Journal of Computational Mathematics
Vol.36, No.6, 2018, 881–902.

http://www.global-sci.org/jcm
doi:10.4208/jcm.1706-m2016-0721

A GENERAL TWO-LEVEL SUBSPACE METHOD FOR NONLINEAR OPTIMIZATION*

Cheng Chen
University of Chinese Academy of Sciences, Beijing 100190, China
LSEC, ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences
Beijing, China
Email: cchen@lsec.cc.ac.cn
Zaiwen Wen
Beijing International Center for Mathematical Research, Peking University, Beijing 100871, China
Email: wenzw@pku.edu.cn
Yaxiang Yuan
LSEC, ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences,
Beijing 100190, China
Email: yyx@lsec.cc.ac.cn

Abstract

A new two-level subspace method is proposed for solving the general unconstrained minimization formulations discretized from infinite-dimensional optimization problems. At each iteration, the algorithm executes either a direct step on the current level or a coarse subspace correction step. In the coarse subspace correction step, we augment the traditional coarse grid space by a two-dimensional subspace spanned by the coordinate direction and the gradient direction at the current point. Global convergence is proved and convergence rate is studied under some mild conditions on the discretized functions. Preliminary numerical experiments on a few variational problems show that our two-level subspace method is promising.

Mathematics subject classification: 65N06, 65B99.
Key words: Nonlinear optimization, Convex and nonconvex problems, Subspace technique, Multigrid/multilevel method, Large-scale problems.

1. Introduction

Consider an infinite-dimensional minimization problem

$$\min_{u\in\mathcal{V}} \ \mathcal{F}(u), \tag{1.1}$$

where $\mathcal{F}$ is a mapping from $\mathcal{V}$ to $\mathbb{R}$ and $\mathcal{V}$ is the infinite-dimensional space where u lives in. Infinite-dimensional optimization problems are a major source of large-scale finite dimensional optimization problems, such as partial differential equations (PDEs) and optimal control problems governed by PDEs. Since it is very hard or almost impossible to obtain explicit solutions for these problems, they are usually solved numerically either by a "discretize-then-optimize" strategy or an "optimize-then-discretize" strategy. For these kind of problems, usually a very

* Received August 17, 2016 / Revised version received March 14, 2017 / Accepted June 28, 2017 /
Published online August 7, 2018 /

fine discretization is needed to obtain a satisfactory discretization error, but the computational cost is much expensive. In this paper, we follow the second strategy and propose a new numerical scheme to solve them.

Quite a few numerical optimization methods for large-scale problems have been developed using a fundamental technique named subspace optimization directly or indirectly. It attracts more and more attention in recent years [1–3]. The conjugate gradient method arose originally in [4] to solve linear systems and were introduced in nonlinear minimization in [5]. It defines a new search direction by a given linear combination of the negative gradient direction and the previous search direction. Yuan and Stoer [6] viewed the conjugate gradient method from the subspace point of view, namely, to find a best trial direction, even an approximate minimum, in the 2-dimensional subspace spanned by the two conjugate directions. Another popular method in nonlinear programming is the limited-memory quasi-Newton method proposed by Shanno [7] and Nocedal [8]. It generates the quasi-Newton matrix by using some historical information. The block coordinate descent (BCD) and the alternating direction method of multipliers (ADM-M) are de facto subspace techniques. More general subspace methods and latest developments are referred to [3, 9–11].

Although existing optimization methods can be applied to solve problem (1.1), they make little use of its underlying hierarchical structure. In contrast, multigrid/multilevel method is a more natural concept. It was originally proposed for solving linear elliptic partial differential equations with simple boundary value and proved to work well [12–15]. It takes advantage of different levels discretization of infinite-dimensional problems to execute the coarse grid corrections recursively with a combination of smoothing steps on fine grid. It not only reduces the computational cost but also accelerates the convergence rate. It is well-known that good performance of iterative methods may depend on a good initial guess. The mesh refinement, or full multigrid method [14, 15], uses the nested iteration idea to solve fine grid problems with an initial point interpolated from the solution of the next coarser grid. Multigrid methods were also extended to solve nonlinear PDE problems. One approach is called Newton-MG method [14–16], in which a linear expansion at the current iterate is used in outer iterations and multigrid methods were used for Jacobian systems in inner iterations. Another extension is full approximation scheme (FAS) [15–17], in which the multigrid methodology is directly applied to the original system of nonlinear equations and its corresponding system of nonlinear residual equations. It obtains a full approximation rather than an error correction term in coarse grid problems. A combination of Newton-MG and FAS was proposed by Yavneh and Dardyk [18]. The other extension is projection multilevel method [19–21], which regards a series of discretization spaces as projections from the infinite-dimensional space, and represents them with nodal or finite element. Taking projections onto various subspaces, it solves the problems by correcting the current iterate.

Multigrid method has also be applied to infinite-dimensional optimization problems, especially optimal control problems governed by PDEs [22–25]. It is used for solving the KKT systems derived from optimality conditions and inner loops of optimization scheme derived from original problems. An approach was proposed by Nash [26] and developed in [27–30] for solving the unconstrained convex infinite-dimensional optimization problems, in which a linear term is added in the discretized nonlinear problems at each level other than the finest one to enforce first order coherence in the neighborhood of current iterate between the neighboring levels of grid. This is a new reinterpretation of multigrid from an optimization point of view and uses it as outer iterative scheme [25]. Based on this scheme, Wen and Goldfarb [31] proposed a method using a line search approach by adding an additional condition to keep the coarser step to be a descent direction at the current level. Gratton, Sartenaer and Toint [32,33] proposed a recursive trust region method using trust region technique in the multilevel scheme recursively. Both of these methods converge for nonconvex problems and especially the latter one can deal with box-constraints. Some numerical results can be found in [34]. Ziems and Ulbrich [35] coupled this method with adaptive mesh refinement and established rigorous posteriori error estimators in multilevel scheme. In Frandi and Papini [36,37], this approach is also used for solving non-differentiable problems. Especially, for convex infinite-dimensional problems, Xu, Tai and their collaborators in [38,39] developed a subspace correction framework and proved the convergence rate. The multigrid method is included as a specical case. This framework has also been extended to solve constrained minimizations [39,40] and applied in image processing [41]. It also makes the concept of multigrid to be implemented in parallel with theoretical convergent guarantee, taking advantages of modern high performance computing hardware structures. A review of the recent developments is referred to [25].

In this paper, we propose a new two-level subspace optimization method by combining the idea of multigrid optimization and subspace technique. Our algorithm executes either a direct step at the current level or a coarse level correction step when the coarser gird satisfies some certain criteria, while the coarse level construction is different from previous multigrid optimization methods. Taking advantage of the subspace framework of Yuan and Stoer's [6], we augment the traditional coarse grid space by a two-dimensional subspace spanned by the coordinate direction and the gradient direction of the current point in the construction of coarse space correction step and try to find an approximate minimum in this new subspace. For coarse level correction step, we follow the "optimize-then-discretize" strategy to derive the formulation of infinite-dimensional coarse space sub-problem first and solve its discretized version. Without adding the linear term as in Nash's scheme [26], our coarse subspace model keeps the zeroth-order coherence with fine level discretization of the problem, which guarantees the algorithm to be a monotone descent one with coarse subspace correction step. We also prove the global convergence and convergence rate which is at least R-linear for strongly convex case and $O(1/\epsilon^2)$ for nonconvex case. Numerical experiments show that our two-level subspace method performs comparably efficiently in solving very large-scale dimensional problems.

This paper is organized as follows. In Section 2, we give the problem statement and a detailed description of our new two-level subspace method. The global convergence, R-linear convergence rate for strongly convex case and $O(1/\epsilon^2)$ for nonconvex case are proved in Section 3. Numerical experiments are showed in Section 4. Finally, some conclusion and future works are given in the last section.

2. A New Two-level Subspace Method

2.1. Problem statement and notations

We first introduce the hierarchical properties of our discretization. For the levels $\ell = N_0, N_0+1, \cdots, N$, there exists a set of nested finite dimensional girds spaces $\mathcal{V}_{N_0} \subset \mathcal{V}_{N_0+1} \subset \cdots \subset \mathcal{V}_N \subset \mathcal{V}$. For every $\mathcal{V}_\ell$, let $\Phi_\ell = \{\phi_\ell^{(j)}\}_{j=1}^{n_\ell}$ be a suitable basis of $\mathcal{V}_\ell$, where n_ℓ is the dimension of $\mathcal{V}_\ell$. The discretization of u at level ℓ is denoted as

$$u_\ell = \Phi_\ell x_\ell = \sum_{j=1}^{n_\ell} x_\ell^{(j)} \phi_\ell^{(j)}, \tag{2.1}$$

where $x_\ell = (x_\ell^{(1)}, \dots, x_\ell^{(n_\ell)})^\top \in \mathbb{R}^{n_\ell}$. Letting $f_\ell(x_\ell) = \mathcal{F}(u_\ell)$, the discretized version of problem (1.1) becomes

$$\min_{x_\ell \in \mathbb{R}^{n_\ell}} f_\ell(x_\ell). \tag{2.2}$$

Let $\mathcal{DF}(u)$ be the gradient of $\mathcal{F}(u)$. We can also discretize $\mathcal{DF}(u_\ell)$ at level ℓ in the same discretized space $\mathcal{V}_\ell$ as

$$\mathcal{D}_\ell\mathcal{F}(u_\ell) = \Phi_\ell z_\ell = \sum_{j=1}^{n_\ell} z_\ell^{(j)}\phi_\ell^{(j)}, \tag{2.3}$$

where $z_\ell = (z_\ell^{(1)}, \cdots, z_\ell^{(n_\ell)})^\top$ satisfies the weak formulation

$$\langle \mathcal{D}_\ell\mathcal{F}(u_\ell), v_\ell\rangle = \langle \mathcal{DF}(u_\ell), v_\ell\rangle, \quad \forall v_\ell \in \mathcal{V}_\ell, \tag{2.4}$$

where $\langle u, v\rangle$ is a given inner product on $\mathcal{V}$. By letting $v_\ell = \phi_\ell^{(1)}, \dots, \phi_\ell^{(n_\ell)}$, we obtain a system of n_ℓ equations as

$$\left\langle \mathcal{D}_\ell\mathcal{F}(u_\ell), \phi_\ell^{(i)}\right\rangle = \left\langle \mathcal{DF}(u_\ell), \phi_\ell^{(i)}\right\rangle, \quad i = 1, \dots, n_\ell. \tag{2.5}$$

