alpha}}\right), \tag{7.4}$$

implying that $\forall\, \varepsilon \in (0,1)$, we may choose a sufficiently large N such that $\mathcal{F}(U_N^{(\alpha,\beta)}) < \varepsilon$. This constitutes the first reason for us to take into account the frequency indicator (2.6).

Secondly the frequency indicator $\mathcal{F}(U_N^{(\alpha,\beta)})$ can be used to measure the decay of the reference solution's derivatives as x tends to infinity. According to inequality (7.4), if $|u|_{\hat{A}^r_{\alpha-1,\beta}}/\|U_N^{(\alpha,\beta)}\|_{\hat{\omega}_\alpha}$ is fixed, a larger $\mathcal{F}(U_N^{(\alpha,\beta)})$ implies a larger $|u|_{\hat{A}^r_{\alpha,\beta}}/\|U_N^{(\alpha,\beta)}\|_{\hat{\omega}_\alpha}$. Given any $s \in \Lambda$ (e.g., $s = \sqrt{2x_N^{(\alpha,\beta)}}$), if

$$\mathcal{F}\left(U_N^{(\alpha,\beta)}\right) > c_F(\beta N)^{\frac{1-r}{2}}\frac{|u|_{\hat{A}^r_{\alpha-1,\beta}}}{\|U_N^{(\alpha,\beta)}\|_{\hat{\omega}_\alpha}}\left(\beta^{-1} + s\left(1+\beta^{-\frac{1}{2}}\right)(\ln N)^{\frac{1}{2}}\right), \tag{7.5}$$

then

$$\int_0^{\frac{s^2}{2}} (\hat{\partial}_x^r u(x))^2 x^{\alpha+r}\mathrm{d}x < \int_{\frac{s^2}{2}}^{+\infty} (\hat{\partial}_x^r u(x))^2 x^{\alpha+r}\mathrm{d}x. \tag{7.6}$$

The verification of the inequality (7.6) can be finished by contradiction. First we can combine (7.4) and (7.5) to find

$$s|u|_{\hat{A}^r_{\alpha-1,\beta}} < |u|_{\hat{A}^r_{\alpha,\beta}}. \tag{7.7}$$

If (7.6) does not hold, we would have

$$\begin{aligned}|u|^2_{\hat{A}^r_{\alpha,\beta}} &= \int_0^{+\infty} (\hat{\partial}_x^r u(x))^2 x^{\alpha+r}\mathrm{d}x \le 2\int_0^{\frac{s^2}{2}} (\hat{\partial}_x^r u(x))^2 x^{\alpha+r}\mathrm{d}x \\ &\le 2\cdot\frac{s^2}{2}\int_0^{\frac{s^2}{2}} (\hat{\partial}_x^r u(x))^2 x^{\alpha+r-1}\mathrm{d}x \le s^2\int_0^{+\infty} (\hat{\partial}_x^r u(x))^2 x^{\alpha+r-1}\mathrm{d}x = s^2|u|^2_{\hat{A}^r_{\alpha-1,\beta}},\end{aligned}$$

which would contradict the inequality (7.7). Intuitively, basis functions of higher degree decay more slowly than those of lower degree, so an increase in the frequency indicator implies slower decay at infinity. This slower spatial decay as time increases requires using a larger computational domain which is achieved by decreasing β. In other words, as the frequency indicator increases, the norm of $\hat{\partial}_x^r u(x)\cdot \mathbb{I}_{(s^2/2,+\infty)}(x)$ becomes larger than that of $\hat{\partial}_x^r u(x)\cdot \mathbb{I}_{(0,s^2/2)}(x)$, implying scaling is indeed needed to enlarge the computational domain because $\|\hat{\partial}_x^r u\cdot \mathbb{I}_{(x>s^2/2)}\|_{\hat{\omega}_\alpha}$ is the dominant component of $\|\hat{\partial}_x^r u\|_{\hat{\omega}_\alpha}$.

