



**HAL**  
open science

# A NEW APPROACH TO IMPROVE ILL-CONDITIONED PARABOLIC OPTIMAL CONTROL PROBLEM VIA TIME DOMAIN DECOMPOSITION

Mohamed-Kamel Riahi

► **To cite this version:**

Mohamed-Kamel Riahi. A NEW APPROACH TO IMPROVE ILL-CONDITIONED PARABOLIC OPTIMAL CONTROL PROBLEM VIA TIME DOMAIN DECOMPOSITION. 2020. hal-00974285v3

**HAL Id: hal-00974285**

**<https://inria.hal.science/hal-00974285v3>**

Preprint submitted on 4 Nov 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright



29 refer to [19, 34, 38] and also [55] as a few standard literatures in the topic. In the past decade,  
 30 following the increasing advances of computing power many algorithms based parallel techniques  
 31 and implementations have been proposed to the approximation of the solution of optimal control  
 32 problems with PDE constrains, see for instance [48, 5, 57, 52] and also see [9, 13, 53] for Stocks  
 33 and Navier-Stokes flow problem. We may also refer to [28] as a recent survey in this field.

34 It is well-known that the steepest descent algorithm has a slow convergence rate with ill-  
 35 conditioned problems because the number of iterations is proportional to the condition number  
 36 of the problem. The method of J.Barzila and J.Borwein [6] based on two-point step-length for  
 37 the steepest-descent method for approximating the secant equation avoids this handicap. Our  
 38 method is very different because first, it is based on a decomposition of the unknown and proposes  
 39 a set of bloc descent directions, and second because it is general where it can be coupled together  
 40 with any least-square-like optimization procedure.

41 The theoretical basis of our approach is presented and applied to the optimization of a positive  
 42 definite quadratic form. Then we apply it on a complex engineering problem involving control of  
 43 system governed by PDEs. We consider the optimal heat control which is known to be ill-posed  
 44 in general (and well-posed under some assumptions) and presents some particular theoretical  
 45 and numerical challenges. We handle the ill-posedness degree of the heat control problem by  
 46 varying the regularization parameter and apply our methods in the handled problem to show the  
 47 efficiency of our algorithm. The distributed- and boundary-control cases are both considered.

48 This paper is organized as follows: In Section 2, we present our method in a linear algebra  
 49 framework to highlight its generality. Section 3 is devoted to the introduction of the optimal  
 50 control problem with constrained PDE on which we will apply our method. We present the  
 51 Euler-Lagrange-system associated to the optimization problem and give the explicit formulation  
 52 of the gradient in both cases of distributed- and boundary-control. Then, we present and explain  
 53 the parallel setting for our optimal control problem. In Section 4, we perform the convergence  
 54 analysis of our parallel algorithm. In Section 5, we present the numerical experiments that  
 55 demonstrate the efficiency and the robustness of our approach. We make concluding remarks in  
 56 Section 6. For completeness, we include calculus results in the Appendix.

57 Let  $\Omega$  be a bounded domain in  $\mathbb{R}^3$ , and  $\Omega_c \subset \Omega$ , the boundary of  $\Omega$  is denoted by  $\partial\Omega$ . We  
 58 denote by  $\Gamma \subset \partial\Omega$  a part of this boundary. We denote  $\langle \cdot, \cdot \rangle_2$  (respectively  $\langle \cdot, \cdot \rangle_c$  and  $\langle \cdot, \cdot \rangle_\Gamma$ ) the  
 59 standard  $L^2(\Omega)$  (respectively  $L^2(\Omega_c)$  and  $L^2(\Gamma)$ ) inner-product that induces the  $L^2(\Omega)$ -norm  $\|\cdot\|_2$   
 60 on the domain  $\Omega$  (respectively  $\|\cdot\|_c$  on  $\Omega_c$  and  $\|\cdot\|_\Gamma$  on  $\Gamma$ ).

61 In the case of finite dimensional vector space in  $\mathbb{R}^m$ , the scalar product  $a^T b$  of  $a$  and  $b$  (where  
 62  $a^T$  stands for the transpose of  $a$ ) is denoted by  $\langle \cdot, \cdot \rangle_2$  too. The scalar product with respect to  
 63 the matrix  $A$ , i.e.  $\langle x, Ax \rangle_2$  is denote by  $\langle x, x \rangle_A$  and its induced norm is denoted by  $\|x\|_A$ . The  
 64 transpose of the operator  $A$  is denoted by  $A^T$ . The Hilbert space  $L^2(0, T; L^2(\Omega_c))$  (respectively  
 65  $L^2(0, T; L^2(\Gamma))$ ) is endowed by the scalar product  $\langle \cdot, \cdot \rangle_{c,I}$  (respectively  $\langle \cdot, \cdot \rangle_{\Gamma,I}$ ) that induces the  
 66 norm  $\|\cdot\|_{c,I}$  (respectively  $\|\cdot\|_{\Gamma,I}$ ).

67

## 2. ENHANCED STEEPEST DESCENT ITERATIONS

68 The steepest descent algorithm minimizes at each iteration the quadratic function  $q(x) =$   
 69  $\|x - x^*\|_A^2$ , where  $A$  is assumed to be a symmetric positive definite (SPD) matrix and  $x^*$  is the  
 70 minimum of  $q$ . The vector  $-\nabla q(x)$  is locally the descent direction that yields the fastest rate of  
 71 decrease of the quadratic form  $q$ . Therefore all vectors of the form  $x + \theta \nabla q(x)$ , where  $\theta$  is a suitable  
 72 negative real value, minimize  $q$ . The choice of  $\theta$  is found by looking for the  $\min_{s < 0} q(x + s \nabla q(x))$   
 73 with the use of a line-search technique. In the case where  $q$  is a quadratic form  $\theta$  is given by  
 74  $-\|\nabla q(x)\|_2^2 / \|\nabla q(x)\|_A^2$ . We recall in Algorithm 1 the steepest descent algorithm; *Convergence*  
 75 is a boolean variable based on estimation of the residual vector  $r^k < \epsilon$ , where  $\epsilon$  is the stopping  
 76 criterion.

**Algorithm 1:** Steepest descent.

---

**Input:**  $x^0, \epsilon$ ;  
**1**  $k = 0$ ;  
**2 repeat**  
**3**  $r^k = \nabla q^k := \nabla q(x^k)$ ;  
**4** Compute  $Ar^k$ ;  
**5** Compute  $\theta^k = -\|r^k\|_2^2 / \|r^k\|_A^2$ ;  
**6**  $x^{k+1} = x^k + \theta^k r^k$ ;  
**7**  $k = k + 1$ ;  
**8 until**  $\|r\|^k \leq \epsilon$ ;

---

77 Our method proposes to modify the step 5. of Algorithm 1. It considers the step-length  
78  $\theta \in \mathbb{R}_+ \setminus \{0\}$  as a vector in  $\mathbb{R}^{\hat{n}} \setminus \{\mathbf{0}\}$  where  $\hat{n}$  is an integer such that  $1 \leq \hat{n} \leq \text{size}(x)$ , we shall  
79 denote this new vector as  $\Theta_{\hat{n}}$ .

In the following, it is assumed that for a giving vector  $x \in \mathbb{R}^m$ , the integer  $\hat{n}$  divides  $m$  with  
with null remainder. In this context, let us introduce the identity operators  $I_{\mathbb{R}^m}$  witch is an  
 $m$ -by- $m$  matrix and its partition (partition of unity) given by the projection operators  $\{\pi_n\}_{n=1}^{\hat{n}}$  :  
projectors from  $\mathbb{R}^m$  into a set of canonical basis  $\{e_i\}_i$ . These operators are defined for  $1 \leq n \leq \hat{n}$   
by

$$\pi_n(x) = \sum_{i=(n-1) \times \frac{m}{\hat{n}} + 1}^{n \times \frac{m}{\hat{n}}} \langle e_i, x \rangle_2 e_i.$$

80 For reading conveniences, we define  $\tilde{x}_n$  a vector in  $\mathbb{R}^{\frac{m}{\hat{n}}}$  such that  $\tilde{x}_n := \pi_n(x)$ . The concatenation  
81 (denoted for simplicity by  $\sum$ ) of  $\tilde{x}_n$  for all  $1 \leq n \leq \hat{n}$  reads  $\hat{x}_{\hat{n}} \equiv (\tilde{x}_1^T, \dots, \tilde{x}_{\hat{n}}^T)^T$ .

82 Recall the gradient  $\nabla_x := (\frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_m})^T$ , and define the bloc gradient  $\nabla_{\hat{x}_{\hat{n}}} := (\nabla_{\tilde{x}_1}^T, \dots, \nabla_{\tilde{x}_{\hat{n}}}^T)^T$ ,  
83 where obviously  $\nabla_{\tilde{x}_n} := (\frac{\partial}{\partial x_{(n-1) \times \frac{m}{\hat{n}} + 1}}, \dots, \frac{\partial}{\partial x_{n \times \frac{m}{\hat{n}}}})^T$ . In the spirit of this decomposition we in-  
84 vestigate, in the sequel, the local descent directions as the bloc partial derivatives with respect to  
85 the bloc-variables  $(\tilde{x}_n)_{n=1}^{\hat{n}}$ . We aim, therefore, at finding  $\Theta_{\hat{n}} = (\theta_1, \dots, \theta_{\hat{n}})^T \in \mathbb{R}^{\hat{n}}$  that ensures  
86 the  $\min_{(\theta_n)_n < 0} q(\hat{x}_{\hat{n}}^k + \sum_{n=1}^{\hat{n}} \theta_n \nabla_{\tilde{x}_n} q(\hat{x}_{\hat{n}}^k))$ .

87 We state hereafter a motivating result, which its proof is straightforward because the spaces  
88 are embedded. Let us first, denote by

$$(1) \quad \Phi_{\hat{n}}(\Theta_{\hat{n}}) = q\left(\hat{x}_{\hat{n}} + \sum_{n=1}^{\hat{n}} \theta_n \nabla_{\tilde{x}_n} q(\hat{x}_{\hat{n}})\right)$$

89 which is quadratic because  $q$  is.

**Theorem 2.1.** *According to the definition of  $\Phi_{\hat{n}}(\Theta_{\hat{n}})$  (see Eq.(1)) we immediately have*

$$\min_{\mathbb{R}^p} \Phi_p(\Theta_p) \leq \min_{\mathbb{R}^q} \Phi_q(\Theta_q) \quad \forall q < p.$$

The new algorithm we discuss in this paper proposes to define a sequence  $(\hat{x}_{\hat{n}}^k)_k$  of vectors  
that converges to  $x^*$  unique minimizer of the quadratic form  $q$ . The update formulae reads:

$$\tilde{x}_n^{k+1} = \tilde{x}_n^k + \theta_n^k \nabla_{\tilde{x}_n} q(\hat{x}_{\hat{n}}^k),$$

90 where we recall that  $\hat{n}$  is an arbitrarily chosen integer. Then  $\hat{x}_{\hat{n}}^{k+1} = \sum_{n=1}^{\hat{n}} \tilde{x}_n^{k+1}$ .

91 We shall explain now how one can accurately computes the vector step-length  $\Theta_{\hat{n}}^k$  at each  
 92 iteration  $k$ . It is assumed that  $q$  is a quadratic form. From Eq.(1) using the chain rule, we obtain  
 93 the Jacobian vector  $\Phi'_{\hat{n}}(\Theta_{\hat{n}}) \in \mathbb{R}^{\hat{n}}$  given by

$$(2) \quad (\Phi'_{\hat{n}}(\Theta_{\hat{n}}))_j = (\nabla_{\hat{x}_j} q(\hat{x}_{\hat{n}}^k))^T \nabla_{\hat{x}_j} q \left( \hat{x}_{\hat{n}}^k + \sum_{n=1}^{\hat{n}} \theta_n \nabla_{\hat{x}_n} q(\hat{x}_{\hat{n}}^k) \right) \in \mathbb{R},$$

and the Hessian matrix  $\Phi''_{\hat{n}}(\Theta_{\hat{n}}) \in \mathbb{R}^{\hat{n} \times \hat{n}}$  is given by

$$(\Phi''_{\hat{n}}(\Theta_{\hat{n}}))_{i,j} = (\nabla_{\hat{x}_j} q(\hat{x}_{\hat{n}}^k))^T \left( \nabla_{\hat{x}_i} \nabla_{\hat{x}_j} q \left( \hat{x}_{\hat{n}}^k + \sum_{n=1}^{\hat{n}} \theta_n \nabla_{\hat{x}_n} q(\hat{x}_{\hat{n}}^k) \right) \right) \nabla_{\hat{x}_j} q(\hat{x}_{\hat{n}}^k).$$

94 It is worth noticing that the matrix  $\nabla_{\hat{x}_i} \nabla_{\hat{x}_j} q \left( \hat{x}_{\hat{n}}^k + \sum_{n=1}^{\hat{n}} \theta_n \nabla_{\hat{x}_n} q(\hat{x}_{\hat{n}}^k) \right)$  is a bloc portion of the  
 95 Hessian matrix  $A$ . However if the gradient  $\nabla_{\hat{x}_n} q \in \mathbb{R}^{\frac{m}{n}}$  assumes an extension by zero (denoted  
 96 by  $\tilde{\nabla}_{\hat{x}_i} q$ ) to  $\mathbb{R}^m$  so the matrix  $\Phi''_{\hat{n}}(\Theta_{\hat{n}})$  has therefore the simplest implementable form

$$(3) \quad (\Phi''_{\hat{n}}(\Theta_{\hat{n}}))_{i,j} = (\tilde{\nabla}_{\hat{x}_j} q(\hat{x}_{\hat{n}}^k))^T A \tilde{\nabla}_{\hat{x}_i} q(\hat{x}_{\hat{n}}^k).$$

97 We thus have the expansion  $\Phi_{\hat{n}}(\Theta_{\hat{n}}^k) = \Phi_{\hat{n}}(\mathbf{0}) + (\Theta_{\hat{n}}^k)^T \Phi'_{\hat{n}}(\mathbf{0}) + \frac{1}{2} (\Theta_{\hat{n}}^k)^T \Phi''_{\hat{n}}(\mathbf{0}) \Theta_{\hat{n}}^k$ , with  $\mathbf{0} :=$   
 98  $(0, \dots, 0)^T \in \mathbb{R}^{\hat{n}}$ . Then the vector  $\Theta_{\hat{n}}^k$  that annuls the gradient writes:

$$(4) \quad \Theta_{\hat{n}}^k = -\Phi''_{\hat{n}}(\mathbf{0})^{-1} \Phi'_{\hat{n}}(\mathbf{0}).$$

