Purdue University

Purdue e-Pubs

Department of Computer Science Technical Reports

Department of Computer Science

1999

Automated Estimation of Relaxation Parameters for Interface Relaxation

John R. Rice
Purdue University, jrr@cs.purdue.edu

P. Tsompanopoulou

E. Vavalis

Report Number:
99-015

Rice, John R.; Tsompanopoulou, P.; and Vavalis, E., "Automated Estimation of Relaxation Parameters for Interface Relaxation" (1999). *Department of Computer Science Technical Reports.* Paper 1446.
https://docs.lib.purdue.edu/cstech/1446

AUTOMATED ESTIMATION OF RELAXATION PARAMETERS FOR INTERFACE RELAXATION

John R. Rice
P. Tsompanopoulou
E. Vavalis

Department of Computer Sciences
Purdue University
West Lafayette, IN   47907

Automated estimation of relaxation parameters for interface relaxation [1]

J.R. Rice P. Tsompanopoulou [2] and E. Vavalis [2]

Purdue University, Computer Science Department, West Lafayette, IN 47907.

Abstract

An adaptive procedure, based on Automatic Differentiation, for estimating "good" values for the relaxation paramaters for general multi-dimensional problems is proposed.

1 Introduction

Interface Relaxation (IR) methods [11] have been recently attracted a lot of attention. This is due to their increased capability in splitting Multi-Physics Multi-Domain (MPMD) PDE problems into a collaborative pool of simple problems [8]. This splitting is done in a natural way and leads to effective numerical methods which fully utilize existing PDE solvers through software reuse practices. The resulting PDE sub-problems are coupled together thought relaxation mechanisms that try to mimic the physical world. There already exist several such interface relaxation techniques in the literature [11]. Most of them involve relaxation parameters that can significantly improve the convergence properties of the various IR methods if their values are properly chosen. These parameters depend on many components (PDE operator, PDE domain, splitting, ...) of the original problem. This, unfortunately makes the selection of "optimum" values for the relaxation parameters a hard and challenging problem. On the other hand the local subdomain discretization schemae do not affect the convergence properties of the IR schemae giving them the great versatility to let one select the most appropriate discretization parameters or method for the differential problem define on each subdomain.

Several papers have been devoted in theoretically obtaining optimum values for the interface relaxation parameters [14,16]. Nevertheless the analysis is done for model problems and subdomain splittings and although they provide important information fail to assist a non-expert user to select values for those parameters for problems with moderate complexion. When this complication is increased even experts can not effectively select parameters.

The main objective of our study is to provide an adaptive heuristic mechanism for automatically selecting "good" relaxation parameters for general differential equations and domain decompositions. These proposed mechanism uses Automatic Differentiation (AD) and utilizes the existing analytically estimated values

The rest of this paper is organized as following. In the next section we briefly present the general methodology of IR. An adaptive procedure for estimating "good" values for the relaxation parameters is proposed in section 3.

2   Domain decomposition with iterated interface relaxation

Currently the domain decomposition world consists of two parts –overlapping and non–overlapping– both living in prosperity. Overlapping, known also as Schwartz, methods were the first considered and have already proved themselves as very efficient numerical procedures enjoying certain very desirable convergence properties. Nevertheless it has been also observed that they might have several serious drawbacks which will prohibit their use for certain applications. For example, almost all of the many proposed domain decomposition methods for solving wave propagation models (that consist of the Helmholtz equation coupled with various absorbing or reflecting boundary conditions) are non–overlapping and of interface relaxation type [1,3,7,10].

Non–overlapping methods exhibit certain advantages compared to overlapping ones. Specifically:

- They are not sensitive to jumps on the operator coefficients. Their convergence behavior and theoretical error estimates remain the same even if the differential operator includes discontinuous coefficients provided that the jumps occur along the interface lines [15].
- They have smaller communication overhead in a parallel implementation on distributed memory multiprocessor systems. Their communication overhead is proportional to the length of the interface lines while it is proportional to the overlapping area in the case of overlapping methods [4].
- The bookkeeping is rather easy for the decomposition and manipulations of the associated data structures compared to the more complicated and

