

# Diffusive Realization of a Lyapunov Equation Solution, and Parallel Algorithms Implementation

Huu-Quan Do\*, Michel Lenczner<sup>†</sup>, Raphaël Couturier\* and Youssef Yakoubi<sup>†</sup>

## Abstract

In Yakoubi [11] and Lenczner et al. [5] the authors developed a theoretical framework of diffusive realization for state-realizations of some linear operators. Those are solutions to certain linear operator differential equations posed in one-dimensional bounded domains. They illustrate the theory on a Lyapunov equation arising from the optimal control theory of the heat equation. In principle their method might be very efficient for real-time computations, however it suffers from strong limitations. Here, we present significant improvements and report numerical results. A method of contour optimization is provided. It is based on a theoretical error estimate of the solution. Finally, we discuss expected gains if the method is implemented on different parallel computer topologies. The envisioned applications are for real-time distributed control on distributed computing architectures.

## 1 Introduction

The low computational cost of the diffusive realization (DR) of linear operators has been extensively investigated, see the papers by G. Montseny and of D. Matignon, e.g. Laudebat et al. [3], and Hélie et al. [2], on various pseudodifferential operators. Those of C. Lubich and collaborators, e.g. López-Fernández et al. [6], apply a similar idea to convolution operators and develop error estimates as well as optimized numerical methods. In the related field of numerical inversion of the Laplace transform, Weideman and Trefethen [10] provided an error estimate and its application to contour optimization for computation cost reduction. It is based on the balancing of discretization and truncation errors. In our work, we focus on distributed control applications of the diffusive application where real-time implementations improved by contour optimization are

an essential feature. We also observe that the DR decomposes an operator as a linear combination of local operators namely of differential operators. Such decomposition seems to be well suited for implementations on distributed computing architectures. Combining the two features opens a new direction to develop embedded real-time computation for distributed systems on distributed architectures with parallel implementation. Hence, it yields embedded, massive, scalable and low-cost computation.

The realization of a linear operator  $u \rightarrow z = Pu$ , by the DR method, is addressed to causal operators when the kernel  $p$  of  $P$  is explicitly known. When it is analytic in its second variable then see Montseny [8]. The case where  $P$  is the solution of an operator equation, so  $p$  is neither explicitly given nor analytic, is reported in Yakoubi et al. [12]. Their method was announced in Lenczner and Montseny [8] and fully developed in Lenczner et al. [5]. The numerical method is implemented for an operator  $P$  solution of the Lyapunov equation issued from optimal control of the heat equation. An overview of the theory and of the numerical results was presented in the PhD thesis of Yakoubi [11]. We notice that their results were rather inaccurate. Precisely, the DR theory requires an analytic extension of the kernel  $p$  which was obtained after approximation by Legendre polynomials and extension out of the interval  $\hat{\omega} = (-1, 1)$ . This leads to non-uniformly bounded extensions with respect to the number of polynomials which causes high numerical errors.

Four contributions are reported in this paper. First, we apply an additional change of variables to the kernel  $p$  and its extension so that they become defined in  $\hat{\omega}$  which eliminates an important source of error. Then, as in [10], but with a significantly different theoretical approach, we provide, for the first time, an error estimate of the DR and use it in a contour optimization method. Moreover, the method has been extensively tested and some significant results are reported. Finally, we present results for three parallel topologies: line topology, hypercube topology, binary topology. We also compare the computation times between our algorithm with these topologies and a direct spectral method with

\*Femto-ST Institute, University of Franche Comte, IUT Belfort-Montbéliard, 19 av. du Maréchal Juin, BP 527, 90016 Belfort, France [quan.do\\_huu@univ-fcomte.fr](mailto:quan.do_huu@univ-fcomte.fr), [raphael.couturier@univ-fcomte.fr](mailto:raphael.couturier@univ-fcomte.fr)

<sup>†</sup>Femto-ST, Time-Frequency Department, 26, chemin de l'Épitaphe, 25030 Besançon, France [michel.lenczner@utbm.fr](mailto:michel.lenczner@utbm.fr)

a line topology. All these results are carried out on the same operator  $P$  as in [5]. However, we do not envision any limitation for its extension to more complex cases as, for example, solutions to Riccati equations.

This paper is organized as follows. The definition of the DR is recalled in Section 2, then in Section 3 a numerical method is detailed. Error estimates and simulation results are reported in Sections 4 and 5 respectively. Finally, estimates of numbers of operations and computation times on parallel architectures are discussed in Section 6.

## 2 Diffusive realization

For more detailed definitions and elementary properties, we refer to [4] and [8]. We consider a kernel operator  $P$ ,

$$Pu(x) = \int_0^1 p(x, y)u(y)dy,$$

and its unique decomposition  $(Pu) = z^+ + z^-$  into causal and anti-causal parts,

$$z^+(x) = \int_0^x p(x, y)u(y)dy, z^-(x) = \int_x^1 p(x, y)u(y)dy.$$

Throughout this paper, we shall use the superscripts  $+$  or  $-$  to refer to causal or anti-causal operators, and the convention  $\mp = -(\pm)$ . The realizations of  $z^\pm$  are formulated thanks to the diffusive representation, in the form

$$(2.1) \quad z^+(x) = \int_{\mathbb{R}} \mu^+(x, \xi)\psi^+(x, \xi)d\xi$$

and  $z^-(x) = \int_{\mathbb{R}} \mu^-(x, \xi)\psi^-(x, \xi)d\xi,$

where  $\psi^\pm$  stores a part of the history of the input data  $u$ . They are respectively solution to the forward and backward ordinary differential equation in  $x$ ,

$$\partial_x \psi^+(x, \xi) + \theta^+(\xi)\psi^+(x, \xi) = u(x) \text{ with } \psi^+(0, \xi) = 0,$$

$$\partial_x \psi^-(x, \xi) - \theta^-(\xi)\psi^-(x, \xi) = u(x) \text{ with } \psi^-(1, \xi) = 0,$$