Consequently, the gradient of the discretized function $f_\ell(x_\ell)$ at level ℓ and the discretized gradient $\mathcal{D}_\ell\mathcal{F}(u_\ell)$ can be related as

$$M_\ell z_\ell \;=\; \nabla f_\ell(x_\ell), \tag{2.6}$$

where M_ℓ is the mass matrix at the level ℓ, whose (i,j) element is $\left\langle \phi_\ell^{(j)}, \phi_\ell^{(i)}\right\rangle$. In fact, it follows from (2.5) that we only need to prove

$$\left\langle \mathcal{DF}(u_\ell), \phi_\ell^{(i)}\right\rangle = \;(\nabla f_\ell(x_\ell))^{(i)}, \quad \text{for each} \quad i = 1, \dots, n_\ell.$$

Let $e_\ell^{(i)}$ be a n_ℓ dimensional vector whose ith component is equal to 1 and others are equal to 0. According to the definition of directional derivative, for any i, we have

$$\begin{aligned}(\nabla f_\ell(x_\ell))^{(i)} &= \lim_{t\to 0}\frac{f_\ell(x_\ell + te_\ell^{(i)}) - f_\ell(x_\ell)}{t}\\ &= \lim_{t\to 0}\frac{\mathcal{F}(u_\ell + t\phi_\ell^{(i)}) - \mathcal{F}(u_\ell)}{t}\\ &= \left\langle \mathcal{DF}(u_\ell), \phi_\ell^{(i)}\right\rangle.\end{aligned} \tag{2.7}$$

We further adopt the following notations in this paper. We denote the kth iterate on level ℓ by $x_{\ell,k}$, where the first subscript ℓ denotes the discretization level and the second subscript k denotes the iteration count. If a vector has only one subscript, as for example x_ℓ, the subscript ℓ refers to the level of the multigrid. The component of a vector is marked by superscripts, for example, $x^{(i)}$ means the ith component of the vector x. The same notations are applied to $z_{h,k}$ as well. We also use $f_{\ell,k} \equiv f_\ell(x_{\ell,k})$ and $\nabla f_{\ell,k} \equiv \nabla f_\ell(x_{\ell,k})$. For convenience, we use h to denote a finer level and H to denote the coarser level with respect to h, which stand for any two of $N_0, N_0+1, \cdots, N$, respectively.

2.2. A general two-level subspace optimization framework

Instead of simply finding a point $u_{h,k+1}$ in the coarser grid space $\mathcal{V}_H$, we seek a point $u_{h,k+1}$ in $S_{h,k}+\mathcal{V}_H$, satisfying some conditions, where $S_{h,k}$ is a subspace including descent information, such as the coordinate direction of current iteration and the previous iterations or gradients, in $\mathcal{V}_h$. The choice of $S_{h,k}$ is, of course, not unique. We first consider

$$S_{h,k} = \operatorname{span}\{u_{h,k}, \mathcal{D}_h\mathcal{F}(u_{h,k})\} \subseteq \mathcal{V}_h, \tag{2.8}$$

which consists of the current point and the current gradient. We compute the gradient $\mathcal{D}\mathcal{F}(u_{h,k})$ in the original infinite dimensional space at $u_{h,k}$ for each step and then discretize it as $\mathcal{D}_h\mathcal{F}(u_{h,k})$ in the weak formulation (2.4) in the same finite space $\mathcal{V}_h$.

Since the computational cost of the subspace minimization problem usually is dominate, a small subspace is preferred. In addition, the subspace should be able to provide as much information as possible and the chosen subspace should be helpful for us to make progress on decreasing function values. Based on these two rules, we properly choose the coarsest level H only if it is helpful, then a new solution $u_{h,k+1}$ is obtained by

$$u_{h,k+1} \approx \arg\min \ \mathcal{F}(u), \quad \text{s.t.} \quad u \in S_{h,k} + \mathcal{V}_H. \tag{2.9}$$

A general algorithm framework is outlined in Algorithm 2.1.

Algorithm 2.1. A general two-level subspace optimization framework

```
Choose $h$ to be the finest level $N$. Set $u_{h,0} \in \mathcal{V}_h$, $S_{h,0} = \emptyset$ and $k = 0$.
while stopping conditions not met do
  if $u_{h,k}$ is not optimal at a coarser level $H \in \{N_0, N_0+1, \dots, N-1\}$ then
    compute $u_{h,k+1}$ by solving (2.9).
  else
    find a point $u_{h,k+1} \in \mathcal{V}_h$ on level $h$.
  Set $S_{h,k+1}$ as (2.8).
```

In the next few subsections, we specify the conditions on whether $u_{h,k}$ is optimal, the approaches for choosing the coarse level H and the methods for computing $u_{h,k+1}$.

2.3. Switching conditions for coarse level correction

Since our algorithm works on different levels of grid, we need to give the relationship of points on different levels. Firstly we define the prolongation operator from the coarser level to finer level from our discretization. For any $u_{\ell-1}$ in the next coarser level, we have $u_{\ell-1} = \sum_{i=1}^{n_{\ell-1}} x_{\ell-1}^{(i)}\phi_{\ell-1}^{(i)}$. Since for every $i = 1, \cdots, n_{\ell-1}$, $\phi_{\ell-1}^{(i)}$ also lives in the finer level, we have $\phi_{\ell-1}^{(i)} = \sum_{j=1}^{n_\ell} p_\ell^{(j,i)}\phi_\ell^{(j)}$. Therefore, we have

$$u_{\ell-1} = \sum_{i=1}^{n_{\ell-1}} x_{\ell-1}^{(i)}\Big(\sum_{j=1}^{n_\ell} p_\ell^{(j,i)}\phi_\ell^{(j)}\Big) = \sum_{j=1}^{n_\ell}\Big(\sum_{i=1}^{n_{\ell-1}} p_\ell^{(j,i)}x_{\ell-1}^{(i)}\Big)\phi_\ell^{(j)}. \tag{2.10}$$

From (2.10) we can summarize that

$$x_\ell = P_\ell x_{\ell-1}, \tag{2.11}$$

by letting P_ℓ be the matrix form of $(p_\ell^{(j,i)})$. Then we define R_ℓ to be the restriction operator from the finer level to the coarser level and assume the following relationship between them.

Assumption 2.1. *The prolongation operator P_ℓ and the restriction operator R_ℓ satisfy*

$$P_\ell = \sigma_\ell R_\ell^\top, \tag{2.12}$$

where σ_ℓ is a constant at level ℓ, and we always suppose that $\|R_\ell\| \leq 1$.

We let this assumption hold for every level ℓ in the multilevel case. For a more extensive coverage we refer to see [14, 15].

Similar to the traditional multigrid method, for each iteration step, our algorithm alternates between two kinds of steps, a direct search step, which is generated in the current level, and a coarse subspace correction step, which is generated from the proposed subspace. Therefore, it is important to decide when the algorithm executes a coarse level correction. Here we use

$$\|R_\ell \nabla f_{\ell,k}\| \geq \max\{\kappa_g \|\nabla f_{\ell,k}\|, \epsilon_\ell\}, \tag{2.13}$$

to avoid the discretized gradient $\nabla f_{\ell,k}$ falling into the null space of the coarser grid, where $\kappa_g \leq \min\{1, \min_\ell R_\ell\}$ and $\epsilon_\ell \in (0,1)$ denotes the tolerance for first order optimality condition at the fine grid. It was first used in [42]. The reason is that if $\nabla f_{\ell,k}$ lives in the null space of coarse grid, i.e., $R_\ell \nabla f_{\ell,k} = 0$, it can make little progress from the current iterator $x_{\ell,k}$ to perform a coarse level correction.

2.4. Coarse subspace construction

If a coarse subspace correction step is chosen in Nash's scheme [26], a new function is defined by adding a linear term to the discretized function on coarser level, i.e., $\varphi_H(x_H) = f_H(x_H) - v_H^\top x_H$, where $v_H = \nabla f_{H,0} - R_h \nabla f_{h,k}$. Following this kind of formulation, the first order coherence between the two adjoining levels around the current iterate can be satisfied [26, 31, 32].

Different from this scheme, as what is mentioned in Subsection 2.1, we define a new subspace by augmenting the coarse grid space with some information on fine level. Note that $\mathcal{D}_h(u_{h,k})$ is the gradient of the original functional, which is the best descent direction locally. The addition of $\mathcal{D}_h(u_{h,k})$ in the searching subspace yields a nice descent property. Adding $u_{h,k}$ helps the coarse subspace correction step to keep the zeroth order coherence with the finer level at the iterate $u_{h,k}$ and makes the procedure monotonically decreasing.

