Department
of
APPLIED MATHEMATICS

Parallel Function Decomposition Methods
and Numerical Applications

by

Xue-Cheng Tai

Report no. 111 September 1997

UNIVERSITY OF BERGEN
Bergen, Norway

Department of Mathematics
University of Bergen
5007 Bergen
Norway

ISSN 0084-778x

Parallel Function Decomposition Methods and Numerical Applications

by

Xue-Cheng Tai

Report No. 111

September 1997

PARALLEL FUNCTION DECOMPOSITION METHODS AND NUMERICAL APPLICATIONS

XUE-CHENG TAI

Abstract

We consider a convex minimization problem. If the minimization function can be decomposed into a sum of convex functions, several parallel minimization algorithms can be derived. These algorithms are used to get parallel ADI and domain decomposition algorithms for nonlinear partial differential equations. Numerical experiments are presented.

KEYWORDS: Parallel, domain decomposition, splitting, ADI, function decomposition.

1 Introduction

We consider the minimization problem

$$\min_{v \in K} F(v), \quad K \subset B \ . \tag{1}$$

Above, the function F is convex and it is defined in a reflexive Banach space B. The set K is convex and $K \cap B$ is closed in B. The following partial differential equations:

$$a) \begin{cases} -\Delta u = & f \text{ in } \Omega \subset R^d \ , \\ u \geq & \phi \text{ in } \Omega \ , \\ u = & 0 \text{ and } \phi \leq 0 \text{ on } \partial\Omega \ , \end{cases} \qquad b) \begin{cases} - & \nabla \cdot (|\nabla u|^{s-2} \nabla u) = f \text{ in } \Omega \ , \\ & u = 0 \text{ on } \partial\Omega \ . \end{cases} \ . \tag{2}$$

can be written as a minimization problem of form (1).

The essential idea we are going to discuss is that if the function $F(\cdot)$ can be decomposed into the sum of suitable convex functions, then we can have some parallel algorithms for (1). These algorithms not only reduce a large and complicated problem into smaller and simpler sub-problems, and also enable us to use different ways to decompose a complicated constraint into simpler constraints and use parallel processor to solve the decomposed sub-problems. This idea was first discussed in report [8], and was later published in papers [9] [10].

2 Conditions for function decomposition

We assume that

$$F(v) = F_1(v) + F_2(v) + \cdots + F_m(v). \tag{3}$$

In the above, each F_i is a convex functional defined on a Banach space B_i. Moreover, we assume that there are convex subsets K_i such that

$$K = \cap_{i=1}^m K_i, \quad B = \cap_{i=1}^m B_i. \tag{4}$$

Under these conditions, the minimization problem (1) can be regarded as minimizing a separable function under constraint $v_i \in K_i,\ i = 1, 2, \cdots, m$ and $v_1 = v_2 = \cdots = v_m$. We define

$$\begin{array}{rll} X & = \{(v_1, v_2, \cdots, v_m) & \mid\ v_i \in B_i,\ i = 1, 2, \cdots, m\} = \prod_{i=1}^m B_i\ , \\ W & = \{(v, v, \cdots, v) & \mid\ v \in K_i,\ i = 1, 2, \cdots, m\} \\ & = \{(v_1, v_2, \cdots, v_m) & \mid\ v_i \in K_i,\ v_i = v,\ i = 1, 2, \cdots, m\}\ . \end{array} \tag{5}$$

Evidently, W is a convex subset in the diagonal subspace of X, and (1) is equivalent to

$$\min_{(v_1, v_2, \cdots, v_m) \in W} \sum_{i=1}^m F_i(v_i)\ . \tag{6}$$

3 A projection method

We first use projection methods to deal with the constraint $v_1 = v_2 = \cdots = v_m$. We define the functional $F_s : \prod_{i=1}^m B_i \mapsto R$ as

$$F_s(v_1, v_2, \cdots, v_m) = \sum_{i=1}^m F_i(v_i)\ . \tag{7}$$

Let I_W behe indicator function of the set W. Then the minimization problem (6) can be written as

$$\min_{v \in X} (F_s(v) + I_W(v))\ . \tag{8}$$

In case that each F_i is differentiable in B_i, (8) can be solved by

$$0 \in F_s{}'(u) + \partial I_W(u)\ . \tag{9}$$

Here ∂I_W is the subgradient of the function I_W, for the definition see Ekeland and Temam [2, p.20]. If each F_i is convex, then $F_s{}'$ is maximal monotone. Under very weak extra conditions on $F_s{}'$, Lions and Mercier [4] proved that

$$u^{n+1} = (I + \rho F_s{}')^{-1}(I - \rho \partial I_W)(I + \rho \partial I_W)^{-1}(I - \rho F_s{}')u^n \tag{10}$$

converges to the minimizer of (1) for any $\rho > 0$.