parameterized by  $\xi \in \mathbb{R}$ . We notice that the functions  $\psi^\pm$  are defined independently of  $P$ . Conversely, the coefficients  $\mu^\pm$ , called *diffusive symbols*, depend on  $P$  but not on  $u$ . The functions  $\xi \mapsto \theta^\pm(\xi)$  parameterize two closed paths in the complex plane, satisfying the cone condition, and enlacing the singularities of the Laplace transform  $\mathcal{P}^\pm$  defined hereafter. The functions

$$(2.2) \quad y \mapsto p(x, x - y) \text{ and } y \mapsto p(x, x + y),$$

restricted to  $y < x$  and  $y > x$  respectively, correspond to the impulse response of the kernels of the causal and

anti-causal parts of the operator  $P$ . They are assumed to be analytic and their analytic extensions to  $\mathbb{R}^+$  admit Laplace transforms  $\mathcal{P}^\pm$ . Then the latter are assumed to admit holomorphic extensions to the complex domains bounded to the left by  $-\theta^\pm$  and vanish at infinity. Finally, the expression of the diffusive symbols turns to be  $\mu^\pm(x, \xi) = \mp \frac{\theta^{\pm'}(\xi)}{2i\pi} \mathcal{P}^\pm(x, -\theta^\pm(\xi))$ .

## 3 Diffusive realization approximation

In Subsection 3.1, we approximate the symbols  $\mu^\pm$  based on a dedicated spectral approximation of the kernel  $p$ . The spectral method combines the basis of Legendre polynomials in both  $x$  and  $y$  with an extension in  $y$ . Then, computational algorithms for history functions  $\psi^\pm$  and for  $z^\pm$  are detailed in Subsection 3.2 and 3.3. In this paper, we only use the hyperbolic contours  $-\theta^\pm$  proposed by Weideman and Trefethen in [10], in the context of inverse Laplace transform computation, namely

$$-\theta^\pm(\xi) = \theta_h(1 + \sin(i\xi - \alpha)) \text{ for } \xi \in \mathbb{R},$$

where  $\theta_h > 0$  regulates the width of the contours and  $\alpha$  defines the hyperbola's asymptotic angle.

**3.1 Approximation of diffusive symbols** Obviously, impulse responses (2.2) are generally neither analytic nor known, since  $p$  is the unknown of a boundary value problem. The method, adopted in this paper, circumvents the problem of extension to  $\mathbb{R}^+$  by directly building approximations on  $\mathbb{R}^+$ . To this end, both changes of variable  $r^+ = 2e^y - 1$  and  $r^- = -2e^{-y} + 1$  are used to map  $\mathbb{R}^+$  into  $\hat{\omega}$ . Thus, the transformed kernels are approximated over  $(-z^*, 1)$  and  $(-1, z^*)$  with  $z^* = 1 - 2e^{-1}$  and then extend naturally  $\hat{\omega}$ . For the sake of the method's uniformity, the change of variable  $s = 2x - 1$  is used to map  $(0, 1)$  into  $(-1, 1)$ , so the discretization with respect to  $x$  is done with the basis of the modified Legendre polynomials, vanishing at the boundaries,  $\eta_k(s) = (L_{k-2}(s) - L_k(s))/\sqrt{2(2k-1)}$  for  $k \geq 2$  with  $\eta_0(s) = \frac{1-s}{2}$  and  $\eta_1(s) = \frac{1+s}{2}$ , where  $L_k$  is the  $k^{\text{th}}$  usual Legendre polynomial. The extension is thus of the form  $\hat{p}^{N\pm}(s, r^\pm) = \sum_{k=2}^{N_1} \sum_{l=0}^{N_2} p_{kl}^\pm \eta_k(s) L_l(r^\pm)$ . The coefficients  $p_{kl}^\pm$  are computed through a weak formulation involving a reflection principle for the parts where  $r^\pm$  corresponds to  $y \notin (0, 1)$ . The final expression of the diffusive coefficient approximation is

$$\mu^{N\pm}(x, \xi) = \mp \frac{\theta^{\pm'}}{2i\pi} \mathcal{P}^{N\pm}(x, -\theta^\pm(\xi))$$

with

$$\mathcal{P}^{N\pm}(x, \lambda) = \sum_{k=2}^{N_1} \sum_{l=0}^{N_2} p_{kl}^\pm \nu_k(x) \zeta_l^\pm(x, \lambda)$$

where  $\nu_k(x) = \eta_k(2x - 1)$  and  $\zeta_l^\pm(x, \lambda) = \mathcal{L}_y(L_l(\beta^\pm(x)e^{-y} + \gamma^\pm))(\lambda)$  which can be determined by the recurrence equation

$$(3.3) \quad \zeta_0^\pm(x, \lambda) = \frac{1}{\lambda}, \quad \zeta_1^\pm(x, \lambda) = \frac{\beta^\pm(x)}{\lambda + 1} + \frac{\gamma^\pm}{\lambda},$$

$$\zeta_{k+1}^\pm(x, \lambda) = \frac{2k+1}{k+1} (\beta^\pm(x)\zeta_k^\pm(x, \lambda+1) + \gamma^\pm\zeta_k^\pm(x, \lambda))$$

$$- \frac{k}{k+1} \zeta_{k-1}^\pm(x, \lambda).$$

where  $\beta^+(x) = 2e^{x-1}$ ,  $\beta^-(x) = -2e^{-x}$  and  $\gamma^\pm = \mp 1$ .

