Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 216 (2006) 326–336
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A fast algorithm for convolution integrals with space and time variant kernels

Erez Gilad [a,b,*,1], Jost von Hardenberg [c,d]

[a] Department of Physics, Ben Gurion University, Ben Gurion Road, P.O. Box 653, Beer Sheva 84105, Israel
[b] Department of Solar Energy and Environmental Physics, BIDR, Ben Gurion University, Sede Boker Campus 84990, Israel
[c] ISAC–CNR, Corso Fiume 4, I-10133 Torino, Italy
[d] CIMA, Università di Genova, via Cadorna 7, I-17100 Savona, Italy

Received 28 September 2005; received in revised form 5 December 2005; accepted 5 December 2005
Available online 19 January 2006

Abstract

Mathematical models in many fields of the physical sciences involve nonlocal terms which are formally similar to convolution integrals. We show that it is possible to approximate a particular class of such integrals, which by themselves are not convolutions, as a linear combination of convolution integrals, allowing for their efficient numerical computation as an $\mathcal{O}(N \log N)$ process.

PACS: 02.30.Jr; 02.60.Nm; 02.70.Hm; 02.30.Mv

Keywords: Convolution; Space and time variant kernels; Nonlocal terms; Integro-differential equations

1. Introduction

The numerical integration of convolution integrals was made very efficient by the introduction of the fast Fourier transform algorithm [1], allowing for their evaluation as an $\mathcal{O}(N \log N)$ process by the use of the convolution theorem [2,3]. This has led, since the mid-1960s, to a large number of applications in various areas of the physical sciences. In particular, the integration of partial differential equations containing nonlocal integral terms (i.e., integro-differential equations) has profited from this technique: examples include mathematical biology [4–6], population dynamics [7,8], nonlinear optics [9–11], epidemics [12,13] and superfluidity [14,15]. There are cases though, where the nonlocal terms appearing in the problem are not convolutions, even if formally very similar, preventing this efficient approach.

* Corresponding author. Tel.: +972 8 6461853/5 44640521; fax: +972 8 6472904.
E-mail addresses: gilade@bgu.ac.il (E. Gilad), j.vonhardenberg@isac.cnr.it (J. von Hardenberg).
URL: http://physics.bgu.ac.il/~gilade (E. Gilad).
[1] E. Gilad was partially supported by the James S. McDonnell Foundation (Grant No. 220020056) and by ISAC–CNR and CIMA.

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.12.003

One particular example of such a situation, which motivated this work, is given by the integral terms appearing in [16–19], describing a two-dimensional spatially extended dynamical model for water-vegetation interactions in drylands. The nonlocal terms in that model are of the form

$$I_1(\mathbf{r}) = \int_D \mathscr{K}[\mathbf{r}-\mathbf{r}';\phi(\mathbf{r}')]\psi(\mathbf{r}')\,\mathrm{d}\mathbf{r}' \tag{1}$$

or

$$I_2(\mathbf{r}) = \int_D \mathscr{K}[\mathbf{r}-\mathbf{r}';\phi(\mathbf{r})]\psi(\mathbf{r}')\,\mathrm{d}\mathbf{r}', \tag{2}$$

where, in that example, ψ represents either a biomass density field (in the case of Eq. (1)) or a soil-water density field (in the case of Eq. (2)), and the integration is taken over a domain D (the time dependence of the fields is omitted for convenience). The kernel function $\mathscr{K}$ represents in this example the effect of the root system and was chosen as a two-dimensional Gaussian with a characteristic width ϕ, which is a linear function of the biomass density. This feature models the dependence of the root system extension on biomass (the larger the biomass, the longer its roots). The scalar field $I_1(\mathbf{r})$ describes the consumption rate of soil water at a point $\mathbf{r}$ by the biomass, whereas the field $I_2(\mathbf{r})$ describes the growth rate of a biomass patch at a point $\mathbf{r}$. Both integrals $I_1(\mathbf{r})$ and $I_2(\mathbf{r})$ cannot be written as convolutions due to the kernel dependence on ϕ, which implies a dependence of the kernel on either $\mathbf{r}$ or $\mathbf{r}'$. Similar integrals also arise in the fields of image processing [20,21] and of nonstationary linear signal filtering [22].

In this paper, we present a fast algorithm for approximating nonlocal terms of the form of Eqs. (1) and (2) for different kernel functions, including the examples of Exponential, Gaussian, and Lorentzian kernels, in one, two and three dimensions. The paper is organized as follows: in Section 2 we present schematically the algorithm with some remarks on its computational cost. Due to the heuristic approach of this section, some aspects of the algorithm are discussed briefly, and a more detailed description is deferred to subsequent sections. In Sections 3 and 4 we show in detail how the approximation coefficients required by the algorithm can be computed, with a few specific examples. The accuracy which can be achieved using this algorithm and the relevant factors controlling the error in the approximation are discussed in Section 5. In Section 6 we show some examples of numerical performances and we conclude with some final remarks in Section 7.

2. The algorithm

Our goal is to evaluate integral terms of the form of Eq. (1) or (2), where both ϕ and ψ are known scalar fields over a $\mathscr{D}$-dimensional domain D. We assume that D allows for the computation of Fourier transforms and that the kernel function $\mathscr{K}$ is well defined in the sense that its integral over D converges.

Clearly, unless the dependence on ϕ can be factored out of the kernel, in general the integrals in Eq. (1) or (2) are not convolutions since, due to the dependence of the field ϕ on $\mathbf{r}$ or $\mathbf{r}'$, $\mathscr{K}$ is not a function of $\mathbf{r}-\mathbf{r}'$ alone. This prevents us from using the convolution theorem to evaluate the terms $I_1(\mathbf{r})$ and $I_2(\mathbf{r})$ trivially as a product of two fields in Fourier space. The main idea described in this paper is to approximate the kernel, for any value of ϕ, as a linear superposition of instances of the same kernel, evaluated at fixed values of ϕ. This shifts the implicit kernel dependence on space to the linear approximation coefficients, thus enabling the evaluation of the integral as a linear combination of convolutions.