Next, we show that increasing x_L in the moving technique may control the errors in the exterior domain when the generalized Laguerre functions are adopted. After increasing x_L to $x_L + d$, $U_{N,x_L}^{(\alpha,\beta)}(x)e^{\beta x/2}$ and $U_{N,x_L+d}^{(\alpha,\beta)}(x)e^{\beta x/2}$ are two identical polynomials of order N since they pass through the same $N+1$ different points: $(x_i^{(\alpha,\beta)}+d,\ U_{N,x_L}^{(\alpha,\beta)}(x_i^{(\alpha,\beta)}+d)e^{\beta(x_i^{(\alpha,\beta)}+d)/2})$, $i = 0,\ldots,N$, i.e., $U_{N,x_L+d}^{(\alpha,\beta)}(x) = U_{N,x_L}^{(\alpha,\beta)}(x)$ for any $x \in (x_L+d,\infty)$. Then we have

$$\left\|U^{(\alpha,\beta)}_{N,x_L+d}(x,t)\mathbb{I}_{(x>x_L+d)}\right\|^2_{\hat{\omega}_\alpha} = \left\|U^{(\alpha,\beta)}_{N,x_L}(x,t)\mathbb{I}_{(x>x_L+d)}\right\|^2_{\hat{\omega}_\alpha}$$

$$\leq \left\|U^{(\alpha,\beta)}_{N,x_L}(x,t)\mathbb{I}_{(x>x_L)}\right\|^2_{\hat{\omega}_\alpha}, \tag{7.8}$$

$$\left\|(u(x,t)-U^{(\alpha,\beta)}_{N,x_L}(x,t))\mathbb{I}_{(x>x_L+d)}\right\|_{\hat{\omega}_\alpha} \leq \left\|(u(x,t)-U^{(\alpha,\beta)}_{N,x_L}(x,t))\mathbb{I}_{(x>x_L)}\right\|_{\hat{\omega}_\alpha}. \tag{7.9}$$

That is, both the norm of $U^{(\alpha,\beta)}_{N,x_L}$ and the error $\|(u(x)-U^{(\alpha,\beta)}_{N,x_L})\mathbb{I}_{(x>x_L)}\|_{\hat{\omega}_\alpha}$ will not increase as x_L increases. Furthermore, when solving a time-dependent PDE,

$$u_t(t,x) = D_x(t)u(t,x), x,t \in \Lambda, \tag{7.10}$$

where $D_x(t)$ represents a differential operator that only involves spatial derivatives. Let $U^{(\alpha,\beta)}_{N,x_L}(x,t_n)$ denote its numerical solution at t_n. Then we have that there exists $d \geq 0$ such that

$$\left\|U^{(\alpha,\beta)}_{N,x_L+d}(x,t_{n+1})\mathbb{I}_{(x>x_L+d)}\right\|_{\hat{\omega}_\alpha} \leq \left\|U^{(\alpha,\beta)}_{N,x_L}(x,t_n)\mathbb{I}_{(x>x_L)}\right\|_{\hat{\omega}_\alpha}, \tag{7.11}$$

the verification of which is given in the following.