**REMARK 3.1.** *We have used a spectral method to discretize both the  $x$ - and  $y$ -directions. In the  $y$ -direction the global basis function is the root of the analytic extension. On the contrary, there is no restriction regarding approximations in the  $x$ -direction. For instance a local basis as a finite element basis might be used.*

**3.2 Discretization of  $\psi$  with respect to  $x$**  The  $x$ -discretization of the history function is based on the interpolation of discrete inputs  $(u_\ell)_\ell$  located at regularly spaced nodes  $(x_\ell)_\ell$  separated by a distance  $h$ . In each interval  $[x_\ell - h/2, x_\ell + h/2]$ ,  $u(x)$  is equal to the constant  $u_\ell$ . In Yakoubi et al. [12] the calculations for both causal and anti-causal parts have been detailed, we simply recall them without repeating their justification. Defining the parameters  $\alpha^\pm(\xi) = e^{-\theta^\pm(\xi)h}$ , and  $\beta^\pm(\xi) = \frac{\alpha^\pm(\xi)-1}{-\theta^\pm(\xi)}$ , the recurrence relations yield

$$(3.4) \quad \psi^+(x_{\ell+1}, \xi) \simeq \psi^{H^+}(x_{\ell+1}, \xi) = \alpha^+(\xi)\psi^+(x_\ell, \xi)$$

$$+ \beta^+(\xi)u_\ell, \quad \psi^+(0, \xi) = 0,$$

and  $\psi^-(x_\ell, \xi) \simeq \psi^{H^-}(x_{\ell+1}, \xi) = \alpha^-(\xi)\psi^-(x_{\ell+1}, \xi)$

$$- \beta^-(\xi)u_\ell, \quad \psi^-(1, \xi) = 0$$

where  $H = 1/h$  is the number of intervals in the  $x$ -variable. Notice that this recurrence relation has already been established by C. Casenave [1] for the causal part.

**3.3 Approximation of the integrals  $z^\pm$**  A direct application of the residue theorem allows to eliminate the terms without exponential,

$$(3.5) \quad \int_{\mathbb{R}} \frac{\mu^{N^\pm}(x, \xi)}{\theta^\pm(\xi)} d\xi = 0.$$

Posing  $\gamma^\pm(\xi) = \frac{\alpha^\pm(\xi)}{-\theta^\pm(\xi)}$ , from the recurrence relation (3.4), evaluating the integrals thanks to the trapezoidal rule with  $2M + 1$  quadrature nodes regularly

spaced at a distance  $h_\xi$  yields the final approximations  $z_{\ell+1}^+$  and  $z_\ell^-$  of  $z^+$  and  $z^-$  at the input nodes,  $z_{\ell+1}^+ = h_\xi \sum_{k=-M}^M \mu_{\ell+1,k}^{N^+} (\alpha_k^+ \psi_{\ell,k}^+ + \gamma_k^+ u_\ell)$  and  $z_\ell^- = h_\xi \sum_{k=-M}^M \mu_{\ell,k}^{N^-} (\alpha_k^- \psi_{\ell+1,k}^- - \gamma_k^- u_\ell)$ .

#### 4 Error estimate and contour optimization

There are three sources of errors in the algorithm. They originate from the discretization of the contour integral, the piecewise constant interpolation of the input  $u$  and the application of the Galerkin method in the kernel computation. The latter is not taken into account in this paper. The realization  $z^\pm$  is approximated with respect to  $x$  as

$$z_h^+(x_\ell) = \int_0^{x_\ell} p(x_\ell, y) \tilde{u}(y) dy$$

and

$$z_h^-(x_\ell) = \int_{x_\ell}^1 p(x_\ell, y) \tilde{u}(y) dy$$

where  $u(x)$  is replaced by its piecewise constant approximation  $\tilde{u}(x)$  equal to  $u(x_\ell)$  on each interval  $[x_\ell - h/2, x_{n+1} + h/2]$ . The realization of  $z_h^\pm$  is formulated, thanks to the diffusive representation and the formula (3.5), in the form

$$z_h^\pm(x_\ell) = \int_{\mathbb{R}} \mu^{N^\pm}(x_\ell, \xi) \tilde{\psi}^{H^\pm}(x_\ell, \xi) d\xi$$

where  $\tilde{\psi}^{H^+}(x_\ell, \xi) = \psi^{H^+}(x_\ell, \xi) - \frac{u_{\ell-1}}{\theta^+(\xi)}$  and  $\tilde{\psi}^{H^-}(x_\ell, \xi) = \psi^{H^-}(x_\ell, \xi) + \frac{u_\ell}{\theta^-(\xi)}$ . The discretization  $\psi^{H^\pm}$  of  $\psi^\pm$  with respect to  $x$  is rewritten as

$$\psi^{H^\pm}(x_\ell, \xi) = -\frac{1 - e^{\pm\theta^\pm(\xi)h}}{\theta^\pm(\xi)} \sum_{j \in J_\ell^\pm} e^{-\theta^\pm(\xi)(\pm t_{\ell,j})} u_j.$$

Taking into account the approximation with respect to  $\xi$  by the trapezoidal rule yields

$$z_{h_\xi}^\pm(x_n) = h_\xi \sum_{k=-M}^M \mu^{N^\pm}(x_n, kh_\xi) \tilde{\psi}^{H^\pm}(x_n, kh_\xi).$$

Then, the approximation  $z_h^+$  is written as a linear combination of inverse Laplace transform  $\mathcal{L}^{-1}$ ,

$$z_h^+(x_\ell) = \sum_{j \in J_\ell^+} u_j (\mathcal{L}^{-1}[F^+(-\theta^+)])(\pm t_{\ell,j})$$

$$- \mathcal{L}^{-1}[F^+(-\theta^+)](\pm t_{\ell,j+1})$$

for all  $\ell \in \{0, 1, \dots, H\}$  with  $J_\ell^+ = \{0, \dots, \ell - 1\}$ ,  $J_\ell^- = \{\ell, \dots, H - 1\}$ ,  $x_j = jh$ ,  $u_j = u(x_j)$ ,  $t_{\ell,j} = x_\ell - x_j$ ,