To be more specific, we write the following approximation to the kernel function $\mathscr{K}$:

$$\mathscr{K}(\mathbf{s};\phi) \approx \widetilde{\mathscr{K}}(\mathbf{s};\phi) \equiv \sum_{l=1}^{N_l} \alpha_l(\phi)\mathscr{K}(\mathbf{s};\phi_l), \tag{3}$$

where N_l is the number of terms in the approximation, $\{\phi_l\}$ is a series of constants and $\alpha_l(\phi)$ are unknown functions of ϕ. Note that the basis functions in this expansion are given by the same kernel function we wish to approximate, computed at fixed values of ϕ. This is a natural choice, as it allows for an exact reproduction of the kernel whenever $\phi = \phi_l$ (see Section 5 for more details). Since the field ϕ depends on space, the approximation coefficients, being functions of ϕ, have an implicit dependence on space, $\alpha_l(\phi) = \alpha_l[\phi(\mathbf{r})]$ or $\alpha_l(\phi) = \alpha_l[\phi(\mathbf{r}')]$. As we will show in Section 3, optimal coefficients $\alpha_l(\phi)$ can be easily computed by minimization of

an appropriately defined error, and, as we will show in Section 4, in some cases exact closed form formulas can be derived. The series $\{\phi_l\}$ and the number of terms used in the approximation, N_l, represent free parameters of the approximation which should be chosen according to the distribution of ϕ values in the specific problem: we will discuss this issue in Section 5.

For simplicity we start our illustration with the case of Eq. (1), when ϕ is a function of $\mathbf{r}'$. Substituting Eq. (3) into Eq. (1) gives

$$I_1(\mathbf{r}) \approx \int_D \left[\sum_{l=1}^{N_l} \alpha_l(\phi)\mathscr{K}(\mathbf{r}-\mathbf{r}';\phi_l)\right]\psi(\mathbf{r}')\,\mathrm{d}\mathbf{r}' = \sum_{l=1}^{N_l}\int_D \mathscr{K}(\mathbf{r}-\mathbf{r}';\phi_l)\alpha_l(\phi)\psi(\mathbf{r}')\,\mathrm{d}\mathbf{r}', \tag{4}$$

which we rewrite as

$$I_1(\mathbf{r}) \approx \sum_{l=1}^{N_l}\int_D f_l(\mathbf{r}-\mathbf{r}')g_l(\mathbf{r}')\,\mathrm{d}\mathbf{r}' = \sum_{l=1}^{N_l} f_l * g_l, \tag{5}$$

where

$$f_l(\mathbf{r}-\mathbf{r}') = \mathscr{K}(\mathbf{r}-\mathbf{r}';\phi_l), \quad g_l(\mathbf{r}') = \alpha_l[\phi(\mathbf{r}')]\psi(\mathbf{r}') \tag{6}$$

and the operator "$*$" denotes a convolution of two fields over the domain D. As shown by Eq. (5), the field $I_1(\mathbf{r})$ is now approximated by a sum of N_l convolutions which can be efficiently evaluated in Fourier space thanks to the convolution theorem.

In case the field ϕ in the kernel depends on $\mathbf{r}$ instead on $\mathbf{r}'$, as in Eq. (2), the resulting approximation can be similarly written as

$$I_2(\mathbf{r}) \approx \sum_{l=1}^{N_l}\alpha_l(\phi)\int_D f_l(\mathbf{r}-\mathbf{r}')g_l'(\mathbf{r}')\,\mathrm{d}\mathbf{r}' = \sum_{l=1}^{N_l}\alpha_l[\phi(\mathbf{r})](f_l * g_l'), \tag{7}$$

where now

$$f_l(\mathbf{r}-\mathbf{r}') = \mathscr{K}(\mathbf{r}-\mathbf{r}';\phi_l), \quad g_l'(\mathbf{r}') = \psi(\mathbf{r}'). \tag{8}$$

The algorithm's different steps can be summarized as follows, using the convolution theorem for evaluating the integrals:

Algorithm 1. Approximation of convolution integrals with space–time variant kernels:

(1) Choose N_l constants $\{\phi_l\}$;
(2) Compute the approximation coefficients $\alpha_l(\phi)$ as described in Section 3;
(3) Compute

$$I_1(\mathbf{r}) \approx \mathscr{F}^{-1}\left[\sum_{l=1}^{N_l}\mathscr{F}(f_l)\mathscr{F}(g_l)\right] \tag{9}$$

or

$$I_2(\mathbf{r}) \approx \mathscr{F}^{-1}\left[\sum_{l=1}^{N_l}\alpha_l(\phi)\mathscr{F}(f_l)\mathscr{F}(g_l')\right], \tag{10}$$

where the operators $\mathscr{F}$ and $\mathscr{F}^{-1}$ indicate a direct and inverse Fourier transform, respectively, and the fields f_l, g_l and g_l' are defined above.

In case the evaluation of $I_1(\mathbf{r})$ or $I_2(\mathbf{r})$ needs to be iterated, as is usually the case when the problem is time dependent, some of the terms above need to be computed only once. In particular, since f_l is simply the kernel function evaluated at a fixed ϕ_l, its Fourier transform $\mathscr{F}(f_l)$ can be evaluated once for all at the beginning of the code. In case the evaluation of the coefficients $\alpha_l(\phi)$ is time consuming, they can be evaluated for discretized values of ϕ once for all at the beginning.

The computational complexity of this algorithm is of order $\mathcal{O}[N_l N \log(N)]$, where N is the number of grid points in the spatial domain of interest; This has to be compared with a complexity of $\mathcal{O}(N^2)$ if a direct, brute force, approach is used. Since a fairly small number of terms in the approximation ($N_l \ll N$) is enough to obtain a good approximation, as is shown in the following, the advantage for large values of N is obvious.

