ITERATIVE METHODS FOR NONLINEAR ELLIPTIC EQUATIONS

LONG CHEN

CONTENTS

In this chapter we discuss iterative methods for solving the finite element discretization of semi-linear elliptic equations of the form: find $u \in H_0^1(\Omega)$ such that

$$-\Delta u = f(x,u) \quad \text{in } H^{-1}(\Omega). \tag{1}$$

In most places, we shall drop the dependence on x in f to emphasize the nonlinearity in u. Denoted by $A : H_0^1(\Omega) \to H^{-1}(\Omega)$ is the elliptic operator defined by $-\Delta$ and $\|v\|_A^2 = \langle Av, v\rangle$. Equation (1) can be also written as

$$\mathcal{L}(u) := Au - f(u) = 0.$$

Let $\mathbb{V}_h \subset H_0^1(\Omega)$ be the linear finite element space associated to a quasi-uniform mesh $\mathcal{T}_h$. The nonlinear Garlerkin method is: find $u_h \in \mathbb{V}_h$ such that

$$Au_h = f(u_h) \quad \text{in } \mathbb{V}_h', \tag{2}$$

We assume f is smooth enough, for example, f is Lipchitz w.r.t u

$$\|f(u) - f(v)\|_{-1} \leq L\|u - v\|_A.$$

We shall assume the existence and locally uniqueness of the solution to the nonlinear equation (1). Then it can be shown that (c.f. [13] and the references cited therein) if h is sufficiently small, the equation (2) for the nonlinear Garlerkin method has a (locally unique) solution u_h satisfying the error estimate

$$\|u - u_h\|_{0,p} + h\|u - u_h\|_{1,p} \lesssim h^2\|u\|_{2,p} \quad \text{for all } 2 \leq p < \infty. \tag{3}$$

We are interested in iterative methods to compute u_h or a good approximation of u_h. We shall drop the subscript h when the size h does not play a role in the method.

Date: 11/20/2013, Latest update: March 18, 2022.

1. Basic Iterative Methods

Starting from an initial guess u_0, for $k = 1, 2 \cdots$, we look for a better approximation u_{k+1} (of u_h) from an existing one u_k. Here we keep the iteration counter k and drop the subscript h to simplify the notation.

A general form of iterative methods consists of three steps:

(1) form the residual $r_k = f(u_k) - Au_k$;
(2) solve the residual equation $\hat{e}_k = B^{-1} r_k$,
(3) update the solution $u_{k+1} = u_k + \hat{e}_k$

What is the ideal choice of B? For linear equation $Au = f$, the ideal one is $B = A^{-1}$ since the difference $e_k := u - u_k$ satisfies the equation $Ae_k = r_k$. But for nonlinear equation $Ae_k = f(u) - Au_k \neq r_k$. We should figure out what the ideal equation for e_k is in the nonlinear sitting.

By Taylor theorem, we have

$$A(u - u_k) = f(u) - Au_k = f(u_k) + f'(\xi)(u - u_k) - Au_k,$$

where $\xi = tu_k + (1-t)u$ for some $t \in (0, 1)$. Rearrange the term implies

$$J_\xi e_k = r_k, \tag{4}$$

where $J_\xi = A - f'(\xi)$, the Jacobian of $\mathcal{L}(u) = Au - f(u)$ at ξ. Therefore the ideal choice is $B = J_\xi^{-1}$. But ξ is unknown even u is known. Different methods will be obtained by different choices of B.

We write derivatives and equations in the calculus style. A more rigorous lagrange is the calculus of variation. Given the current approximation u_k, we perturb it as $u_k + \varepsilon e$, where $\varepsilon \in \mathbb{R}$ and $e \in H_0^1$, and take the derivative with respect to ε

$$\frac{d}{d\varepsilon} \mathcal{L}(u_k + \varepsilon e)|_{\varepsilon = 0} = \mathcal{L}'(u_k) e.$$

The residual equation $\mathcal{L}'(u_k)e = r_k$ in $\mathbb{V}'$ is understood as the weak form:

$$(\mathcal{L}'(u_k)e, \phi) = (r_k, \phi) \quad \forall \phi \in \mathbb{V}. \tag{5}$$

For $\mathcal{L}(u) = Au - f(u)$, the Jacobian is $\mathcal{L}'(u_k) = A - f'(u_k)$ and equation (5) becomes

$$(Ae, \phi) + (f'(u_k)e, \phi) = (r_k, \phi) \quad \forall \phi \in \mathbb{V}.$$

