Optimal boundary control parareal algorithm for cooling electronics circuits

Juan R. Fleitas[1], Diego H. Stalder[1], Christian E. Schaerer[2]

[1]Engineering Faculty, National University of Asuncion, SL, Paraguay
[2]LCCA, Polytechnic Faculty, National University of Asuncion, SL, Paraguay
LCCA: SCIENTIFIC AND APPLIED COMPUTING LABORATORY.

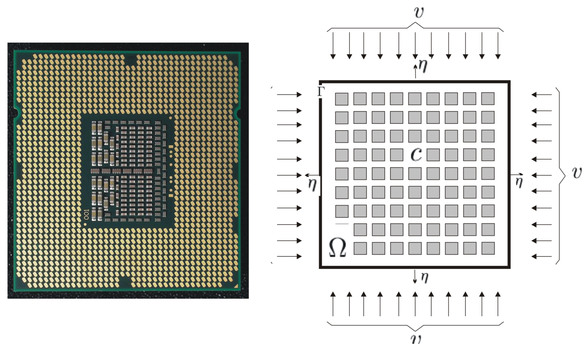
Abstract

In this work we consider the Parareal Algorithm, developed in [3], as a preconditioner for solving the optimal boundary control problem for cooling an electronic circuit. The constitutive equations are obtained by modeling the circuit as a parabolic partial differential equation, where the cooling mechanisms are modeled using Neumann boundary conditions (control variables) [1]. The optimal control problem is obtained by associating to the state and control variables, a linearized quadratic cost function like in [4]. The classical finite element method - (Galerkin)- was implemented for the spatial dicretization resulting in a large scale system ODEs [2].

For the implementation of the preconditioner the time domain is divided into coarse time intervals, where the initials conditions are arbitrary. Each of these coarse time intervals are divided into fine time intervals that are solved simultaneously in several processors. The Conjugate Gradient method is used to update the initial conditions and the control variables, and the Parareal method is used as a preconditioner to accelerate its convergence.

Mathematical model

The electronic circuits temperature subject to internal heating sources with cooling mechanisms are modelled by the heat equation using Neumann boundary conditions [1].

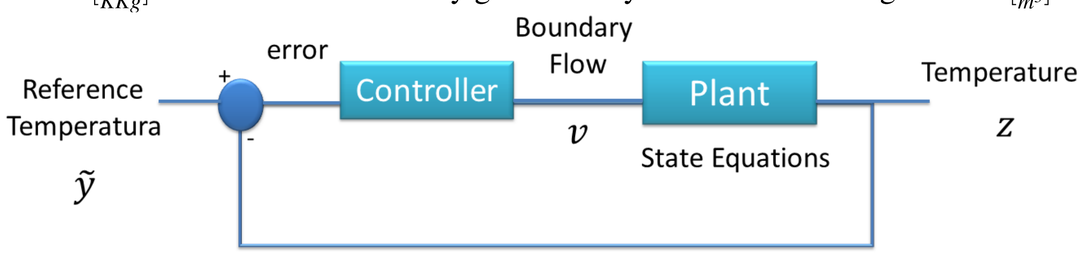


Optimal control problem

The optimal control problem is formulated as a constrained minimization problem. To design the controller we define a linear-quadratic cost function that assess the quality of the solución according to the parameters r, q and s:

$$\begin{cases} \min \quad \mathcal{J}(z(v),v) = \dfrac{1}{2}\displaystyle\int_{t_o}^{t_f} \left(r\|v\|^2_{L^2(\Gamma)} + q\|(z(t)-\tilde{y}(t))\|^2_{L^2(\Omega)} \right) dt \\ \qquad\qquad +\frac{1}{2}s\|z(t_f)-\tilde{y}(t_f)\|^2_{L^2(\Omega)}, \\ \text{s. t.} \begin{cases} \varrho c_p \frac{\partial z}{\partial t} = \lambda \triangle z + c, \quad \text{en} \quad \Omega \times [t_o, t_f], \\ \frac{\partial z}{\partial \eta} = v \quad \text{en} \quad \Gamma \times [t_o, t_f], \\ z(t_o, \Omega) = z_o, \end{cases} \end{cases} \tag{1}$$