3. Computing the approximation coefficients

We compute optimal approximation coefficients $\alpha_l(\phi)$ for a kernel function $\mathcal{K}(\mathbf{s};\phi)$, by applying a standard minimization procedure to a measure of the approximation error. For any given, fixed, ϕ, we write a measure of the error in the kernel approximation, $\widetilde{\mathcal{K}}(\mathbf{s};\phi)$, as a functional of $\alpha_l(\phi)$ and of N_l:

$$\mathfrak{F}[\alpha_l(\phi), N_l] \equiv \int_\Phi \int_D \left[\widetilde{\mathcal{K}}(\mathbf{s};\phi,N_l) - \mathcal{K}(\mathbf{s};\phi)\right]^2 \mathrm{d}\mathbf{s}\,\mathrm{d}\phi, \tag{11}$$

where D is the physical domain and Φ is the domain spanned by the values of ϕ. Having fixed N_l, we wish to minimize the functional $\mathfrak{F}[\alpha_l(\phi)]$ with respect to all α_l. We use the method of variations [23,24], and define a small variation in $\alpha_l(\phi)$ as $\alpha_l(\phi) + \delta\alpha_l(\phi)$. Substituting this into Eqs. (3) and (11) and expanding up to first-order in $\delta\alpha_l(\phi)$ gives

$$\begin{aligned}
\mathfrak{F} + \delta\mathfrak{F} &= \int_\Phi \int_D \left\{\sum_l \{[\alpha_l(\phi) + \delta\alpha_l(\phi)]\mathcal{K}(\mathbf{s};\phi_l)\} - \mathcal{K}(\mathbf{s};\phi)\right\}^2 \mathrm{d}\mathbf{s}\,\mathrm{d}\phi \\
&= \int_\Phi \int_D \left[\sum_l \alpha_l(\phi)\mathcal{K}(\mathbf{s};\phi_l) - \mathcal{K}(\mathbf{s};\phi) + \sum_l \delta\alpha_l(\phi)\mathcal{K}(\mathbf{s};\phi_l)\right]^2 \mathrm{d}\mathbf{s}\,\mathrm{d}\phi \\
&= \int_\Phi \int_D \left[\sum_l \alpha_l(\phi)\mathcal{K}(\mathbf{s};\phi_l) - \mathcal{K}(\mathbf{s};\phi)\right]^2 \mathrm{d}\mathbf{s}\,\mathrm{d}\phi \\
&\quad + 2\int_\Phi \int_D \left[\sum_l \alpha_l(\phi)\mathcal{K}(\mathbf{s};\phi_l) - \mathcal{K}(\mathbf{s};\phi)\right]\left[\sum_j \delta\alpha_j(\phi)\mathcal{K}(\mathbf{s};\phi_j)\right]\mathrm{d}\mathbf{s}\,\mathrm{d}\phi + \mathcal{O}(\delta\alpha^2).
\end{aligned} \tag{12}$$

Hence, we obtain an expression for the variation in $\mathfrak{F}$ with respect to variations in α_l

$$\begin{aligned}
\delta\mathfrak{F} &= 2\int_\Phi \sum_j \left\{\int_D \left[\sum_l \alpha_l(\phi)\mathcal{K}(\mathbf{s};\phi_l)\mathcal{K}(\mathbf{s};\phi_j) - \mathcal{K}(\mathbf{s};\phi)\mathcal{K}(\mathbf{s};\phi_j)\right]\mathrm{d}\mathbf{s}\right\}\delta\alpha_j(\phi)\,\mathrm{d}\phi \\
&= 2\int_\Phi \sum_j \left[\sum_l M_{jl}\alpha_l(\phi) - b_j(\phi)\right]\delta\alpha_j(\phi)\,\mathrm{d}\phi,
\end{aligned} \tag{13}$$

where we defined

$$M_{jl} \equiv \int_D \mathcal{K}(\mathbf{s};\phi_j)\mathcal{K}(\mathbf{s};\phi_l)\,\mathrm{d}\mathbf{s}, \quad b_j(\phi) \equiv \int_D \mathcal{K}(\mathbf{s};\phi)\mathcal{K}(\mathbf{s};\phi_j)\,\mathrm{d}\mathbf{s}. \tag{14}$$

The requirement $\delta\mathfrak{F} = 0$ for any $\delta\alpha_l(\phi)$ yields a set of N_l linear equations for the approximation coefficients

$$\sum_{l=1}^{N_l} M_{jl}\alpha_l(\phi) = b_j(\phi), \quad j = 1,\ldots,N_l, \tag{15}$$

which in some cases can be solved analytically, providing an exact, compact expression for α_l as a function of ϕ. Alternatively this linear system can be solved numerically for discrete values of ϕ.

4. Examples for some common kernel functions

In this section we will demonstrate the method described above to compute the approximation coefficients, applying it to some common kernel functions and deriving analytical expressions for the dependence of α_l on ϕ.

Typical shapes of the resulting approximation coefficients $\alpha_l(\phi)$ as a function of ϕ are summarized in Fig. 1, with different choices of the series $\{\phi_l\}$. Notice the property that, whenever $\phi = \phi_l$, then $\alpha_l(\phi_l) = 1$, while $\alpha_{j\neq l}(\phi_l) = 0$, which results from having used as basis functions for the approximation instances of the same kernel we wish to approximate, computed at $\{\phi_l\}$.