$F^\pm(-\theta^\pm) = \frac{\mathcal{P}^\pm(x_\ell, -\theta^\pm(\xi))}{-\theta^\pm(\xi)}$ . Weideman and Trefethen [10] have developed a contour optimization based on a balance between the truncation error estimate and the discretization error for the numerical integration of the Laplace inversion at points  $t_{\ell,j} \in I = \{h, 2h, \dots, 1\}$  excluding point 0. This approach is not efficient in the present case since the ratio between the upper and lower bounds of the set  $I$ , i.e.  $H = \frac{1}{h}$ , is very large for a fine mesh and the numerical inversion of Laplace transform is relatively expensive. To circumvent this problem, the error estimates from Theorem 4.1 of Stenger [9] are applied to the evaluation of the discretization error of each pair  $\mathcal{L}^{-1}[F^\pm(-\theta^\pm)](\pm t_{\ell,j}) - [F^\pm(-\theta^\pm)](\pm t_{\ell,j+1})$  of Laplace inverses. The global error

$$e(x) = |z(x) - z_{h_\xi}(x)|$$

is majorized by the sum  $e_h(x) + e_{h_\xi}(x)$ . The error  $e_h(x) = |z(x) - z_h(x)|$  is the approximation with respect to  $x$  and the error  $e_{h_\xi}(x) = |z_h(x) - z_{h_\xi}(x)|$  is the approximation with respect to  $\xi$ , where  $z = z^+ + z^-$ ,  $z_h = z_h^+ + z_h^-$  and  $z_{h_\xi} = z_{h_\xi}^+ + z_{h_\xi}^-$ . Thus the  $L^2$ -norm of global error

$$\|e\|_{L^2(0,1)} \leq \sqrt{2}(\|e_h\|_{L^2(0,1)} + \|e_{h_\xi}\|_{L^2(0,1)})$$

and both terms on the right side of this inequality are evaluated hereafter.

In this paper, we only present the main results without detailing the proofs.

**THEOREM 4.1.** *If we assume that  $u \in H^1(0, 1)$  and the kernel  $p \in L^2((0, 1)^2)$ , then there holds*

$$\|e_h\|_{L^2(0,1)} \leq \frac{\sqrt{2}}{2} h \|u'\|_{L^2(0,1)} \times \|p\|_{L^2((0,1)^2)}.$$

We denote by

$$g_N^\pm(x, \xi, t, h) = \frac{(1 - e^{\pm\theta^\pm(\xi)h}) e^{-\theta^\pm(\xi)(\pm t)}}{h} \frac{e^{-\theta^\pm(\xi)(\pm t)}}{-\theta^\pm(\xi)} \mu^{N^\pm}(x, \xi),$$

$$g_N^{\pm 0}(x, \xi, h) = \pm \frac{e^{-\theta^\pm(\xi)h}}{h} \frac{\mu^{N^\pm}(x, \xi)}{-\theta^\pm(\xi)}.$$

We introduce the function  $f^\pm(a, \theta_h^\pm, \alpha^\pm, x, t, d_1, d_2, h)$

$$\begin{aligned} &= \frac{\int_{\mathbb{R}} |g_N^\pm(x, \xi - id_1, t, h)| d\xi}{e^{2\pi d_1/h_\xi} - 1} + \frac{\int_{\mathbb{R}} |g_N^\pm(x, \xi + id_2, t, h)| d\xi}{e^{2\pi d_2/h_\xi} - 1} \\ &+ \left( \sum_{k=M+1}^{\infty} |g_N^\pm(x, kh_\xi, t, h)| + \sum_{k=-\infty}^{-(M+1)} |g_N^\pm(x, kh_\xi, t, h)| \right), \end{aligned}$$

and  $f^{\pm 0}(a, \theta_h^\pm, \alpha^\pm, x, d_1, d_2, h)$

$$\begin{aligned} &= \frac{\int_{\mathbb{R}} |g_N^{\pm 0}(x, \xi - id_1, h)| d\xi}{e^{2\pi d_1/h_\xi} - 1} + \frac{\int_{\mathbb{R}} |g_N^{\pm 0}(x, \xi + id_2, h)| d\xi}{e^{2\pi d_2/h_\xi} - 1} \\ &+ \left( \sum_{k=M+1}^{\infty} |g_N^{\pm 0}(x, kh_\xi, h)| + \sum_{k=-\infty}^{-(M+1)} |g_N^{\pm 0}(x, kh_\xi, h)| \right). \end{aligned}$$

As [10], for each  $x, t, h$ , we assume that  $g_N^\pm(x, \xi + iw, t, h)$ ,  $g_N^{\pm 0}(x, \xi + iw, h)$  are analytic in the strip  $-d_1 < w < d_2$  for some  $d_1 > 0$  and  $d_2 > 0$ , with  $g_N^\pm(x, \xi + iw, t, h)$  and  $g_N^{\pm 0}(x, \xi + iw, h) \rightarrow 0$  uniformly as  $|\xi + iw| \rightarrow \pm\infty$  in that strip.

For simplicity's sake, we only use a contour  $\theta^+ = \theta^- = \theta$ ,

$$G(a, \theta_h, \alpha, d_1, d_2, h)$$

$$= \left[ \sum_{n=0}^{H-1} \left( \sum_{j \in J_{n-1}^+} f^+(a, \theta_h, \alpha, x_n, d_1, d_2, t_{n,j}, h)^2 \right. \right.$$

$$\left. \left. + f^{+0}(a, \theta_h, \alpha, x_n, d_1, d_2, h)^2 + f^{-0}(a, \theta_h, \alpha, x_n, d_1, d_2, h)^2 \right. \right.$$

$$\left. \left. + \sum_{j \in J_{n+1}^-} f^-(a, \theta_h, \alpha, x_n, d_1, d_2, t_{n,j}, h)^2 \right) \right]^{1/2}.$$

**THEOREM 4.2.** *If we assume that  $u \in L^2(0, 1)$  then*

$$\|e_{h_\xi}\|_{L^2(0,1)} \leq h \|u\|_{L^2(0,1)} \times G(a, \theta_h, \alpha, d_1, d_2, h).$$

Finally, optimal contours are sought with parameter solution to the optimization problem,

$$(4.6) \quad \begin{aligned} &G(a^*, \theta_h^*, \alpha^*, d_1, d_2, h) \\ &= \min_{\substack{a \in (0, +\infty) \\ \theta_h \in (0, +\infty) \\ \alpha \in (d_1, \pi/2 - d_2)}} G(a, \theta_h, \alpha, d_1, d_2, h). \end{aligned}$$

## 5 Simulation results

We recall the model problem used in [5]: the operator  $P$  is solution to the Lyapunov equation,

$$(5.7) \quad \frac{d^2}{dx^2} Pu + P \frac{d^2}{dx^2} u = Qu$$

in  $\omega = (0, 1)$ , for all  $u$  vanishing at the boundary, and where  $Q$  is a  $L^2(\omega)$ -bounded linear kernel operator. This problem originates in optimal filtering or control theory of the heat equation,  $\frac{\partial T}{\partial t} - \frac{\partial^2 T}{\partial x^2} = q$  in  $\omega$  with

Dirichlet boundary conditions. We recall that the kernel  $p$  is the unique solution to the boundary value problem  $\Delta p = q$  in the square  $(0, 1)^2$ ,  $p = 0$  on the boundary and  $q$  is the kernel of  $Q$ .