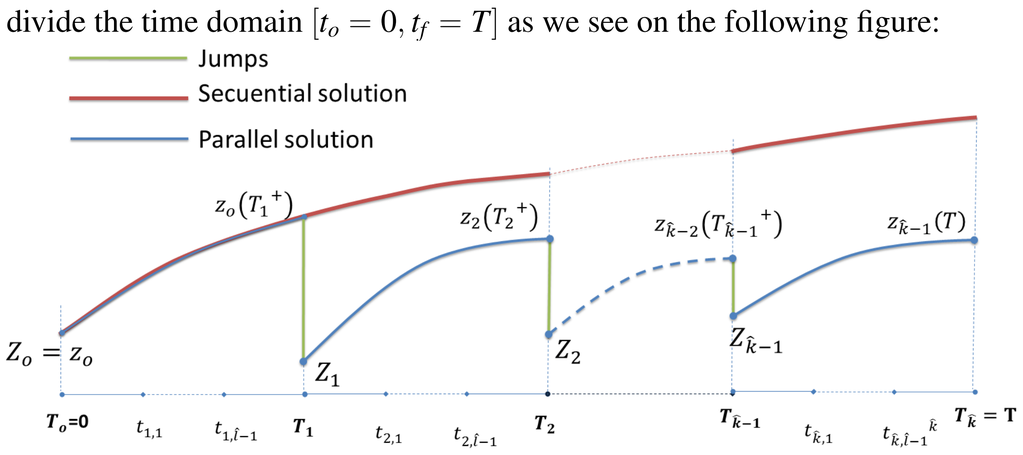
where ϱ is the mass density of the material $[\frac{Kg}{m^3}]$, c_p is the internal energy density of the material $[\frac{W}{KKg}]$ and c is the heat density generated by the internal heating sources $[\frac{W}{m^3}]$.



For the spatial discretization we choose a set of equal triangular elements as in [2] and we get a large scale ODEs by applying the standard finite element methods to the problem (1).

Time domain decomposition (Parallelization)

We divide the time domain $[t_o = 0, t_f = T]$ as we see on the following figure:



$\hat{k}$ =Number of coarse grid sub-intervals (Processors) $\quad \Delta T = \frac{T}{\hat{k}} \quad \tau = \frac{\Delta T}{\hat{l}}$

$\hat{l}$ =Number of fine grid sub-intervals

We modify slightly the cost functional, to have the same solution that we have using the secuencial algorithm. It consists in the introduction of a penalization proportional to the jumps:

$$\begin{cases} \min \mathcal{J}_\varepsilon(\underline{z_k}, \underline{v_k}, Z_k) = \mathcal{J}(z(v),v) + \dfrac{1}{2\,\varepsilon\,\Delta T}\displaystyle\sum_{k=1}^{\hat{k}-1} \|\underline{z_{k-1}}(T_k^-) - Z_k\|^2_{M_h}, \\ \text{s. t.} \left\{ \begin{cases} M_h \underline{\dot{z}_k} = A_h \underline{z_k} + B_h \underline{v_k} + c_h, \\ \underline{z_k}(T_k^+) = Z_k \end{cases} \quad t \in [T_k, T_{k+1}], \right\} \quad k = 0,\ 1,\ \cdots,\ \hat{k}-1. \end{cases} \tag{2}$$

where $\varepsilon > 0$ is the penalization parameter and $\underline{z} \in \mathbb{R}^{\hat{q}}$, $\underline{v} \in \mathbb{R}^{\hat{p}}$, $M_h = M_h^T,\ A_h = A_h^T \in \mathbb{R}^{\hat{q}\times\hat{q}}$; $B_h \in \mathbb{R}^{\hat{q}\times\hat{p}}$; $R_h \in \mathbb{R}^{\hat{p}\times\hat{p}}$; $c_h \in \mathbb{R}^{\hat{q}}$.

Unconstrained Problem

The functional on the equation (2) yields the following Lagrangian:

$$\mathcal{L}(\underline{z},\underline{v},Z,\underline{p},\underline{\eta}) = \mathcal{J}_\varepsilon(\underline{z},\underline{v},Z) + \sum_{k=0}^{\hat{k}-1} \int_{T_k}^{T_{k+1}} \underline{p_k}^T \left(M_h \underline{\dot{z}_k} - A_h \underline{z_k} - B_h \underline{v_k} - c_h\right) dt + \sum_{k=0}^{\hat{k}-1} \underline{\eta_k}^T \left(\underline{z_k}(T_k^+) - Z_k\right).$$

where p_k y η_k^T are Lagrange multipliers.