For our general framework, we can define

$$T(h) = \max_\ell \left\{ \ell \mid \ell \leq h, \left\| \prod_{j=\ell}^{h} R_j \nabla f_{\ell,k} \right\| < \kappa_g \|\nabla f_{\ell,k}\| \; or \; \left\| \prod_{j=\ell}^{h} R_j \nabla f_{\ell,k} \right\| < \epsilon_\ell \right\}. \tag{2.14}$$

If $T(h) = h$, it means that it can make little progress even at the next coarser level. Hence, a direct step should be chosen at the current iterate. Otherwise, a coarse level correction step is executed at the coarse level $H = T(h)$. After choosing the coarse level H, we specify $u_{h,k+1} = \arg\min_{u \in S_{h,k} + \mathcal{V}_H} \mathcal{F}(u)$. We follow the "optimize-then-discretize" strategy to get $\mathcal{DF}(u_{h,k+1})$ firstly and compute the discretized gradient $z_{\ell,k}$ in (2.5). Then there exists a set of coefficients $x_H^{(i)} (i = 1, \cdots, n_H), t_1, t_2$ such that

$$\begin{aligned} u_{h,k+1} &= \sum_{i=1}^{n_H} x_H^{(i)} \phi_H^{(i)} + t_1 u_{h,k} + t_2 \mathcal{D}_h \mathcal{F}(u_{h,k}) \\ &= \sum_{i=1}^{n_H} x_H^{(i)} \sum_{j=1}^{n_h} P_{H,h}^{(i,j)} \phi_h^{(j)} + t_1 \sum_{i=1}^{n_h} x_{h,k}^{(i)} \phi_h^{(i)} + t_2 \sum_{i=1}^{n_h} z_{h,k}^{(i)} \phi_h^{(i)} \\ &= \Phi_h P_{H,h} x_H + t_1 \|x_{h,k}\| \Phi_h \frac{x_{h,k}}{\|x_{h,k}\|} + t_2 \|z_{h,k}\| \Phi_h \frac{z_{h,k}}{\|z_{h,k}\|} \\ &= \Phi_h \tilde{P}_{H,h} \tilde{x}_H, \end{aligned} \tag{2.15}$$

where

$$\tilde{P}_{H,h} = [P_{H,h}, \frac{x_{h,k}}{\|x_{h,k}\|}, \frac{z_{h,k}}{\|z_{h,k}\|}] \tag{2.16}$$

is a new prolongation matrix and $\tilde{x}_H = (x_H^\top, t_1', t_2')^\top$. We can easily derive that $P_{H,h} = \prod_{\ell=H+1}^{h} P_{h+H+1-\ell}$ and $t_1' = t_1 \|x_{h,k}\|$, $t_2' = t_2 \|z_{h,k}\|$, respectively.

According to our construction of the subspace, the objective function at the coarse level H is not simply the discretized function $f_H(x_H)$, but rather

$$\tilde{f}_H(\tilde{x}_H) = f_h(\tilde{P}_{H,h} \tilde{x}_H). \tag{2.17}$$

We choose the first iteration point to be $\tilde{x}_{H,0} = (0, \cdots, 0, \|x_{h,k}\|, 0)^\top$, then $\tilde{P}_{H,h} \tilde{x}_{H,0} = x_{h,k}$. Thus, we have the zeroth order coherence

$$\tilde{f}_H(\tilde{x}_{H,0}) = f_h(x_{h,k}). \tag{2.18}$$

Moreover, we always minimize (2.17) to find $\tilde{x}_H^*$ such that $\tilde{f}_H(\tilde{x}_H^*) \le \tilde{f}_H(\tilde{x}_{H,0})$. After obtaining $\tilde{x}_H^*$, we interpolate it to the finer level by letting $x_{h,k+1} = \tilde{P}_{H,h} \tilde{x}_H^*$. This leads to another zeroth order coherence $\tilde{f}_H(\tilde{x}_H^*) = f_h(x_{h,k+1})$. Gathering the mechanics of coarse correction steps and the zeroth order coherence makes our algorithm monotonically decreasing.

We should mention that since the definition of $\tilde{P}_{H,h}$ depends on the point $x_{h,k}$ at the next finer level h, the objective function $\tilde{f}_H$ is different at different points. We omit this dependence by using $\tilde{f}_H(\cdot)$ to simplify our notation. Actually, this is an extension from optimization perspective of FAS scheme [15], where we update for a full approximation of the new point rather than an error term.

2.5. Direct search step

If condition (2.13) is not satisfied, a direct search step is to execute from $x_{h,k}$. The construction of search steps is based on the basic iterative scheme, such as gradient method, conjugate method, Newton method or quasi-Newton method for just one single level.

A descent direction $d_{h,k}$ can be computed by many unconstrained optimization algorithms which are mentioned before. For proving the global convergence, we firstly choose the descent direction $d_{h,k}$ satisfying

$$\|d_{h,k}\| \le \beta \|\nabla f_{h,k}\| \quad \text{and} \quad -\nabla f_{h,k}^\top d_{h,k} \ge \eta \|\nabla f_{h,k}\|^2, \tag{2.19}$$

where β and η are positive constants.

Then we find a proper step size $\alpha_{h,k}$ along this direction. On one hand, we require the step size $\alpha_{h,k}$ to satisfy the Armijo condition

$$f_h(x_{h,k} + \alpha_{h,k} d_{h,k}) \leq f_{h,k} + \rho \alpha_{h,k} \nabla f_{h,k}^{\top} d_{h,k}, \tag{2.20}$$

where $0 < \rho < \frac{1}{2}$. It helps the the next iterate $x_{h,k+1}$ to generate a sufficient decrease of function value from the current point $x_{h,k}$. On the other hand, $\alpha_{h,k}$ should avoid being a "too-short" step. It is required to satisfy the Wolfe condition

$$\nabla f_{h,k+1}^{\top} d_{h,k} \geq \mu \nabla f_{h,k}^{\top} d_{h,k}, \tag{2.21}$$

where $\rho < \mu < 1$, or the Goldstein condition

$$f_h(x_{h,k} + \alpha_{h,k} d_{h,k}) \geq f_{h,k} + (1-\rho) \alpha_{h,k} \nabla f_{h,k}^{\top} d_{h,k}. \tag{2.22}$$

Alternatively, a backtracking strategy can be used with

$$\alpha = \tau^p \alpha_0, \tag{2.23}$$

where $0 < \tau < 1$, α_0 is the initial step size and p is the smallest integer satisfying condition (2.20).

Finally, we give the Algorithm 2.2 as the detailed version for our two-level subspace algorithm (Algorithm 2.1) below.

Algorithm 2.2. Two-level Subspace Method $x_h^* = TLS(h, x_{h,0})$

```
Choose $h$ to be the finest level $N$, initialize the parameters.
for $k = 1, 2, \ldots$ do
  if $\|\nabla f_{h,k}\| < \epsilon_h$ or $k > K_h$ then
    return solution $x_{h,k}$.
  if $T(h) == h$ then
    Direct Search Computation.
      Compute a descent search direction $d_{h,k}$.
      Find a proper stepsize $\alpha_{h,k}$.
      $x_{h,k+1} = x_{h,k} + \alpha_{h,k} d_{h,k}$.
  else
    Coarse Subspace Correction Computation
      Set $H = T(h)$.
      Construct $\tilde{P}_{H,h} = [P_{H,h}, \frac{x_{h,k}}{\|x_{h,k}\|}, \frac{z_{h,k}}{\|z_{h,k}\|}]$ and $\tilde{f}_H$.
      Compute $\tilde{x}_H^*$ by solving $\min_{\tilde{x}_H} \tilde{f}_H(\tilde{x}_H)$.
      $x_{h,k+1} = \tilde{P}_{H,h} \tilde{x}_H^*$.
```

2.6. Full Multigrid Skill

Since starting from a good initial point may reduce the number of iterations in most occasions, we use the so-called full multigrid skill or mesh refinement technique in our implementation. Firstly we start at the coarsest level $\ell = N_0$ where the problem is easily solved. After getting a minimum x_ℓ^* at the current level, we prolongate the solution to the next finer level $\ell + 1$ with interpolation as an initial point, and then apply Algorithm 2.2 to the discretized problem at this new level to obtain the minimum $x_{\ell+1}^*$. We repeat this process until we reach the finest level and get the final solution. The detailed method is described in Algorithm 2.3.

Algorithm 2.3. Full Two-level Subspace Method

```
Set an initial approximation $x_{N_0,0}$.
for $\ell = N_0, N_0+1, \ldots, N$t do
    Call $x_\ell^* = TLS(\ell, x_{\ell,0})$, i.e., apply Alg. 2.2 to solve the discretized problem
    $\min_{x_\ell} f_\ell(x_\ell)$ on level $\ell$.
    if $\ell < N$ then
        Compute the initial point $x_{\ell+1,0} = P_{\ell+1} x_\ell^*$ on level $\ell + 1$.
```

3. Convergence Analysis

3.1. Assumptions and properties for functions of all levels

In this section, we make the following assumptions.