The above Approximation of the Diffusive Realization (ADR) with the contour optimization is compared to a Direct Spectral method (DS) using the usual Legendre polynomials to estimate  $p$  and a trapezoidal rule for  $z$ . Experiments have been carried out for two kernels  $q$  and two inputs  $u$ . The kernels are a two-dimensional Gaussian density,  $q_1(x, y) = C e^{-((x-x_0)^2+(y-y_0)^2)/2\sigma^2}/\sigma^2 2\pi$  with  $\sigma = 0.2$ ,  $x_0 = y_0 = 0.4$ ,  $C = 10$ , and an oscillating function,  $q_2(x, y) = 2a^2\pi^2 \sin(b\pi x)\sin(a\pi y)$  with  $a = 3$  and  $b = 5$ . The inputs are both oscillating functions,  $u_1(y) = 3y \sin(6\pi y)^2$  and  $u_2(y) = 15 \sin(4\pi y^4)$ . The errors are evaluated using a reference solution  $z^{ref}$  computed with the DS method with  $30 \times 30$  polynomials and with inputs discretized with 1,024 nodes. Three parameters of discretization are used, the number  $N$  of Legendre polynomials in each  $x$ - and  $y$ -directions to discretize the kernel  $p$ , the number  $H$  of discretization points  $x_\ell$  of the input, and the number  $M$  of quadrature nodes along the contours. The contour parameters are chosen as the solution of the optimization problem  $\min_{a, \theta_h, \alpha} G(a, \theta_h, \alpha, d_1, d_2, h)$ . The discrete outputs  $z_\ell$  are evaluated at the same nodes as the input. The error function is evaluated in the discrete  $\ell^2$ -norm evaluated at the nodes  $x_\ell$  of the input mesh

$$Error(z) = \left( \sum_{\ell=1}^H (z^{ref}(x_\ell) - z_\ell)^2 \right)^{1/2} / \left( \sum_{\ell=1}^H (z^{ref}(x_\ell))^2 \right)^{1/2}$$

Four values  $N = 10, 15, 20$  and  $25$  have been tested when the other parameters are very large,  $H = 1,024$  and  $M = 100$ . The DS method yields errors varying from  $10^{-6}$  to  $10^{-12}$  for  $q_1$  and from  $10^{-2}$  to  $10^{-15}$  for  $q_2$ . For the ADR they vary from  $10^{-5}$  to  $10^{-6}$  for  $q_1$  and from  $10^{-1}$  to  $10^{-3}$  for  $q_2$  when  $N$  varies from 10 to 20, and then increases. This lack of convergence may come from the operations of extension of  $p^\pm$  which exhibits oscillations and also from the very high values of coefficients of the Legendre polynomials yielding large truncature errors in the Laplace domain. In the following experiment, we consider that  $N = 20$  is the optimal value for the ADR.

To test the effect of the number  $H$  of discretization points in the input, experiments have been carried out with  $H = 16, 32, 64, 128$  and  $256$  while  $N = 20$  and  $M = 100$ . For  $q_1$  (resp.  $q_2$ ) the minimal error of  $10^{-6}$  (resp.  $10^{-3}$ ) is reached for the ADR with  $H = 256$  (resp.  $H = 128$ ). So, for  $H$  varying between 16 and 256 (resp. 128) the errors of the ADR and of the DS method

change from  $10^{-3}$  to  $10^{-6}$  and from  $10^{-4}$  to  $10^{-9}$  (resp. from  $10^{-2}$  to  $10^{-3}$  and from  $10^{-2}$  to  $10^{-6}$ ). In all cases, we observe a better decay of the error for the DS method with respect to  $H$  than for the ADR. The error of the DS method is with infinite order, whereas for the ADR method the error follows the one of a quadrature method.

Three values  $M = 20, 40$  and  $60$  have been compared when  $N = 20$  and  $H = 1,024$ . The ADR method has errors varying between  $10^{-3}$  to  $10^{-6}$  for  $q_1$  and between  $10^{-2}$  to  $10^{-3}$  for  $q_2$ . This confirms a very fast decay of the error with respect to the number of quadrature points in the contours. For the ADR method, which involves complex numbers, a significant gain in terms of computation cost only exists if the number  $M$  can be as small as discussed in the next section. With an optimization of the contours to solve the optimization problem (4.6), for  $M = 10$ , errors in the range of  $10^{-3}$  for  $q_1$  and of  $10^{-2}$  for  $q_2$  have been reached with the ADR method. Figure 1 shows a contour plot of the relative error with  $q_1(x, y)$  and  $u_1(y)$  in the  $(\theta_h, \alpha)$ -parameter plane. Darker shades represent higher accuracy.

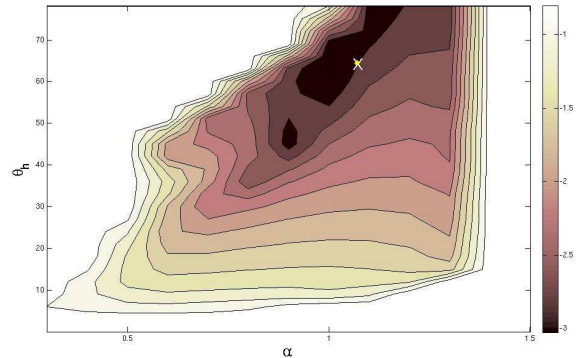


Figure 1: Level lines of the logarithm  $\log_{10} \|e\|_{L^2(0,1)}$  of the error function for  $q_1(x, y)$  and  $u_1(y)$  at a fixed  $a = 1$ . The cross represents the solution to the optimization problem  $\min_{\alpha, \theta_h} G(a, \theta_h, \alpha, d_1, d_2, h)$  with  $d_1 = d_2 = 0.5$ ,  $h = 0.02$ .