The relation between this splitting method and the projection method is due to the following fact, see Gabay [3, p.328]

$$2P_W - I = (I - \rho \partial I_W)(I + \rho \partial I_W)^{-1}\ , \tag{11}$$

where P_W is the projection operator from X to W. Therefore (10) has the form

$$(I + \rho F_s{}')u^{n+1} = (2P_W - I)(I - \rho F_s{}')u^n\ . \tag{12}$$

As shown in Lions and Mercier [4], this method can be used not only for solving elliptic problems, but also for hyperbolic problems.

4 The penalization methods

We use penalization methods to deal with the constraint $v_1 = v_2 = \cdots = v_m$. In order to be able to use parallel methods, we introduce one extra variable v and put the constraints $v_i = v,\ i = 1, 2, \cdots, m$ into the penalization terms, i.e. we use the penalization functional

$$F_r(v, v_1, v_2, \cdots, v_m) = \sum_{i=1}^{m} F_i(v_i) + \frac{r}{2m} \sum_{i=1}^{m} \|v_i - v\|_{V_i}^2 , \tag{13}$$

where the penalization is done in weaker Hilbert spaces $V_i,\ i = 1, 2, \cdots, m$. We define $V = \cap_{i=1}^{m} V_i$. We expect that when $r \to \infty$, the minimizer of F_r over $V \times \prod_{i=1}^{m}(K_i \cap B_i)$ or over $(K \cap V) \times \prod_{i=1}^{m} B_i$ will converge to the minimizer of (1). One algorithm to search for the minimizer for F_r over $V \times \prod_{i=1}^{m}(K_i \cap B_i)$ is:

Algorithm 4.1 *Choose an initial approximation $u^0 \in V$ and a parameter r large enough.*

Step 1. For $n \geq 1$, if u^n is known, find $u_i^n \in K_i \cap B_i$ in parallel for $i = 1, 2, \cdots, m$ such that

$$F_i(u_i^n) + \frac{r}{2m}\|u_i^n - u^n\|_{V_i}^2 \leq F_i(v_i) + \frac{r}{2m}\|v_i - u^n\|_{V_i}^2, \quad \forall v_i \in K_i \cap B_i . \tag{14}$$

Step 2. Find $u^{n+1} \in V$ such that

$$\sum_{i=1}^{m} \|u^{n+1} - u_i^n\|_{V_i}^2 \leq \sum_{i=1}^{m} \|v - u_i^n\|_{V_i}^2, \quad \forall v \in V . \tag{15}$$

If we minimize F_r over $(K \cap V) \times \prod_{i=1}^{m} B_i$, then the constraints K_i from (14) is moved to the projection step (15), we will get a different algorithm:

Algorithm 4.2 . *Choose an initial approximation $u^0 \in V$ and a parameter r large enough.*

Step 1. For $n \geq 1$, if u^n is known, find $u_i^n \in B_i$ in parallel for $i = 1, 2, \cdots, m$ such that

$$F_i(u_i^n) + \frac{r}{2m}\|u_i^n - u^n\|_{V_i}^2 \leq F_i(v_i) + \frac{r}{2m}\|v_i - u^n\|_{V_i}^2, \quad \forall v_i \in B_i . \tag{16}$$

Step 2. Find $u^{n+1} \in K \cap V$ such that

$$\sum_{i=1}^{m} \|u^{n+1} - u_i^n\|_{V_i}^2 \leq \sum_{i=1}^{m} \|v - u_i^n\|_{V_i}^2, \quad \forall v \in K \cap V . \tag{17}$$