A.1 (smoothness) $f_N(x_N)$, the discretized function at the uppermost level, is continuously differentiable with Lipschitz gradient, i.e., there exists $0 < L_N < \infty$ such that

$$\|\nabla f_N(x_N) - \nabla f_N(y_N)\| \le L_N \|x_N - y_N\|. \tag{3.1}$$

A.2 (boundedness) $f_N(x_N)$, the discretized function at the uppermost level, has a bounded level set $\{x_N | f_N(x_N) \le f_N(x_{N,0})\}$ for any $x_{N,0}$.

A.3 (convexity) $f_N(x_N)$, the discretized function at the uppermost level, is strongly convex, i.e., there exists $0 < m_N < \infty$ such that

$$f_N(y_N) \ge f_N(x_N) + \nabla f_N(x_N)^\top (y_N - x_N) + \frac{1}{2} m_N \|y_N - x_N\|^2. \tag{3.2}$$

From the assumptions above, we can prove the smoothness, boundedness and convexity of discretized functions at all levels in the following three lemmas.

Lemma 3.1. *Suppose that assumption* A.1 *holds. Let $f_\ell(x_\ell)$ be the discretized function at the level ℓ other than the uppermost level. Then $f_\ell(x_\ell)$ is continuously differentiable and its gradient is Lipschitz continuous, i.e., there exists $0 < L_\ell < \infty$ such that*

$$\|\nabla f_\ell(x_\ell) - \nabla f_\ell(y_\ell)\| \le L_\ell \|x_\ell - y_\ell\|, \tag{3.3}$$

where $L_\ell \le (\prod_{i=\ell+1}^N \sigma_i)^2 L_N$.

Proof. From the definition of $f_\ell(\cdot)$, we have

$$f_\ell(x_\ell) = \mathcal{F}(\Phi_\ell x_\ell) = \mathcal{F}(\Phi_N P_{\ell,N} x_\ell) = f_N(P_{\ell,N} x_\ell).$$

So $f_\ell(\cdot)$ is continuous differentiable. For any x_ℓ and y_ℓ, we have

$$\begin{aligned}
&\|\nabla f_\ell(x_\ell) - \nabla f_\ell(y_\ell)\| = \|P_{\ell,N}^\top (\nabla f_N(P_{\ell,N} x_\ell) - \nabla f_N(y_\ell))\| \\
\le& \|P_{\ell,N}\| L_N \|P_{\ell,N}(x_\ell - y_\ell)\| \le \|P_{\ell,N}\|^2 L_N \|(x_\ell - y_\ell)\| \\
=& \left\| \prod_{i=\ell+1}^{N} P_{N+\ell+1-i} \|^2 L_N \|(x_\ell - y_\ell) \right\| \le \left(\prod_{i=\ell+1}^{N} \sigma_i \right)^2 L_N \|x_\ell - y_\ell\|,
\end{aligned}$$

where the last inequality is derived from Assumption 2.1. This completes the proof. □

Lemma 3.2. *Suppose that assumption* A.2 *holds. Let $f_\ell(x_\ell)$ be the discretized function at the level ℓ other than the uppermost level. Then $f_\ell(x_\ell)$ also has the bounded level set $\{x_\ell | f_\ell(x_\ell) \leq f_\ell(x_{\ell,0})\}$ for any $x_{\ell,0}$.*

Proof. From the proof of the former lemma, for any level ℓ, we have $f_\ell(x_\ell) = f_N(P_{\ell,N}x_\ell)$. Assume that there exists $x_{\ell,0}$ such that the level set $\{x_\ell | f_\ell(x_\ell) \leq f_\ell(x_{\ell,0})\}$ is unbounded. It means that $\{P_{\ell,N}x_\ell | f_N(P_{\ell,N}x_\ell) \leq f_N(P_{\ell,N}x_{\ell,0})\}$ is unbounded. This contradicts with the fact that $\{x_N | f_N(x_N) \leq f_N(P_{\ell,N}x_{\ell,0})\}$ is bounded. □

Lemma 3.3. *Suppose that assumption* A.3 *holds. Let $f_\ell(x_\ell)$ be the discretized function at the level ℓ other than the uppermost level. Then $f_\ell(x_\ell)$ is strongly convex, i.e., there exists $0 < m_\ell < \infty$ such that*

$$f_\ell(y_\ell) \geq f_\ell(x_\ell) + \nabla f_\ell(x_\ell)^\top (y_\ell - x_\ell) + \frac{1}{2} m_\ell \|y_\ell - x_\ell\|^2. \tag{3.4}$$

Proof. Since P_ℓ is column full rank for any ℓ, $P_{\ell,N} = \prod_{i=\ell+1}^{N} P_{N+\ell+1-i}$ is also column full rank and $P_{\ell,N}^\top P_{\ell,N}$ is positive definite. Let λ_{min}^P be the minimal eigenvalue of the latter matrix.

According to the strong convexity of $f_N(\cdot)$, for any y_ℓ, we have

$$\begin{aligned} f_\ell(y_\ell) &= f_N(P_{\ell,N}y_\ell) \\ &\geq f_N(x_N) + \nabla f_N(x_N)^\top (P_{\ell,N}y_\ell - x_N) + \frac{1}{2} m_N \|P_{\ell,N}y_\ell - x_N\|^2. \end{aligned}$$

Letting $x_N = P_{\ell,N}x_\ell$, we have

$$\begin{aligned} f_\ell(y_\ell) &\geq f_N(P_{\ell,N}x_\ell) + \nabla f_N(P_{\ell,N}x_\ell)^\top (P_{\ell,N}(y_\ell - x_\ell)) + \frac{1}{2} m_N \|P_{\ell,N}(y_\ell - x_\ell)\|^2 \\ &= f_\ell(x_\ell) + \nabla f_\ell(x_\ell)^\top (y_\ell - x_\ell) + \frac{1}{2} m_N (y_\ell - x_\ell)^\top (P_{\ell,N}^\top P_{\ell,N})(y_\ell - x_\ell) \\ &\geq f_\ell(x_\ell) + \nabla f_\ell(x_\ell)^\top (y_\ell - x_\ell) + \frac{1}{2} m_N \lambda_{min}^P \|y_\ell - x_\ell\|^2. \end{aligned}$$

This completes the proof. □

Based on the lemmas above, furthermore, we let $L = \max_\ell\ L_\ell$ and $m = \min_\ell\ m_\ell$. In the following lemma, we show that the step size generated by any of the three kinds of line search methods has a lower bound.

Lemma 3.4. *Suppose that assumption* A.1 *holds, $d_{\ell,k}$ is a descent direction at $x_{\ell,k}$ and Armijo condition* (2.20) *is satisfied. The step size $\alpha_{\ell,k}$ has a lower bound for any kind of line search method referred in Section* 2*, specifically,* $\min\{\alpha_0, \frac{2\tau(\rho-1)d_{\ell,k}^\top \nabla f_{\ell,k}}{L_\ell \|d_{\ell,k}\|^2}\}$ *for the backtracking strategy,* $\frac{-2\rho d_{\ell,k}^\top \nabla f_{\ell,k}}{L_\ell \|d_{\ell,k}\|^2}$ *for Goldstein condition and* $\frac{(\mu-1)d_{\ell,k}^\top \nabla f_{\ell,k}}{L_\ell \|d_{\ell,k}\|^2}$ *for Wolfe condition.*

Proof. 1. For backtracking strategy (2.23), we refer to see [31].

2. For Goldstein condition (2.22), since z_ℓ is Lipschitz continuous, it follows from Taylor's theorem (Theorem 1.2.22 in [43]) that

$$f_\ell(x_{\ell,k} + \alpha d_{\ell,k}) \leq f_{\ell,k} + \alpha \nabla f_{\ell,k}^\top d_{\ell,k} + \frac{1}{2} L_\ell \alpha^2 \|d_{\ell,k}\|^2.$$

Combining with (2.22), we can derive that $\alpha \geq \frac{-2\rho \nabla f_{\ell,k}^\top d_{\ell,k}}{L_\ell \|d_{\ell,k}\|^2}$.

3. For Wolfe condition (2.21), we subtract a term $\nabla f_{\ell,k}^\top d_{\ell,k}$ to the both sides of (2.21) and get

$$L_\ell \alpha \|d_{\ell,k}\|^2 \geq (\nabla f_{\ell,k+1} - \nabla f_{\ell,k})^\top d_{\ell,k} \geq (\mu - 1)\nabla f_{\ell,k}^\top d_{\ell,k},$$

where the first inequality comes from the Lipschitz continuous gradient of f_ℓ. This implies the lower bound of Wolfe condition. □

From this Lemma, we can always have a lower bound of step size $\bar{\alpha}_\ell$ as

$$\bar{\alpha}_\ell = \min\{\alpha_0, \frac{2\tau(1-\rho)}{L_\ell}, \frac{2\rho}{L_\ell}, \frac{1-\mu}{L_\ell}\}, \tag{3.5}$$

if the negative gradient direction is used, or

$$\bar{\alpha}_\ell = \min\{\alpha_0, \frac{2\tau(1-\rho)\eta}{L_\ell \beta^2}, \frac{2\rho\eta}{L_\ell \beta^2}, \frac{(1-\mu)\eta}{L_\ell \beta^2}\}, \tag{3.6}$$

if a descent direction other than the negative gradient is used, where β and η are constants in (2.19).

The following lemma gives some properties of strongly convex functions.