Denoting $\mathbf{u}_{N,x_L} = (U^{(\alpha,\beta)}_{N,x_L}(x^{(\alpha,\beta)}_0),\ldots,U^{(\alpha,\beta)}_{N,x_L}(x^{(\alpha,\beta)}_N))^T$, we define the translation operator matrix $T^{(\alpha,\beta)}_N(s)$ acting on $\mathbf{u}_{N,x_L}$ such that $T^{(\alpha,\beta)}_N(s)\mathbf{u}_{N,x_L} = (U^{(\alpha,\beta)}_{N,x_L}(x^{(\alpha,\beta)}_0+s),\ldots,U^{(\alpha,\beta)}_{N,x_L}(x^{(\alpha,\beta)}_N+s)^T$. It is readily observed that $T^{(\alpha,\beta)}_N(s+t) = T^{(\alpha,\beta)}_N(s)T^{(\alpha,\beta)}_N(t)$ for $s,t \in \Lambda$, and $T^{(\alpha,\beta)}_N(0)$ is an identity matrix. That is, $T^{(\alpha,\beta)}_N(s \in \Lambda)$ forms a semigroup. We denote its generator, $\hat{\mathcal{L}}^{(\alpha,\beta)}_{N,p} := \lim_{s\to\infty}\frac{T^{(\alpha,\beta)}_N(s)-T^{(\alpha,\beta)}_N(0)}{s}$, and then $T^{(\alpha,\beta)}_N(s) = \mathrm{e}^{s\hat{\mathcal{L}}^{(\alpha,\beta)}_{N,p}}$. Since the Laguerre functions tend to 0 at $+\infty$, $\lim_{s\to+\infty}T^{(\alpha,\beta)}_N(s) = 0$, indicating $\|\mathrm{e}^{\hat{\mathcal{L}}^{(\alpha,\beta)}_{N,p}}\|_{\hat{w}^{\alpha,\beta}} < 1$ where the matrix norm $\|\cdot\|_{\hat{w}^{\alpha,\beta}}$ is induced from the vector norm $\|\mathbf{u}_{N,x_L}\|^2_{\hat{w}^{\alpha,\beta}} := \sum_{\ell=0}^N (U^{(\alpha,\beta)}_{N,x_L}(x^{(\alpha,\beta)}_\ell))^2\hat{w}^{(\alpha,\beta)}_\ell$. After discretizing $D_x(t)$ in (7.10) with some numerical schemes, we will obtain $\mathbf{u}_{N,x_L}(t_{n+1}) = D^{(\alpha,\beta)}_N(t_n)\mathbf{u}_{N,x_L}(t_n)$, $D^{(\alpha,\beta)}_N(t_n) \in \mathbb{R}^{(N+1)\times(N+1)}$. Let $\mathbf{u}_{N,x_L+d}(t_{n+1}) := T^{(\alpha,\beta)}_N(d)\mathbf{u}_{N,x_L}(t_{n+1})$ where we choose $d = -\ln\|D^{(\alpha,\beta)}_N(t_n)\|_{\hat{w}^{\alpha,\beta}}/\ln\|e^{\hat{\mathcal{L}}^{(\alpha,\beta)}_{N,p}}\|_{\hat{w}^{\alpha,\beta}}$. Then we are able to readily verify that $\|\mathbf{u}_{N,x_L+d}(t_{n+1})\|_{\hat{w}^{\alpha,\beta}} \leq \|\mathbf{u}_{N,x_L}(t_{n+1})\|_{\hat{w}^{\alpha,\beta}}$, which directly gives (7.11) using $\|\mathbf{u}_{N,x_L}(t)\|_{\hat{w}^{\alpha,\beta}} = \|U^{(\alpha,\beta)}_{N,x_L}(x,t)\mathbb{I}_{(x>x_L)}\|_{\hat{\omega}_\alpha}$.

7.2. Sensitivity analysis. The scaling in Algorithm 2.1 adopts two parameters q,ν, and the moving in Algorithm 3.1 adopts two parameters δ,μ, too. Below we would like to use two examples to investigate how these parameters affect the performance and then discuss some intuitive rules for setting them.

Example 9. First we discuss the scaling technique's dependence on q and ν via solving

$$\begin{cases} \dfrac{\partial u}{\partial t} + \dfrac{x-5}{2+t}\cdot\dfrac{\partial u}{\partial x} = 0, \\ u(x,0) = \dfrac{1}{1+\mathrm{e}^{\frac{x-5}{2}}}. \end{cases} \tag{7.12}$$

Its analytic solution is $u(x,t) = 1/(1+\mathrm{e}^{\frac{x-5}{2+t}})$, diffusive over time, and thus requires decreasing β. Table 3 records the scaling factor β in the bottom-left and the error in the upper-right at $t = 10$ where the improved Euler scheme is called in EVOLVE of Algorithm 2.1, and we have set $\Delta t = 0.004$, $\beta_0 = 3$, and $N = 24$. It can be observed

TABLE 3

Errors (upper-right) and scaling factors (bottom-left) at $t = 10$ for different ν and q in solving (7.12) with Algorithm 2.1. A smaller ν facilitates scaling and results in more timely scaling and a smaller error, while a larger q can scale the basis functions in a more "continuous" manner and let the scaling factor better match the function. That is, setting $q \lesssim 1$ and $\nu \gtrsim 1$ will be a good choice for the scaling technique.