In total, the new extension of  $p^\pm$  yields a viable numerical method that might be used when a limited precision is required, as, for example, applications in real-time distributed control. Limitations of the ADR method have been well identified.

## 6 Parallel computation

In this section we describe the implementations of our algorithm on different parallel topologies. The algorithm is based on the history functions  $\psi^+$  for the

diffusive realizations of the causal part  $z^+$ . Equation (3.4) is rewritten as  $\psi_{\ell+1,k}^+ = \sum_{i=0}^{\ell} (\alpha_k^+)^{\ell-i} \beta_k^+ u_i$ . So if we define  $\varphi_{\ell,k}^+$  as  $\varphi_{\ell,k}^+ = \sum_{i=0}^{\ell} (\alpha_k^+)^{-i} \beta_k^+ u_i$  then  $\psi_{\ell+1,k}^+ = \varphi_{\ell,k}^+ (\alpha_k^+)^{\ell}$ . The variable  $\varphi^+$  is used to make a cumulative sum. The resulting algorithm is summarized in Algorithm 1.

**Offline computation** of diffusive symbol  
 $\mu^{N^+}(x, \xi)$ ,  $(\alpha^+)^{-\ell} \beta^+$  and  $(\alpha^+)^{\ell}$ ,  
 $\ell \in \{0, \dots, N-1\}$ .  
**Online computation**  
**for**  $\ell = 0, \dots, N-2$  **do**  
    **for**  $k=1, \dots, M$  **do**  
         $\varphi_{\ell+1,k}^+ = \varphi_{\ell,k}^+ + (\alpha_k^+)^{-\ell-1} \beta_k^+ u_{\ell+1}$ ,  
         $\varphi_{0,k}^+ = 0$   
    **end**  
**end**  
**for**  $\ell = 0, \dots, N-2$  **do**  
    **for**  $k=1, \dots, M$  **do**  
         $\psi_{\ell+1,k}^+ = \varphi_{\ell,k}^+ (\alpha_k^+)^{\ell}$   
    **end**  
     $z_{\ell+1}^+ = 2h_{\xi} \Re \left( \sum_{k=1}^M \mu_{\ell+1,k}^{N^+} (\alpha_k^+ \psi_{\ell,k}^+ + \gamma_k^+ u_{\ell}) \right)$   
**end**

**Algorithm 1:** Diffusive realization of  $z^+(x)$ .

Note that the implementation of the anti-causal part is done in a similar way, and will not be described in detail. Consequently, we will drop all upper indices “+” without any risk of confusion.

To be able to implement this algorithm in parallel, the first loop, which intends to compute a cumulative sum, needs to be executed in parallel. Then the estimate of the contour is a computation where no communication between the nodes is required. The computation time is optimized by adapting the topologies in order to implement efficiently a cumulative sum. In Section 6.1-6.3, we derive parallel topologies, well suited for diffusive realization, and the line topology for the direct spectral method in Section 6.4.

**6.1 DR with a line topology** We consider that a processor can only communicate with its two neighbors. In order to maximize parallelism we consider that each of the  $k$  processors controls  $m$  sensors (i.e. inputs). So the input  $u$  is discretized by  $k \times m$  points. The computation is achieved in four steps: first step, the local cumulated sums are computed on all processors; second step, processors (except the first one) add  $n$  complex values of the previous processors to the last cumulated sums, then the results of the last cumulated sums (except the last processor) are sent to the next pro-

cessors; third step, previous cumulated sums are added to all the local elements of all the processors except the first one, the fourth step is the computation of the causal part with the evaluation of the integrals along the contour. Here is the total number of operations and communications for the line topology,  $\mathcal{O}_{DRline} = (28k \times m \times n) \otimes + (4(7k-1) \times m \times n) \oplus + (4(k-1) \times n) \textcircled{S}$ , where the symbols  $\otimes$ ,  $\oplus$  and  $\textcircled{S}$  refer to multiplications, additions and real number transmission. We denote by  $T_j$  for  $j = 1, \dots, 4$  the computation time of the  $j^{\text{th}}$  step and by  $t_j$ , for  $j = 1, \dots, 3$  the time taken by a multiplication, an addition and a transmission respectively. We find that  $T_1 = 4m \times n \times t_1 + 4m \times n \times t_2$ ,  $T_2 = 2n \times t_2 + 2n \times t_3$ ,  $T_3 = 2n \times (m-1) \times t_2$ ,  $T_4 = 10m \times n \times t_1 + 8m \times n \times t_2$ . If  $T_1 \geq \frac{k-4}{3} \times T_2$  then the computation of causal and anti-causal parts cannot be overlapped, so the computation time is  $T_{DRline\_no\_overlap} = 28m \times n \times t_1 + 2n \times (14m+k-3) \times t_2 + 2n \times (k-1) \times t_3$ . If  $T_1 < \frac{k-4}{3} \times T_2$ , and  $T_3 \geq \frac{k-2}{3} \times T_2 - \frac{T_1}{3}$  then the computation of step 1 can be overlapped and the computation time is  $T_{DRline\_overlap\_step1} = 24m \times n \times t_1 + 2n \times (12m+k-3) \times t_2 + 2n \times (k-1) \times t_3$ . If  $T_1 < \frac{k-4}{3} \times T_2$ , and  $T_3 < \frac{k-2}{3} \times T_2 - \frac{T_1}{3}$  then the computation of step 1 and step 3 can be overlapped and the computation time is  $T_{DRline\_overlap\_step3} = 24m \times n \times t_1 + 2n \times (11m+k-2) \times t_2 + 2n \times (k-1) \times t_3$ .