Preconditioned Conjugate Gradient

Adjoint State

Taking $\left.\frac{\partial \mathcal{L}}{\partial \underline{z_k}}\right|_* = 0, \forall k = 0, 1, \cdots, \hat{k}-1$ we have the adjoint state:

$$\begin{aligned} M_h \underline{\dot{p}_k} &= -A_h \underline{p_k} + q\, M_h(\underline{z_k} - \underline{\tilde{y}_k}), \\ \underline{p_k}(T_{k+1}^-) &= -\frac{1}{\varepsilon\,\Delta T}\left(\underline{z_k}(T_{k+1}^-) - Z_{k+1}\right) \quad \forall k = 0, 1, \cdots, \hat{k}-2, \\ \underline{p_{\hat{k}-1}}(T_{\hat{k}}^-) &= -s\left(\underline{z_{\hat{k}-1}}(T_{\hat{k}}^-) - \underline{\tilde{y}}\right) \quad \forall\, k = \hat{k}-1, \end{aligned} \tag{3}$$

these expressions are used for getting an expression of the functional gradient.

The iterative algorithm (Corrections)

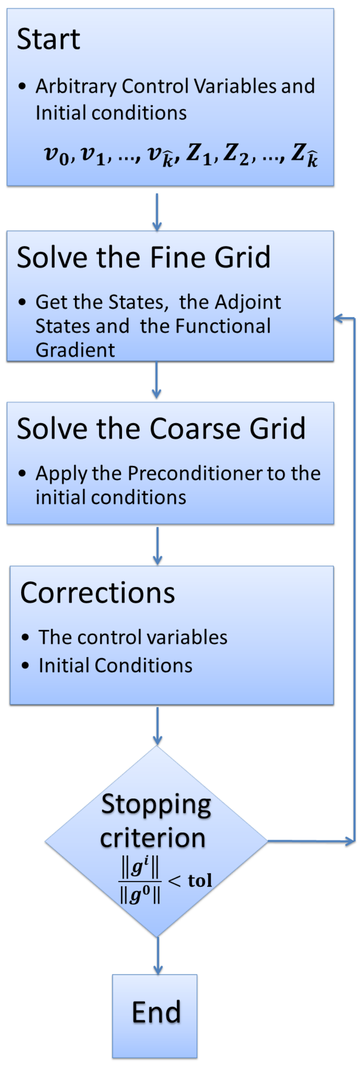
We start the CG with arbitrary values of the control variables and initial conditions. These initial values will be corrected by the following scheme:

$$v^{i+1} = v^i +, \alpha^i d_v^i, \qquad Z^{i+1} = Z^i + \alpha^i d_Z^i, \tag{4}$$

where $v, d_v \in \mathbb{R}^{\hat{p}\hat{l}\hat{k}}$, $Z, d_Z \in \mathbb{R}^{\hat{q}(\hat{k}-1)}$ and $\alpha \in \mathbb{R}$ that is obtained applying the steepest Descent condition. The vectors d_v and d_Z are components of the search directions The search directions $d^{i+1} = [d_v^{i+1\,T} \quad d_Z^{i+1\,T}]^T$ are generated as a linear combination of the functional gradient and the previous search direction:

$$d^{i+1} = -\nabla \mathcal{J}_\varepsilon{}^{i+1} + \beta^i d^i, \quad \beta^i = \frac{(\nabla \mathcal{J}_\varepsilon{}^{i+1}, Hd^i)}{(d^i, Hd^i)} \tag{5}$$

where $Hd^i \in \mathbb{R}^{(\hat{p}\hat{l}\hat{k}+\hat{q}(\hat{k}-1))}$ is the product of the functional Hessian and the search direction. The functional gradient is obtained by the adjoint method.