99 Algorithm 1 has therefore a bloc structure which can be solved in parallel. This is due to the fact  
 100 that partial derivatives can be computed independently. The new algorithm is thus as follows  
 101 (see Algorithm 2)

---

**Algorithm 2:** Enhanced steepest descent.

---

```

k = 0;
Input:  $\hat{x}_{\hat{n}}^0 \in \mathbb{R}^m, \epsilon;$ 
1 repeat
2   forall the  $1 \leq n \leq \hat{n}$  do
3      $\tilde{x}_n^k = \pi_n(\hat{x}_{\hat{n}}^k);$ 
4      $r_n = \nabla_{\tilde{x}_n^k} q(\hat{x}_{\hat{n}}^k);$ 
5      $resize(r_n)$  (i.e. extension by zero means simply project on  $\mathbb{R}^m$ );
6   end
7   Assemble  $\Phi'_{\hat{n}}(\mathbf{0})$  with element  $(\Phi'_{\hat{n}}(\mathbf{0}))_j = r_j^T r_j$  according to Eq.(2);
8   Assemble  $\Phi''_{\hat{n}}(\mathbf{0})$  with element  $(\Phi''_{\hat{n}}(\mathbf{0}))_{i,j} = r_i^T A r_j$  according to Eq.(3);
9   Compute  $\Theta_{\hat{n}}^k$  solution of Eq.(4);
10  Update  $\hat{x}_{\hat{n}}^{k+1} = \hat{x}_{\hat{n}}^k + \sum_n \theta_n \nabla_{\hat{x}_n} q(\hat{x}_{\hat{n}}^k);$ 
11   $r^k = \sum_n \|r_n\|_2;$ 
12   $k = k + 1;$ 
13 until  $r^k \leq \epsilon;$ 

```

---

102 It is noteworthy that algorithm 2 reduces to the usual steepest descent method when  $\hat{n} = 1$   
 103 and reduces to the usual Newton's method for  $\hat{n} = m$ . The computational effort of the pre-  
 104 sented method, although spread across processor in parallel fashion, could represent an efficiency-  
 105 damper when the parameter  $\hat{n}$  is close to the size of the original matrix. This is very apparent  
 106 at the step 9 on the inversion of the Hessian  $\Phi''_{\hat{n}}(\mathbf{0})$ (which becomes the global Hessian if  $\hat{n} = m$ ).  
 107

108 In the sequel we shall discuss a case where the global Hessian is not given explicitly (only  
 109 matrix-vector product operation is known). Our method uses a new fashion of decomposition of  
 110 the time-domain in order to provide a small representative of the global Hessian via a projection  
 111 upon a well chosen time-basis. That will improve the optimization iteration scheme. We shall  
 discuss in details this situation with optimal control problem with PDE constrains.

### 112 3. APPLICATION TO A PARABOLIC OPTIMAL CONTROL PROBLEM

113 In this part we are interested in the application of Algorithm 2 in a finite element computa-  
 114 tional engineering problem involving optimization with constrained PDE. Because of the large  
 115 scale system arising from the discretization of the control problem, specially in the case of con-  
 116 strained time dependent PDE, its is necessary to divide the problem in order to deal with a  
 117 set of small systems, which their resolutions together may lead to (or to an approximation of)  
 118 the solution of the initial problem. Those small systems may i) have the same structure of the  
 119 global system, and this method are known as receding horizon control e.g. [14, 35, 26] from  
 120 which inherits the instantaneous control technique e.g. [13, 27, 43] which is based on the time  
 121 discretiation scheme, or ii) could be a reduced version of the global system using a proper re-  
 122 duced basis projection (see for instance [30, 29]), typically via proper orthogonal decomposition  
 123 (POD) [24, 3, 2, 1, 32] or a snapshot variant of the POD (see for instance [20, 21, 47]). All  
 124 the aforementioned techniques (and many others) could be interpreted as "suboptimal" methods  
 125 for the control problem. The last decade has known much interest on time-domain decomposi-  
 126 tion [41, 39, 56, 33] related to optimal control with PDE constrains. In [25] it has been proposed  
 127 in a distributed optimal control problem context a time-domain decomposition algorithm. In  
 128 the light of the snapshots technique and multiple-shooting algorithm, the continuity conditions  
 129 (at the breakpoints resulting from the time-domain decomposition) has been reformulated as  
 130 a discrete time optimal control. The Krylov-type resolution of the whole system is therefore  
 131 preconditioned via block Gauss-Seidel resolution of the tridiagonal system of the optimality  
 132 condition. This method [25] points out the strong relationship between the aforementioned in-  
 133 stantaneous control technique and the application of one step forward Gauss-Seidel method. In  
 134 spite of the very promising numerical experiments of this method, a general convergence theory  
 135 is lacking.

136 In our previous effort [40] we have proposed a time-domain decomposition method based on  
 137 iterations that require an approximated solution of optimal control problem restricted to time  
 138 subdomains (the approximated solutions could be at least the results of one iteration of the  
 139 gradient method). It is worth noticing that the method proposed in [40] like the one proposed  
 140 in [25] uses snapshot technique. However, in [25] the author used the fact that the target is  
 141 provided during the integration time domain (because the cost functional is a mixed integral  
 142 and final value), in [40], since the cost functional provides only the final value as a target func-  
 143 tion, it was necessary to improvise intermediate targets in order to define small optimal control  
 144 problems (of the same type as the original problem) restricted to time subdomains. The method  
 145 presented in this paper, is completely different from these techniques, where one iterations in  
 146 our new algorithm require several resolution of the same problem but with control variable only  
 147 defined on subdomains. This step, even though it increases the complexity compared to stan-  
 148 dard approach, but this excess in the complexity is achieved through parallel simulations. This  
 149 technique somehow enables us to approximate the second derivative of the infinite dimensional

150 cost functional by a finite dimensional matrix with a relatively small size (i.e.  $\hat{n}$ ). The method  
 151 that we propose could be interpreted therefore as “suboptimal” control technique regarding the  
 152 reduction made on the Hessian. It could be viewed also as preconditioner to the Krylov-type  
 153 methods, where it is hard to theoretically prove that the condition number is improved, however  
 154 numerical results support our claims.

155 In this paper, we deal with the optimal control problem of a system, which is governed by the  
 156 heat equation. We shall present two types of control problems. The first concerns the distributed  
 157 optimal control and the second concerns the Dirichlet boundary control. The main difference  
 158 from the algorithm just presented in linear algebra is that the decomposition is applied on the  
 159 time-domain that involve only the control variable. This technique is not classical, we may refer  
 160 to a similar approaches that has been proposed for the time domain decomposition in application  
 161 to the control problem, for instance [41, 39, 40] which basically they use a variant of the parareal  
 162 in time algorithm [36].

163 **3.1. Distributed optimal control problem.** Let us briefly present the steepest descent method  
 164 applied to the following optimal control problem: find  $v^*$  such that

$$(5) \quad J(v^*) = \min_{v \in L^2(0,T;L^2(\Omega_c))} J(v),$$

165 where  $J$  is a quadratic cost functional defined by

$$(6) \quad J(v) = \frac{1}{2} \|y(T) - y^{target}\|_2^2 + \frac{\alpha}{2} \int_I \|v\|_c^2 dt,$$

166 where  $y^{target}$  is a given target state and  $y(T)$  is the state variable at time  $T > 0$  of the heat  
 167 equation controlled by the variable  $v$  over  $I := [0, T]$ . The Tikhonov regularization parameter  $\alpha$   
 168 is introduced to penalize the control’s  $L^2$ -norm over the time interval  $I$ . The optimality system  
 169 of our problem reads:

$$(7) \quad \begin{cases} \partial_t y - \sigma \Delta y = \mathcal{B}v, & \text{on } I \times \Omega, \\ y(t=0) = y_0. \end{cases}$$

$$(8) \quad \begin{cases} \partial_t p + \sigma \Delta p = 0, & \text{on } I \times \Omega, \\ p(t=T) = y(T) - y^{target}. \end{cases}$$

$$(9) \quad \nabla J(v) = \alpha v + \mathcal{B}^T p = 0, \text{ on } I \times \Omega.$$

170 In the above equations,  $\sigma$  stands for the diffusion coefficient, the operator  $\mathcal{B}$  stands for the  
 171 characteristic function of  $\Omega_c \subset \Omega$ , the state variable  $p$  stands for the Lagrange multiplier (adjoint  
 172 state) solution of the backward heat equation Eq.(8), Eq.(7) is called the forward heat equation.  
 173 The optimal control problem (5) is known to be well-posed [37]. The boundary conditions for  
 174 the Eqs.(7)-(8) does’t affect the results of our algorithm. The reader could apply his own choice  
 175 of boundary condition.

176 **3.2. Dirichlet boundary optimal control problem.** In this subsection we are concerned with  
 177 the PDE constrained Dirichlet boundary optimal control problem, where we aim at minimizing  
 178 the cost functional  $J_\Gamma$  defined by

$$(10) \quad J_\Gamma(v_\Gamma) = \frac{1}{2} \|y_\Gamma(T) - y^{target}\|_2^2 + \frac{\alpha}{2} \int_I \|v_\Gamma\|_\Gamma^2 dt,$$

179 where the control variable  $v_\Gamma$  is only acting on the boundary  $\Gamma \subset \partial\Omega$ . Here too,  $y^{target}$  is a given  
 180 target state (not necessary equal the one defined in the last subsection ! ) and  $y_\Gamma(T)$  is the state  
 181 variable at time  $T > 0$  of the heat equation controlled by the variable  $v_\Gamma$  during the time interval

182  $I := [0, T]$ . This kind of optimal control problem are known to be ill-posed [11, 7]. As before  $\alpha$   
 183 is a regularization term. The involved optimality system reads

$$(11) \quad \begin{cases} \partial_t y_\Gamma - \sigma \Delta y_\Gamma &= f & \text{on } I \times \Omega \\ y_\Gamma &= v_\Gamma & \text{on } I \times \Gamma \\ y_\Gamma &= g & \text{on } I \times \{\partial\Omega \setminus \Gamma\} \\ y_\Gamma(0) &= y_0 \end{cases}$$

$$(12) \quad \begin{cases} \partial_t p_\Gamma + \sigma \Delta p_\Gamma &= 0 & \text{on } I \times \Omega \\ p_\Gamma &= 0 & \text{on } I \times \partial\Omega \\ p_\Gamma(T) &= y_\Gamma(T) - y^{target} \end{cases}$$

$$(13) \quad \nabla J_\Gamma(v_\Gamma) = \alpha v_\Gamma - (\nabla p_\Gamma)^T \vec{n} = 0 \quad \text{on } I \times \Gamma,$$

184 where  $f \in L^2(\Omega)$  is any source term,  $g \in L^2(\Gamma)$  and  $\vec{n}$  is the outward unit normal on  $\Gamma$ . the  
 185 adjoint state  $p_\Gamma$  stands for the Lagrange multiplier, and is the solution of the backward heat  
 186 equation Eq.(12). Both functions  $f$  and  $g$  will be given explicitly for each numerical test that  
 187 we consider in the numerical experiment section.

188 **3.3. Steepest descent algorithm for optimal control of constrained PDE.** In the optimal  
 189 control problem, the evaluation of the gradient as it is clear in Eq.(9) (respectively (13)) requires  
 190 the evaluation of the time dependent Lagrange multiplier  $p$  (respectively  $p_\Gamma$ ). This fact, makes the  
 191 steepest descent optimization algorithm slightly different from the Algorithm 1 already presented.

192 Let us denote by  $k$  the current iteration superscript. We suppose that  $v^0$  is known. The first  
 193 order steepest descent algorithm updates the control variable as follows:

$$(14) \quad v^k = v^{k-1} + \theta^{k-1} \nabla J(v^{k-1}), \text{ for } k \geq 1,$$

The step-length  $\theta^{k-1} \in \mathbb{R}_+ \setminus \{0\}$  in the direction of the gradient  $\nabla J(v^{k-1}) = \alpha v^{k-1} + \mathcal{B}^T p^{k-1}$   
 (respectively  $\nabla J_\Gamma(v_\Gamma^{k-1}) = \alpha v_\Gamma - (\nabla p_\Gamma)^T \vec{n}$ ) is computed as :

$$\theta^{k-1} = -\|\nabla J(v^{k-1})\|_{c,I}^2 / \|\nabla J(v^{k-1})\|_{\nabla^2 J}^2.$$

194 The above step-length  $\theta^{k-1}$  (respectively  $\theta_\Gamma^{k-1}$ ) is optimal (see e.g. [15]) in the sense that it  
 195 minimizes the functional  $\theta \rightarrow J(v^{k-1} + \theta \nabla J(v^{k-1}))$  (respectively  $\theta \rightarrow J_\Gamma(v_\Gamma^{k-1} + \theta \nabla J_\Gamma(v_\Gamma^{k-1}))$ ).  
 196 The rate of convergence of this technique is  $(\frac{\kappa-1}{\kappa+1})^2$ , where  $\kappa$  is the condition number of the  
 197 quadratic form, namely the Hessian of the cost functional  $J$  (respectively  $J_\Gamma$ ) that writes:

$$\nabla^2 J(v)(\delta v, \delta w) = \langle \delta y(T)(\delta v), \delta y(T)(\delta w) \rangle + \alpha \int_0^T \langle \delta v, \delta w \rangle dt,$$

198 where  $\delta y(T)(\delta v)$  (respectively  $\delta y(T)(\delta v)$ ) is the state variable evaluated at time  $T$  where it has  
 199 been controlled during the entire time domain  $[0, T]$  by the control  $\delta v$  (respectively  $\delta w$ ). It is  
 200 worth noticing that the matrix representation of the Hessian operator is not assembled during  
 201 the optimization procedure even after space discretization. Our method only evaluate it and  
 202 never invert it.

203 **3.4. Time-domain decomposition algorithm.** Consider  $\hat{n}$  subdivisions of the time interval  
 204  $I = \cup_{n=1}^{\hat{n}} I_n$ , consider also the following convex cost functional  $J$ :

$$(15) \quad J(v_1, v_2, \dots, v_{\hat{n}}) = \frac{1}{2} \|\mathcal{Y}(T) - y^{target}\|_2^2 + \frac{\alpha}{2} \sum_{n=1}^{\hat{n}} \int_{I_n} \|v_n\|_c^2 dt,$$

205 where  $v_n, n = 1, \dots, \hat{n}$  are control variables with time support included in  $I_n, n = 1, \dots, \hat{n}$ . The  
 206 state  $\mathcal{Y}(T)$  stands for the sum of state variables  $\mathcal{Y}_n$  which are time-dependent state variable

207 solution to the heat equation controlled by the variable  $v_n$ . Obviously because the control is  
 208 linear the state  $\mathcal{Y}$  depends on the concatenation of controls  $v_1, v_2, \dots, v_{\hat{n}}$  namely  $v = \sum_{n=1}^{\hat{n}} v_n$ .