4.1. One-dimensional decaying exponential

We consider a decaying Exponential kernel in an infinite one-dimensional domain, with a characteristic length determined by ϕ

$$\mathcal{K}(s;\phi) = \mathrm{e}^{-\frac{|s|}{\phi}}. \tag{16}$$

We assume that the range of possible ϕ values in the problems is given by $\phi \in \Phi$ ($\phi > 0$). We define a series $\{\phi_l\}_{l=1}^{N_l} \in \Phi$ and approximate the kernel according to Eq. (3)

$$\mathrm{e}^{-\frac{|s|}{\phi}} \approx \sum_{l=1}^{N_l} \alpha_l(\phi)\mathrm{e}^{-\frac{|s|}{\phi_l}}. \tag{17}$$

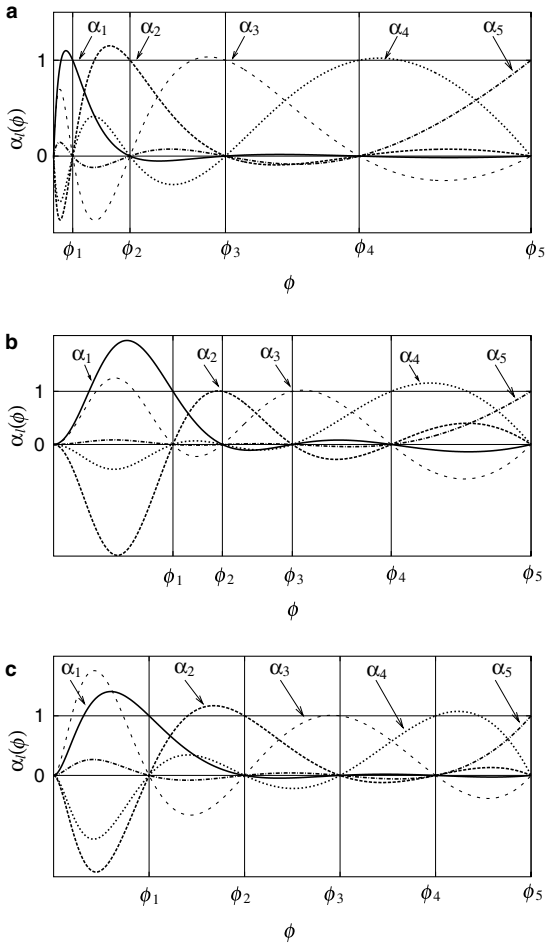

Fig. 1. Typical shapes of the approximation coefficients $\alpha_l(\phi)$ as a function of ϕ for three different kernels and three different choices of the series $\{\phi_l\}$, with $N_l = 5$. The coefficients $\alpha_l(\phi)$ are shown for: (a) one-dimensional decaying Exponential kernel (Eq. (19)), with $\{\phi_l\}$ distributed according to $\phi_l \sim l^2$, (b) two-dimensional Gaussian kernel, (Eq. (23)), with $\{\phi_l\}$ distributed according to $\phi_l \sim 2^{l/2}$ and (c) three-dimensional Lorentzian kernel (Eq. (27)), with $\{\phi_l\}$ distributed according to $\phi_l \sim l$. Notice that whenever $\phi = \phi_l$, $\alpha_l(\phi_l) = 1$, while $\alpha_{j\neq l}(\phi_l) = 0$. All series $\{\phi_l\}$ are distributed over the range $\Phi = [0, 1]$ and are normalized such that $\phi_{N_l} = 1$.

We calculate the entries M_{jl} and $b_j(\phi)$ of the linear system in Eq. (15) using Eq. (14) as follows:

$$\begin{aligned} M_{jl} &= \int_{-\infty}^{\infty} \mathrm{e}^{-\frac{|s|}{\phi_j}} \mathrm{e}^{-\frac{|s|}{\phi_l}} \mathrm{d}s = 2\frac{\phi_j \phi_l}{\phi_j + \phi_l}, \\ b_j(\phi) &= \int_{-\infty}^{\infty} \mathrm{e}^{-\frac{|s|}{\phi}} \mathrm{e}^{-\frac{|s|}{\phi_l}} \mathrm{d}s = 2\frac{\phi \phi_j}{\phi + \phi_j}. \end{aligned} \tag{18}$$

The solution of the linear system equation (15), using these entries, yields the following analytic expression for the approximation coefficients (see Fig. 1(a)):

$$\alpha_l(\phi) = \frac{2\phi}{\phi + \phi_l} \prod_{j \neq l} \frac{\phi - \phi_j}{\phi_l - \phi_j} \frac{\phi_l + \phi_j}{\phi + \phi_j}. \tag{19}$$

4.2. Two-dimensional Gaussian

We consider a Gaussian kernel in an infinite two-dimensional domain, with a characteristic width determined by ϕ,

$$\mathscr{K}(\mathbf{s}; \phi) = \mathrm{e}^{-\frac{|\mathbf{s}|^2}{2\phi^2}}. \tag{20}$$

We assume that $\phi \in \Phi$ ($\phi > 0$) and define the series $\{\phi_l\}_{l=1}^{N_l} \in \Phi$. The kernel can be approximated using Eq. (3)

$$\mathrm{e}^{-\frac{|\mathbf{s}|^2}{2\phi^2}} \approx \sum_{l=1}^{N_l} \alpha_l(\phi) \mathrm{e}^{-\frac{|\mathbf{s}|^2}{2\phi_l^2}}. \tag{21}$$

We calculate the entries M_{jl} and $b_j(\phi)$ of the linear system in Eq. (15) using Eq. (14) as follows:

$$\begin{aligned} M_{jl} &= \int_D \mathrm{e}^{-\frac{|\mathbf{s}|^2}{2\phi_j^2}} \mathrm{e}^{-\frac{|\mathbf{s}|^2}{2\phi_l^2}} \mathrm{d}\mathbf{s} = 2\pi \int_0^{\infty} \mathrm{e}^{-\frac{s^2}{2}\left(\frac{1}{\phi_j^2} + \frac{1}{\phi_l^2}\right)} s \, \mathrm{d}s = 2\pi \frac{\phi_j^2 \phi_l^2}{\phi_j^2 + \phi_l^2}, \\ b_j(\phi) &= \int_D \mathrm{e}^{-\frac{|\mathbf{s}|^2}{2\phi^2}} \mathrm{e}^{-\frac{|\mathbf{s}|^2}{2\phi_j^2}} \mathrm{d}\mathbf{s} = 2\pi \frac{\phi^2 \phi_j^2}{\phi^2 + \phi_j^2}. \end{aligned} \tag{22}$$