ν \ q	0.6	0.8	0.9	0.95
1.05	5.7526e-07 0.3888	5.8297e-07 0.3221	5.2818e-07 0.2659	5.2500e-07 0.3663
q^{-1}	6.9708e-07 0.2333	5.8293e-07 0.3221	5.3177e-07 0.2954	5.2488e-07 0.3663
$\sqrt{2}$	6.4207e-07 0.3888	5.8117e-07 0.2557	5.2836e-07 0.2954	5.2640e-07 0.3855
$\sqrt{3}$	7.1155e-07 0.3888	6.1260e-07 0.3221	5.4353e-07 0.3283	5.2370e-07 0.3855
2	7.5526e-07 0.3888	6.5640e-07 0.3221	5.5429e-07 0.3647	5.2492e-07 0.4058

TABLE 4

Errors in Λ_e (upper-right) and x_L (bottom-left) at $t = 5$ for different μ and δ in solving (7.13) with Algorithm 3.1 and a fixed $d_{\max} = 0.05$. Increasing μ makes the moving less sensitive to translation and results in a smaller x_L given the same δ. On the other hand, increasing δ to be too large overincreases x_L and leads to unnecessary additional computational cost. So setting $\mu \gtrsim 1$ and $\delta \ll 1$ could work well for the moving.

μ \ δ	0.0005	0.001	0.002	0.005
1.0001	2.0687e-07 9.9985	2.0687e-07 9.9990	2.0687e-07 10.0000	8.3063e-07 25.0000
1.0002	5.2547e+12 3.0320	2.0687e-07 9.9990	2.0687e-07 10.0000	8.3063e-07 25.0000
1.0005	8.9856e+16 0.3635	9.3629e+15 1.0420	2.0688e-07 9.9980	8.3062e-07 24.9950
1.001	1.1390e+17 0.1745	6.2077e+16 0.3650	5.3087e+15 1.0880	8.3061e+08 24.9900

there that given the same q, the numerical solution tends to be less accurate with a larger ν. This is because if we lift the scaling threshold, the time to do scaling will be delayed and error will accumulate. On the other hand, if we compare errors in each row of Table 3, it can be discovered that with the same ν, larger q will give better results, since adjusting scaling factor in a more "continuous" manner could prevent overadjusting the scaling factor within one single time step or failing to adjust the scaling factor timely, and therefore adjust the scaling factor to best match diffusivity of the solution. Therefore, we could conclude that setting $q \lesssim 1$ and $\nu \gtrsim 1$ for the scaling technique would generate good results.

Example 10. In this example we would like to test how different choices of μ and δ affect Algorithm 3.1's capability in capturing translation through solving

$$\begin{cases} \partial_t u + 2\partial_x u = 0, \\ u(x,0) = \dfrac{1}{1+\mathrm{e}^{\frac{x}{2}}}. \end{cases} \tag{7.13}$$

Its analytic solution is $u(x,t) = 1/(1+\mathrm{e}^{\frac{x-2t}{2}})$ and moves right at the speed of 2. Table 4 records x_L in the bottom-left and the errors in Λ_e in the upper-right at $t = 5$, where the improved Euler scheme is used in EVOLVE of Algorithm 3.1 and we have set $N = 40$, $\Delta t = 0.001$, and $\beta_0 = 3$. It should be noted that we have fixed $d_{\max} = 0.05$, which is much larger than the actual translation with one single time step: 0.002, for the purpose of testing. According to the numerical results in

Table 4, we observed that, when fixing δ, increasing μ makes the moving less sensitive to translation and eventually fail to capture the translation, which leads to huge errors and is not desired. On the other hand, by fixing μ and comparing x_L in each row for different δ, we discovered that increasing δ allows x_L to increase more in a single time step and leads to a larger x_L. Yet when μ is too small or δ is too large, the moving mechanism may generate an x_L which is larger than real translation, which will result in an unnecessarily large interior domain $(0, x_L)$ and efforts in solving the interior domain problem in Λ_i as more nodes are required in Λ_i. Hence we can set $\mu \gtrsim 1$ and $\delta \ll 1$ in implementing Algorithm 3.1 for achieving more accurate capture of translation.