209 Let us define  $\Theta_{\hat{n}} := (\theta_1, \theta_2, \dots, \theta_{\hat{n}})^T$  where  $\theta_n \in \mathbb{R}_- \setminus \{0\}$ . For any admissible control  $w =$   
 210  $\sum_{n=1}^{\hat{n}} w_n$ , we also define  $\varphi_{\hat{n}}(\Theta_{\hat{n}}) := J(v + \sum_{n=1}^{\hat{n}} \theta_n w_n)$ , which is quadratic. We have:

$$(16) \quad \varphi_{\hat{n}}(\Theta_{\hat{n}}) = \varphi_{\hat{n}}(\mathbf{0}) + \Theta_{\hat{n}}^T \nabla \varphi_{\hat{n}}(\mathbf{0}) + \frac{1}{2} \Theta_{\hat{n}}^T \nabla^2 \varphi_{\hat{n}}(\mathbf{0}) \Theta_{\hat{n}},$$

211 where  $\mathbf{0} = (0, \dots, 0)^T$ . Therefore we can write  $\nabla \varphi_{\hat{n}}(\Theta_{\hat{n}}) \in \mathbb{R}^{\hat{n}}$  as  $\nabla \varphi_{\hat{n}}(\Theta_{\hat{n}}) = D(v, w) +$   
 212  $H(v, w) \Theta_{\hat{n}}$ , where the Jacobian vector and the Hessian matrix are given respectively by:

$$\begin{aligned} D(v, w) &:= (\langle \nabla J(v), \boldsymbol{\pi}_1(w) \rangle_c, \dots, \langle \nabla J(v), \boldsymbol{\pi}_{\hat{n}}(w) \rangle_c)^T \in \mathbb{R}^{\hat{n}}, \\ H(v, w) &:= (H_{n,m})_{n,m}, \text{ for } H_{n,m} = \langle \boldsymbol{\pi}_n(w), \boldsymbol{\pi}_m(w) \rangle_{\nabla^2 J}. \end{aligned}$$

213 Here,  $(\boldsymbol{\pi}_n)$  is the restriction over the time interval  $I_n$ , indeed  $\boldsymbol{\pi}_n(w)$  has support on  $I_n$  and  
 214 assumes extension by zero in  $I$ . The solution  $\Theta_{\hat{n}}^*$  of  $\nabla \varphi_{\hat{n}}(\Theta_{\hat{n}}) = \mathbf{0}$  can be written in the form:

$$(17) \quad \Theta_{\hat{n}}^* = -H^{-1}(v, w) D(v, w).$$

215 In the parallel distributed control problem, we are concerned with the following optimality sys-  
 216 tem:

$$(18) \quad \begin{cases} \partial_t \mathcal{Y}_n - \sigma \Delta \mathcal{Y}_n = \mathcal{B} v_n, & \text{on } I \times \Omega, \\ \mathcal{Y}_n(t=0) = \delta_n^0 y_0. \end{cases}$$

$$(19) \quad \mathcal{Y}(T) = \sum_{n=1}^{\hat{n}} \mathcal{Y}_n(T)$$

$$(20) \quad \begin{cases} \partial_t \mathcal{P} + \sigma \Delta \mathcal{P} = 0, & \text{on } I \times \Omega, \\ \mathcal{P}(t=T) = \mathcal{Y}(T) - y^{target}. \end{cases}$$

$$(21) \quad \nabla J(\sum_{n=1}^{\hat{n}} v_n) = \mathcal{B}^T \mathcal{P} + \alpha \sum_{n=1}^{\hat{n}} v_n = 0, \text{ on } I \times \Omega.$$

217 where  $\delta_n^0$  stands for the function taking value "1" only if  $n = 1$ , else it takes the value "0". The  
 218 Dirichlet control problem we are concerned with:

$$(22) \quad \begin{cases} \partial_t \mathcal{Y}_{n,\Gamma} - \sigma \Delta \mathcal{Y}_{n,\Gamma} = f & \text{on } I \times \Omega \\ \mathcal{Y}_{n,\Gamma} = v_{n,\Gamma} & \text{on } I \times \Gamma \\ \mathcal{Y}_{n,\Gamma} = g & \text{on } I \times \{\partial\Omega \setminus \Gamma\} \\ \mathcal{Y}_{n,\Gamma}(0) = \delta_n^0 y_0. \end{cases}$$

$$(23) \quad \mathcal{Y}_{\Gamma}(T) = \sum_{n=1}^{\hat{n}} \mathcal{Y}_{n,\Gamma}(T)$$

$$(24) \quad \begin{cases} \partial_t \mathcal{P}_{\Gamma} + \sigma \Delta \mathcal{P}_{\Gamma} = 0 & \text{on } I \times \Omega \\ \mathcal{P}_{\Gamma} = 0 & \text{on } I \times \partial\Omega \\ \mathcal{P}_{\Gamma}(T) = \mathcal{Y}_{\Gamma}(T) - y^{target}. \end{cases}$$

$$(25) \quad \nabla J_{\Gamma}(\sum_{n=1}^{\hat{n}} v_{n,\Gamma}) = -(\nabla \mathcal{P}_{\Gamma})^T \vec{n} + \alpha \sum_{n=1}^{\hat{n}} v_{n,\Gamma} = 0 \quad \text{on } I \times \Gamma.$$

It is noteworthy that each of the forward problems Eq (18) and (22) with respect to  $1 \leq n \leq \hat{n}$  are fully performed in parallel over the entire time interval  $I$ , and that the time domain decomposition is only for the control variable  $v$ . More precisely; in order to evaluate the gradient Eq (21) (respectively (25)) we solve  $\hat{n}$  forward problem in parallel, then we compute the solution  $\mathcal{Y}(T)$  at Eq (23) (respectively (23)). Finally we evaluate the adjoint solution of the backward

problem Eq (20)(respectively Eq (24) ). It is recalled that the superscript  $k$  denotes the iteration index. The update formulae for the control variable  $v^k$  is given by:

$$v_n^k = v_n^{k-1} + \theta_n^{k-1} \nabla J(v_n^{k-1}).$$

219 We show hereafter how to assemble vector step-length  $\Theta_{\hat{n}}^k$  at each iteration. For the purposes  
220 of notation we denote by  $H_k$  the  $k$ -th iteration of the Hessian matrix  $H(\nabla J(v^k), \nabla J(v^k))$  and by  
221  $D_k$  the  $k$ -th iteration of the Jacobian vector  $D(\nabla J(v^k), \nabla J(v^k))$ . The line-search is performed  
222 with quasi-Newton techniques that uses at each iteration  $k$  a Hessian matrix  $H_k$  and Jacobian  
223 vector  $D_k$  defined respectively by:

$$(26) \quad D_k := \left( \langle \nabla J(v^k), \pi_1(\nabla J(v^k)) \rangle_c, \dots, \langle \nabla J(v^k), \pi_{\hat{n}}(\nabla J(v^k)) \rangle_c \right)^T,$$

$$(27) \quad (H_k)_{n,m} := \langle \pi_n(\nabla J(v^k)), \pi_m(\nabla J(v^k)) \rangle_{\nabla^2 J}.$$

224 The spectral condition number of the Hessian matrix  $\nabla^2 J$  is denoted as:  $\underline{\kappa} = \underline{\kappa}(\nabla^2 J) :=$   
225  $\lambda_{max} \lambda_{min}^{-1}$ , with  $\lambda_{max} := \lambda_{max}(\nabla^2 J)$  the largest eigenvalue of  $\nabla^2 J$  and  $\lambda_{min} := \lambda_{min}(\nabla^2 J)$   
226 its smallest eigenvalue.

227 According to Eq.(17) we have

$$(28) \quad \Theta_n^k = -H_k^{-1} D_k.$$

228 From Eq.(16) we have:

$$(29) \quad J(v^{k+1}) = J(v^k) + (\Theta_{\hat{n}}^k)^T D_k + \frac{1}{2} (\Theta_{\hat{n}}^k)^T H_k \Theta_{\hat{n}}^k.$$

229 Our parallel algorithm to minimize the cost functional Eq.(15), is stated as follows (see Algo-  
rithm 3).

---

**Algorithm 3: Enhanced steepest descent algorithm for the optimal control problem.**

---

0 **Input:**  $v^0, \epsilon$   
1 **repeat**  
2   **forall the**  $1 \geq n \geq \hat{n}$  **do**  
3     | Solve  $\mathcal{Y}_n(T)(v_n^k)$  of Eq.(18)(respectively Eq.(22)) **in parallel** for all  $1 \leq n \leq \hat{n}$ ;  
4   **end**  
5   Compute  $\mathcal{P}(t)$  with the backward problem according to Eq.(20) (respectively Eq.(24)) ;  
6   **forall the**  $1 \geq n \geq \hat{n}$  **do**  
7     | Compute  $(D_k)_n$  of Eq.(26) **in parallel** for all  $1 \leq n \leq \hat{n}$ ;  
8   **end**  
9   Gather  $(D_k)_n$  from processor  $n$ ,  $2 \leq n \leq \hat{n}$  to **master processor**;  
10   Assemble the Hessian matrix  $H_k$  according to Eq.(27) with **master processor**;  
11   Compute the inversion of  $H_k$  and calculate  $\Theta_{\hat{n}}^k$  using Eq.(28);  
12   Broadcast  $\theta_n^k$  from **master processor** to all slaves processors;  
13   Update time-window-control variable  $v_n^{k+1}$  **in parallel** as :  

$$v_n^{k+1} = v_n^k + \theta_n^k \pi_n(\nabla J(v^k)) \quad \text{for all } 1 \leq n \leq \hat{n},$$
  

$$r^k = \|\nabla J(v^k)\|_2;$$
  
14    $k = k + 1$ ;  
15 **until**  $r^k \leq \epsilon$ ;

---

231 Since  $(v_n)_n$  has disjoint time-support, thanks to the linearity, the notation  $e_n(\nabla J(v^k))$  is  
 232 nothing but  $\nabla J(v_n^k)$ , where  $v^k$  is the concatenation of  $v_1^k, \dots, v_{\hat{n}}^k$ . In Algorithm 3 steps 9, 10, 11,  
 233 12 and 13 are trivial tasks in regards to computational effort. This is because the aforementioned  
 234 steps involve only communications between processors in term of vector-message passing (of size  
 235  $\hat{n} \ll \text{size of the finite element solution}$ ). Also the inversion of the matrix  $H_k$  is trivial task on  
 236 nowadays computer since it is low ranked matrix where direct inverse technique could be applied.

237

#### 4. CONVERGENCE ANALYSIS OF ALGORITHM 3

238 This section provides the proof of convergence of Algorithm 3. In the quadratic optimization  
 239 framework many convergence analysis are provided in the literature. We shall follow a standard  
 240 technique given in [15]. The key ingredient of the convergence theorem is the following results.  
 241 In the sequel, we suppose that  $\|\nabla J(v^k)\|_c$  does not vanish; otherwise the algorithm has already  
 242 converged.

243 **Proposition 4.1.** *The increase in value of the cost functional  $J$  between two successive controls*  
 244  *$v^k$  and  $v^{k+1}$  is bounded below by:*

$$(30) \quad J(v^k) - J(v^{k+1}) \geq \frac{1}{2\kappa(H_k)} \frac{\|\nabla J(v^k)\|_c^4}{\|\nabla J(v^k)\|_{\nabla^2 J}^2}.$$

245 *Proof.* Using Eq.(28) and Eq.(29), we can write:

$$(31) \quad J(v^k) - J(v^{k+1}) = \frac{1}{2} D_k^T H_k^{-1} D_k.$$

246 From the definition of the Jacobian vector  $D_k$  we have

$$\begin{aligned} \|D_k\|_2^2 &= \sum_{n=1}^{\hat{n}} \langle \nabla J(v^k), \pi_n(\nabla J(v^k)) \rangle_c^2, \\ &= \sum_{n=1}^{\hat{n}} \langle \pi_n(\nabla J(v^k)), \pi_n(\nabla J(v^k)) \rangle_c^2, \\ &= \sum_{n=1}^{\hat{n}} \|e_n(\nabla J(v^k))\|_c^4, \\ &= \|\nabla J(v^k)\|_c^4. \end{aligned}$$

247 Furthermore since  $H_k$  is an SPD matrix we have  $\lambda_{\min}(H_k^{-1}) = \frac{1}{\lambda_{\max}(H_k)}$ . In addition we have  
 248  $\frac{1}{\lambda_{\min}(H_k)} \geq \frac{1}{\frac{1}{\hat{n}} \mathbf{1}_{\hat{n}}^T H_k \mathbf{1}_{\hat{n}}}$ , where we have used the fact that  $\frac{1}{\sqrt{\hat{n}}} \mathbf{1}_{\hat{n}} = \frac{1}{\sqrt{\hat{n}}} (1, \dots, 1)^T$  is unit vector in  
 249  $\mathbb{R}^{\hat{n}}$  and Rayleigh-Quotient property. Moreover, we have:

$$\begin{aligned} D_k^T H_k^{-1} D_k &= \frac{D_k^T H_k^{-1} D_k}{\|D_k\|_2^2} \|D_k\|_2^2 \geq \lambda_{\min}(H_k^{-1}) \|D_k\|_2^2 \\ &= \lambda_{\min}(H_k^{-1}) \lambda_{\min}(H_k) \frac{\|\nabla J(v^k)\|_c^4}{\lambda_{\min}(H_k)} \\ &\geq \frac{\lambda_{\min}(H_k)}{\lambda_{\max}(H_k)} \frac{\|\nabla J(v^k)\|_c^4}{\frac{1}{\hat{n}} \mathbf{1}_{\hat{n}}^T H_k \mathbf{1}_{\hat{n}}} \\ &= \frac{\hat{n}}{\kappa(H_k)} \|\nabla J(v^k)\|_{\nabla^2 J}^{-2} \|\nabla J(v^k)\|_c^4. \end{aligned}$$

250 Since the partition number  $\hat{n}$  is greater than or equal to 1, we conclude that :

$$(32) \quad D_k^T H_k^{-1} D_k \geq \frac{\|\nabla J(v^k)\|_{\nabla^2 J}^{-2} \|\nabla J(v^k)\|_c^4}{\kappa(H_k)}.$$

251 Hence, using Eq.(31) we get the stated result.  $\square$

**Theorem 4.2.** *Under the hypothesis of the well-posedness (in the sense of Hadamard) of the optimal control problem, the control sequence  $(v^k)_{k \geq 1}$  of Algorithm 3 converges, for any partition  $\hat{n}$  of sub intervals, to the optimal control  $v^k$  unique minimizer of the quadratic functional  $J$ . Furthermore we have:*

$$\|v^k - v^*\|_{\nabla^2 J}^2 \leq r^k \|v^0 - v^*\|_{\nabla^2 J}^2,$$

252 where the rate of convergence  $r := \left(1 - \frac{4\underline{\kappa}}{\kappa(H_k)(\underline{\kappa}+1)^2}\right)$  satisfies  $0 \leq r < 1$ , where  $\underline{\kappa}$  stands for the  
253 spectral condition number of the hessian  $\nabla^2 J$ .