Lemma 3.5 (Theorem 5.3.4 in [43]) . *Suppose that assumptions* A.1 *and* A.3 *hold, for all x_ℓ, it holds*

$$\frac{m}{2}\|x_\ell - x_\ell^*\|^2 \leq f_\ell(x_\ell) - f_\ell(x_\ell^*) \leq \frac{1}{2m}\|\nabla f_\ell(x_\ell)\|^2, \tag{3.7}$$

where x_ℓ^* is the unique minimizer of $f_\ell(x_\ell)$ on level ℓ.

3.2. Some properties for coarse subspace correction

In this subsection, we analyze the properties in coarse subspace correction. The following lemma shows that the new constructed prolongation matrix $\tilde{P}_{H,h}$ is bounded.

Lemma 3.6. *The prolongation matrix $\tilde{P}_{H,h}$ has bounded ℓ_2 norm, i.e., there exists an upper bound B such that*

$$\|\tilde{P}_{H,h}\| \leq B. \tag{3.8}$$

Proof. From the relationship of the transfer matrix (2.12) and the definition of $\tilde{P}_{H,h}$ (2.16), we have

$$\|\tilde{P}_{H,h}\|^2 \leq \|P_{H,h}\|^2 + \left\|\frac{x_{h,k}}{\|x_{h,k}\|}\right\|^2 + \left\|\frac{z_{h,k}}{\|z_{h,k}\|}\right\|^2.$$

Then $\|\tilde{P}_{H,h}\| \leq \sqrt{(\prod_{\ell=H+1}^{h} \sigma_\ell)^2 + 2}$. Since there are finite levels of grids, there exists a positive constant B such that $\|\tilde{P}_{H,h}\| \leq B$. □

From this lemma and the fact that ∇f_h is Lipschitz continuous, we can prove that $\nabla \tilde{f}_H$ is also Lipshchitz continuous in the following lemma.

Lemma 3.7. *Suppose assumption* A.1 *hold for f_h. At any point $x_{h,k}$, $\tilde{f}_H$ also has Lipschitz continuous gradient, and the Lipschitz constant $\tilde{L}_H$ satisfies*

$$\tilde{L}_H \leq B^2 L_h. \tag{3.9}$$

Proof. From assumption A.1 and Lemma 3.1, we have

$$\|\nabla f_h(x_h) - \nabla f_h(y_h)\| \le L_h \|x_h - y_h\|, \quad \forall x_h, y_h.$$

Then according to the definition of $\tilde{f}_H$, we can derive that

$$\begin{aligned}\|\nabla \tilde{f}_H(\tilde{x}_H) - \nabla \tilde{f}_H(\tilde{y}_H)\| &= \|\tilde{P}_{H,h}^\top \nabla f_h(\tilde{P}_{H,h}\tilde{x}_H) - \tilde{P}_{H,h}^\top \nabla f_h(\tilde{P}_{H,h}\tilde{y}_H)\| \\ &\le \|\tilde{P}_{H,h}\| L_h \|\tilde{P}_{H,h}(\tilde{x}_H - \tilde{y}_H)\| \\ &\le B^2 L_h \|\tilde{x}_H - \tilde{y}_H\|.\end{aligned}$$

This proves the lemma and implies (3.9). □

The following lemma gives the progress with one coarse subspace correction step.

Lemma 3.8. *Suppose assumption* A.1 *holds for* $\tilde{f}_H$ *and the Lipschitz constant is* $\tilde{L}_H$. *If the algorithm executes a coarse subspace correction step at the point* $x_{h,k}$, *we have*

$$f_{h,k} - f_{h,k+1} \ge \rho\bar{\alpha}((\kappa_g \prod_{\ell=H+1}^{h} \sigma_\ell)^2 + (\frac{\lambda_{min}^M}{\lambda_{max}^M})^2)\|\nabla f_{h,k}\|^2, \tag{3.10}$$

where λ_{min}^M *and* λ_{max}^M *are the minimal and maximal eigenvalues of the mass matrix* M_h, *respectively.*

Proof. When a coarse subspace correction step is chosen, we have

$$f_{h,k} - f_{h,k+1} \ge \tilde{f}_H(\tilde{x}_{H,0}) - \tilde{f}_H(\tilde{x}_{H,1}) \ge \rho\bar{\alpha}\|\nabla \tilde{f}_H(\tilde{x}_{H,0})\|^2. \tag{3.11}$$

The first inequality is due to the descent property of coarse subspace correction and the second inequality comes from the Armijo condition (2.20) by choosing the negative gradient as the descent direction.

According to the definition of $\nabla \tilde{f}$, we have

$$\begin{aligned}\nabla \tilde{f}_H(\tilde{x}_{H,0}) &= \tilde{P}_{H,h}^\top \nabla f_h(\tilde{P}_{H,h}\tilde{x}_{H,0}) \\ &= \tilde{P}_{H,h}^\top \nabla f_{h,k} \\ &= \left((P_{H,h}^\top \nabla f_{h,k})^\top, \ \frac{1}{\|x_{h,k}\|} x_{h,k}^\top \nabla f_{h,k}, \ \frac{1}{\|z_{h,k}\|} z_{h,k}^\top \nabla f_{h,k} \right)^\top \\ &= \left(\prod_{\ell=H+1}^{h} \sigma_\ell (\prod_{\ell=H+1}^{h} R_\ell \nabla f_{h,k})^\top, \ \frac{1}{\|x_{h,k}\|} x_{h,k}^\top \nabla f_{h,k}, \ \frac{1}{\|z_{h,k}\|} z_{h,k}^\top \nabla f_{h,k} \right)^\top.\end{aligned}$$

This relation, together with (2.13) and (2.6), implies that

$$\|\nabla \tilde{f}_H(\tilde{x}_{H,0})\|^2 \ge \left((\kappa_g \prod_{\ell=H+1}^{h} \sigma_\ell)^2 + (\frac{\lambda_{min}^M}{\lambda_{max}^M})^2 \right) \|\nabla f_{h,k}\|^2. \tag{3.12}$$

Consequently, our lemma follows from (3.11) and (3.12). □

3.3. Convergence properties and complexity

The following theorem shows the global convergence for general functions.

Theorem 3.1. *Suppose* (2.19) *is satisfied for all direct search directions. Then under assumptions* A.1-2 *the iterative sequence* $\{x_{h,k}\}$ *generated by Algorithm* 2.2 *at the finer level converges to the minimizer or first order stationary point of* $f_h(x_h)$*.*

Proof. For direct search step, the step size $\alpha_{h,k}$ has a lower bound $\bar{\alpha}$ from Lemma 3.4. From Armijo condition (2.20) we have

$$f_{h,k} - f_{h,k+1} \geq -\rho\bar{\alpha}\nabla f_{h,k}^{\top} d_{h,k}.$$

From (2.19), we have

$$-\nabla f_{h,k}^{\top} d_{h,k} \geq \eta_h \|\nabla f_{h,k}\|^2.$$

Hence,

$$f_{h,k} - f_{h,k+1} \geq \rho\bar{\alpha}\eta_h \|\nabla f_{h,k}\|^2.$$

For coarse subspace correction step, we have

$$f_{h,k} - f_{h,k+1} \geq \rho\bar{\alpha}\kappa_g^2 \|\nabla f_{h,k}\|^2$$

from Lemma 3.8. Let $\delta = \min\{\eta_h, (\kappa_g^2 + \lambda_{min}^M)\}$. The above two inequalities imply that

$$f_{h,k} - f_{h,k+1} \geq \rho\bar{\alpha}\delta \|\nabla f_{h,k}\|^2.$$

Since $f_h(x_h)$ is bounded below by assumption **A.2**, we have

$$\lim_{k\to\infty} \|\nabla f_{h,k}\| = 0,$$

which completes the proof. □

In the following theorem and corollary, we prove the R-linear convergence rate for strongly convex functions.

Theorem 3.2. *For convex functions, suppose assumptions* A.1-3 *hold and* (2.19) *is satisfied for all direct search directions. Assume that the iterative sequence* $\{x_{h,k}\}$ *generated by Algorithm* 2.2 *at the finer level converges to a unique minimizer* x_h^**. Then the convergence rate is at least R-linear.*

Proof. Similar to Theorem 3.1, we can derive that

$$f_{h,k} - f_{h,k+1} \geq \rho\bar{\alpha}\delta \|\nabla f_h(x_{h,k})\|^2$$

from (2.19), Lemma 3.4, Lemma 3.8 and the definition of δ. From the second inequality of (3.7) in Lemma 3.5, we get

$$\|\nabla f_h(x_{h,k})\|^2 \geq m(f_h(x_{h,k}) - f_h(x_h^*)).$$

Hence,

$$f_h(x_{h,k}) - f_h(x_{h,k+1}) \geq \rho\bar{\alpha}\delta m(f_h(x_{h,k}) - f_h(x_h^*)),$$

where $0 < \rho\bar{\alpha}\delta m < 1$. By adding $f_h(x_h^*)$ to both sides of the above inequality, we have

$$f_h(x_{h,k+1}) - f_h(x_h^*) \leq (1 - \rho\bar{\alpha}\delta m)(f_h(x_{h,k}) - f_h(x_h^*)).$$

From the first inequality of (3.7) in Lemma 3.5, we have that

$$f_h(x_{h,k}) - f_h(x_h^*) \geq \frac{m}{2}\|x_{h,k} - x_h^*\|^2.$$

From all above, we can derive that

$$\begin{aligned}\|x_{h,k} - x_h^*\| &\leq \sqrt{\frac{2}{m}}(f_h(x_{h,k}) - f_h(x_h^*))^{\frac{1}{2}} \\ &\leq \sqrt{\frac{2}{m}}(1-\rho\bar{\alpha}\delta m)^{\frac{1}{2}}(f_h(x_{h,k-1}) - f_h(x_h^*))^{\frac{1}{2}} \\ &\leq \sqrt{\frac{2}{m}}(1-\rho\bar{\alpha}\delta m)^{\frac{k}{2}}(f_h(x_{h,0}) - f_h(x_h^*))^{\frac{1}{2}}.\end{aligned} \tag{3.13}$$

This completes the proof. □

Corollary 3.1. *For any $\epsilon > 0$, suppose assumptions* A.1-3 *hold, after $t = \frac{2\log c}{\log(1-\rho\bar{\alpha}\delta m)}$ iterations at most, where*

$$0 < c = \sqrt{\frac{m\epsilon^2}{2(f_h(x_{h,0}) - f_h(x_h^*))}},$$

we have $\|x_{h,k} - x_h^*\| \leq \epsilon$.