The solution for the linear system given in Eq. (15), using these entries yields the following analytic expression for the approximation coefficients (see Fig. 1(b)):

$$\alpha_l(\phi) = \frac{2\phi^2}{\phi^2 + \phi_l^2} \prod_{j \neq l} \frac{\phi^2 - \phi_j^2}{\phi_l^2 - \phi_j^2} \frac{\phi_l^2 + \phi_j^2}{\phi^2 + \phi_j^2}. \tag{23}$$

4.3. Three-dimensional Lorentzian

We consider a Lorentzian kernel in an infinite three-dimensional domain, with a characteristic width determined by ϕ

$$\mathscr{K}(\mathbf{s}; \phi) = \frac{1}{1 + \frac{|\mathbf{s}|^2}{\phi^2}}. \tag{24}$$

We assume that $\phi \in \Phi$ ($\phi > 0$) and define the series $\{\phi_l\}_{l=1}^{N_l} \in \Phi$. We approximate the kernel according to Eq. (3)

$$\frac{1}{1 + \frac{|\mathbf{s}|^2}{\phi^2}} \approx \sum_{l=1}^{N_l} \alpha_l(\phi) \frac{1}{1 + \frac{|\mathbf{s}|^2}{\phi_l^2}}. \tag{25}$$

We calculate the entries M_{jl} and $b_j(\phi)$ of the linear system in Eq. (15) using Eq. (14) as follows:

$$M_{jl} = \int_D \frac{1}{1+\frac{|\mathbf{s}|^2}{\phi_j^2}} \frac{1}{1+\frac{|\mathbf{s}|^2}{\phi_l^2}} \mathrm{d}\mathbf{s} = 4\pi \int_0^\infty \frac{1}{1+\frac{s^2}{\phi_j^2}} \frac{1}{1+\frac{s^2}{\phi_l^2}} s^2 \,\mathrm{d}s = \frac{\pi}{2} \frac{\phi_j^2 \phi_l^2}{\phi_j + \phi_l},$$
$$b_j(\phi) = \int_D \frac{1}{1+\frac{|\mathbf{s}|^2}{\phi^2}} \frac{1}{1+\frac{|\mathbf{s}|^2}{\phi_j^2}} \mathrm{d}\mathbf{s} = \frac{\pi}{2} \frac{\phi^2 \phi_j^2}{\phi + \phi_j}. \tag{26}$$

The solution for the linear system given in Eq. (15), using the entries calculated in Eq. (26), yields the following analytic expression for the approximation coefficients (see Fig. 1(c)):

$$\alpha_l(\phi) = \frac{2\phi^2}{\phi_l(\phi+\phi_l)} \prod_{j\neq l} \frac{\phi-\phi_j}{\phi_l-\phi_j} \frac{\phi_l+\phi_j}{\phi+\phi_j}. \tag{27}$$

5. Accuracy of the approximation

The accuracy which can be achieved with the algorithm described above depends on an appropriate choice of the distribution of $\{\phi_l\}$ over Φ and of the number of basis function N_l. We illustrate this dependence in the following, by considering separately: (i) the error in approximating the kernel alone using Eq. (3); (ii) the error in approximating integrals of the form Eq. (1), compared with a direct, brute force, algorithm, in a practical application example.

5.1. The error in the kernel approximation

We write the error in the kernel approximation (i.e., Eq. (3)) for an arbitrary ϕ as

$$e^2(\phi; N_l) = \frac{\int_D \left[\widetilde{\mathscr{K}}(\mathbf{s},\phi;N_l) - \mathscr{K}(\mathbf{s},\phi)\right]^2 \mathrm{d}\mathbf{s}}{\int_D \mathscr{K}(\mathbf{s},\phi)^2 \,\mathrm{d}\mathbf{s}}, \tag{28}$$

where here ϕ is regarded as an argument of the error (not a parameter) and the dependence of the error on the parameter N_l is explicitly noted. We illustrate the dependence of the error in the kernel approximation on the particular choice of $\{\phi_l\}$ by fixing $N_l = 12$ and plotting, in Fig. 2, this error as a function of $\phi \in \Phi = (0,1]$. The errors are shown for (a) one-dimensional Exponential kernel (Eq. (17)) and (b) two-dimensional Gaussian kernel (Eq. (21)) using three different distributions of $\{\phi_l\}$ over Φ.

Fig. 2 illustrates how, for a fixed number of basis functions, an appropriate choice of the series $\{\phi_l\}$ can lead to a highly accurate approximation of the kernel over a desired range of ϕ values in a given problem.

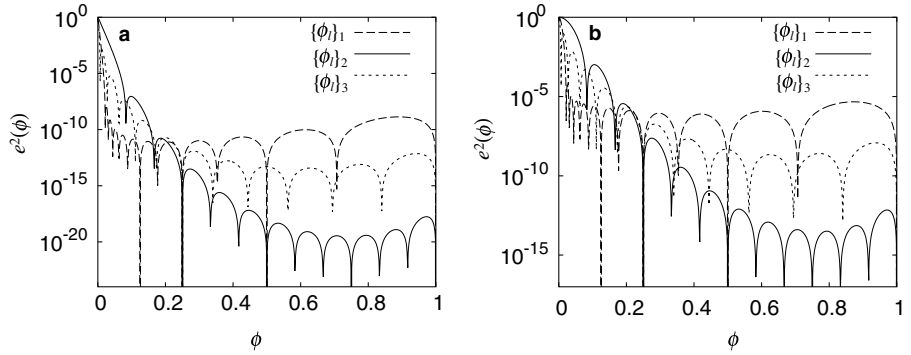

Fig. 2. Plot of the error in the kernel approximation $e^2(\phi)$ (Eq. (28)) as a function of ϕ, for (a) one-dimensional Exponential kernel (Eq. (17)) and (b) two-dimensional Gaussian kernel (Eq. (21)) using three different distributions of $\{\phi_l\}$ over the range $\phi \in (0,1]$. In both cases $N_l = 12$ and the series $\{\phi_l\}$ is distributed according to: $\{\phi_l\}_1 \sim 2^{l/2}$, $\{\phi_l\}_2 \sim l$, $\{\phi_l\}_3 \sim l^2$ and normalized such that $\phi_{N_l} = 1$. Notice that the error drops to 0 whenever $\phi = \phi_l$.