**6.2 DR with a hypercube topology** If a hypercube topology is used to compute the cumulative sum, with  $p = 2^d$  processors, and considering that a processing unit controls  $m$  sensors, we also suggest to proceed in four steps to implement the computation: the first, the third and the fourth steps are similar to those with the line topology. The second step is divided into  $d$  substeps. In the  $j^{\text{th}}$  substep a processor receives  $n$  complex numbers and adds them to the data for the next message. At the same time, it adds them to its last cumulative sum if, for the causal (resp. anti-causal) part, its rank is less (resp. greater) than the rank of its  $j^{\text{th}}$  neighbor. The total number of operations and communications is  $\mathcal{O}_{DRhyper} = 28m \times n \times 2^d \otimes + 2n \times (14m \times 2^d + 3d \times 2^d - 2^{d+1} - 2m + 2) \oplus + (4n \times d \times 2^d) \textcircled{S}$  and the computation time is  $T_{DRhyper} = 28m \times n \times t_1 + 4n \times (7m + 2d - 1) \times t_2 + 4n \times d \times t_3$ .

**6.3 DR with binary tree topology** A binary tree topology can also be used to compute the cumulative sum with  $p = 2^d$  processors each controlling  $m$  sensors. The diffusive realization is implemented in five steps. Step 1 is the same as in the previous topologies; Step 2 is divided into  $d$  substeps. For the causal part, during the  $i^{\text{th}}$  substep ( $1 \leq i \leq d$ ), the  $(2^i j)^{\text{th}}$  processor receives and adds the  $n$  complex values from the  $(2^{i-1}(2j-1))^{\text{th}}$

processor ( $1 \leq j \leq 2^{d-i}$ ) and other processors have no computation. Step 3 is also divided into  $d$  substeps. For the causal part, at the  $i^{\text{th}}$  small step ( $1 \leq i \leq d$ ), the  $(2^{d-i+1}j)^{\text{th}}$  processor receives and adds the  $n$  complex values from the  $(2^{d-i}(2j-1))^{\text{th}}$  processor ( $1 \leq j \leq 2^{i-1}$ ), at the same time the  $(2^{d-i}(2j-1))^{\text{th}}$  processor receives the  $n$  complex values from the  $(2^{d-i+1}j)^{\text{th}}$  processor and replaces its old values and the other processors are not used. The operation of replacement is counted as an addition. Steps 4 and 5 are identical as Steps 3 and 4 of the above other topologies. The anti-causal part is deduced by symmetry. So, the total number of operations and communications for both parts is  $\mathcal{O}_{DRbinary\_tree} = (28m \times n \times 2^d) \otimes + 4n \times (7m \times 2^d + 3 \times 2^d - 3) \oplus + (12n \times (2^d - 1)) \textcircled{S}$ , with the computation time  $T_{DRbinary\_tree} = 28m \times n \times t_1 + 4n \times (7m + 2d) \times t_2 + 12n \times d \times t_3$ .

REMARK 6.1. *It should be noted that there is no overlap in the hypercube and binary tree topologies because it would require to double the number of links to be able to send messages for causal and anti-causal parts simultaneously.*

**6.4 DS method with a line topology** The computation for the direct spectral method follows the formula

$$Pu(x_\ell) \approx h \sum_j p(x_\ell, y_j)u(y_j) \text{ for all } \ell.$$

So, the computation consists in the multiplication between a real matrix  $p$  and a real vector input  $u$ . The network and the computation implementation are similar to the line network of diffusive realization. The number of operations and communications is  $\mathcal{O}_{direct} = m^2 \times k^2 \otimes + m^2 \times k^2 \oplus + m \times k \times (k-1) \textcircled{S}$  with the computation time  $T_{direct} = m^2 \times (1.5k-1) \times t_1 + m^2 \times (1.5k-1) \times t_2 + m \times (1.5k-2) \times t_3$ .

**6.5 Computation time estimate** In the following we report simulation results showing the computation times for two sets of speeds

1.  $t_1 = t_2 = 5 \times 10^{-8}$  s,  $t_3 = 5 \times 10^{-7}$  s,
2.  $t_1 = t_2 = 5 \times 10^{-8}$  s,  $t_3 = 5 \times 10^{-5}$  s,

and three different numbers of nodes  $m = 1$ ,  $m = 16$  or  $m = 64$ . The number of quadrature points is fixed at  $n = 10$ . In the case  $m = 1$ , the computation times of the diffusive realization with a line are not plotted to show the computation times for the hypercube which increase slowly. These simulations are shown in a series of figures in this paper.

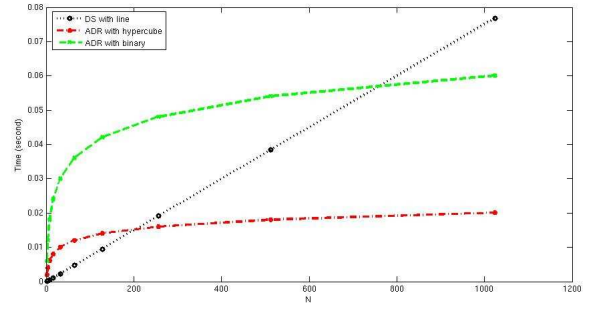


Figure 2: Comparison of computation times in seconds with  $n = 10, m = 1, t_1 = t_2 = 5 \times 10^{-8}(s), t_3 = 5 \times 10^{-5}(s)$ .

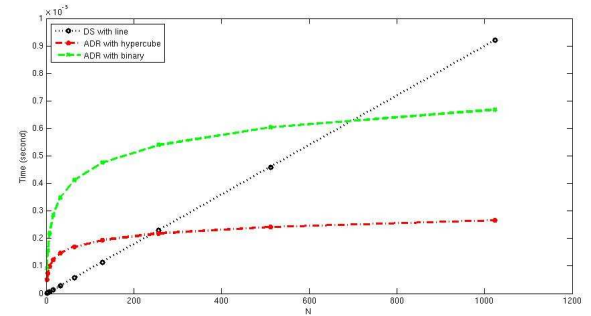


Figure 3: Comparison of computation times in seconds with  $n = 10, m = 1, t_1 = t_2 = 5 \times 10^{-8}(s), t_3 = 5 \times 10^{-7}(s)$ .