2

$$g_{ij}U_i^{new}, \left(U_j^{new}, \frac{\partial U_i^{new}}{\partial n}, \frac{\partial U_j^{new}}{\partial n} \right) = 0$$
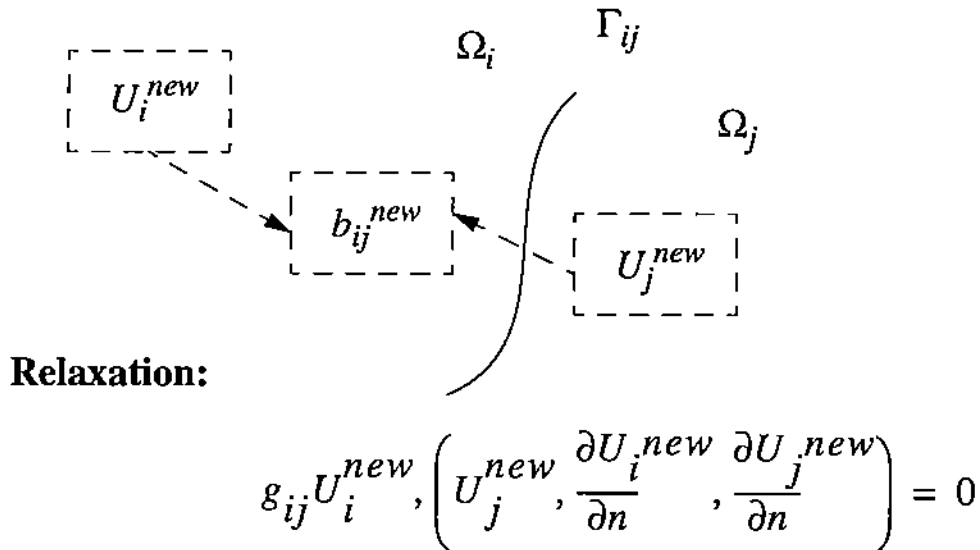
Fig. 1. The interface relaxation mechanism

costly bookkeeping of the overlapping methods [4].

There are two principal viewpoints of non-overlapping methods, preconditioning and interface relaxation. For an in depth and up-to-date survey of non-overlapping domain decomposition methods considered and analyzed from the preconditioning viewpoint the reader is referred to [13] and for a general formulation and analysis of interface relaxation methods to [8]. We give a brief presentation of the interface relaxation method philosophy and practice, in order to identify its main characteristics.

Interface relaxation is a step beyond non–overlapping domain decomposition; it follows Southwell's relaxation of the 1930's – but at the PDE instead of the linear algebra level – to formulate relaxation as iterated interface smoothing procedures. A complex physical phenomenon consists of a collection of simple parts with each one of them obeying a single physical law locally and adjusting its interface conditions with neighbors. Interface relaxation partitions the domain on a set of non–overlapping subdomains, imposes some boundary conditions on the interface among subdomains lines. Given an initial guess, it imitates the physics of the real world by solving the local problems exactly on each subdomain and relaxing boundary values to get better estimates of correct interface conditions. This is illustrated in Figure 1 where the generic relaxation formula $g_{i,j}$ (based on the current solutions U_i^{New} and U_j^{New} of the two local to the neighboring subdomains Ω_i and Ω_j) calculates successive approximations $b_{i,j}^{New}$ to the solution on the interface $\Gamma_{i,j}$ between them.

To formally describe this method we consider the differential problem

$$Du = f \text{ in } \Omega, \quad Bu = c \text{ on } \partial\Omega \tag{1}$$

3

where D is an elliptic, non–linear in general, differential operator and B a condition operator defined on the boundary $\partial\Omega$ of a domain $\Omega \in R^d, d = 1, 2, \dots$. This domain is partitioned into p subdomains $\Omega_i, i = 1, \dots, p$ such that $\Omega = \cup_{i=1}^p \Omega_i$ and $\cap_{i=1}^p \overset{\circ}{\Omega}_i = \emptyset$. For reasons related either to the physical characteristics of this problem or to the computing resources available, one would like to replace (1) with the following system of loosely coupled differential problems

$$D_i u = f_i \text{ in } \Omega_i, \quad G_i u = 0 \text{ on } \partial\Omega_i \backslash \partial\Omega \quad B_i u = c_i \text{ on } \partial\Omega_i \cap \partial\Omega \qquad (2)$$

where $i = 1, \dots, p$. These differential problems are coupled through the interface conditions $G_i u = 0$ and involve the restrictions D_i and B_i of the global differential and boundary operators, D and B, respectively, on each subdomain with some of them possibly linear and some others nonlinear. The functions f_i and c_i are similar restrictions of functions f and c. The local interface operator G_i is associated with the interface relaxation method and different selections for the G_i's lead to different relaxation schemes. In this study we consider several interface relaxation methods that have the following characteristics:

- They first decompose the problem (1) at differential level and then discretize the resulting differential subproblems (2).
- They have the versatility to use the most appropriate discretization scheme for each subproblem.
- They do not overlap the subdomains Ω_i.
- Using good relaxation parameters in G_i, they are fast enough so no preconditioning is needed.
- They simplify the geometry and physics of the computation by considering the subproblems (2) instead of the global differential problem (1).
- They can utilize software parts technology by reusing existing "legacy" software parts for solving the individual subproblems (2).
- They are general and robust.