Proof. From inequality (3.13) and the convergence analysis for convex functions [44], we obtain the result. □

The last theorem shows the convergence rate for general function, including nonconvex functions.

Theorem 3.3. *For functions where assumptions* A.1-2 *hold, suppose* (2.19) *is satisfied for all direct search directions. Assume that Algorithm* 2.2 *generates a sequence $\{x_{h,k}\}$ converging to the first order stationary point x_h^* at the finer level. Then the convergence rate is sub-linear and*

$$\min_k \|\nabla f_{h,k}\|^2 \leq \frac{\rho\bar{\alpha}\delta}{K+1}(f_h(x_{h,0}) - f_h(x_h^*)), \tag{3.14}$$

which guarantees accuracy $\min_k \|\nabla f_{h,k}\| \leq \epsilon$ in $K = O(1/\epsilon^2)$ iterations.

Proof. Similar to Theorem 3.1, we can derive that

$$f_{h,k} - f_{h,k+1} \geq \rho\bar{\alpha}\delta\|\nabla f_h(x_{h,k})\|^2$$

from (2.19), Lemma 3.4, Lemma 3.8 and the definition of δ. By summing up the above inequalities for $k = 0, \ldots, K$, we obtain

$$\rho\bar{\alpha}\delta\sum_{k=1}^{K}\|\nabla f_h(x_{h,k})\|^2 \leq f_{h,0} - f_{h,K+1} \leq f_{h,0} - f_h(x_h^*).$$

From this we can derive (3.14), which completes the proof. □

4. Numerical Experiments

4.1. Test problems

To demonstrate the efficiency of our algorithms, we consider a few variational minimization problems including Bratu problem, nonlinear ellipse problem and nonconvex problem. The parameters in each problem will be specified following its introduction while some common settings are given here. All problems are lived in 2D functional space with a fixed domain $\Omega = (0,1) \times (0,1)$. We discretized Ω into square grids and the term ∇u with finite difference methods. Also, we define P_ℓ as a nine-point prolongation operator and $R_\ell = \frac{1}{4} P_\ell^\top$.

Our three test problems are listed as follows. The first two problems are convex while the last one is nonconvex.

1. Bratu equation

$$\begin{cases} -\Delta u(x,y) + e^{u(x,y)} = 0, \quad (x,y) \in \Omega \\ u(x,y) = 0, \quad (x,y) \in \partial\Omega. \end{cases} \tag{4.1}$$

The variational form of (4.1) is

$$\begin{cases} \min_u \mathcal{F}(u(x,y)) = \int_\Omega \frac{1}{2} |\nabla u(x,y)|^2 + e^{u(x,y)} dxdy \\ \text{s.t. } u(x,y) = 0, \quad (x,y) \in \partial\Omega. \end{cases} \tag{4.2}$$

2. Nonlinear ellipse equation

$$\begin{cases} -\Delta u(x,y) - \lambda u(x,y) e^{u(x,y)} = b(x,y), \quad (x,y) \in \Omega \\ u(x,y) = 0, \quad (x,y) \in \partial\Omega, \end{cases} \tag{4.3}$$

where $b(x,y) = \Big(9\pi^2 + \lambda e^{((x^2-x^3)\sin(3\pi y))}(x^2 - x^3) + 6x - 2\Big) \sin(3\pi y)$ and $\lambda = 10$. The variational form of (4.3) is

$$\begin{cases} \min_u \mathcal{F}(u(x,y)) = \int_\Omega \frac{1}{2} |\nabla u(x,y)|^2 - \lambda(u(x,y)e^{u(x,y)} - e^{u(x,y)}) - b(x,y)u(x,y) dxdy \\ \text{s.t. } u(x,y) = 0, \quad (x,y) \in \partial\Omega. \end{cases} \tag{4.4}$$

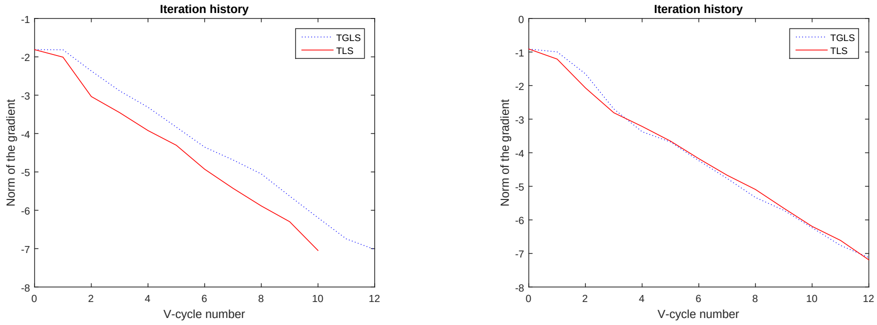
3. Nonconvex variational problem

$$\begin{cases} \min \mathcal{F}(u(x,y)) = \int_\Omega \frac{1}{1+|\nabla u(x,y)|^2} + \gamma |\nabla u(x,y)|^2 dxdy \\ \text{s.t. } u(x,y) = 1000(x-0.5)^2, \quad y = 0 \text{ or } 1, \\ \qquad u(x,y) = 1000(y-0.5)^2, \quad x = 0 \text{ or } 1, \end{cases} \tag{4.5}$$

where $\gamma = 10^{-3}$.

4.2. Evaluation of mesh size independent convergence rates

We first evaluate the convergence rates of our algorithm (TLS) and multigrid method in [31] with conventional multigrid spaces. We change Algorithm 2.1 in [31] to a two-level version with a fixed coarse level, and refer it as Two-Grid Line Search (TGLS). The initial point of all unconstrained problems are chosen as zero vectors. The algorithms stop when $\|\nabla f_h(x_h)\| \leq \epsilon_h = 10^{-7}$. For each V-cycle, the numbers of presmoothing and postsmoothing are set to 2. Both presmoothing and postsmoothing operators are L-BFGS. The level ℓ means there are 2^ℓ



(a) Problem (4.2)

(b) Problem (4.4)

Fig. 4.1. The gradient norms of TGLS and TLS on Problems (4.2) and (4.4) at the level $h = 6$.

Table 4.1: The numbers of V-cycles performed by TGLS and TLS over different fine spaces on Problems (4.2) and (4.4).

Prob.	Problem (4.2)		Problem (4.4)	
h	TGLS	TLS	TGLS	TLS
6	12	10	12	12
7	12	10	13	12
8	12	10	15	14
9	14	12	20	15
10	16	10	20	17

grids in each direction. For convenience, coarse spaces in both methods are always chosen as $H = h - 3$.

Then we test the convergence rate by taking $h = 6$ which means a grid of size $2^6 \times 2^6$. We compare the decrease of the gradients norms over V-cycles of TGLS and TLS on problems (4.2) and (4.4). The results are given in Fig. 4.1. They show that both algorithms converge linearly. Furthermore, we compare the numbers of V-cycles over different fine level spaces from 6 to 10. The results are listed in Table 4.1. It shows that the numbers of V-cycles on each level are almost the same. Hence, our multigrid methods can produce a mesh size independent linear convergence rate.

4.3. Performance evaluation with full multigrid methods

We compare our algorithm with the state-of-art full multigrid with Line Search method proposed in [31]. FMLS-LBFGS denotes the Algorithm 2.3 in [31]. It uses LBFGS in direct steps and coarse level corrections. For the implementation of our Algorithm 2.3, we use LBFGS in direct steps and gradient method with BB stepsize for convex cases and LBFGS for nonconvex cases in coarse level corrections. We denote them by FTLS-LBFGS-BB and FTLS-LBFGS, respectively. We also use backtracking for all algorithms in implementation for line search. Preventing cycling with similar point sequences, we also add an additional switching condition

$$\|x_{h,k} - x_h^{lc}\| \geq \kappa_x \|x_h^{lc}\| \tag{4.6}$$

for coarse level correction, where x_h^{lc} is the initial point of last coarse step and $\kappa_x \in (0, 1)$. Otherwise we will choose a direct step at a fine grid.

Table 4.2: Summary of computational results of problem (4.2).

Alg.	FMLS-LBFGS			FTLS-LBFGS-BB		
ℓ	nf	ng	nv	nf	ng	nv
3	91	80	0	83	81	0
4	100	76	11	49	46	2
5	85	57	9	30	26	3
6	79	47	6	46	41	3
7	38	29	5	33	28	2
8	44	40	3	57	52	2
9	52	49	1	6	4	1
10	27	26	0	12	11	0
11	3	2	0	3	2	0
time	20.241			9.476		
$\lVert g^* \rVert$	8.3e-8			8.2e-8		

Table 4.3: Summary of computational results of problem (4.4).