*Proof.* We denote by  $v^*$  the optimal control that minimizes  $J$ . The equality

$$J(v) = J(v^*) + \frac{1}{2} \langle v - v^*, v - v^* \rangle_{\nabla^2 J} = J(v^*) + \frac{1}{2} \|v - v^*\|_{\nabla^2 J}^2,$$

254 holds for any control  $v$ ; in particular we have:

$$\begin{aligned} J(v^{k+1}) &= J(v^*) + \frac{1}{2} \|v^{k+1} - v^*\|_{\nabla^2 J}^2, \\ J(v^k) &= J(v^*) + \frac{1}{2} \|v^k - v^*\|_{\nabla^2 J}^2. \end{aligned}$$

255 Consequently, by subtracting the equations above, we obtain

$$(33) \quad J(v^{k+1}) - J(v^k) = \frac{1}{2} \|v^{k+1} - v^*\|_{\nabla^2 J}^2 - \frac{1}{2} \|v^k - v^*\|_{\nabla^2 J}^2.$$

256 Since  $J$  is quadratic, we have  $\nabla^2 J(v^k - v^*) = \nabla^2 J(v^k)$ , that is  $v^k - v^* = (\nabla^2 J)^{-1} \nabla J(v^k)$ .  
257 Therefore we deduce:

$$(34) \quad \begin{aligned} \|v^k - v^*\|_{\nabla^2 J}^2 &= \langle v^k - v^*, v^k - v^* \rangle_{\nabla^2 J} \\ &= \langle v^k - v^*, \nabla^2 J, v^k - v^* \rangle_c \\ &= \langle (\nabla^2 J)^{-1} \nabla J(v^k), \nabla^2 J, (\nabla^2 J)^{-1} \nabla J(v^k) \rangle_c \\ &= \langle \nabla J(v^k), (\nabla^2 J)^{-1}, \nabla J(v^k) \rangle_c \\ &= \|\nabla J(v^k)\|_{(\nabla^2 J)^{-1}}^2. \end{aligned}$$

Because of Eq.(31), we also have

$$J(v^{k+1}) - J(v^k) = -\frac{1}{2} D_k^T H_k^{-1} D_k.$$

Using Eq.(33) and the above, we find that:

$$\|v^{k+1} - v^*\|_{\nabla^2 J}^2 = \|v^k - v^*\|_{\nabla^2 J}^2 - D_k^T H_k^{-1} D_k.$$

258 Moreover, according to Eqs (32)-(34), we obtain the following upper bound:

$$(35) \quad \begin{aligned} \|v^{k+1} - v^*\|_{\nabla^2 J}^2 &\leq \|v^k - v^*\|_{\nabla^2 J}^2 - \frac{1}{\kappa(H_k)} \frac{\|\nabla J(v^k)\|_c^4}{\|\nabla J(v^k)\|_{\nabla^2 J}^2} \\ &\leq \|v^k - v^*\|_{\nabla^2 J}^2 \left(1 - \frac{1}{\kappa(H_k)} \frac{\|\nabla J(v^k)\|_c^4}{\|\nabla J(v^k)\|_{\nabla^2 J}^2 \|\nabla J(v^k)\|_{(\nabla^2 J)^{-1}}^2}\right). \end{aligned}$$

Using the Kantorovich inequality [31, 4] (see also The Appendix) :

$$(36) \quad \frac{\|\nabla J(v^k)\|_c^4}{\|\nabla J(v^k)\|_{\nabla^2 J}^2 \|\nabla J(v^k)\|_{(\nabla^2 J)^{-1}}^2} \geq \frac{4\lambda_{max}\lambda_{min}}{(\lambda_{max} + \lambda_{min})^2}.$$

Then

$$1 - \frac{1}{\kappa(H_k)} \frac{\|\nabla J(v^k)\|_c^4}{\|\nabla J(v^k)\|_{\nabla^2 J}^2 \|\nabla J(v^k)\|_{(\nabla^2 J)^{-1}}^2} \leq 1 - \frac{4\underline{\kappa}}{\kappa(H_k)(\underline{\kappa} + 1)^2}.$$

Finally we obtain the desired results for any partition to  $\hat{n}$  subdivision, namely

$$\|v^k - v^*\|_{\nabla^2 J}^2 \leq \left(1 - \frac{4\underline{\kappa}}{\kappa(H_k)(\underline{\kappa} + 1)^2}\right)^k \|v^0 - v^*\|_{\nabla^2 J}^2.$$

The proof is therefore complete.  $\square$

**Remark 4.1.** Remark that for  $\hat{n} = 1$ , we immediately get the condition number  $\kappa(H_k) = 1$  and we recognize the serial steepest gradient method, which has convergence rate  $\left(\frac{\underline{\kappa}-1}{\underline{\kappa}+1}\right)^2$ .

It is difficult to pre-estimate the spectral condition number  $\kappa(H_k)(\hat{n})$  (is a function of  $\hat{n}$ ) that play an important role and contribute to the evaluation of the rate of convergence as our theoretical rate of convergence stated. We present in what follows numerical results that demonstrate the efficiency of our algorithm. Tests consider examples of well-posed and ill-posed control problem.

## 5. NUMERICAL EXPERIMENTS

We shall present the numerical validation of our method in two stages. In the first stage, we consider a linear algebra framework where we construct a random matrix-based quadratic cost function that we minimize using Algorithm 2. In the second stage, we consider the two optimal control problems presented in sections 3.1 and in 3.2 for the distributed- and Dirichlet boundary-control respectively. In both cases we minimize a quadratic cost function properly defined for each handled control problem.

**5.1. Linear algebra program.** This subsection treat basically the implementation of Algorithm 2. The program was implemented using the scientific programming language Scilab [51]. We consider the minimization of a quadratic form  $q$  where the matrix  $A$  is an SPD  $m$ -by- $m$  matrix and a real vector  $b \in \mathbb{R}^m \cap \text{rank}(A)$  are generated by hand (see below for their constructions). We aim at solving iteratively the linear system  $Ax = b$ , by minimizing

$$(37) \quad q(x) = \frac{1}{2}x^T Ax - x^T b.$$

Let us denote by  $\hat{n}$  the partition number of the unknown  $x \in \mathbb{R}^m$ . The partition is supposed to be uniform and we assume that  $\hat{n}$  divides  $m$  with a null rest.

We give in Table 1 a SCILAB function that builds the vector step-length  $\Theta_n^k$  as stated in Eq. (4). In the practice we randomly generate an SPD sparse matrix  $A = (\alpha + \gamma m)I_{\mathbb{R}^m} + R$ , where  $0 < \alpha < 1$ ,  $\gamma > 1$ ,  $I_{\mathbb{R}^m}$  is the  $m$ -by- $m$  identity matrix and  $R$  is a symmetric  $m$ -by- $m$  random matrix. This way the matrix  $A$  is symmetric and diagonally dominant, hence SPD. It is worthy noticing that the role of  $\alpha$  is regularizing when rapidly vanishing eigenvalues of  $A$  are generated randomly. This technique helps us to manipulate the coercivity of the handled problem hence its spectral condition number.

For such matrix  $A$  we proceed to minimize the quadratic form defined in Eq.(37) with several  $\hat{n}$ -subdivisions.

The improvement quality of the algorithm against the serial case  $\hat{n} = 1$  in term of iteration number is presented in Figure. 1. In fact, the left hand side of Figure. 1 presents the cost function

```

1 function [P]=Build_Hk(n,A,b,xk,dJk)
2 m=size(A,1);l=m/n;ii=modulo(m,n);
3 if ii~=0 then
4     printf("Please chose an other n!");
5     abort;
6 end
7 dJkn=zeros(m,n); Dk=[];
8 for i=1:n
9     dJkn((i-1)*l+1:i*l,i)=dJk((i-1)*l+1:i*l);
10    Dk(i)=dJkn(:,i)'*(A*xk-b);
11 end
12 Hk=[,];
13 for i=1:n
14     for j=i:n
15         Hktmp=A*dJkn(:,j);
16         Hk(i,j)=dJkn(:,i)'*Hktmp;
17         Hk(j,i)=Hk(i,j);
18     end
19 end
20 theta=-Hk\Dk;
21 P=eye(m,m);
22 for i=1:n
23     P((i-1)*l+1:i*l,(i-1)*l+1:i*l)=theta(i).*eye(l,l);
24 end
25 endfunction

```

TABLE 1. Scilab function to build the vector step length, for the linear algebra program.

293 minimization versus the iteration number of the algorithm where several choices of partition on  
 294  $\hat{n}$  are carried out. In the right hand side of the Figure. 1 we give the logarithmic representation  
 295 of the relative error  $\frac{\|x^k - x^*\|_2}{\|x^*\|_2}$ , where  $x^*$  is the exact solution of the linear system at hand.

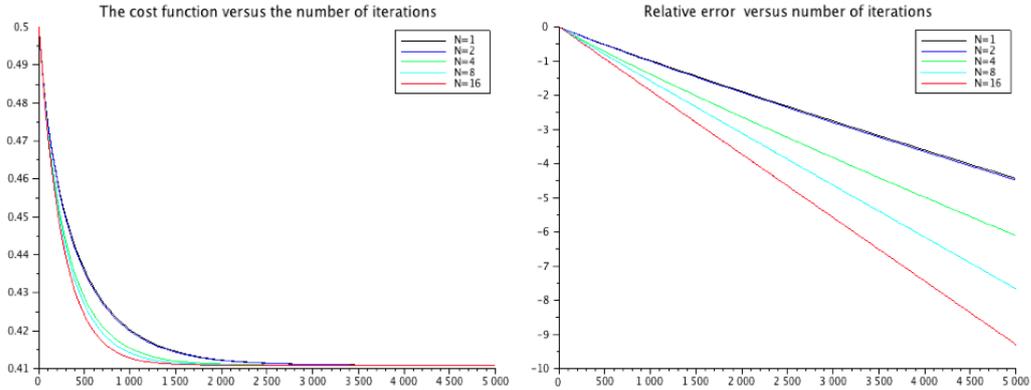


FIGURE 1. Performance in term of iteration number: Several decomposition on  $\hat{n}$ . Results from the linear algebra Scilab program.

296 **5.2. Heat optimal control program.** We discuss in this subsection the implementation results  
 297 of Algorithm 3 for the optimization problems presented in section 3. Our tests deal with the 2D-  
 298 heat equation on the bounded domain  $\Omega = [0, 1] \times [0, 1]$ . We consider, three types of test problems

299 in both cases of distributed and Dirichlet controls. Tests vary according to the theoretical  
 300 difficulty of the control problem [11, 7, 17]. Indeed, we vary the regularization parameter  $\alpha$  and  
 301 also change the initial and target solutions in order to handle more severe control problems as  
 302 has been tested for instance in [11].

303 Numerical tests concern the minimization of the quadratic cost functionals  $J(v)$  and  $J_\Gamma(v_\Gamma)$   
 304 using Algorithm 3. It is well known that in the case  $\alpha$  vanishes the control problem becomes  
 305 an "approximated" controllability problem. Therefore the control variable tries to produce a  
 306 solution that reaches as close as it "can" the target solution. With this strategy, we accentuate  
 307 the ill-conditioned degree of the handled problem. We also consider an improper-posed problems  
 308 for the controllability approximation, where the target solution doesn't belong to the space of  
 309 the reachable solutions. No solution exists thus for the optimization problem i.e. no control  
 310 exists that enables reaching the given target !

311 For reading conveniences and in order to emphasize the role of the parameter  $\alpha$  on numerical  
 312 tests, we tag problems that we shall consider as  $\mathcal{P}_i^\alpha$  where the index  $i$  refers to the problem  
 313 among  $\{1, 2, 3, 4\}$ . The table below resumes all numerical test that we shall experiences

	Minimize $J(v)$ distributed control	Minimize $J_\Gamma(v_\Gamma)$ boundary control
Moderate $\alpha = 1 \times 10^{-02}$	well-posed problem corresponding data in $(\mathcal{P}_1^\alpha), (\mathcal{P}_2^\alpha)$	ill-posed problem corresponding data in $(\mathcal{P}_3^\alpha)$
314 Vanishing $\alpha = 1 \times 10^{-08}$	ill-posed problem corresponding data in $(\mathcal{P}_1^\alpha), (\mathcal{P}_2^\alpha)$	sever ill-posed problem corresponding data in $(\mathcal{P}_3^\alpha)$
Solution does not exist	sever ill-posed problem corresponding data in $(\mathcal{P}_4^\alpha)$	sever ill-posed problem corresponding data in $(\mathcal{P}_4^\alpha)$

315 We suppose from now on that the computational domain  $\Omega$  is a polygonal domain of the  
 316 plane  $\mathbb{R}^2$ . We then introduce a triangulation  $\mathcal{T}_h$  of  $\Omega$ ; the subscript  $h$  stands for the largest  
 317 length of the edges of the triangles that constitute  $\mathcal{T}_h$ . The solution of the heat equation at a  
 318 given time  $t$  belongs to  $H^1(\Omega)$ . The source terms and other variables are elements of  $L^2(\Omega)$ .  
 319 Those infinite dimensional spaces are therefore approximated with the finite-dimensional space  
 320  $V_h$ , characterized by  $\mathbb{P}_1$  the space of the polynomials of degree  $\leq 1$  in two variables  $(x_1, x_2)$ .  
 321 We have  $V_h := \{u_h | u_h \in C^0(\bar{\Omega}), u_{h|_K} \in \mathbb{P}_1, \text{ for all } K \in \mathcal{T}_h\}$ . In addition, Dirichlet boundary  
 322 conditions (where the solution is in  $H_0^1(\Omega)$  i.e. vanishing on boundary  $\partial\Omega$ ) are taken into account  
 323 via penalization of the vertices on the boundaries. The time dependence of the solution is  
 324 approximated via the implicit Euler scheme. The inversion operations of matrices is performed  
 325 by the UMFPACK solver. We use the trapezoidal method in order to approximate integrals defined  
 326 on the time interval.