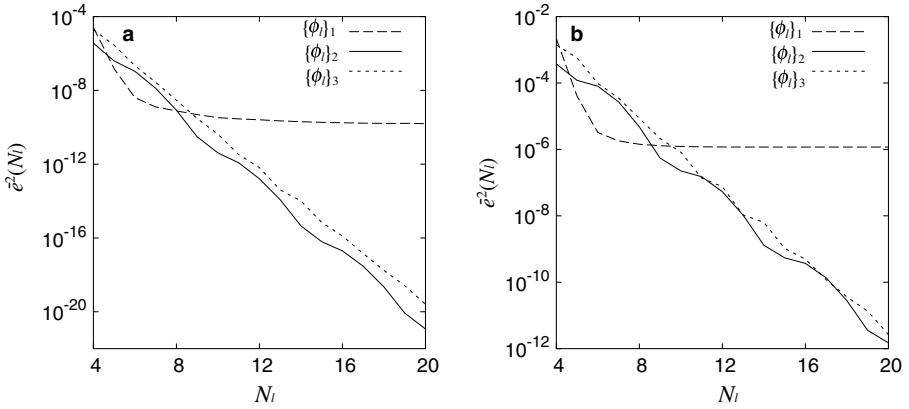

Fig. 3. Plots of the average error in the kernel approximation (Eq. (29)) as a function of N_l, for the cases of (a) one-dimensional Exponential kernel (Eq. (17)) and (b) two-dimensional Gaussian kernel (Eq. (21)), using the same three distributions of $\{\phi_l\}$ over Φ as described in the caption of Fig. 2. In this figure we used $\Phi = [0.2, 1]$ for the evaluation of Eq. (29).

Obviously, no universally valid choice of the distribution of $\{\phi_l\}$ over Φ can be given as this depends on the particular distribution of values of ϕ in the problem at hand. In theory, if the distribution of ϕ in the problem were known in advance, it would be possible to define an optimal set of ϕ_l for a given N_l, but we will not attempt this exercise here.

In order to study the dependence of the kernel approximation accuracy on the number of basis function N_l, we define an average error in the approximation, averaging Eq. (28) over Φ

$$\bar{e}^2(N_l) = \frac{\int_\Phi e^2(\phi; N_l)\,\mathrm{d}\phi}{\int_\Phi \mathrm{d}\phi}. \tag{29}$$

Fig. 3 shows plots of this average error as a function of N_l for the cases of one-dimensional Exponential kernel (Eq. (17)) and two-dimensional Gaussian kernel (Eq. (21)), using the three distributions of $\{\phi_l\}$ described in the caption of Fig. 2. This plot demonstrates the performance of the algorithm, but also the importance of choosing appropriately the distribution ϕ_l: Two of the choices of $\{\phi_l\}$, i.e., $\{\phi_l\}_2 \sim l$ and $\{\phi_l\}_3 \sim l^2$ lead to an error which decreases exponentially in a wide range of N_l whereas in the case $\{\phi_l\}_1 \sim 2^{l/2}$ the error decreases more rapidly initially, but it reaches quite early a plateau.[2]

5.2. The error in the integral approximation

In the following, we explore numerically and in a practical application example, the difference between an approximated integral, $\widetilde{I}(\mathbf{r})$, computed using the approximation procedure described above, and an integral, $I(\mathbf{r})$, computed using an alternative, standard, brute force approach. To this purpose we define an error between the two as

$$e_I^2(N_l) = \frac{\int_D \left[\widetilde{I}(\mathbf{r}; N_l) - I(\mathbf{r})\right]^2 \mathrm{d}\mathbf{r}}{\int_D I(\mathbf{r})^2\,\mathrm{d}\mathbf{r}}. \tag{30}$$

This error is calculated for the case of Eq. (1) with a two-dimensional Gaussian kernel and for the three different distributions of $\{\phi_l\}$ defined above. The field $\psi(\mathbf{r})$ is taken from [17] where it represents a spatial distribution of biomass density and assumes values in the range $\psi \in [0, 1]$. The field $\phi(\mathbf{r})$ in this example repre-

[2] For this particular distribution of ϕ_l, the number of values in the range Φ saturates for $N_l > 5$, with all new values in the sequence being added below the lowest limit of the range Φ: While in general we found that in order to have a good approximation it is necessary to include a few ϕ_l values also outside the range of ϕ, in this case these new values have a negligible effect on the accuracy of the approximation.