In addition to the stiffness matrix, one needs to assemble a weighted mass matrix for the weighted L^2 inner product $(f'(u_k)e, \phi)$ for which the function $f'(u_k)$ defines a variable coefficient of the inner product. When the linear finite element is used, one can use three vertices quadrature rule i.e.

$$\int_\tau g(x) \, \mathrm{d}x \approx \frac{1}{3} \sum_{i=1}^{3} g(x_i) |\tau|.$$

Then the mass matrix becomes diagonal $M = \operatorname{diag}(m_1, \cdots, m_N)$. This is known as the mass lumping. Taking the test function ϕ as one basis function ϕ_i at the i-th vertex, we obtain the equation at the i-th vertex

$$\sum_j a_{ij} e_j + m_i f'(u_{k,i}) e_i = m_i r_{k,i}, \tag{6}$$

For 2-D uniform grids, $m_i = h^2$ and (6) can be interpret as the finite difference discretization of the Jacobian equation at each vertex.

Note that even u could have non-zero Dirichlet boundary condition, in the residual equation, the correction e is always zero on the Dirichlet boundary. In view of calculus of variation, the perturbation $u_k + \varepsilon e$ still satisfies the imposed boundary condition and thus e should vanish on the Dirichlet boundary and the residual equation is posed for free nodes `freeNode` only. Thus the linear algebraic system is solved for all interior nodes only. The boundary condition of u is included when computing the residual $f(u) - Au$. Let `A` be the full stiffness matrix and `u` is the vector evaluated at all nodes, then `(A*u)(freeNode)` will bring the boundary effect to the interior.

Algorithm 1.1 (Fixed point/Picard's iteration)**.** *We choose $B = A^{-1}$. Namely solving*

$$Au_{k+1} = f(u_k). \tag{7}$$

Obviously if the sequence $\{u_k\}$ converges, the limit is the solution u. From

$$\|u - u_{k+1}\|_1 = \|f(u) - f(u_k)\|_{-1} \leq L\|u - u_k\|_A,$$

the fixed point iteration converges if $L < 1$, which restricts its usage.

Algorithm 1.2 (Pseudo-transient continuation/Gradient flow)**.** *We choose $B = \alpha I$.*

$$u_{k+1} = u_k + \alpha(f(u_k) - Au_k). \tag{8}$$

It can be thought as the forward Euler method to compute the steady state solution of the nonlinear parabolic equation

$$u_t + Au = f(u). \tag{9}$$

One iteration in (8) is cheap since only the action of A not A^{-1} is needed. But the method is not recommend to use for large size problems since the step size α should be small enough (in the size of h^2 even for the linear problem) and thus it takes large iteration steps to converge to the steady state. The overall cost will be very high.

Along this line, more accurate and stable explicit methods, e.g. Runge-Kutta methods, or implicit methods, can be used to solve (9) with large time step. Notice that we are interested the equilibrium point not the trajectory. So one can rescale the pseudo time variable to speed up the iteration. For example, let $\tau = \mu \ln(1 + t)$ and (9) becomes

$$u_t = \frac{\mu}{1+t}(f(u) - Au). \tag{10}$$

In [7], it shows time discretiztion of (10) is more robust and converges faster.

When the nonlinear equation is the first order equation of an optimization problem, i.e., there exists an energy $E(u)$ s.t. $\mathcal{L}(u) = \nabla E(u)$, then the residual $f(u_k) - Au_k = -\nabla E(u_k)$ and scheme (8) can be interpret as the gradient flow. An optimal choice of step size α can be obtained by a linear search; see [9] for several linear search methods. We still use the name 'gradient flow' even the nonlinear equation is not a gradient.

Algorithm 1.3 (Gradient flow in H^{-1} inner product/Relaxed fixed-point method)**.** *We choose $B = \alpha A^{-1}$.*

$$u_{k+1} = u_k + \alpha A^{-1}(f(u_k) - Au_k) = (1 - \alpha)u_k + \alpha A^{-1} f(u_k). \tag{11}$$

We can interpret the above scheme as a gradient flow $u_t = f(u) - Au$ but imposed in the inner product H^{-1} which is represented by A^{-1}. The second formula shows it is also a relaxed fixed point iteration, i.e, a weighted average of the current approximation and the update obtained by the fixed point iteration.

Exercise 1.4. Write the error equation and derive a range of α such that the result iteration is a contraction and thus the true solution u is a fixed point iteration of (11).