327 The numerical experiments were run using a parallel machine with 24 CPU's AMD with 800  
 328 MHz in a Linux environment. We code two FreeFem++ [45] scripts for the distributed and  
 329 Dirichlet control. We use MPI library in order to achieve parallelism.

330 Tests that concern the distributed control problem are produced with control that acts on  
 331  $\Omega_c \subset \Omega$ , with  $\Omega_c = [0, \frac{1}{3}] \times [0, \frac{1}{3}]$ , whereas Dirichlet boundary control problem, the control acts  
 332 on  $\Gamma \subset \partial\Omega$ , with  $\Gamma = \{(x_1, x_2) \in \partial\Omega, |x_2 = 0\}$ . The time horizon of the problem is fixed to  
 333  $T = 6.4$  and the small time step is  $\tau = 0.01$ , the finite element triangulation has a size  $h = 1./20$   
 334 as elements edge length. In order to have a better control of the time evolution we put the  
 335 diffusion coefficient  $\sigma = 0.01$ .

336 **Remark 5.1.** *Regarding these parameters of discretization, we have the number of degree of*  
 337 *freedom for the spacial variable is  $400 = 20^2$  and the temporal degree of freedom is  $T/\tau = 640$ .*

338 5.2.1. *First test problem: Moderate Tikhonov regularization parameter  $\alpha$ .* We consider an opti-  
 339 mal control problem on the heat equation. The control is considered first to be distributed and  
 340 then Dirichlet. For the distributed optimal control problem we first use the functions

$$(\mathcal{P}_1^\alpha) \quad \begin{aligned} y_0(x_1, x_2) &= \exp(-\gamma 2\pi((x_1 - .7)^2 + (x_2 - .7)^2)) \\ y^{target}(x_1, x_2) &= \exp(-\gamma 2\pi((x_1 - .3)^2 + (x_2 - .3)^2)), \end{aligned}$$

341 as initial condition and target solution respectively. The real valued  $\gamma$  is introduced to force the  
 342 Gaussian to have support strictly included in the domain and verify the boundary conditions.  
 The aim is to minimize the cost functional defined in Eq. (6). The decay of the cost function

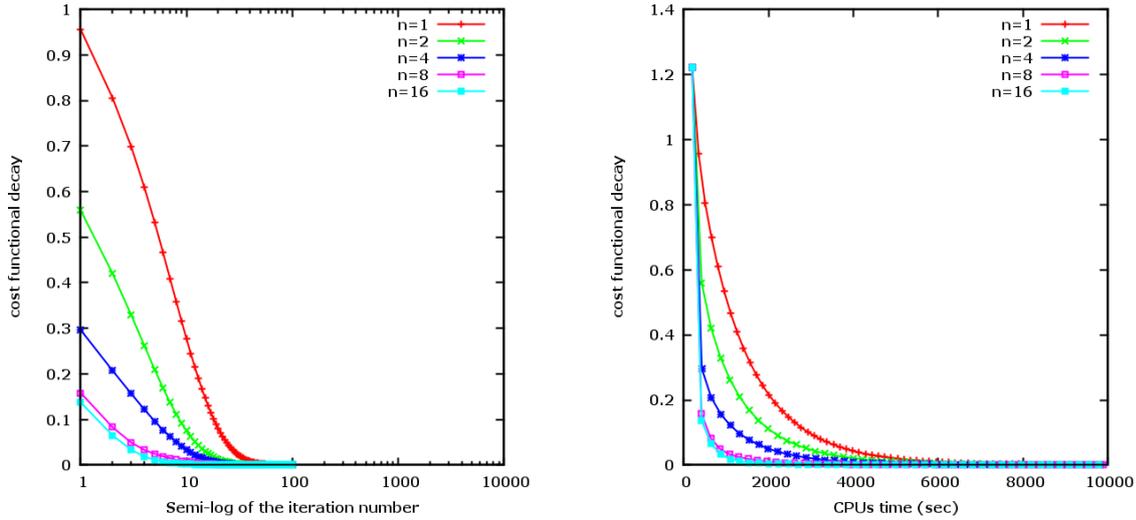


FIGURE 2. First test problem, for  $\mathcal{P}_1^\alpha$ : Normalized and shifted cost functional values versus iteration number (left) and versus computational time (right) for several values of  $\hat{n}$  (i.e. the number of processors used).

343 with respect to the iterations of our algorithm is presented in Figure. 2 on the left side, and  
 344 the same results are given with respect to the computational CPU's time (in sec) on the right  
 345 side. We show that the algorithm accelerates with respect to the partition number  $\hat{n}$  and also  
 346 preserves the accuracy of the resolution. Indeed, all tests independently of  $\hat{n}$  always converge  
 347 to the unique solution. This is in agreement with Theorem 4.2, which proves the convergence  
 348 of the algorithm to the optimal control (in the well-posedness framework [37]) for an arbitrary  
 349 partition choice  $\hat{n}$ .  
 350

351 We test a second problem with an a priori known solution of the heat equation. The considered  
 352 problem has

$$(\mathcal{P}_2^\alpha) \quad \begin{aligned} y_0(x_1, x_2) &= \sin(\pi x_1) \sin(\pi x_2) \\ y^{target}(x_1, x_2) &= \exp(-2\pi^2 \sigma T) \sin(\pi x_1) \sin(\pi x_2), \end{aligned}$$

353 as initial condition and target solution respectively. Remark that the target solution is taken as  
 354 a solution of the heat equation at time  $T$ . The results of this test are presented in Figure. 4,  
 355 which shows the decay in values of the cost functional versus the iterations of the algorithm on  
 356 the left side and versus the computational CPU's time (in sec) on the right side.

357 We give in Figure. 3 and Figure. 5 several rows value snapshots (varying the  $\hat{n}$ ) of the control  
 358 and its corresponding controlled final solution  $y(T)$ . Notice the stability and the accuracy of the

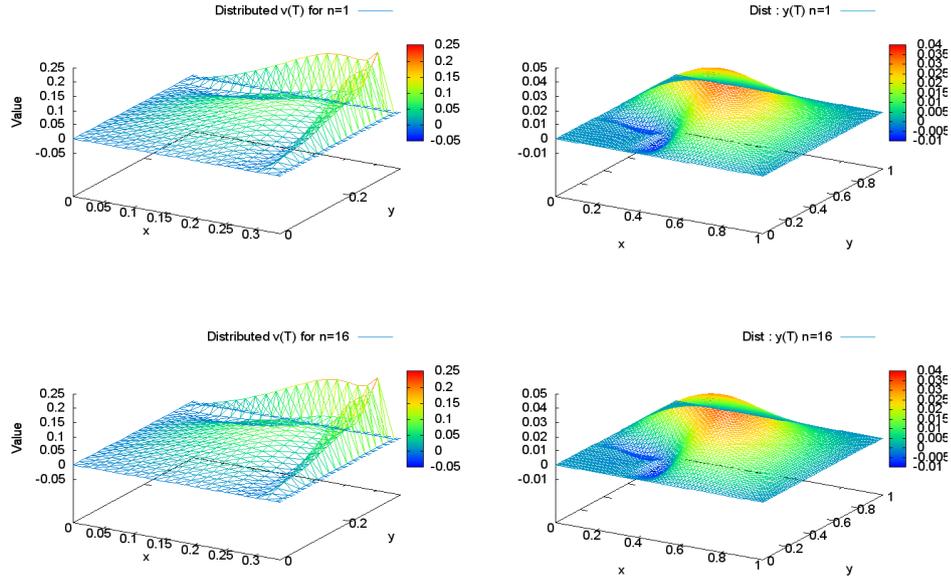


FIGURE 3. Snapshots in  $\hat{n} = 1, 16$  of the distributed optimal control on the left columns and its corresponding controlled final state at time T:  $y(T)$  on the right columns. The test case corresponds to the control problem  $\mathcal{P}_1^\alpha$ , where  $\alpha$  is taken as  $\alpha = 1 \times 10^{-02}$ . Same result apply for different choice of  $\hat{n}$ .

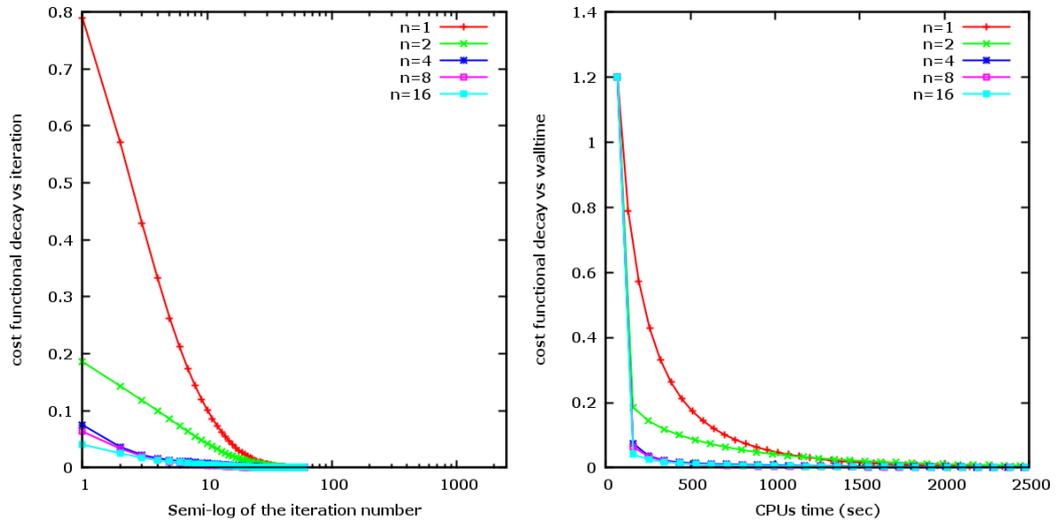


FIGURE 4. First test problem, for  $\mathcal{P}_2^\alpha$ : Normalized cost functional values versus computational CPU time for several values of  $\hat{n}$  (i.e. the number of processors used).

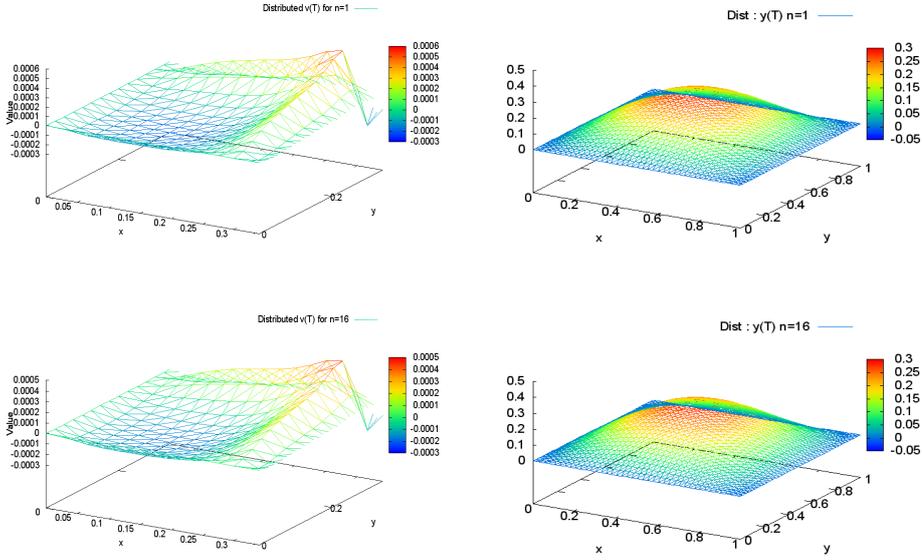


FIGURE 5. Snapshots in  $\hat{n} = 1, 16$  of the distributed optimal control on the left columns and its corresponding controlled final state at time T:  $y(T)$  on the right columns. The test case corresponds to the control problem  $\mathcal{P}_2^\alpha$ , where  $\alpha = 1 \times 10^{-02}$ . Same results apply for different choice of  $\hat{n}$ .

359 method with any choice of  $\hat{n}$ . In particular the shape of the resulting optimal control is unique  
 360 as well as the controlled solution  $y(T)$  doesn't depend on  $\hat{n}$ .

361 For the Dirichlet boundary control problem we choose the following functions as source term,  
 362 initial condition and target solution:

$$\begin{aligned}
 f(x_1, x_2, t) &= 3\pi^3 \sigma \exp(2\pi^2 \sigma t) (\sin(\pi x_1) + \sin(\pi x_2)) \\
 (\mathcal{P}_3^\alpha) \quad y_0(x_1, x_2) &= \pi (\sin(\pi x_1) + \sin(\pi x_2)) \\
 y^{target}(x_1, x_2) &= \pi \exp(2\pi^2 \sigma) (\sin(\pi x_1) + \sin(\pi x_2)),
 \end{aligned}$$

363 respectively. Because of the ill-posed character of this problem, its optimization leads to results  
 364 with high contrast in scale. We therefore preferred to summarize the optimizations results in  
 365 Table 3 instead of Figures.

366 **Remark 5.2.** *Because of the linearity and the superposition property of the heat equation, it can*  
 367 *be shown that problems  $(\mathcal{P}_2^\alpha$  and  $\mathcal{P}_3^\alpha$ ) mentioned above are equivalent to a control problem which*  
 368 *has null target solution.*

369 5.2.2. *Second test problem: vanishing Tikhonov regularization parameter  $\alpha$ .* In this section, we  
 370 are concerned with the "approximate" controllability of the heat equation, where the regulariza-  
 371 tion parameter  $\alpha$  vanishes, practically we take  $\alpha = 1 \times 10^{-08}$ . In this case, problems  $\mathcal{P}_2^\alpha$  and  $\mathcal{P}_3^\alpha$ ,  
 372 in the continuous setting are supposed to be well posed (see for instances [16, 44]). However, may  
 373 not be the case in the discretized settings; we refer for instance to [17] (and reference therein)  
 374 for more details.