Alg.	FMLS-LBFGS			FTLS-LBFGS-BB		
ℓ	nf	ng	nv	nf	ng	nv
3	160	144	0	148	147	0
4	131	105	17	32	28	3
5	118	76	12	42	31	3
6	65	48	7	64	60	4
7	66	38	5	50	45	3
8	38	22	4	16	14	2
9	37	21	3	12	9	1
10	17	14	2	7	5	1
11	11	6	0	1	1	0
time	17.643			5.547		
$\lVert g^* \rVert$	1.4e-7			3.8e-8		

For parameters in practical consideration, we set

$$\epsilon_\ell = 10^{-7}/5^{h-\ell}, \quad \kappa_g = 10^{-2}, \quad \kappa_x = 10^{-2}, \quad \rho = 10^{-3}$$

on all levels. We add another two termination rules to prevent stagnating of algorithms,

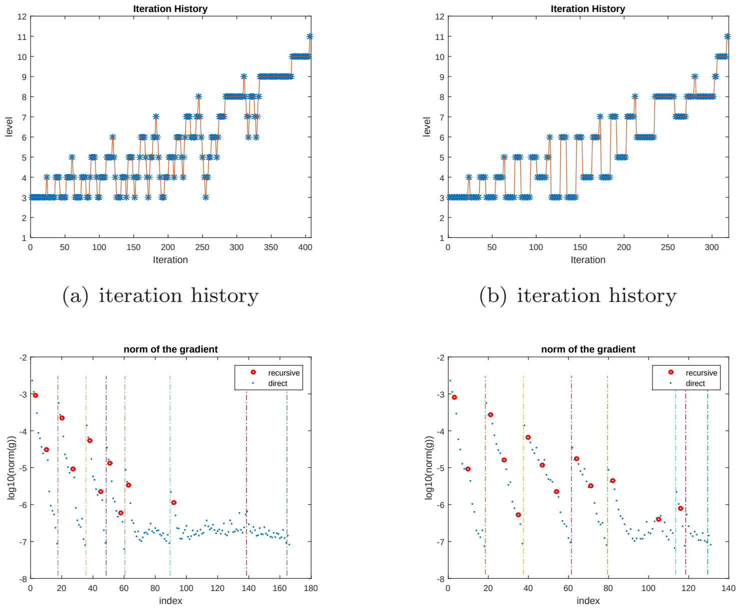
$$\frac{f_{\ell,k}(\tilde{f}_{\ell,k}) - f_{\ell,k+1}(\tilde{f}_{\ell,k+1})}{\max\left(|f_{\ell,k}(\tilde{f}_{\ell,k})|, |f_{\ell,k+1}(\tilde{f}_{\ell,k+1})|, 1\right)} \leq 10^{-16} \quad \text{or} \quad \|x_{\ell,k} - x_{\ell,k+1}\| \leq 10^{-12}$$

Both conditions are also used in [31]. As for the maximal iteration number, we set $K = 1000$ for direct search steps. For coarse level correction, K always equals to 10 in LBFGS while in BB, $K = 10$ if the level ℓ is no finer than 5 and the difference between the finer level h and the coarser level H is no greater than 3, otherwise $K = 20$. The algorithms described above were coded in MATLAB 2015a (Release 8.5.0) and all experiments below were run on a Dell Optiplex 9020 with an Intel-Core i7-4790 3.60 GHz CPU and 8 GB RAM.

We summarize the computational costs on different levels of both methods in Table 4.2-4.4. We report the numbers of function evaluations and gradient evaluations at different levels

Table 4.4: Summary of computational results for nonconvex problem (4.5).

Alg.	FMLS-LBFGS			FTLS-LBFGS		
ℓ	nf	ng	nv	nf	ng	nv
3	1456	1099	0	358	241	0
4	3135	1970	270	250	164	8
5	4739	2989	504	452	257	11
6	4409	3020	550	505	328	19
7	3236	2212	383	632	382	36
8	3357	1470	221	954	787	97
9	1836	1014	173	478	449	8
10	833	573	102	142	124	5
11	372	318	46	69	67	4
time	1049.071			282.648		
$\|g^*\|$	6.5e-7			4.8e-7		



(a) iteration history

(b) iteration history

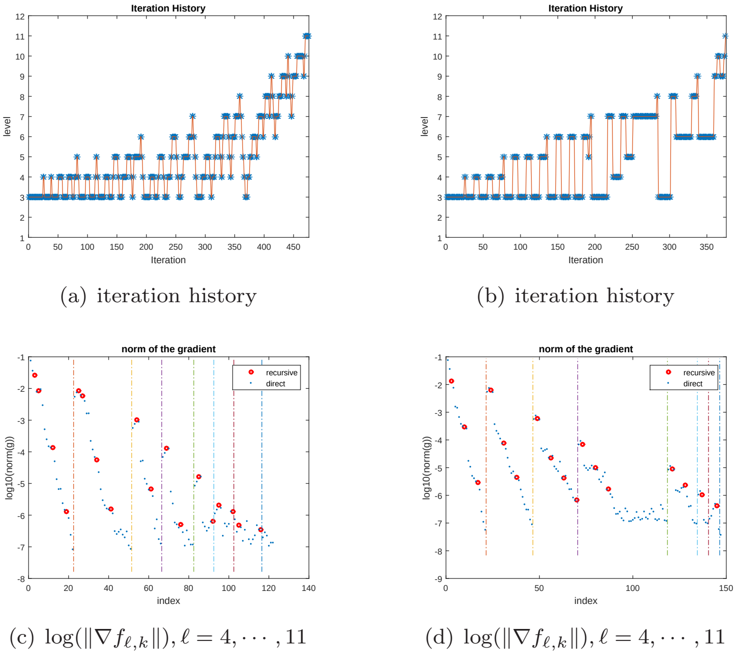
(c) $\log(\|\nabla f_{\ell,k}\|), \ell = 4, \cdots, 11$

(d) $\log(\|\nabla f_{\ell,k}\|), \ell = 4, \cdots, 11$

Fig. 4.2. Performance plots for iterations and gradient values for problem (4.2). The first line corresponds to comparison of iteration history and the second line corresponds to comparison of logarithm of gradient where recursive steps are marked by “o”.

and total CPU time measured by seconds, and the attained accuracies which are measured by the Euclidean norm of gradient $\|g^*\|$. We also compare the iteration behavior between two algorithms in Fig. 4.2-Fig. 4.4. Iteration histories of two algorithms are showed in Fig. 4.x(a) and Fig. 4.x(b), respectively. The norm of gradients basing on the common logarithm are depicted in Fig. 4.x(c) and Fig. 4.x(d).

In convex occasions, our algorithm FTLS-LBFGS-BB takes less than half and one third of the time FMLS-LBFGS uses, respectively. FMLS-LBFGS fails to satisfy the target gradient



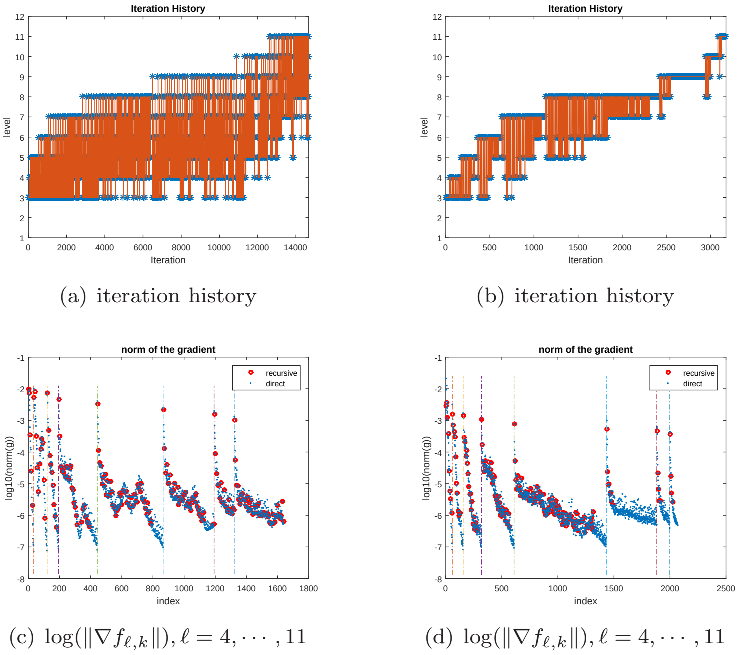
(a) iteration history

(b) iteration history

(c) $\log(\|\nabla f_{\ell,k}\|), \ell = 4, \cdots, 11$

(d) $\log(\|\nabla f_{\ell,k}\|), \ell = 4, \cdots, 11$

Fig. 4.3. Performance plots for iterations and gradient values for problem (4.4). The first line corresponds to comparison of iteration history and the second line corresponds to comparison of logarithm of gradient where recursive steps are marked by "o".



(a) iteration history

(b) iteration history

(c) $\log(\|\nabla f_{\ell,k}\|), \ell = 4, \cdots, 11$

(d) $\log(\|\nabla f_{\ell,k}\|), \ell = 4, \cdots, 11$

Fig. 4.4. Performance plots for iterations and gradient values for problem (4.5). The first line corresponds to comparison of iteration history and the second line corresponds to comparison of logarithm of gradient where recursive steps are marked by "o".

tolerance in the nonlinear ellipse example, while FTLS-LBFGS-BB reaches the target tolerance successfully and efficiently. In nonconvex case, FMLS-LBFGS fails to reach the target gradient tolerance after stopping the algorithm because it can hardly make progress on function value. FTLS-LBFGS just uses about one fourth of the time to reach the similar achieved gradient norm with FMLS-LBFGS.