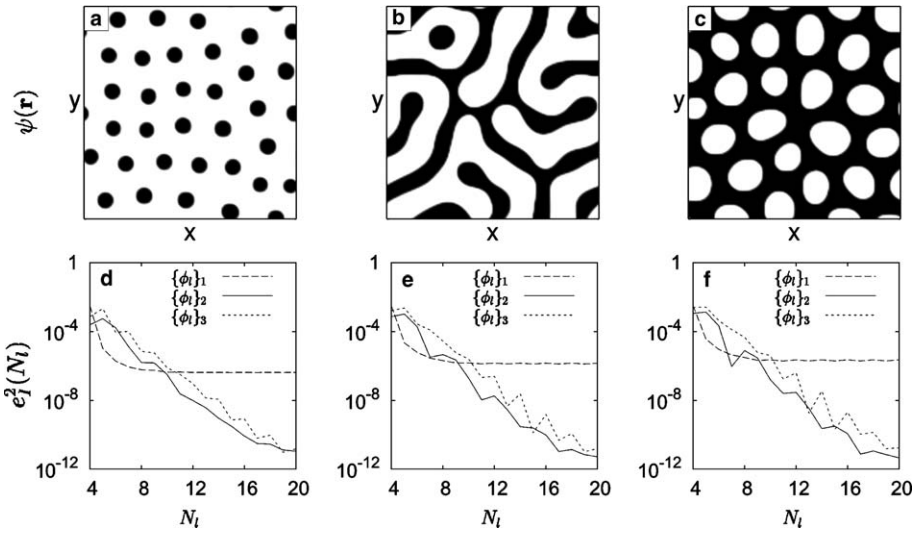

Fig. 4. Three different spatial patterns of the field ψ: (a) spots, (b) stripes and (c) gaps and the corresponding error in the integral approximation $e_I^2(N_l)$ (Eq. (30)) as a function of the number of basis functions N_l (panels (d)–(f)). The field $\psi(\mathbf{r})$ is taken from [17], where it represents the spatial distribution of a biomass density and assumes values over the range $\psi \in [0,1]$ (darker shades of gray represent higher biomass density) and $\phi(\mathbf{r}) = 1 + \psi(\mathbf{r})$. The error $e_I^2(N_l)$ is calculated for each spatial pattern of ψ using a two-dimensional Gaussian kernel and three different distributions of $\{\phi_l\}$, defined previously in the caption of Fig. 2.

sents the root length and is a linear function of the biomass according to $\phi(\mathbf{r}) = 1 + \psi(\mathbf{r})$, so that $\phi \in [1,2]$. We calculate the error defined in Eq. (30) for three different spatial patterns of biomass density shown in the upper panels of Fig. 4: (a) spots, (b) stripes and (c) gaps (see also [25]). The domain size of these patterns is $L = 6$ and the series $\{\phi_l\}$ are chosen in the range $\Phi = (0, L]$, with $\phi_{N_l} = L$.

All continuous fields are discretized over a 128×128 points grid with periodic boundary conditions. The field $\widetilde{I}(\mathbf{r})$ is calculated according to the approximation algorithm described in Section 2, while the field $I(\mathbf{r})$ is calculated using a direct brute-force method of integration, specifically the trapezoidal rule [26], as follows:

$$I(x_i, y_j) = \Delta x \Delta y \sum_{k,l=-N/2}^{N/2} \mathrm{e}^{-\frac{[(i-k)\Delta x]^2 + [(j-l)\Delta y]^2}{2\phi(x_k, y_l)^2}} \psi(x_k, y_l), \quad i, j = 1, \ldots, N, \tag{31}$$

where N is the number of grid points in each direction and $\Delta x = \Delta y = L/N$. This expression can also be considered simply a discrete version, on a finite grid, of Eq. (1) with a Gaussian kernel.

The resulting error is plotted in panels (d)–(f) in Fig. 4 as a function of the number of basis function used in the approximation, N_l. Note that for the choices $\{\phi_l\}_2$ and $\{\phi_l\}_3$ the error can be made arbitrarily small by choosing a small, finite value of N_l. The error for the $\{\phi_l\}_1$ case reaches an asymptotic value as $N_l \sim 8$, for reasons similar to those already discussed in footnote 2 for Fig. 3.

To summarize this section, Figs. 2–4 show that by tuning correctly both the number of basis functions N_l and the distribution of the series $\{\phi_l\}$, a highly accurate approximation can be achieved for both the kernel function and for the integral itself over the desired range of ϕ values in the problem. Furthermore, only a small number of basis functions $N_l \ll N$ is enough to achieve good accuracy.

6. Numerical performance of the algorithm

We demonstrate the advantage of using the approximation described in this paper by comparing the speed of a numerical implementation of the algorithm with a direct brute force approach (see Eq. (31)). We computed the single processor CPU time needed to perform one integration of integral I_1 on two-dimensional grids, ranging in resolution from 64×64 points up to 1024×1024 points, using a Gaussian kernel function, with periodic boundary conditions and setting $N_l = 16$.

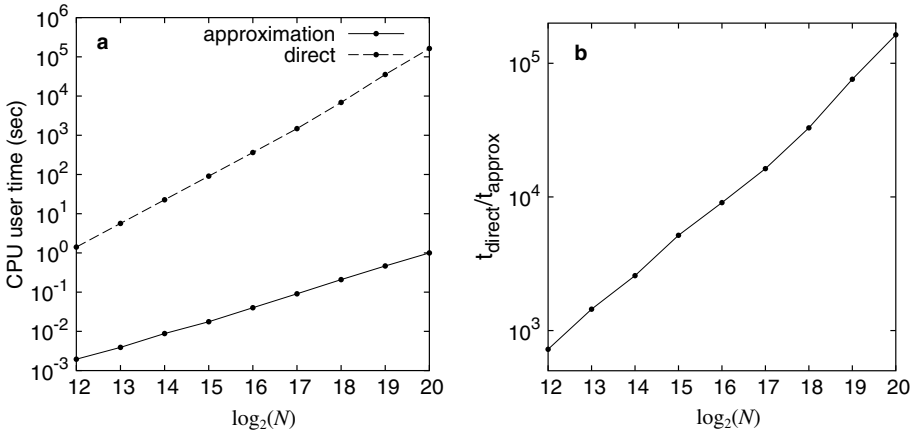

Fig. 5. (a) CPU user time required to complete a single integral calculation as a function of grid size, using two different algorithms: a direct brute-force algorithm (dashed line) and the approximation algorithm (solid line). (b) CPU user time ratio between the two algorithms obtained as a ratio of the two curves in panel (a). The number of basis functions in these tests is $N_I = 16$.

The numerical simulations were run using Fortran implementations of the algorithms, on an SGI-Altix 3700 machine with Intel Itanium2 1.3 GHz processors and 32 GB memory, running 64-bit linux OS [27]. Intel Compilers [28] were used for compilation and the FFTW library [29] was used for the Fourier transforms. All simulations used double precision arithmetic.

Fig. 5(a) shows the CPU user time required to complete a single calculation of the integral as a function of the grid size, using both a direct brute-force algorithm (dashed line) and the approximation algorithm (solid line). Fig. 5(b) shows the ratio between these two curves. Note that the speed-up obtained by using the approximation algorithm with respect to the direct brute-force algorithm ranges between 10^3 and 10^5, whereas the error in the approximation remains small.