8. Applications to structured cell population models. One example of an application requiring the solution of PDEs in an unbounded domain is the structured population models that track populations of cells endowed with attributes such as their size. The standard sizer-timer model for the density of cells with age near a and size near x is formulated in [11], and generalizations to include stochasticity in growth rate are studied in [19, 4]. Here we address a continuum model describing a stochastic model for cell populations [24]:

$$\frac{\partial n}{\partial t} + \frac{\partial n}{\partial a} + \frac{\partial (ng)}{\partial x} - \frac{1}{2}\frac{\partial^2(\sigma^2 n)}{\partial x^2} = -D(x,a,t)n(x,a,t), \quad (x,a) \in \Lambda \times \Lambda, \tag{8.1}$$

where $n(a,x,t)$ describes the density of cells with respect to age a and size x at time t, $g(a,x,t)$ is the mean growth rate of an individual cell, and $\sigma^2(a,x,t)$ is the variance of stochasticity in the growth rate, i.e., $\mathrm{d}x = g\mathrm{d}t + \sigma \mathrm{d}B_t$, for an individual cell. The fluctuating growth rate manifests itself as a diffusive term. The right-hand side of (8.1) represents cell division occurring with division rate $D(x,a,t)$. Dirichlet boundary conditions are imposed at $x = 0$, $n(0,a,t) = n_0(a,t)$, and at $x = +\infty$, $n(+\infty,a,t) = 0$ if we assume that there are no cells of infinite size. More importantly, the boundary condition at $a = 0$ should account for two daughter cells (one of size x and one of size $y - x$) from the binary fission of a mother cell of size $y > x$:

$$n(x,0,t) = 2\int_0^{+\infty} \mathrm{d}a \int_x^{+\infty} \mathrm{d}y\, \tilde{D}(y,x,a,t)n(y,a,t), \tag{8.2}$$

where $\tilde{D}(y,x,a,t)$ is the differential division rate representing the rate that a cell of age a and size y gives birth to a daughter cell of size $x < y$. Integrating over the daughter cell's size x, D and $\tilde{D}$ satisfy $D(y,a,t) = \int_0^y \tilde{D}(y,x,a,t)\mathrm{d}x$, reflecting cell number conservation. Finally, to maintain biomass conservation during division, $\tilde{D}(y,x,a,t) = \tilde{D}(y-x,x,a,t)$. The prefactor 2 in (8.2) indicates that a cell of size y gives birth to one daughter cell of size $y - x$ and another of size x.

The nonlocal boundary condition (8.2) for cell proliferation plays an essential role in depicting how cell division affects the cell population size and age structure and presents a major obstacle in numerical computation as the integration is taken in the unbounded domain $(x, +\infty) \times (0, +\infty)$. Another numerical challenge arises from a possible blowup behavior in which

$$\lim_{t\to+\infty} \langle x(t) \rangle = \lim_{t\to+\infty} \frac{\int_0^{+\infty}\int_0^{+\infty} xn(a,x,t)\mathrm{d}a\mathrm{d}x}{\int_0^{+\infty}\int_0^{+\infty} n(a,x,t)\mathrm{d}a\mathrm{d}x} = +\infty. \tag{8.3}$$

Whether blowup can occur is of biological interest [3, 25] and has been predicted within certain cell proliferation models (8.1) under specific conditions [3].