375 Table 2 contains the summarized results for the convergence of the distributed control problem.  
 376 On the one hand, we are interested in the error given by our algorithm for several choices of

Test problem	Results					
$\mathcal{P}_1^\alpha$	$\alpha = 1 \times 10^{-02}$					
	Quantity	$\hat{n} = 1$	$\hat{n} = 2$	$\hat{n} = 4$	$\hat{n} = 8$	$\hat{n} = 16$
	Number of iterations $k$	100	68	63	49	27
	walltime in sec	15311.6	15352.3	14308.7	10998.2	6354.56
	$\ \mathcal{Y}^k(T) - y^{target}\ _2 / \ y^{target}\ _2$	0.472113	0.472117	0.472111	0.472104	0.472102
$\int_{(0,T)} \ v^k\ _c^2 dt$	0.0151685	0.0151509	0.0151727	0.0152016	0.015214	
$\mathcal{P}_2^\alpha$	$\alpha = 1 \times 10^{-02}$					
	Quantity	$\hat{n} = 1$	$\hat{n} = 2$	$\hat{n} = 4$	$\hat{n} = 8$	$\hat{n} = 16$
	Number of iterations $k$	60	50	45	40	35
	walltime in sec	3855.21	3726.28	4220.92	3778.13	3222.78
	$\ \mathcal{Y}^k(T) - y^{target}\ _2 / \ y^{target}\ _2$	$8.26 \times 10^{-08}$	$8.26 \times 10^{-08}$	$8.15 \times 10^{-08}$	$8.15 \times 10^{-08}$	$8.14 \times 10^{-08}$
$\int_{(0,T)} \ v^k\ _c^2 dt$	$1.68 \times 10^{-07}$	$1.68 \times 10^{-07}$	$1.72 \times 10^{-07}$	$1.72 \times 10^{-07}$	$1.72 \times 10^{-07}$	
$\mathcal{P}_2^\alpha$	$\alpha = 1 \times 10^{-08}$					
	Quantity	$\hat{n} = 1$	$\hat{n} = 2$	$\hat{n} = 4$	$\hat{n} = 8$	$\hat{n} = 16$
	Number of iterations $k$	60	50	40	30	20
	walltime in sec	3846.23	4654.34	3759.98	2835.31	1948.4
	$\ \mathcal{Y}^k(T) - y^{target}\ _2 / \ y^{target}\ _2$	$3.93 \times 10^{-08}$	$1.14 \times 10^{-08}$	$5.87 \times 10^{-09}$	$2.04 \times 10^{-09}$	$1.76 \times 10^{-09}$
$\int_{(0,T)} \ v^k\ _c^2 dt$	$5.42 \times 10^{-07}$	$4.13 \times 10^{-06}$	$2.97 \times 10^{-04}$	$3.64 \times 10^{-03}$	$2.51 \times 10^{-03}$	

TABLE 2. Results' summary of Algorithm 3 applied on the distributed control problems  $\mathcal{P}_1^\alpha$  and  $\mathcal{P}_2^\alpha$ .

Test problem	Results					
$\mathcal{P}_3^\alpha$	$\alpha = 1 \times 10^{-02}$					
	Quantity	$\hat{n} = 1$	$\hat{n} = 2$	$\hat{n} = 4$	$\hat{n} = 8$	$\hat{n} = 16$
	Number of iterations	40	40	30	18	10
	walltime in sec	12453.9	12416.1	9184.28	5570.54	3158.97
	$\ \mathcal{Y}_\Gamma(T) - y^{target}\ _2 / \ y^{target}\ _2$	$8.54 \times 10^{+06}$	0.472488	0.0538509	0.0533826	0.0534024
$\int_{(0,T)} \ v\ _\Gamma^2 dt$	$2.79 \times 10^{+08}$	$1.96 \times 10^{+07}$	31.4193	138.675	275.08	
$\mathcal{P}_3^\alpha$	$\alpha = 1 \times 10^{-08}$					
	Quantity	$\hat{n} = 1$	$\hat{n} = 2$	$\hat{n} = 4$	$\hat{n} = 8$	$\hat{n} = 16$
	Number of iterations	40	40	30	27	10
	walltime in sec	1248.85	1248.97	916.232	825.791	325.16
	$\ \mathcal{Y}_\Gamma(T) - y^{target}\ _2 / \ y^{target}\ _2$	$8.85 \times 10^{+06}$	0.151086	0.0292072	0.0278316	0.0267375
$\int_{(0,T)} \ v\ _\Gamma^2 dt$	$7.92 \times 10^{+08}$	$2.30 \times 10^{+07}$	$1.27 \times 10^{+07}$	$1.47 \times 10^{+07}$	$1.58 \times 10^{+06}$	

TABLE 3. Results' summary of Algorithm 3 applied on the Dirichlet boundary control problem  $\mathcal{P}_3^\alpha$ .

377 partition number  $\hat{n}$ . On the other hand, we give the  $L^2(0, T; L^2(\Omega_c))$  of the control. We notice the  
378 improvement in the quality of the algorithm in terms of both time of execution and control energy  
379 consumption, namely the quantity  $\int_{(0,T)} \|v^k\|_c^2 dt$ . In fact, for the optimal control framework ( $\mathcal{P}_1^\alpha$   
380 and  $\mathcal{P}_2^\alpha$  with  $\alpha = 1 \times 10^{-02}$ ), we see that, for a fixed stopping criterion, the algorithm is faster  
381 and consume the same energy independently of  $\hat{n}$ . In the approximate controllability framework  
382 ( $\mathcal{P}_2^\alpha$  with  $\alpha = 1 \times 10^{-08}$  vanishes), we note first that the general accuracy of the controlled  
383 solution (see the error  $\|\mathcal{Y}^k(T) - y^{target}\|_2 / \|y^{target}\|_2$ ) is improved as  $\alpha = 1 \times 10^{-08}$  compered  
384 with  $\alpha = 1 \times 10^{-02}$ . Second, we note that the error diminishes when increasing  $\hat{n}$ , the energy  
385 consumption rises however. The scalability in CPU's time and number of iteration shows the  
386 enhancement of our method when it is applied (i.e. for  $\hat{n} > 1$ ).

387 Table 3 contains the summarized results at the convergence of the Dirichlet boundary control  
388 problem. This problem is known in the literature for its ill-posedness, where it may be singular  
389 in several cases see [7] and references therein. In fact, it is very sensitive to noise in the data.  
390 We show in Table 3 that for a big value of the regularization parameter  $\alpha$  our algorithm behaves  
391 already as the distributed optimal control for a vanishing  $\alpha$ , in the sense that it consumes  
392 more control energy to produce a more accurate solution with smaller execution CPU's time.  
393 It is worth noting that the serial case  $\hat{n} = 1$  fails to reach an acceptable solution, whereas the  
394 algorithm behaves well as  $\hat{n}$  rises.

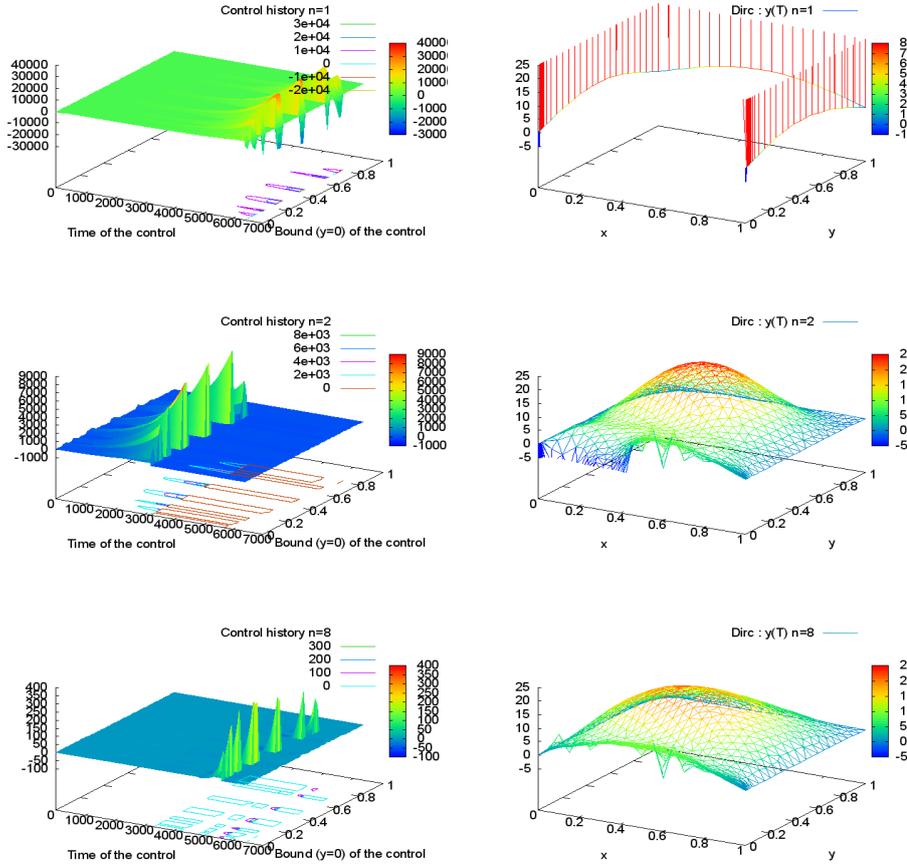


FIGURE 6. Several rows value snapshots in  $\hat{n}$  of the Dirichlet optimal control on the left columns and its corresponding controlled final state at time  $T$ :  $y(T)$  on the right columns. The test case corresponds to the control problem  $\mathcal{P}_3^\alpha$ , where  $\alpha = 1 \times 10^{-02}$ .

395 We give in Figure. 6 and Figure. 9 several rows value snapshots (varying  $\hat{n}$ ) of the Dirichlet  
 396 control on  $\Gamma$ . We present in the first column its evolution during  $[0, T]$  and on the second column  
 397 its corresponding controlled final solution  $y(T)$  at time  $T$ ; we scaled the plot of the z-range of  
 398 the target solution in both Figs.6 and 9.

399 In each row one sees the control and its solution for a specific partition  $\hat{n}$ . The serial case  
 400  $\hat{n} = 1$  leads to a controlled solution which doesn't have the same rank as  $y^{target}$ , whereas as  $\hat{n}$   
 401 rises, we improve the behavior of the algorithm.

402 In a sequential simulation of the optimal control problem, it is worth noting that (based on  
 403 observations) the optimal control is relatively "lazy" during a long period of the simulation time  
 404 and meets its maximum in magnitude only around the final horizon time  $T$ . This observation  
 405 is very apparent in Figure. 6 and Figure. 9 (see the first row i.e. case  $\hat{n} = 1$ ). The nature  
 406 of our algorithm, which is based on time domain decomposition, obliges the control to act in  
 407 subintervals. Hence, the control acts more often and earlier in time (before  $T$ ) and leads to a  
 408 better controlled solution  $y(T)$ .

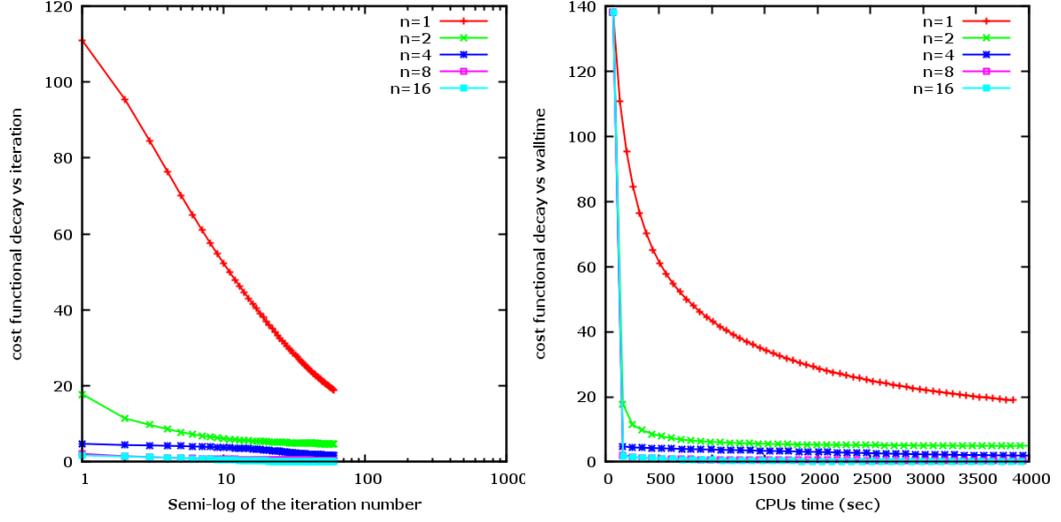


FIGURE 7. Normalized and shifted cost functional values versus computational CPU time for several values of  $\hat{n}$  (i.e. the number of processors used), Distributed control problem  $\mathcal{P}_2^\alpha$  with  $\alpha = 1 \times 10^{-08}$ .

409 5.2.3. *Third test problem: Sever ill-posed problem (no solution).* In this test case, we consider  
 410 a severely ill-posed problem where we are not in the frame of the hypothesis of Theorem 4.2.  
 411 In fact, the target solution is piecewise Lipschitz continuous, so that it is not regular enough  
 412 compared with the solution of the heat equation. This implies that in our control problem, both  
 413 the distributed and the Dirichlet boundary control has no solution. The initial condition and the  
 414 target solution are given by

$$(\mathcal{P}_4^\alpha) \quad \begin{aligned} y_0(x_1, x_2) &= \pi(\sin(\pi x_1) + \sin(\pi x_2)) \\ y^{target}(x_1, x_2) &= \min(x_1, x_2, (1 - x_1), (1 - x_2)), \end{aligned}$$

respectively. A plots of the initial condition and the target solutions are given in Figure. 10.

Test problem	Results					
Distributed control						
$\mathcal{P}_4^\alpha$	$\alpha = 1 \times 10^{-08}$					
	Quantity	$\hat{n} = 1$	$\hat{n} = 2$	$\hat{n} = 4$	$\hat{n} = 8$	$\hat{n} = 16$
	Number of iterations	100	68	60	50	40
	walltime in sec	6381.43	6303.67	5548.16	4676.83	3785.97
	$\ \mathcal{Y}(T) - y^{target}\ _2 / \ y^{target}\ _2$	$8.16 \times 10^{-03}$	$5.3 \times 10^{-03}$	$4.74 \times 10^{-03}$	$3.95 \times 10^{-03}$	$3.76 \times 10^{-03}$
	$\int_{(0,T)} \ v\ _c^2 dt$	0.34	3.01	52.87	52.77	2660.87
Dirichlet control						
$\mathcal{P}_4^\alpha$	$\alpha = 1 \times 10^{-08}$					
	Quantity	$\hat{n} = 1$	$\hat{n} = 2$	$\hat{n} = 4$	$\hat{n} = 8$	$\hat{n} = 16$
	Number of iterations	25	25	20	4	1
	walltime in sec	848.58	655.40	655.40	146.19	62.87
	$\ \mathcal{Y}_\Gamma(T) - y^{target}\ _2 / \ y^{target}\ _2$	$2.85 \times 10^{+10}$	3055	39.3	0.2	0.067
	$\int_{(0,T)} \ v\ _\Gamma^2 dt$	$6.73 \times 10^{+08}$	$2.17 \times 10^{+07}$	141.62	17.84	26758.5

TABLE 4. Results' summary of Algorithm 3 applied on to both distributed and Dirichlet boundary control for the third test problem  $\mathcal{P}_4^\alpha$ .