7. Conclusions

We have illustrated a method for approximating integrals of the form of Eqs. (1) and (2) as a linear combination of a small number of convolutions. Good accuracy can be achieved with this method, with pronounced advantages in terms of computational complexity and, thus, speed of integration.

The coefficients required for the approximation can be easily obtained analytically or numerically. We developed analytical solutions for the coefficients and tested the method for decaying Exponential, Gaussian and Lorentzian kernels in one, two and three dimensions respectively; also other kernels, in arbitrary $\mathscr{D}$-dimensional domains could be explored.

We believe that the method which we present in this paper can be a useful technique for the numerical integration of nonlocal PDE's and of integro-differential problems containing convolution integrals with space and time variant kernels, given its efficient trade-off between speed and accuracy.

References

[1] J.W. Cooley, O.W. Tukey, An algorithm for the machine calculation of complex Fourier series, Math. Comput. 19 (1965) 297–301.
[2] H.J. Nussbaumer, Fast Fourier Transform and Convolution Algorithms, Springer, New York, 1982.
[3] E.O. Brigham, The Fast Fourier Transform, Prentice-Hall, Englewood Cliffs, NJ, 1974.
[4] F. Bagnoli, M. Bezzi, Speciation as pattern formation by competition in a smooth fitness landscape, Phys. Rev. Lett. 79 (1997) 3302–3305.
[5] A. Mogilner, L. Edelstein-Keshet, A non-local model for a swarm, J. Math. Biol. 38 (1999) 534–570.
[6] C. Topaz, A. Bertozzi, Swarming patterns in a two dimensional kinematic model for biological groups, SIAM J. Appl. Math. 65 (1) (2004) 152–174.
[7] M.A. Fuentes, M.N. Kuperman, V.M. Kenkre, Non-local interaction effects on pattern formation in population dynamics, Phys. Rev. Lett. 91 (2003) 158104-1–158104-4.

[8] J. Billingham, Dynamics of a strongly nonlocal reaction-diffusion population model, Nonlinearity 17 (2004) 313–346.
[9] J.T. Sheridan, J.R. Lawrence, Nonlocal-response diffusion model of holographic recording in photopolymer, J. Opt. Soc. Am. A 17 (2000) 1108–1114.
[10] J.T. Sheridan, M. Downey, F.T. O'Neill, Diffusion-based model of holographic grating formation in photopolymers: generalized non-local material responses, J. Opt. A: Pure Appl. Opt. 3 (2001) 477–488.
[11] W. Krolikowski, O. Bang, J.J. Rasmussen, J. Wyller, Modulational instability in nonlocal nonlinear Kerr media, Phys. Rev. E 64 (2001) 016612-1–016612-8.
[12] J. Medlock, M. Kot, Spreading disease: integro-differential equations old and new, Math. Biosci. 184 (2003) 201–222.
[13] V.M. Kenkre, Results from variants of the Fisher Equation in the study of epidemics and bacteria, Physica A 342 (2004) 242–248.
[14] N.G. Berloff, Nonlocal nonlinear Schrödinger equations as models of superfluidity, J. Low Temp. Phys. 31 (1999) 359–380.
[15] N.G. Berloff, P.H. Roberts, Motions in a Bose condensate: VI. Vortices in a nonlocal model, J. Phys. A: Math. Gen. 32 (1999) 5611–5625.
[16] E. Gilad, J. von Hardenberg, A. Provenzale, M. Shachak, E. Meron, Ecosystem engineers: from pattern formation to habitat creation, Phys. Rev. Lett. 93 (2004) 098105-1–098105-4.
[17] E. Gilad, J. von Hardenberg, A. Provenzale, M. Shachak, E. Meron, A model of plants as ecosystem engineers, Am. Naturalist (submitted for publication).
[18] H. Yizhaq, E. Gilad, E. Meron, Banded vegetation: biological productivity and resilience, Physica A 356 (1) (2005) 139–144.
[19] E. Meron, E. Gilad, Dynamics of plant communities in drylands: a pattern formation approach, in: B. Blasius, J. Kurths, L. Stone (Eds.), Complex Population Dynamics: Nonlinear Modeling in Ecology, Epidemiology and Genetics, World-Scientific, Singapore (in press).
[20] E. Chang, S. Mallat, Wavelet foveation, Appl. Comput. Harmonic Anal. 9 (2000) 312–335.
[21] S. Tan, J.L. Dale, A. Johnston, Performance of three recursive algorithms for fast space-varian Gaussian filtering, Real-Time Imaging 9 (2003) 215–228.
[22] G.F. Margrave, Theory of nonstationary linear filtering in the Fourier domain with application to time-variant filtering, Geophysics 63 (1) (1998) 244–259.
[23] A.R. Forsyth, Calculus of Variations, Dover Publications, New York, 1960.
[24] G. Arfken, Mathematical Methods for Physicists, third ed., Academic Press, Orlando, FL, 1985, pp. 925–962.
[25] J. von Hardenberg, E. Meron, M. Shachak, Y. Zarmi, Diversity of vegetation patterns and desertification, Phys. Rev. Lett. 87 (2001) 198101-1–198101-4.
[26] R.L. Burden, J.D. Faires, Numerical Analysis, seventh ed., Brooks-Cole, Pacific Grove, CA, 2000.
[27] 64-Bit sgi Linux 2.4.21-sgi301r1, SGI ProPack(TM) v3.0 Service Pack 1.
[28] Intel(R) Fortran Itanium(R) Compiler for Itanium(R)-based applications. Version 8.1 (Build 20041021). Intel(R) C++ Itanium(R) Compiler for Itanium(R)-based applications. Version 8.1 (Build 20041021). Optimization flags used for compilation: -O3 -tpp2 -fnsplit -ip -ipo -ftz.
[29] M. Frigo, S.G. Johnson, FFTW: An adaptive software architecture for the FFT, Proc. ICASSP 1998 3 (1998) 1381–1384.