Existing numerical methods such as the finite volume method in [25] typically truncate the unbounded domain into a bounded domain and therefore cannot accurately capture long time blowup behavior of $\langle x(t)\rangle$. The need for numerical solutions in the unbounded domain $\Lambda \times \Lambda$ for (8.1) and (8.2) is thus evident. We apply the scaling technique built in to Algorithm 2.1 only in the x-dimension for tracking the increasing $\langle x(t)\rangle$, considering the age distribution is often presumed to be stable since no cell could live too long without division. A standard two-dimensional pseudospectral method with the generalized Laguerre functions is used in (a, x)-space, coupled with a third-order TVD Runge–Kutta time discretization in t.

Example 11. We solve (8.1) and (8.2) with $g(x, a, t) = t+7$, $\sigma^2(x, a, t) = 2(t+6)x$, $D(x, a, t) = x/(t + 5)$, $\tilde{D}(y, x, a, t) = 1/(t + 5)$. These parameters lead to the analytic solution $n(x, a, t) = e^t e^{-2a} \exp(-x/(5 + t))$, which produces the mean size $\langle x(t)\rangle = 5+t$. This result shows that the average size is unbounded as it grows linearly in time and thus, for general cases, requires proper scaling in the x-dimension. Here we adopt the same expansion order N in both size x- and age a-dimensions. For the nonlocal boundary condition given in (8.2), we also use $N + 1$ Laguerre–Robatto collocation points in each dimension to perform the numerical integration.

Figure 8 presents the numerical results with the initial scaling factors $(\beta_x, \beta_a) = (0.9, 1)$ and a time step of 0.002. We observe that the frequency-dependent scaling algorithm, Algorithm 2.1, in the x-dimension shows a faster spectral convergence with N than that of the unscaled algorithm (see Figure 8(a)). That is, both the sizer-timer

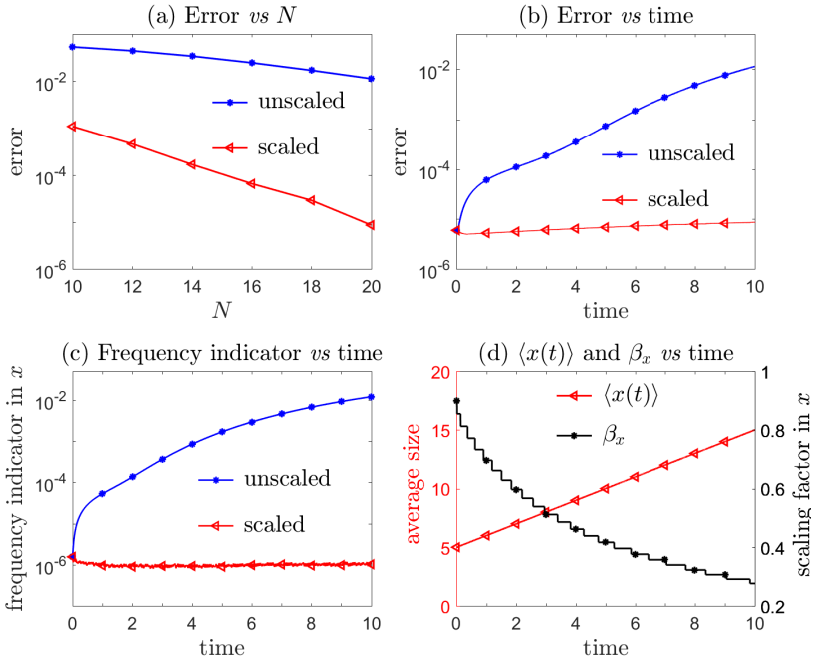

FIG. 8. *Numerical results obtained by the scaling algorithm, Algorithm* 2.1*, for the structured cell population proliferation model* (8.1) *with the nonlocal boundary* (8.2)*: The scaled method gives better results than the unscaled one till* $t = 10$*. The latter experiences a growth in error because inappropriate scaling factors are used, whereas the former gains a faster spectral convergence in the expansion order* N*. We adopt the same* N *in both size* x*- and age* a*-dimensions and set* $N = 20$ *for the last three plots. The frequency-dependent scaling is applied only in the* x*-dimension for tracking the blowup behavior in* (8.3)*. The frequency indicator in the* x*-dimension is kept around* 10^{-6} *through constantly shrinking the scaling factor* β_x *to capture the blowup. The average size of the scaled solution is in good agreement with that of the analytical solution, i.e.,* $\langle x(t)\rangle = 5 + t$*.*