5. Conclusion and Future Work

In this paper, we proposed a new two-level subspace method framework for general nonlinear optimization discretized from the infinite-dimensional problems. The main contribution of our method is the combination of the subspace technique and multigrid. In the coarse subspace correction steps, we augment the coarse grid space with some subspace consisting of descent directions. We establish a two-level subspace multigrid framework based on traditional optimization methods and a new coarse level subspace correction. We prove the linear convergence rate for strongly convex case and the sublinear convergence rate for nonconvex case. We implement the direct search direction step with limited memory BFGS and the coarse subspace correction with gradient method with BB stepsize for convex cases and limited memory BFGS for nonconvex cases. Preliminary numerical experiments show that our algorithm performs efficiently on unconstrained optimization. Our future work includes designing the coarse s-pace with more information of historical descent directions and extending this framework to constrained optimization, especially box-constrained problems as in [33, 39, 40].

Acknowledgements. The first and second authors would like to thank Prof. Hehu Xie for the helpful discussion on multigrid methods. The third author was supported by NSFC grants 11331012 and 11688101. The authors are sincerely grateful to two anonymous referees for their valuable suggestions and comments as well as for pointing out a few relevant literatures.

References

[1] N. Gould, D. Orban and P.L. Toint, Numerical methods for large-scale nonlinear optimization, *Acta Numer.*, **14** (2005), 299-361.

[2] Y.X. Yuan, Subspace techniques for nonlinear optimization, R. Jeltsch, D. Li and I.H. Sloan, editors, Some Topics in Industrial and Applied Mathematics (Series in Contemporary Applied Mathematics CAM 8), pages 206-218, Beijing, 2007, Higher Education Press.

[3] Y.X. Yuan, A review on subspace methods for nonlinear optimization, S.Y. Jang, Y.R. Kim, D.W. Lee and I. Yie, editors, Proceddings of the International Congress of Mathematicians, volume 4, pages 807-827, Soeul, Korea, 2014.

[4] M.R. Hestenes and E. Stiefel, Methods of conjugate gradients for solving linear systems, *J. Res. N.B.S.*, **49** (1952), 409-436.

[5] R. Fletcher and C.M. Reeves, Function minimization by conjugate gradients, *Comput. J.*, **7** (1964), 149-154.

[6] Y.X. Yuan and J. Stoer, A subspace study on conjugate gradient algorithms, *ZAMM Z. angew. Math. Mech.*, **75** (1995), 69-77.

[7] D.F. Shanno, Conjugate gradient methods with inexact searches, *Math. Oper. Res.*, **3** (1978), 244-256.

[8] J. Nocedal, Updating quasi-newton matrices with limited storage, *Math. Comput.*, **35** (1980), 773-782.

[9] Q. Ni and Y.X. Yuan, A subspace limited memory quasi-newton algorithm for large-scale nonlinear bound constrained optimization, *Math. Comput.*, **66** (1997), 1509-1520.

[10] P.E. Gill and M.W. Leonard, Reduced-hessian quasi-newton methods for unconstrained optimization, *SIAM J. Optim.*, **12** (2001), 209-237.

[11] Z.H. Wang and Y.X. Yuan, A subspace implementation of quasi-newton trust region methods for unconstrained optimization, *Numer. Math.*, **104** (2006), 241-269.

[12] W. Hackbusch, Multigrid Methods and Applications, volume 4 of Springer Ser. Comput. Math., Springer-Verlag, Berlin, 1985.

[13] S.F. McCormick, Multigrid Methods, volume 3 of Frontiers Appl. Math., SIAM, Philadelphia, 1987.

[14] W.L. Briggs, V.E. Henson and S.F. McCormick, A Multigrid Tutorial, SIAM, Philadelphia, 2 edition, 2000.

[15] U. Trottenberg, C.W. Oosterlee and A. Schuller, Multigrid, Academic Press, San Diego, CA, 2001.

[16] V.E. Henson, Multigrid methods nonlinear problems: an overview, Electronic Imaging 2003, pages 36-48, International Society for Optics and Photonics, 2003.

[17] A. Brandt, Multi-level adaptive solutions to boundary-value problems, *Math. Comput.*, **31** (1977), 333-390.

[18] I. Yavneh and G. Dardyk, A multilevel nonlinear method, *SIAM J. Sci. Comput.*, **28** (2006), 24-46.

[19] T.A. Manteuffel, S.F. McCormick, O. Röhrle and J. Ruge, Projection multilevel methods for quasilinear elliptic partial di.erential equations: Numerical results, *SIAM J. Numer. Anal.*, **44** (2006), 120-138.

[20] T.A. Manteuffel, S.F. McCormick and O. Röhrle, Projection multilevel methods for quasilinear elliptic partial di.erential equations: Theoretical results, *SIAM J. Numer. Anal.*, **44** (2006), 139-152.

[21] S.F. McCormick, Projection multilevel methods for quasi-linear pdes: V-cycle theory, *Multiscale Model. Simul.*, **4** (2005), 1339-1348.

[22] T. Dreyer, B. Maar and V. Schulz, Multigrid optimization in applications, *J. Comput. Appl. Math.*, **120** (2000), 67-84.

[23] B. Maar and V. Schulz, Interior point multigrid methods for topology optimization, *Struct. Multidiscip. Optim.*, **19** (2000), 214-224.

[24] A. Borzì. and K. Kunisch, A multigrid scheme for elliptic constrained optimal control problems, *Comput. Optim. Appl.*, **31** (2005), 309-333.

[25] A. Borzì. and V. Schulz, Multigrid methods for pde optimization, *SIAM Rev.*, **51** (2009), 361-395.

[26] S.G. Nash, A multigrid approach to discretized optimization problems, *Optim. Methods Softw.*, **14** (2000), 99-116.

[27] R.M. Lewis and S.G. Nash, Model problems for the multigrid optimization of systems governed by di.erential equations, *SIAM J. Sci. Comput.*, **26** (2005), 1811-1837.

[28] R.M. Lewis and S.G. Nash, Factors a.ecting the performance of optimization-based multigrid methods, W.W. Hager, S.J. Huang, P.M. Pardalos and O.A. Prokopyev, editors, Multiscale Optimization Methods and Applications, pages 151-172, New York, 2006, Springer.

[29] R.M. Lewis and S.G. Nash, Using inexact gradients in a multilevel optimization algorithm, *Comput. Optim. Appl.*, **56** (2013), 39-61.

[30] S.G. Nash, Properties of a class of multilevel optimization algorithms for equality-constrained problems, *Optim. Methods Softw.*, **29** (2014), 137-159.

[31] Z. Wen and D. Goldfarb, A line search multigrid method for large-scale nonlinear optimization, *SIAM J. Optim.*, **20** (2009), 1478-1503.

[32] S. Gratton, A. Sartenaer and P.L. Toint, Recursive trust-region methods for multiscale nonlinear optimization, *SIAM J. Optim.*, **19** (2008), 414-444.

[33] S. Gratton, M. Mou.e, P.L. Toint and M. Weber-Mendonça, A recursive-trust-region method for bound-constrained nonlinear optimization, *IMA J. Numer. Anal.*, **28** (2008), 827-861.

[34] S. Gratton, M. Mou.e, A. Sartenaer, P.L. Toint and D. Tomanos, Numerical experience with a recursive trust-region method for multilevel nonlinear bound-constrained optimization, *Optim. Methods Softw.*, **25** (2010), 359-386.

[35] J.C. Ziems and S. Ulbrich, Adaptive multilevel inexact sqp methods for pde-constrained optimization, *SIAM J. Optim.*, **21** (2011), 1-40.

[36] E. Frandi and A. Papini, Coordinate search algorithms in multilevel optimization, *Optim. Methods Softw.*, **29** (2014), 1020-1041.

[37] E. Frandi and A. Papini, Improving direct search algorithms by multilevel optimization techniques, *Optim. Methods Softw.*, **30** (2015), 1077-1094.

[38] X.C. Tai and J. Xu, Global and uniform convergence of subspace correction methods for some convex optimization problems, *Math. Comp.*, **71** (2002), 105-124.

[39] X.C. Tai and P. Tseng, Convergence rate analysis of an asynchronous space decomposition method for convex minimization, *Math. Comp.*, **71** (2002), 1105?135.

[40] X.C. Tai, Rate of convergence for some constraint decomposition methods for nonlinear variational inequalities, *Numer. Math.*, **93** (2003), 755-786.

[41] K. Chen and X.C. Tai, A nonlinear multigrid method for total variation minimization from image restoration, *J. Sci. Comput.*, **33** (2007), 115-138.

[42] S. Gratton, A. Sartenaer and P.L. Toint, Second-order convergence properties of trust-region methods using incomplete curvature information, with an application to multigrid optimization, *J. Comput. Appl. Math.*, **24** (2006), 676-692.

[43] W. Sun and Y.X. Yuan, Optimization Theory and Methods: Nonlinear Programming, volume 1, Springer Science & Business Media, New York, 2006.

[44] Y. Nesterov, Introductory Lectures on Convex Optimization, volume 87, Springer Science & Business Media, New York, 2004.