5 The augmented Lagrangian methods

The Augmented Lagrangian methods combine the multipliers methods with the penalization methods. Compared with the penalization methods, the accuracy of the augmented Lagrangian method is not restricted by the penalization parameter. Define the augmented Lagrangian functional

$$L_r(v, v_i, \mu_i) = \sum_{i=1}^{m} F_i(v_i) + \frac{1}{m}\sum_{i=1}^{m}(\mu_i, v_i - v)_{V_i} + \frac{r}{2m}\sum_{i=1}^{m} \|v_i - v\|_{V_i}^2 \tag{18}$$

and we try to seek a saddle point for L_r. The saddle point (u, u_i, λ_i) of L_r over $(K \cap V) \times \prod_{i=1}^{m} B_i \times \prod_{i=1}^{m} V_i$ satisfies

$$L_r(u, u_i, \mu_i) \leq L_r(u, u_i, \lambda_i) \leq L_r(v, v_i, \lambda_i), \quad \forall v \in K \cap V, v_i \in B_i, \mu_i \in V_i , \tag{19}$$

The following algorithm can be used to search a saddle point for L_r over $(K \cap V) \times \prod_{i=1}^{m} B_i \times \prod_{i=1}^{m} V_i$.

Algorithm 5.1 *Choose initial values* $u_i^0 \in B_i$ *and* $\lambda_i^0 \in V_i$, $i = 1, 2, \cdots, m$.

Step 1. For $n \geq 1$, *find* $u^n \in K \cap V$ *by solving*

$$r \sum_{i=1}^{m} (u^n - u_i^{n-1}, v)_{V_i} - \sum_{i=1}^{m} (\lambda_i^{n-1}, v)_{V_i} = 0 , \quad \forall v \in K \cap V , \tag{20}$$

Step 2. find $u_i^n \in B_i$ *in parallel for* $i = 1, 2, \cdots, m$ *such that*

$$\begin{aligned} & F_i(u_i^n) + \tfrac{1}{m}(\lambda_i^{n-1}, u_i^n)_{V_i} + \tfrac{r}{2m}\|u_i^n - u^n\|_{V_i}^2 \\ \leq \ & F_i(v_i) + \tfrac{1}{m}(\lambda_i^{n-1}, v_i)_{V_i} + \tfrac{r}{2m}\|v_i - u^n\|_{V_i}^2 , \quad \forall v_i \in B_i. \end{aligned} \tag{21}$$

Step 3. Set $\lambda_i^n = \lambda_i^{n-1} + r(u_i^n - u^n)$ *and go to the next iteration.*

We can also search a saddle point of L_r over $V \times \prod_{i=1}^m (K_i \cap B_i) \times \prod_{i=1}^m V_i$ and get a similar algorithm to Algorithm 4.2.

6 Applications to splitting methods

A parallel splitting method from the penalization method

In papers by Lu, Neittaanmäki and Tai [5], and Bensoussan, Lions and Temam [1], some parallel splitting methods were studied. In fact, they coincide with the parallel penalization method when applied to elliptic problems that can be regarded as minimization problems.

We consider an elliptic problem (linear or nonlinear) $Au = 0$ and we assume that this equation is derived from the minimization problem (1). We assume that A is the differential of F in a Hilbert space H. If F can be split as (3) and each F_i has a differential A_i in the Hilbert space H, then we need to solve $\sum_{i=1}^m A_i u = 0$. We assume that Dom(F) and Dom(F_i) are Hilbert spaces and Dom$(F) = \cap_{i=1}^m$Dom(F_i). If we take $K = V =$ Dom(F), $K_i =$ Dom(F_i), $V_i = H$, and use Algorithm 4.1, we get

Algorithm 6.1 *Choose an initial value* $u^0 \in H$ *and a parameter* r *large enough.*

Step 1. For $n \geq 1$, *find* $u_i^n \in D(F_i)$ *from the following problem in parallel for* $i = 1, 2, \cdots, m$:

$$\min_{v_i \in D(F_i)} \left(F_i(v_i) + \frac{r}{2m}\|v_i - u^n\|_{V_i}^2 \right) . \tag{22}$$

This is equivalent to finding $u_i^n \in$ *Dom*(F_i) *such that*

$$\frac{r}{m}(u_i^n - u^n) + A_i u_i^n = 0 . \tag{23}$$

Step 2. Set u^{n+1} *as in (24) and go to the next iteration.*

$$u^{n+1} = \frac{1}{m}\sum_{i=1}^{m} u_i^n. \tag{24}$$

By defining $\tau = \frac{1}{r}$, we can see that this is exactly the algorithm studied in Lu, Neittaanmäki and Tai [5]. The convergence is proved in [5] under the assumption that each A_i is coercive.