model (8.1) in unbounded domain and the nonlocal boundary condition (8.2) are well resolved by the Laguerre spectral approximation with frequency-dependent scaling. When fixing $N = 20$, the unscaled numerical solution experiences an error growth to 1.143e-02 till $t = 10$ for using inappropriate scaling factors, whereas the error of the scaled solution is less than 8.662e-06 (see Figure 8(b)). The frequency indicator in the x-dimension is kept around 10^{-6} (red curve with left-pointing triangles in Figure 8(c)) by continuously shrinking the scaling factor β_x from 0.9 to 0.2766 for tracking the blowup (black curve with asterisks in Figure 8(d)). The average size of the scaled solution behaves almost exactly like $\langle x(t)\rangle = 5 + t$ and the value at $t = 10$ is 15.001 (see the red curve with left-pointing triangles in Figure 8(d)). Note that the scaling in the a-dimension will really not be triggered even when we apply the scaling algorithm for both x- and a-dimensions.

9. Summary and conclusions. The key to making spectral approximations in unbounded domains more efficient is to allocate collocation points in an economical manner such that crucial regimes of unknown solutions can be resolved accurately. This is essentially an adaptive numerical method for PDEs in unbounded domains, for which there are very few studies compared with its bounded-domain counterpart. Using the standard language of adaptive methods, the proposed scaling technique based on the frequency indicator can be regarded as r-adaptivity to redistribute collocation points via adjusting the scaling factor, while the proposed moving technique based on the exterior-error indicator is similar to h-adaptivity to add collocation points in the interior subdomain. Both indicators utilize only the numerical solution and do not require prior knowledge of unknown solutions.

A promising direction will be to extend the scaling and moving techniques to hyperbolic cross spaces [18] which may significantly reduce the cost of numerical simulations in high dimensions. We are also carrying out more rigorous numerical analysis of the proposed techniques as well as more detailed discussion on how to choose the related parameters in solving different problems. Finally, apart from the proposed scaling and moving techniques, a p-adaptive technique [26] which can be applied to time-dependent problems with oscillatory behavior will also be investigated.

Acknowledgments. The authors are sincerely grateful to the handling editor and referees for their patience and very valuable suggestions.

REFERENCES

[1] E. Bernard, M. Doumic, and P. Gabriel, *Cyclic asymptotic behaviour of a population reproducing by fission into two equal parts*, Kinet. Relat. Modles., 12 (2019), pp. 551–571.
[2] N. Boussaid and A. Comech, *Nonlinear Dirac Equation: Spectral Stability of Solitary Waves*, AMS, Providence, RI, 2019.
[3] S. Burov and D. Kessler, *Effective potential for cellular size control*, Bull. Am. Phys. Soc., 63 (2018).
[4] H. Caswell, *Sensitivity analysis of the stochastic growth rate: three extensions*, Aust. N. Z. J. Stati., 47 (2005), pp. 75–85.
[5] O. Coulaud, D. Funaro, and O. Kavian, *Laguerre spectral approximation of elliptic problems in exterior domains*, Comput. Methods Appl. Mech. Engrg., 80 (1990), pp. 451–458.
[6] J. Cuevas–Maraver, P. G. Kevrekidis, A. Saxena, A. Comech, and R. Lan, *Stability of solitary waves and vortices in a 2D nonlinear Dirac model*, Phys. Rev. Lett., 116 (2016), p. 214101.
[7] B. Y. Guo, L. L. Wang, and Z. Q. Wang, *Generalized Laguerre interpolation and pseudospectral method for unbounded domains*, SIAM J. Numer. Anal., 43 (2006), pp. 2567–2589.
[8] T. Y. Hou and R. Li, *Computing nearly singular solutions using pseudo-spectral methods*, J. Comput. Phys., 226 (2007), pp. 379–397.