Preconditioner

The jumps of the adjoint state can be obtained from the residual of the following equations:

$$\begin{bmatrix} I & -\mathcal{F}_{\Delta\tau}^T & & \\ & \ddots & \ddots & \\ & & I & -\mathcal{F}_{\Delta\tau}^T \\ & & & I \end{bmatrix} \begin{bmatrix} \frac{1}{\varepsilon\Delta T}F_0 & & & \\ & \frac{1}{\varepsilon\Delta T}F_0 & & \\ & & \ddots & \\ & & & \frac{1}{\varepsilon\Delta T}F_0 \end{bmatrix} \begin{bmatrix} I & & & \\ -\mathcal{F}_{\Delta\tau} & I & & \\ & \ddots & \ddots & \\ & & -\mathcal{F}_{\Delta\tau} & I \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_{\hat{k}-1} \end{bmatrix} = F$$

$$\mathbf{C^T K C Z} = \mathbf{F}, \tag{6}$$

where $\mathcal{F}_{\Delta\tau} = (F_1^{-1}F_0)^{\hat{l}}$, $F_1 = M_h - \tau A_h$, $F_0 = M_h$ is obtained from the Backward and Fordward Euler schemes to get the state and the adjoint state respectively.

This fact is the reason for using the aproximation of the inverse of the matrix $C^T\,K\,C$ as a preconditioner for the initial conditions. The aproximation of $\widetilde{C}$ is obtained by using ΔT like stepsize. Then $\mathcal{F}_{\Delta\tau} \approx \mathcal{G}_{\Delta\tau} = F_1^{-1}F_0$, $F_1 = M_h - \Delta T A_h$, $F_0 = M_h$

Numerical Results and Concluding Remarks

The tables show the number of iterations of the used algorithm:

$\hat{k}$	4	8	16	32	64
Iter.	35	37	42	44	39

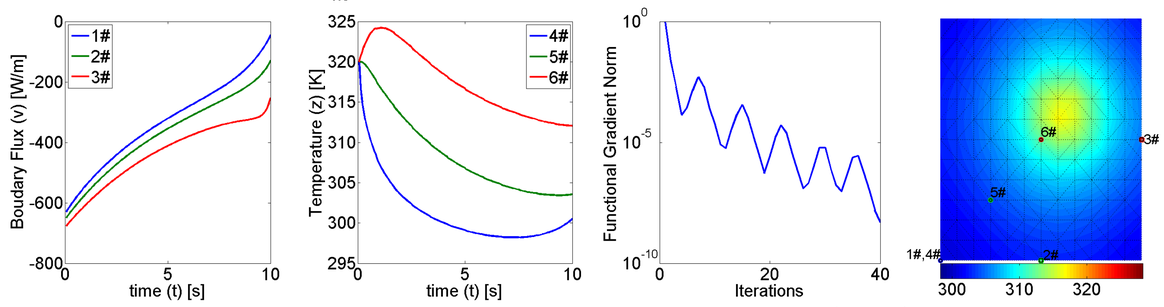
Table: Escalabilidad $\hat{l} = 4$.

$\hat{k}$	4	8	16	32	64
$\hat{l}$	64	32	16	8	4
Iter.	39	39	42	41	38

Table: Speed-up.

The numerical results show that the parallel algorithm reduces the computing time and allows the solving of control problems that demand large memory.

The figures show the control variables, the temperature of the circuite subjected to a heating source of $4x10^5\frac{W}{m^2}$, and the evolution of the norm of functional gradient.



Temperature of the plate ($5x5cm$) $r = 10^{-3}$, $q = 1$, $s = 10$, $z(t_o) = 320^oK$. p/t=T$\hat{k} = 16$ y $\hat{l} = 16$
(169 elements y 48 control elements)

References

[1] Neittaanmäaki, P.;Tiba D. Optimal Control of Nonlinear Parabolic Systems, Marcel Dekker (1994), 339 pags.

[2] Carstensen C., Alberty J., Funken S.; Remarks around 50 lines of Matlab: short finite element Implementation. Numerical Algorithms, Numerical Algorithms 20 (1999), 20:117-137.

[3] Maday I., Turinici G.; "A parareal in time procedure for the control of partial diferential equations. Numerical Analisis", C. R. Acad. Sci. Paris, Ser. I 335 (2002), 387-392. *

[4] Schaerer C. E., Mathew T., Sarkis M.; Block iterative algorithms for the solution of parabolic optimal control problems, Lecture Notes in Computer Science, 4395 (2007) 452-465.

diego.s.eik@gmail.com, juanrofleitas@gmail.com, cschaer@pol.una.py