There are several challenging questions concerning practical applications of such methods (e.g. find the most suitable relaxer for a particular problem of application, determine what is the domain of applicability of each one of them, explain the interaction between the mathematical iteration and the numerical solving method, select "good" or "optimal" values for the relaxation parameters involved, $\cdots$). It is worth to point out that since all the methods decompose and relax interface values at continuum level the convergence analysis of these methods need to be carried out at PDE (continuum) level and therefore is a mathematical and not a numerical analysis problem (see [8] for a discussion).

3 Estimating relaxation paramaters in the general case

As it has been allready observed [11,5] the value of the relaxation paramaters plays a crousial role in the convergence of the various interface relaxation methods. Fine tuning of these parameters can greatly improove the rate of convergence while a bad choice might lead to unacceptable slow convergence or even divergence. They depend on both the eigenvalues of the differential operator and the geometric parameters of the domain. Unfortunately analytic expressions of this dependence are not available and does not seem easy to derive them for general cases. Therefore general and robust heuristic mechanisms that automaticaly compute "good" values for the relaxation paramaters are needed. We next present a framework to adaptively adjust the values of the relaxation paramaters α_i and β_i of the averaging interface relaxation method towards its optimal value. The methodology of the proposed scheme is general enough to treat complicated (possibly non-linear) differential operator and arbitrary domain decompositions in 1,2 or 3 dimensions, has allready be applied with success to the successive over-relaxation (SOR) iterative method [6] and can be readily applied to virtualy any interface relaxation methods.

Consider the particular interface $\Gamma_{i,j} \in R^d, d = 0, 1, 2$ between subdomains Ω_i and Ω_j. The interface relaxation equations on $\Gamma_{i,j}$ are given by

$$u\prime(x) = \beta_{i,j} \left(\frac{\partial u_i^{(2k)}(x)}{\partial \nu} - \frac{\partial u_j^{(2k)}(x)}{\partial \nu} \right) + \frac{\partial u_j^{(2k)}(x)}{\partial \nu} , \quad x \in \Gamma_{i,j} \tag{3}$$

where $\frac{\partial}{\partial \nu}$ represents the outward normal derivative to $\Gamma_{i,j}$ and

$$u(x) = \alpha_{i,j} \left(u_i^{(2k+1)}(x) - u_j^{(2k+1)}(x) \right) + u_j^{(2k+1)}(x) , \quad x \in \Gamma_{i,j}. \tag{4}$$

We are interested in investigating how different values of the relaxation parameters affect the quality of the iterants $u(x)$ and $u\prime(x)$. We can achieve that by computing the derivative of a certain quantity, that reflects the quality of iterants, with respect to these relaxation parameters and use it to update their values. We can consider two such quantities: the 2-norm of the residual $r = \| Du(x) - f(x) \|$ on the interface $\Gamma_{i,j}$ or the size of the Gerr disc associated with that particular interface as considered in the previous section. We should mark that the first choice will not work for interfaces where there are discontinuities and the second one assume that that the iteration matrix M [12] has been obtained.

We first concedrate on the first option Assuming that we are able to compute the sensitivity of the residual with respect to $\alpha_{i,j}$ as its partial derivative $r_\alpha \equiv \frac{\partial r}{\partial \alpha_{i,j}}$ after performing the Neumann (3) relaxation then we can use a

secant method to update the value of the relaxation parameter in the next iteration. More specificaly, if we assume that with the superscripts below we denote the iteration step then we obtain the following simple adaptive formula

$$\alpha_{i,j}^{(k+1)} = \alpha_{i,j}^{(k)} - r_\alpha^{(k)} \frac{\alpha_{i,j}^{(k)} - \alpha_{i,j}^{(k-1)}}{r_\alpha^{(k)} - r_\alpha^{(k-1)}}. \tag{5}$$

while the equivalent formula associated with the Dirichlet (??