[9] R. Li, T. Tang, and P. Zhang, *Moving mesh methods in multiple dimensions based on harmonic maps*, J. Comput. Phys., 170 (2001), pp. 562–588.
[10] H. Ma, W. Sun, and T. Tang, *Hermite spectral methods with a time-dependent scaling for parabolic equations in unbounded domains*, SIAM J. Numer. Anal., 43 (2005), pp. 58–75.
[11] J. A. J. Metz and O. Diekmann, *The Dynamics of Physiologically Structured Populations*, Springer, New York, 1986.
[12] S. A. Orszag, *On the elimination of aliasing in finite-difference schemes by filtering high-wavenumber components*, J. Atmos. Sci., 28 (1971), pp. 1074–1074.
[13] W. Ren and X.-P. Wang, *An iterative grid redistribution method for singular problems in multiple dimensions*, J. Comput. Phys., 159 (2000), pp. 246–273.
[14] S. Shao, N. R. Quintero, F. G. Mertens, F. Cooper, A. Khare, and A. Saxena, *Stability of solitary waves in the nonlinear Dirac equation with arbitrary nonlinearity*, Phys. Rev. E, 90 (2014), 032915.
[15] S. Shao and H. Tang, *Interaction for the solitary waves of a nonlinear Dirac model*, Phys. Lett. A, 345 (2005), pp. 119–128.
[16] J. Shen, T. Tang, and L. L. Wang, *Spectral Methods: Algorithms, Analysis and Applications*, Springer, New York, 2011.
[17] J. Shen and L. L. Wang, *Some recent advances on spectral methods for unbounded domains*, Commun. Comput. Phys., 5 (2009), pp. 195–241.
[18] J. Shen and L. L. Wang, *Sparse spectral approximations of high-dimensional problems based on hyperbolic cross*, SIAM J. Numer. Anal., 48 (2010), pp. 1087–1109.
[19] D. Steinsaltz, S. Tuljapurkar, and C. Horvitz, *Derivatives of the stochastic growth rate*, Theor. Popul. Biol., 80 (2011), pp. 1–15.
[20] H. Tang and T. Tao, *Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws*, SIAM J. Numer. Anal., 41 (2003), pp. 487–515.
[21] T. Tang, *The Hermite spectral method for Gaussian-type functions*, SIAM J. Sci. Comput., 14 (1993), pp. 594–606.
[22] T. Tang, L.-L. Wang, H. Yuan, and T. Zhou, *Rational spectral methods for PDEs involving fractional Laplacian in unbounded domains*, SIAM J. Sci. Comput., 42 (2020), pp. A585–A611.
[23] T. Tang, H. Yuan, and T. Zhou, *Hermite spectral collocation methods for fractional PDEs in unbounded domain*, Commun. Comput. Phys., 24 (2018), pp. 1143–1168.
[24] M. Xia and T. Chou, *Kinetic theory for stochastic sizer-timer models cell size control*, J. Phys. A: Math. Theor., to appear.
[25] M. Xia, C. D. Greenman, and T. Chou, *PDE models of adder mechanisms in cellular proliferation*, SIAM J. Appl. Math., 80 (2020), pp. 1307–1335.
[26] M. Xia, S. Shao, and T. Chou, *A frequency-dependent p-adaptive technique for spectral methods*, J. Comput. Phys., 446 (2021), 110627.
[27] J. Xu, S. Shao, H. Tang, and D. Wei, *Multi-hump solitary waves of a nonlinear Dirac equation*, Commun. Math. Sci., 13 (2015), pp. 1219–1242.
[28] H. Yuan, *An efficient spectral-Galerkin method for fractional reaction-diffusion equations in unbounded domains*, J. Comput. Phys., 428 (2020), 110083.
[29] L. Zheng and X. Zhang, *Variational iteration method and homotopy perturbation method*, Modeling and Analysis of Modern Fluid Problems, Elsevier, Amsterdam, 2017, pp. 253–278.