The parallel splitting does not mean that we can only use m processors. In the dimensional splitting case, each subproblem is again a series of independent one dimensional problems, see Tai and Neittaanmäki [6], and they can be computed again by parallel processors.

The alternating direction method and the local one dimensional method

In this section, we will show that a small change in the penalization functional of the last section will give us the local one dimensional method. Moreover such a small change also turns the splitting method from a parallel one to a sequential one, which means the fractional steps are not independent, but must be solved one after another.

Instead of using penalization functional (13), we put $v_{i-1} = v_i$ as penalization terms into the cost functional, i.e. we define the penalization functional as

$$\sum_{i=1}^{m} F_i(v_i) + \frac{r}{2m} \sum_{i=2}^{m} \|v_{i-1} - v_i\|_{V_i}^2 \ . \tag{25}$$

If we use the Gauss–Seidel method to minimize this function, we get

Algorithm 6.2 *Choose initial values* $u^0 \in D(F)$.

Step 1. For $n \geq 0$, *set* $u^{n+1} = u_m^n$, $u_0^n = u^{n-1}$ *and find* $u_i^n \in Dom(F_i)$ *sequentially for* $i = 1, 2, \cdots, m$ *by solving*

$$\frac{r}{m}(u_i^n - u_{i-1}^n) + A_i u_i^n = 0 \ . \tag{26}$$

If we take $\tau = \frac{m}{r}$, this is the well-known local one dimensional method, see Yanenko [11].

Augmented parallel splitting methods for variational inequalities

As one example of application, we will use Algorithm 5.1 to solve the obstacle problem (2.a). We split f as $f = \sum_{i=1}^{d} f_i$ and define F, F_i, V and V_i as

$$\begin{array}{ll} F(v) = \int_\Omega \left(\frac{1}{2}|\nabla v|^2 - fv\right) dx \ , & F_i(v) = \int_\Omega \left(\frac{1}{2}|D_i v|^2 - f_i v\right) dx \ , \\ V = H_0^1(\Omega), & V_i = \{v| \quad v, \ D_i v \in L^2(\Omega) \ , \ v|_{\partial\Omega} = 0\} \ . \end{array} \tag{27}$$

Let us take

$$K = \{v| \quad v \in L^2(\Omega), \ v \geq \phi \text{ a.e. in } \Omega\}, \quad K_i = K, \ \forall i. \tag{28}$$

If we use Algorithm 5.1, it gives

Algorithm 6.3 *Choose initial values* $\lambda_i^0 \in L^2(\Omega)$, $u_i^0 \in V_i$ *for* $i = 1, 2, \cdots, d$.

Step 1. For $n \geq 1$, *set*

$$u^n = \max(\phi, \frac{1}{d} \sum_{i=1}^{d} u_i^{n-1} + \frac{1}{rd} \sum_{i=1}^{d} \lambda_i^{n-1}) \ . \tag{29}$$

Step 2. Find $u_i^n \in V_i$ *in parallel for* $i = 1, 2, \cdots, d$ *such that*

$$\frac{r}{d}(u_i^n - u^n) - D_i^2 u_i^n = f_i - \frac{1}{d}\lambda_i^{n-1} \ . \tag{30}$$

Step 3. Set $\lambda_i^n = \lambda_i^{n-1} + r(u_i^n - u^n)$ *go to the next iteration.*

Above, step 1 is the projection from $L^2(\Omega)$ to the constraint set K. The operator "max" is in the distribution sense. In step 2, (30) is an independent two point boundary problem with a homogeneous Dirichlet boundary condition in every line in the x_i–direction. Each one dimensional problem is as simple as a Laplace equation. They can be solved by parallel processors, see Tai [7]. In Figure 1, we show a computational result at iteration 10 for an obstacle problem with an analytical solution. Zero initial values are used and $r = 10$. The average convergence rate for this test is 0.7.