Obviously the transmission time has a significant effect on the performance of all the parallel implementations and the number of sensors per processor is also very important. The line topology is especially influenced by the number of sensors. Moreover, the binary tree is not very efficient. As mentioned in the previous part, the only solution to improve it, is to use more links so that some communications can be performed in parallel. In all these simulations, the hypercube technology is, without surprise, the most efficient. This topology has been used in many algorithms since it offers a good compromise between the number of communication links used and the efficiency. The next step will be to implement a real distributed control algorithm.

## 7 Conclusion

The diffusive realization of operators was mainly applied to operators with analytically known kernels. From the references in the field, it is known to be a very effi-

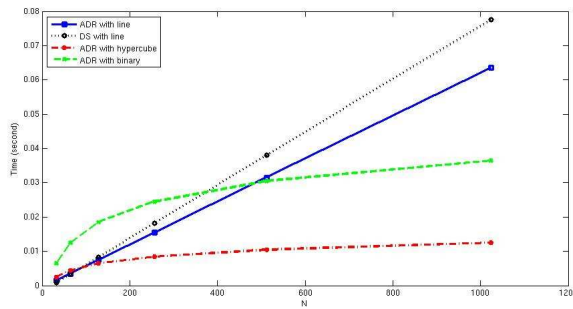


Figure 4: Comparison of computation time in seconds with  $n = 10, m = 16, t_1 = t_2 = 5 \times 10^{-8}(s), t_3 = 5 \times 10^{-5}(s)$ .

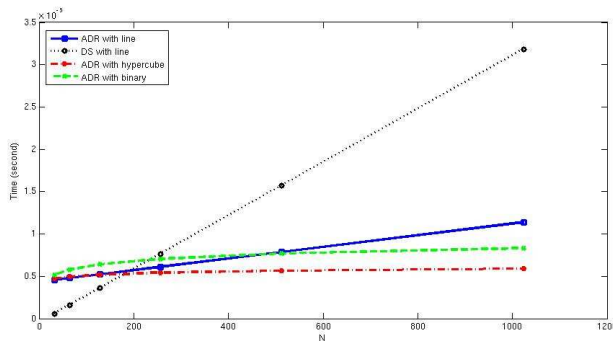


Figure 5: Comparison of computation time in seconds with  $n = 10, m = 16, t_1 = t_2 = 5 \times 10^{-8}(s), t_3 = 5 \times 10^{-7}(s)$ .

cient method requiring little computation for real time realizations since small numbers of contour quadrature points are generally enough to yield good approximations. In Lenczner et al. [5] a mathematical framework was introduced to derive diffusive realization of operators solution to linear operator partial differential equations in one-dimensional domains. Only the premises of a numerical method were produced with very few numerical tests. Here, a more complete numerical method is discussed and its performances are investigated. In particular, it uses another method for analytic extension and introduces a contour optimization based on a theoretical error estimate which reduces computations. The approach is presented through a simple example of a Lyapunov equation arising in optimal control theory from the one-dimensional heat equation. However, the special features of this example are not taken into account and we do not expect further limitations in broader applications. In view of real-time applications,

we have also discussed the cost of this method for a parallel implementation.

## Acknowledgements

This work is funded by the Labex ACTION program (contract ANR-11-LABX-01-01).

## References

- [1] C. Casenave. Représentation diffusive et inversion opératoire pour l'analyse et la résolution de problèmes dynamiques non locaux. PhD thesis, Université Toulouse III - Paul Sabatier, 2009.
- [2] T. Hélie, D. Matignon, and R. Mignot. Criterion design for optimizing low-cost approximations of infinite dimensional systems: towards efficient real-time simulation. *Int. J. Tomogr. Stat.*, 7(F07):13-18, 2007.
- [3] L. Laudebat, P. Bidan, and G. Montseny. Modeling and optimal identification of pseudodifferential electrical dynamics by means of diffusive representation - Part 1: Modeling. *IEEE Transactions on Circuits and Systems I- Regular Papers*, 51(9):1801-1813, 2004.
- [4] M. Lenczner and G. Montseny. Diffusive realization of operator solutions of certain operational partial differential equations. *C.R. Math. Acad. Sci. Paris*, 341(12): 737-740, 2005.
- [5] Michel Lenczner, Gerard Montseny, and Youssef Yakoubi. Diffusive realizations for solutions of some operator equations: the one-dimensional case. *Mathematics of Computation*, 81(277):319-344, Jan 2012.
- [6] M. López-Fernández, C. Lubich, C. Palencia, and A. Schädle. Fast Runge-Kutta approximation of inhomogeneous parabolic equations. *Numer. Math.*, 102(2): 277-291, 2005.
- [7] López-Fernández, M., and C. Palencia. "On the numerical inversion of the Laplace transform of certain holomorphic mappings." *Applied Numerical Mathematics* 51, Issues 2-3 (2004): 289-303.
- [8] G. Montseny. Représentation diffusive. Hermès-Sciences, 2005.
- [9] F. Stenger. Approximations via Whittaker's cardinal function. *Journal of Approximation Theory*, 17(3):222-240, 1976.
- [10] J. A. C. Weideman and L. N. Trefethen. Parabolic and hyperbolic contours for computing the Bromwich integral. *Math. Comp.*, 76(259):1341-1356, 2007.
- [11] Y. Yakoubi. Deux Méthodes d'Approximation pour un Contrôle Optimal Semi-Décentralisé pour des Systèmes Distribués. PhD thesis, Université de Franche-Comté, 2010.
- [12] Yakoubi, Y., Lenczner, M., Goavec-Merou, G., Couturier, R., and Friedt, J. M. (2011, August). Diffusive Realization of a Lyapunov Equation Solution, and Its FPGA Implementation. In *IFAC Conference* (pp. 5477-5482).