415

416 In Figures 11 and 12 we plot the controlled solution at time  $T$  for the distributed and Dirichlet  
 417 control problems respectively. We remark that for the distributed control problem the controlled  
 418 solution is smooth except in  $\Omega_c$ , where the control is able to fit with the target solution.

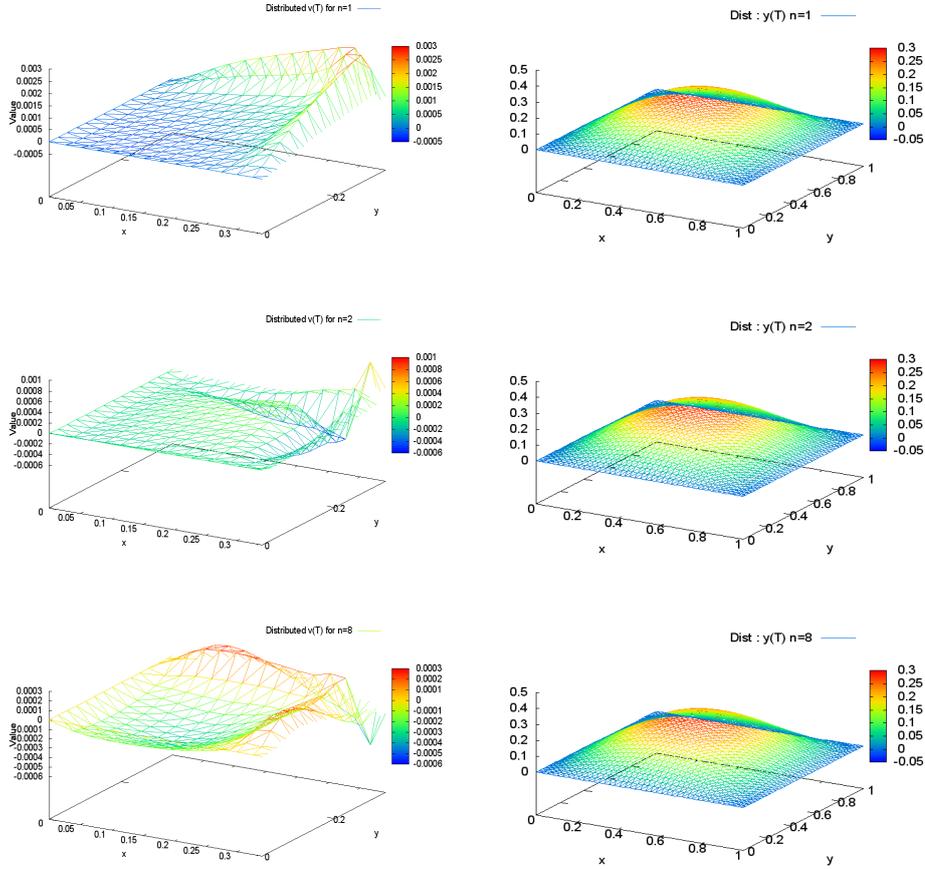


FIGURE 8. Several rows value snapshots in  $\hat{n}$  of the distributed optimal control on the left columns and its corresponding controlled final state at time T:  $(\mathcal{Y}(T))$  on the right columns. The test case corresponds to the control problem  $\mathcal{P}_2^\alpha$ , where  $\alpha = 1 \times 10^{-08}$ .

419 **Remark 5.3.** *Out of curiosity, we tested the case where the control is distributed on the whole*  
 420 *domain. We see that the control succeeds to fit the controlled solution to the target even if it*  
 421 *is not  $C^1(\Omega)$ . This is impressive and shows the impact on the results of the regions where the*  
 422 *control is distributed.*

423 We note the stability of the method of the distributed test case. However, the Dirichlet  
 424 problem test case presents hypersensitivity. In fact, in the case of  $\hat{n} = 1$  the algorithm succeeds  
 425 to produce a similar shape of the controlled solution, although still have a big error. We note  
 426 that the time domain decomposition leads to a control which gives a good shape of the controlled  
 427 solution that has a small error comparing to the target solution.

428 In this severely ill-posed problem, we see that some partitions may fail to produce a control  
 429 that fit the controlled solution to the target. There is an exemption for the case of  $\hat{n} = 8$   
 430 partitions, where we have a good reconstruction of the target. The summarized results are given  
 431 in Tables 4.

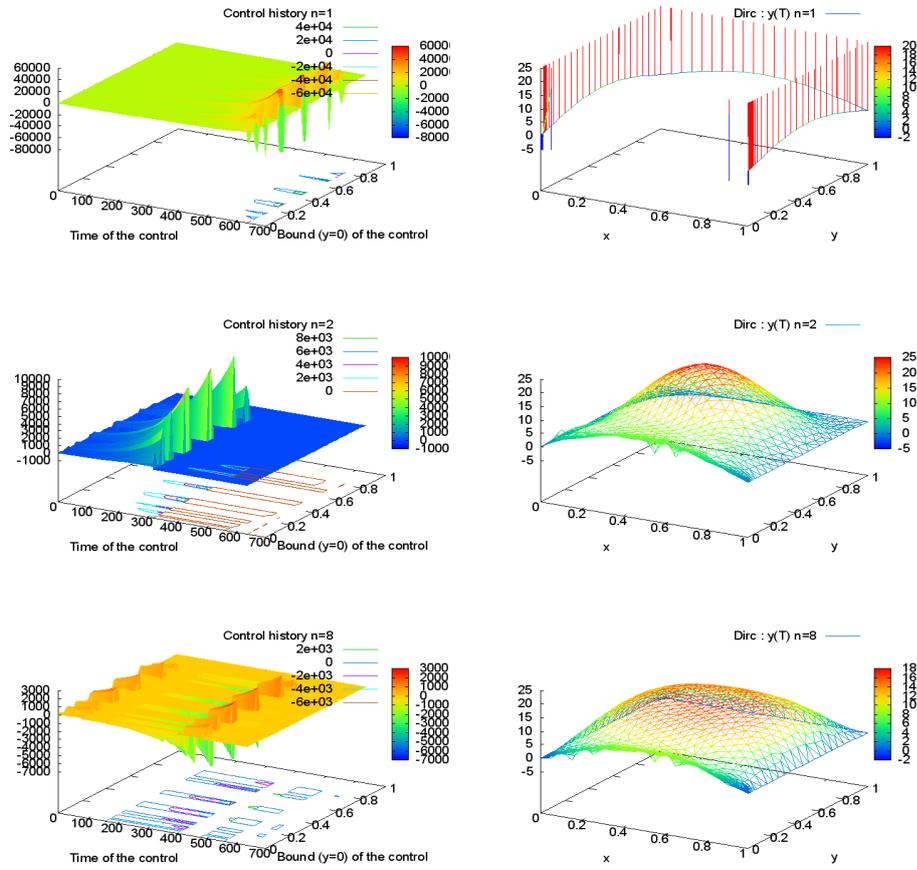


FIGURE 9. Several rows value snapshots in  $\hat{n}$  of the Dirichlet optimal control on the left columns and its corresponding controlled final state at time T:  $\mathcal{Y}_\Gamma(T)$  on the right columns. The test case corresponds to the control problem  $\mathcal{P}_3^\alpha$ , where  $\alpha = 1 \times 10^{-08}$ .

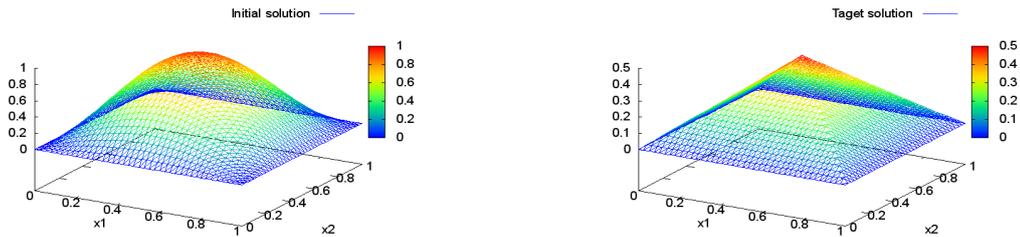


FIGURE 10. Graph of initial and target solution for both distributed and Dirichlet boundary control problem.

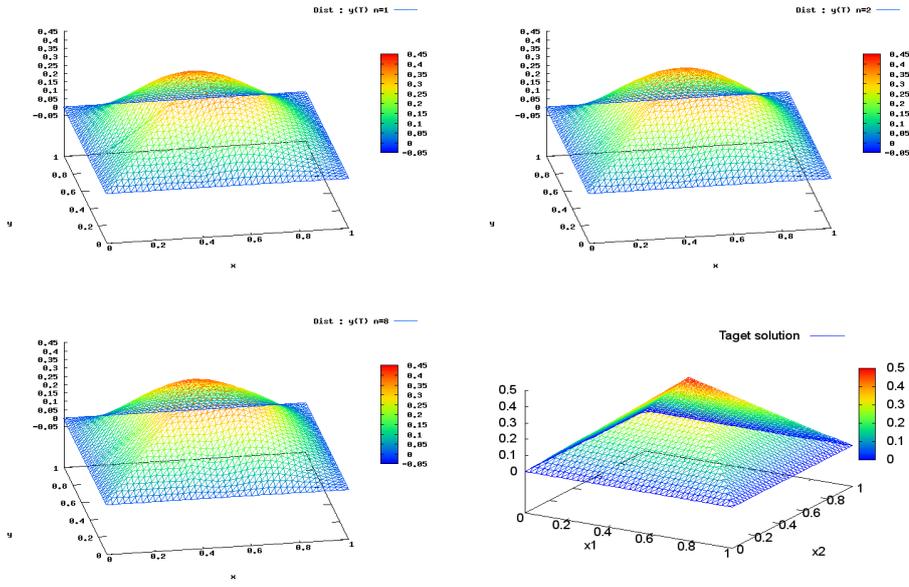


FIGURE 11. Several snapshots in  $\hat{n}$  of final state at time T:  $\mathcal{Y}(T)$ . The test case corresponds to Distributed control sever Ill-posed problem  $\mathcal{P}_4^\alpha$ .

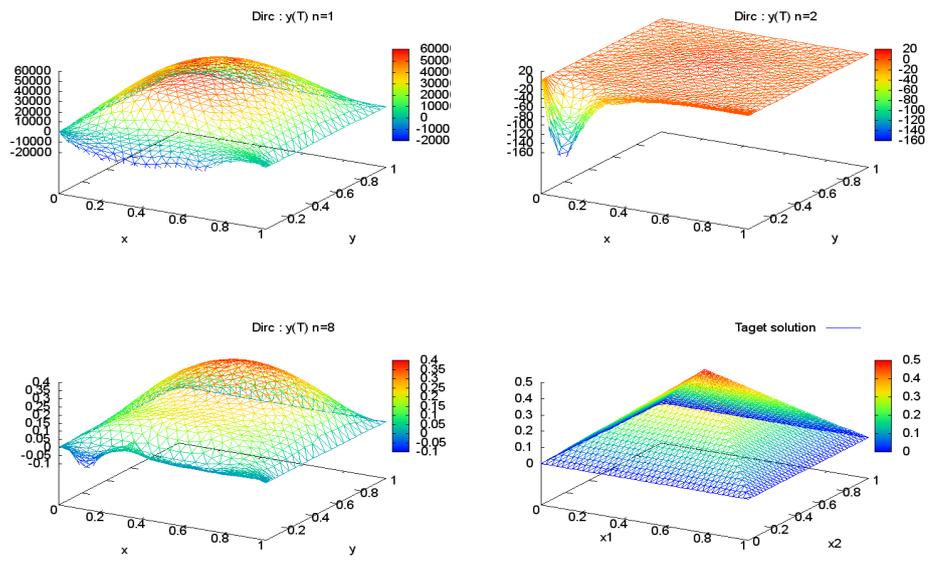


FIGURE 12. Several snapshots in  $\hat{n}$  of final state at time T:  $\mathcal{Y}_\Gamma(T)$ . The test case corresponds to Dirichlet control sever Ill-posed problem  $\mathcal{P}_4^\alpha$ .

432 5.2.4. *Regularization based on the optimal choice of partition.* The next discussion concerns the  
 433 kind of situation where the partition leads to multiple solutions, which is common in ill-posed  
 434 problems. In fact, we discuss a regularization procedure used as an exception handling tool to  
 435 choose the best partition, giving the best solution of the handled control problem.

It is well known that ill-posed problems are very sensitive to noise, which could be present due to numerical approximation or to physical phenomena. In that case, numerical algorithm may blow-up and fail. We present several numerical tests for the Dirichlet boundary control, which is a non trivial problem numerically. The results show that in general time domain decomposition may improve the results in several cases. But scalability is not guaranteed as it is for the distributed control. We propose a regularization procedure in order to avoid the blow-up and also to guarantee the optimal choice of partition of the time domain. This procedure is based on a test of the monotony of the cost function. In fact, suppose that we possess 64 processors to run the numerical problem. Once we have assembled the Hessian  $H_k$  and the Jacobian  $D_k$  for the partition  $\hat{n} = 64$ , we are actually able to get for free the results of the Hessian and the Jacobian for all partitions  $\hat{n}$  that divide 64. Hence, we can use the quadratic property of the cost functional in order to predict and test the value of the cost function for the next iteration without making any additional computations. The formulae is given by:

$$J(v^{k+1}) = J(v^k) - \frac{1}{2} D_k^T H_k^{-1} D_k.$$

436 We present in Algorithm 4 the technique that enables us to reduce in rank and compute a series  
 437 of Hessians and Jacobians for any partition  $\hat{n}$  that divide the available number of processors. An  
 example of the applicability of these technique, on a 4-by-4 SPD matrix, is given in Appendix.