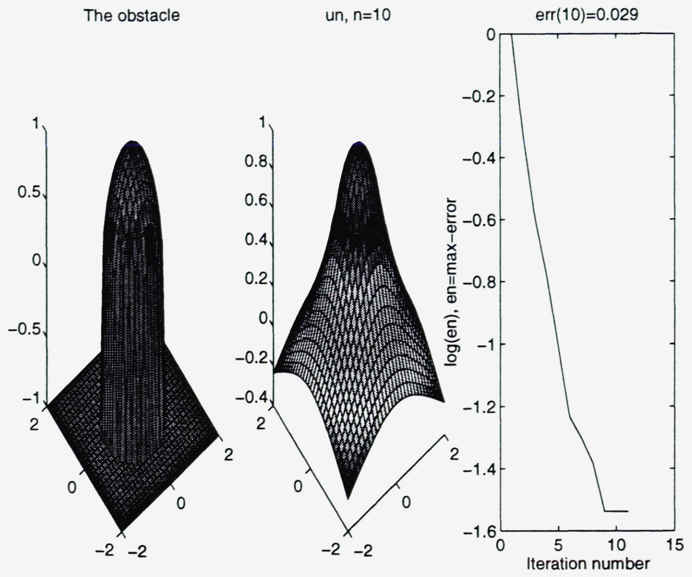

Figure 1: The computed solution by the augmented splitting method

7 Applications to domain decomposition methods

We use domain decomposition to solve the nonlinear problem (2.b). We divide the domain into nonoverlapping subdomains $\Omega_i,\ i = 1, 2, \cdots, m$ and define

$$
\begin{array}{ll}
F(v) = \int_\Omega \left(\frac{|\nabla v|^s}{s} - fv \right) dx, & F_i(v) = \int_{\Omega_i} \left(\frac{|\nabla v|^s}{s} - fv \right) dx, \\
B = \{v| \quad v \in W^{1,s}(\Omega_i),\ \forall i,\ v = 0 \text{ on } \partial\Omega\}, & K = \left\{v| \quad [v]_{\partial\Omega_i \cap \partial\Omega_j} = 0, \quad \forall i, j \right\}, \\
B_i = \{v| \quad v \in W^{1,s}(\Omega_i), \quad v = 0 \text{ on } \partial\Omega\} & V_i = H^1(\Omega_i) \cap \{v|\ v = 0 \text{ on } \partial\Omega\}.
\end{array}
$$

In the above, space B contains functions that are piecewise $W^{1,s}$ and so the functions may have jump along the interfaces between the subdomains. Set K contains functions that have traces on the subdomain interfaces and moreover the jumps $[v]$ should be zero. Thus $B \cap K = W_0^{1,s}(\Omega)$. It is well known that problem (1) is equivalent to (2.b) with F, B and K defined as in the above. The functional F_i are defined over B_i. Moreover, $F(v) = \sum_{i=1}^m F_i(v)$. By applying Algorithm 5.1 to this decomposition, we obtain:

Algorithm 7.1 *Choose $\lambda_i^0 \in V$, $u_i^0 \in B_i$ and $r > 0$.*

Step 1. Solve $u^n \in H_0^1(\Omega)$ from

$$(u^n, v)_{H_0^1(\Omega)} = \sum_{i=1}^m (u_i^n, v)_{H^1(\Omega_i)} + \frac{1}{r} \sum_{i=1}^m (\lambda_i^n, v)_{H^1(\Omega_i)}, \quad \forall v \in H_0^1(\Omega).$$

Step 2. Find $u_i^n \in B_i$ in parallel in each subdomain such that

$$u_i^n = arg \min_{v_i \in B_i} \left(F_i(v_i) + r\|v_i - u^n\|_{H^1(\Omega_i)}^2 + (\lambda_i^n, v_i)_{H^1(\Omega_i)} \right)$$

Step 3. Set $\lambda_i^n = \lambda_i^{n-1} + r(u_i^n - u^n)$ and go to the next iteration.