$$\beta_{i,j}^{(k+1)} = \beta_{i,j}^{(k)} - r_\beta^{(k)} \frac{\beta_{i,j}^{(k)} - \beta_{i,j}^{(k-1)}}{r_\beta^{(k)} - r_\beta^{(k-1)}}. \tag{6}$$

can be derived similarly. Obviously there is no way to analyticaly calculate r_α since there is no closed form of r, as a function of the α's and β's, avaialble. Nevertheless such function is given in a form of an algorithm while relaxing each interface. Therefore a possible solution is to use *automatic differentiation* [9] to obtain "good" values for the relaxation parameters. In the automatic differentiation framework one apllies a process that generates code which uses lookup tables and mechanicaly apply the chain rule to compute the derivative of a function given in a form of a computer program. Among the various software infrastructures that exploit the automatic differentiation idea we have select the ADIFOR package [2] for our implementation.

To move to our second option we consider the Gerschorin disk associated with the ith interface and try to simultaniously minimize its area and move its center as close to the origin as possible. This can be done by minimizing the sum of the absolute value of the elements in the ith row of the iteration matrix M with respect to α and β.

Thus we can use automatic differention to calculate the Jacobian matrix and solve the associated equations for the optimum values of the relaxation parameters.

References

[1] A. Bamberger, R. Glowinski, and Q.H. Tran. A domain decomposition method for the acoustic wave equation with discontinuous coefficients and grid change. *SIAM J. Numer. Anal.*, 34:???–??, 1997.

[2] C. Bischof, A. Carle, P. Khademi, and Mauer A. ADIFOR 2.0: Automatic differentiation of Fortran 77 programs. *IEEE Computational Science & Engineering*, 3:18–32, 1996.

6

[3] B. Despres. Domain decomposition method and the Helmholtz problem. In L. Halpern G. Cohen and P. Joly, editors, *Mathematical and Numerical Aspects of Wave Propagation Phenomena*, pages 44–52. SIAM, 1991.

[4] T.T. Drashansky. *An agent–based approach to building multidisciplinary problem solving enviroments*. PhD thesis, Purdue University, Computer Science Department, December 1996.

[5] D. Funaro, A. Quarteroni, and P. Zanolli. An iterative procedure with interface relaxation for domain decomposition methods. *SIAM J. Numer. Anal.*, 25(6):1213–1236, 1988.

[6] P. Hovland and M. Heath. Adaptive SOR: A case study in automatic defferentiation of algorithm parameters. Technical Report ANL/MCS-P672-0697, Argonne National Laboratory, 1997.

[7] S. Kim. A parallelizable iterative procecedure for the Helmholtz problem. *Appl. Numer. Math.*, 17:411–??, 1995.

[8] M. Mu. Solving composite problems with interface relaxation. *SIAM J. Sci. Comput.*, 1998 (to appear).

[9] L.B. Rall. *Automatic differentiation: Techniques and applications*, volume 120 of *Lecture Notes in Computer Science*. Springer-Verlag, New York, 1981.

[10] J.R. Rice, P. Tsombanopoulou, E. Vavalis, and D. Yang. Domain decomposition methods for underwater acoustics problems. Technical report, Computer Science Department, Purdue University, W. Lafayette, IN, in preparation.

[11] J.R. Rice, P. Tsompanopoulou, and E.A. Vavalis. Review and performance interface relaxation methods for elliptic pdes. Technical Report CSD-TR-97-004, Purdue University, W. Lafayette. IN, 1997.

[12] J.R. Rice, P. Tsompanopoulou, and E.A. Vavalis. Fine tuning interface relaxation methods for elliptic differential equations. Technical Report CSD-TR-98-017, Purdue University, W. Lafayette. IN, 1998.

[13] J. Xu and J. Zou. Non–overlapping domain decomposition methods. Technical report, Mathematics Department, Pennsylvania State University, University Park, PA, 1996.

[14] D. Yang. A parallel iterative nonovelapping domain decomposition procedure for elliptic problems. *IMA J. Numer. Anal.*, 16:75–91, 1996.

[15] D. Yang. A non-overlapping domain decomposition method for elliptic interface prroblems. Technical Report IMA # 1472, University of Minnesota, 1997.

[16] D. Yang. A parallel domain decomposition algorithm for elliptic problems. *J. Comp. Math.*, 16:141–151, 1998.