---

**Algorithm 4:** Reduce in rank of the partition  $\hat{n}$

---

```

0 Input:  $\hat{n}, H_{\hat{n}}^k, D_{\hat{n}}^k$ ;
1  $n = \hat{n}$ ;
2  $J_{n/2}^{k+1} = J_n^{k+1}$ ;
3 while  $J_{n/2}^{k+1} > J_n^k$  do
4   for  $i = 0; i \leq n; i + 2$  do
5      $(D_{n/2}^k)_i = (D_n^k)_i + (D_n^k)_{i+1}$ ;
6     for  $j = 0; j \leq n; j + 2$  do
7        $(H_{n/2}^k)_{i,j} = (H_n^k)_j + (H_n^k)_{j+1}$ ;
8     end
9   end
10  Estimation of the cost  $J_{n/2}^k$ ;
11   $n = n/2$ ;
12 end

```

---

438

439

## 6. CONCLUSION

440 We have presented in this article a new optimization technique to enhance the steepest descent  
 441 algorithm via domain decomposition in general and we applied our new method in particular to  
 442 time-parallelizing the simulation of an optimal heat control problem. We presented its perfor-  
 443 mance (in CPU time and number of iterations) versus the traditional steepest descent algorithm  
 444 in several and various test problems. The key idea of our method is based on a quasi-Newton  
 445 technique to perform efficient real vector step-length for a set of descent directions regarding the

446 domain decomposition. The originality of our approach consists in enabling parallel computation  
 447 where its vector step-length achieves the optimal descent direction in a high dimensional space.  
 448 Convergence property of the presented method is provided. Those results are illustrated with  
 449 several numerical tests using parallel resources with MPI implementation.

#### 450 ACKNOWLEDGEMENT

451 I would like to thank the anonymous referees for their careful reviewing and thorough feedback  
 452 on this paper and for drawing my attention to the work of Heinkenschloss [25]. I would also like  
 453 to thank Professor Yvon Maday for earlier discussions.

#### 454 REFERENCES

- 455 [1] A Alla and M Falcone. A time-adaptive pod method for optimal control problems. In *Proceedings of the 1st*  
 456 *IFAC Workshop on Control of Systems Modeled by Partial Differential Equations*, pages 245–250, 2013.
- 457 [2] Alessandro Alla and Maurizio Falcone. An adaptive pod approximation method for the control of advection-  
 458 diffusion equations. In *Control and Optimization with PDE Constraints*, pages 1–17. Springer, 2013.
- 459 [3] Eyal Arian, Marco Fahl, and Ekkehard W Sachs. Trust-region proper orthogonal decomposition for flow  
 460 control. Technical report, DTIC Document, 2000.
- 461 [4] Jerzy K Baksalary and Simo Puntanen. Generalized matrix versions of the cauchy-schwarz and kantorovich  
 462 inequalities. *Aequationes Mathematicae*, 41(1):103–110, 1991.
- 463 [5] Randolph E. Bank, Bruno D. Welfert, and Harry Yserentant. A class of iterative methods for solving saddle  
 464 point problems. *Numer. Math.*, 56(7):645–666, 1990.
- 465 [6] Jonathan Barzilai and Jonathan M Borwein. Two-point step size gradient methods. *IMA Journal of Numerical*  
 466 *Analysis*, 8(1):141–148, 1988.
- 467 [7] Faker Ben Belgacem and Sidi Mahmoud Kaber. On the Dirichlet boundary controllability of the one-  
 468 dimensional heat equation: semi-analytical calculations and ill-posedness degree. *Inverse Problems*,  
 469 27(5):055012, 19, 2011.
- 470 [8] Petter Bjorstad and William Gropp. *Domain decomposition: parallel multilevel methods for elliptic partial*  
 471 *differential equations*. Cambridge University Press, 2004.
- 472 [9] A. Borzi, K. Ito, and K. Kunisch. An optimal control approach to optical flow computation. *Internat. J.*  
 473 *Numer. Methods Fluids*, 40(1-2):231–240, 2002. ICFD Conference on Numerical Methods for Fluid Dynamics  
 474 (Oxford, 2001).
- 475 [10] Richard H. Byrd, Gabriel Lopez-Calva, and Jorge Nocedal. A line search exact penalty method using steering  
 476 rules. *Math. Program.*, 133(1-2, Ser. A):39–73, 2012.
- 477 [11] C. Carthel, R. Glowinski, and J.-L. Lions. On exact and approximate boundary controllabilities for the heat  
 478 equation: a numerical approach. *J. Optim. Theory Appl.*, 82(3):429–484, 1994.
- 479 [12] Augustin-Louis Cauchy. Méthode générale pour la résolution des systèmes d'équations simultanées. *Compte*  
 480 *Rendu des Scieances de L'Académie des Sciences XXV*, S'erie A(25):536–538, October 1847.
- 481 [13] Haecheon Choi, Michael Hinze, and Karl Kunisch. Instantaneous control of backward-facing step flows. *Appl.*  
 482 *Numer. Math.*, 31(2):133–158, 1999.
- 483 [14] Haecheon Choi, Roger Temam, Parviz Moin, and John Kim. Feedback control for unsteady flow and its  
 484 application to the stochastic Burgers equation. *J. Fluid Mech.*, 253:509–543, 1993.
- 485 [15] Philippe G. Ciarlet. *Introduction à l'analyse numérique matricielle et à l'optimisation*. Collection  
 486 *Mathématiques Appliquées pour la Maîtrise*. [Collection of Applied Mathematics for the Master's Degree].  
 487 Masson, Paris, 1982.
- 488 [16] Jean-Michel Coron and Emmanuel Trélat. Global steady-state controllability of one-dimensional semilinear  
 489 heat equations. *SIAM J. Control Optim.*, 43(2):549–569, 2004.
- 490 [17] Sylvain Ervedoza and Enrique Zuazua. The wave equation: Control and numerics. In *Control of Partial*  
 491 *Differential Equations*, Lecture Notes in Mathematics, pages 245–339. Springer Berlin Heidelberg, 2012.
- 492 [18] David J Evans. *Preconditioning Methods: Theory and Applications*. Gordon and Breach Science Publishers,  
 493 Inc., 1983.
- 494 [19] H. O. Fattorini. *Infinite-dimensional optimization and control theory*, volume 62 of *Encyclopedia of Mathe-*  
 495 *matics and its Applications*. Cambridge University Press, Cambridge, 1999.
- 496 [20] W. R. Graham, J. Peraire, and K. Y. Tang. Optimal control of vortex shedding using low-order models. I.  
 497 Open-loop model development. *Internat. J. Numer. Methods Engrg.*, 44(7):945–972, 1999.
- 498 [21] W. R. Graham, J. Peraire, and K. Y. Tang. Optimal control of vortex shedding using low-order models. II.  
 499 Model-based control. *Internat. J. Numer. Methods Engrg.*, 44(7):973–990, 1999.

- 500 [22] Luigi Grippo, Francesco Lampariello, and Stephano Lucidi. A nonmonotone line search technique for newton's  
501 method. *SIAM Journal on Numerical Analysis*, 23(4):707–716, 1986.
- 502 [23] Marcus J Grote and Thomas Huckle. Parallel preconditioning with sparse approximate inverses. *SIAM Jour-*  
503 *nal on Scientific Computing*, 18(3):838–853, 1997.
- 504 [24] Martin Gubisch and Stefan Volkwein. POD a-posteriori error analysis for optimal control problems with  
505 mixed control-state constraints. *Comput. Optim. Appl.*, 58(3):619–644, 2014.
- 506 [25] Matthias Heinkenschloss. A time-domain decomposition iterative method for the solution of distributed linear  
507 quadratic optimal control problems. *J. Comput. Appl. Math.*, 173(1):169–198, 2005.
- 508 [26] M Hinze and K Kunisch. Suboptimal control strategies for backward facing step ows. In *Proceedings of 15th*  
509 *IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics*, Editor A. Sydow,  
510 volume 3, pages 53–58. Citeseer, 1997.
- 511 [27] Michael Hinze. Optimal and instantaneous control of the instationary navier-stokes equations. 2000.
- 512 [28] Michael Hinze and Karl Kunisch. Three control methods for time-dependent fluid flow. *Flow Turbul. Com-*  
513 *bust.*, 65(3-4):273–298, 2000.
- 514 [29] K. Ito and S. S. Ravindran. A reduced basis method for control problems governed by PDEs. In *Control*  
515 *and estimation of distributed parameter systems (Vorau, 1996)*, volume 126 of *Internat. Ser. Numer. Math.*,  
516 pages 153–168. Birkhäuser, Basel, 1998.
- 517 [30] K. Ito and S. S. Ravindran. Reduced basis method for optimal control of unsteady viscous flows. *Int. J.*  
518 *Comput. Fluid Dyn.*, 15(2):97–113, 2001.
- 519 [31] L. V. Kantorovich. *Functional analysis and applied mathematics*. NBS Rep. 1509. U. S. Department of  
520 Commerce National Bureau of Standards, Los Angeles, Calif., 1952. Translated by C. D. Benster.
- 521 [32] K. Kunisch, S. Volkwein, and L. Xie. HJB-POD-based feedback design for the optimal control of evolution  
522 problems. *SIAM J. Appl. Dyn. Syst.*, 3(4):701–722, 2004.
- 523 [33] J. E. Lagnese and G. Leugering. Time-domain decomposition of optimal control problems for the wave  
524 equation. *Systems Control Lett.*, 48(3-4):229–242, 2003. Optimization and control of distributed systems.
- 525 [34] Irena Lasiecka and Roberto Triggiani. *Control theory for partial differential equations: continuous and ap-*  
526 *proximation theories. I-II*, volume 47-75 of *Encyclopedia of Mathematics and its Applications*. Cambridge  
527 University Press, Cambridge, 2000. Abstract hyperbolic-like systems over a finite time horizon.
- 528 [35] Changhoon Lee, John Kim, and Haecheon Choi. Suboptimal control of turbulent channel flow for drag  
529 reduction. *Journal of Fluid Mechanics*, 358:245–258, 1998.
- 530 [36] J Lions, Yvon Maday, and Gabriel Turinici. A”parareal” in time discretization of pde’s. *Comptes Rendus de*  
531 *l’Academie des Sciences Series I Mathematics*, 332(7):661–668, 2001.
- 532 [37] J.-L. Lions. *Optimal control of systems governed by partial differential equations*. Translated from the French  
533 by S. K. Mitter. Die Grundlehren der mathematischen Wissenschaften, Band 170. Springer-Verlag, New York,  
534 1971.
- 535 [38] J.-L. Lions. *Contrôlabilité exacte, perturbations et stabilisation de systèmes distribués. Tome 1-2*, volume  
536 8-9 of *Recherches en Mathématiques Appliquées [Research in Applied Mathematics]*. Masson, Paris, 1988.  
537 Contrôlabilité exacte. [Exact controllability], With appendices by E. Zuazua, C. Bardos, G. Lebeau and J.  
538 Rauch.
- 539 [39] Y. Maday, J. Salomon, and G. Turinici. Monotonic parareal control for quantum systems. *SIAM Journal on*  
540 *Numerical Analysis*, 45(6):2468–2482, 2007.
- 541 [40] Yvon Maday, Mohamed-Kamel Riahi, and Julien Salomon. Parareal in time intermediate targets methods  
542 for optimal control problems. In Kristian Bredies, Christian Clason, Karl Kunisch, and Gregory von Winckel,  
543 editors, *Control and Optimization with PDE Constraints*, volume 164 of *International Series of Numerical*  
544 *Mathematics*, pages 79–92. Springer Basel, 2013.
- 545 [41] Yvon Maday and Gabriel Turinici. A parareal in time procedure for the control of partial differential equa-  
546 tions. *C. R. Math. Acad. Sci. Paris*, 335(4):387–392, 2002.
- 547 [42] Tahir Malas and Levent Gürel. Incomplete lu preconditioning with the multilevel fast multipole algorithm  
548 for electromagnetic scattering. *SIAM Journal on Scientific Computing*, 29(4):1476–1494, 2007.
- 549 [43] Xavier Marduel and Karl Kunisch. Suboptimal control of transient nonisothermal viscoelastic fluid flows.  
550 *Physics of Fluids (1994-present)*, 13(9):2478–2491, 2001.
- 551 [44] Sorin Micu and Enrique Zuazua. Regularity issues for the null-controllability of the linear 1-d heat equation.  
552 *Systems Control Lett.*, 60(6):406–413, 2011.
- 553 [45] Olivier Pironneau, Frédéric Hecht, and Jacques Morice. freefem++, www.freefem.org/, 2013.
- 554 [46] Alfio Quarteroni and Alberto Valli. *Domain decomposition methods for partial differential equations*. Nu-  
555 *merical Mathematics and Scientific Computation*. The Clarendon Press, Oxford University Press, New York,  
556 1999. Oxford Science Publications.
- 557 [47] SS Ravindran. *Proper orthogonal decomposition in optimal control of fluids*, volume 99. National Aeronautics  
558 and Space Administration, Langley Research Center, 1999.

- 559 [48] Tyrone Rees, Martin Stoll, and Andy Wathen. All-at-once preconditioning in PDE-constrained optimization.  
560 *Kybernetika (Prague)*, 46(2):341–360, 2010.
- 561 [49] Ulrich Rüde. *Mathematical and computational techniques for multilevel adaptive methods*. SIAM, 1993.
- 562 [50] Yousef Saad. *Iterative methods for sparse linear systems*. Siam, 2003.
- 563 [51] Scilab Enterprises. *Scilab: Le logiciel open source gratuit de calcul numérique*. Scilab Enterprises, Orsay,  
564 France, 2012.
- 565 [52] David Silvester and Andrew Wathen. Fast iterative solution of stabilised Stokes systems. II. Using general  
566 block preconditioners. *SIAM J. Numer. Anal.*, 31(5):1352–1367, 1994.
- 567 [53] Martin Stoll and Andy Wathen. All-at-once solution of time-dependent Stokes control. *J. Comput. Phys.*,  
568 232:498–515, 2013.
- 569 [54] Andrea Toselli and Olof Widlund. *Domain decomposition methods: algorithms and theory*, volume 3.  
570 Springer, 2005.
- 571 [55] Fredi Tröltzsch. *Optimal control of partial differential equations*, volume 112 of *Graduate Studies in Mathe-*  
572 *matics*. American Mathematical Society, Providence, RI, 2010. Theory, methods and applications, Translated  
573 from the 2005 German original by Jürgen Sprekels.
- 574 [56] Stefan Ulbrich. Generalized sqp-methods with ?parareal? time-domain decomposition for time-dependent  
575 pde-constrained optimization. In *Real-time PDE-constrained optimization*, volume 3, pages 145–168. SIAM  
576 Philadelphia, PA, 2007.
- 577 [57] Andrew Wathen and David Silvester. Fast iterative solution of stabilised Stokes systems. I. Using simple  
578 diagonal preconditioners. *SIAM J. Numer. Anal.*, 30(3):630–649, 1993.
- 579 [58] Gonglin Yuan and Zengxin Wei. The Barzilai and Borwein gradient method with nonmonotone line search  
580 for nonsmooth convex optimization problems. *Math. Model. Anal.*, 17(2):203–216, 2012.