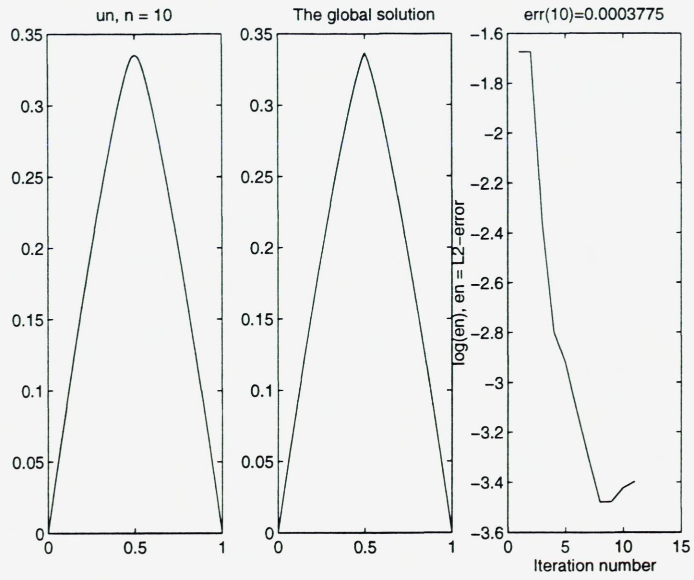

Figure 2: The computed solution by domain decomposition.

Numerical tests for some 1d problems have been done. We choose $f(x) = 1$, $s = 5$ and divide the domain $\Omega = [0, 1]$ into 10 subdomains. Each subdomain contains 10 elements. Linear finite element function spaces are used to approximation the solutions. Zero initial values are being used and $r = 1$. The computed solution at iteration 10 by our domain decomposition algorithm is given in Figure 2.a. The global finite element solution is shown in Figure 2.b. In Figure 2.c, we show the convergence. The computed solution converges to the true solution in about 7 iterations.

References

[1] A. Bensoussan J. L. Lions and R. Temam, Sur les méthodes de décomposition de décentralisation et de coordination et application *Sur les méthodes numériques en sciences physiques et économiques,* J. L. Lions and J. I. Marchuk eds, Dunod-Bordas Paris, 1974. 133-257.

[2] I. Ekeland and R. Temam, *Convex analysis and variational problems*, North–Holland, Amsterdam, 1976.

[3] M. Fortin and R. Glowinski (eds), *Augmented Lagrangian methods: Applications to the numerical solution of boundary value problems*, North–Holland, Amsterdam, 1983.

[4] P. Lions and B. Mercier, Splitting algorithms for the sum of two nonlinear operators, *SIAM J. Numer. Anal.*, **16**, 964–979, 1979.

[5] T. Lu, P. Neittaanmäki and X-C. Tai, Parallel splitting methods for partial differential equations and applications to Navier–Stokes equations, *RAIRO Numer.. Math. and Math. Model.*, **26**, 673–708, 1992.

[6] X.-C. Tai and P. Neittaanmäki, Parallel finite element splitting–up method for parabolic problems, *Numer. Method for Part. Diff. Equat.*, **7**, (1991), 209–225.

[7] X.-C. Tai, Global extrapolation with a parallel splitting method, *Numer. Algorithms* **3**, 1992, 427–440.

[8] X.-C. Tai, Parallel function decomposition and space decomposition methods with application to optimization, splitting and domain decomposition *Preprint no. 231-1992, Institut für Mathematik, Technische Universität Graz*, September, 1992.

[9] X.- C. Tai, Parallel function and space decomposition methods *Finite element methods, Fifty years of the Courant element*, Lecture notes in pure and applied mathematics, vol. 164, P. Neittaanmäki ed, 421–432, 1994, Marcel Dekker Inc..

[10] X.-C. Tai, Parallel function decomposition and space decomposition methods: Part I. Function decomposition, *Beijing Mathematics*, **2, part 2**, (1995), 104–134.

[11] N. N. Yanenko, *The methods of fractional steps*, Springer–Verlag, Berlin, 1971.

Xue-Cheng Tai
University of Bergen, Department of Mathematics,
Johanes Brunsgate 12, 5007, Bergen, Norway.
Tai@mi.uib.no and http://www.mi.uib.no/~tai