Algorithm 1.5 (Newton's method)**.** *We choose* $B = J_k := A - f'(u_k)$.

$$u_{k+1} = u_k + J_k^{-1}(f(u_k) - Au_k).$$

The idea of Newton's method is to solve the linear approximation of the nonlinear equation at the current iteration, i.e., u_{k+1} satisfies the equation

$$\mathcal{L}(u_k) + \mathcal{L}'(u_k)(u_{k+1} - u_k) = 0.$$

To analyze the convergence of Newton's method, we write

$$\begin{aligned} Au_{k+1} &= f(u_k) + f'(u_k)(u_{k+1} - u_k), \\ Au &= f(u_k) + f'(u_k)(u - u_k) + \frac{1}{2}f''(\xi)(u - u_k)^2. \end{aligned}$$

Therefore

$$J_k(u - u_{k+1}) = C(u - u_k)^2.$$

Assume $J_k = A - f'(u_k)$ is still elliptic, i.e., $J_k : H_0^1(\Omega) \to H^{-1}(\Omega)$ is an isomorphism and f'' is uniformly bounded. We then obtain

$$\|u - u_{k+1}\|_1 \leq C_k \|(u - u_k)^2\|_{-1}.$$

Note that $\|(u-u_k)^2\| \neq \|u-u_k\|^2$. We shall use Hölder inequality and Sobolev embedding theorem $L^p \hookrightarrow H^1$ for $1 \leq p \leq 6$ (recall our domain $\Omega \subset \mathbb{R}^d$ for $d \leq 3$) to get

$$|((u - u_k)^2, v)| \leq \|(u - u_k)^2\|_{L^{6/5}} \|v\|_{L^6} = \|u - u_k\|_{L^{12/5}}^2 \|v\|_{L^6} \lesssim \|u - u_k\|_1^2 \|v\|_1.$$

Therefore

$$\|(u - u_k)^2\|_{-1} \lesssim \|u - u_k\|_1^2,$$

and consequently

$$\|u - u_{k+1}\|_1 \leq C\|u - u_k\|_1^2 \leq C^{2(k+1)} \|u - u_0\|_1^{2(k+1)}.$$

If $\|u - u_0\|_1$ is small, i.e., the initial guess is sufficiently close to u, then the Newton's method will converge quadratically. On the other hand, if $\|u - u_0\|_1$ is not small enough, it may diverges. Take an example with $C = 10$ and $\|u - u_0\|_1 = 0.1$. Then the upper bound of the error is always 0.1. Perturbation near this critical value will give either fast convergence or divergence.

The assumption f'' is uniformly bounded can be relaxed to f' is Lipschitz (by using integral form of the remainder in Taylor series). See the general local theorem of convergence of Newton's method established by Kantorovich [5].

The assumption: an initial guess sufficiently close to u can be characterized more qualitatively but is essential and a limitation to the convergence of Newton's method. This is usually coined as the local quadratic convergence.

Remark 1.6. The Newton's method is also refereed as Newton-Raphson method; see [1] on 'who is Raphson'?

2. Newton-type Methods

The constraint of Newton's method are:

(1) Require a good initial guess. It is sensitive to the choice of initial guess.
(2) Computational cost for solving the linear system. Each step we have to invert a large matrix.
(3) Require explicit information of Jacobian matrix. Then the code cannot be applied to a large class of problems since the Jacobian is changing case by case.

(4) Smoothness of the nonlinearity. If the nonlinear term is non-smooth, Newton iteration steps could be large.

We shall discuss the following variants of Newton's method to address some of these difficulties:

(1) Damped Newton: force the decreasing of the residual.
(2) Inexact Newton: solve the Jacobian equation approximately.
(3) Jacobian-free Newton: form the Jacobian numerically not analytically.

2.1. **Damped Newton's method.** In Newton's method, the update could be too large in the searching direction e; see the following figure on a simple example. We thus introduce a step size α to enforce the reduction of the residual in certain norm. The optimal choice of α could be found by line search. Here we use a simple bisection scheme, i.e. $\alpha = 1, 0.5, 0.25 \cdots$ and refer to [4, 9] for more advanced line search.

Algorithm: $u = \mathrm{update}(u_{\mathrm{old}}, e)$
$\alpha = 1;\ r_{\mathrm{old}} = \|f(u_{\mathrm{old}}) - Au_{\mathrm{old}}\|;\ r = r_{\mathrm{old}} + 1;$
while $r > r_{\mathrm{old}}$ **do**
  $u = u_{\mathrm{old}} + \alpha\, e;$
  $r = \|f(u) - Au\|;$
  $\alpha = \alpha/2;$
end

Algorithm 1: Damped Newton Update

2.2. **Inexact Newton's method.** In each step of Newton's method, the most time consuming part is solving the Jacobian equation $Je = r$. Since J is elliptic, the most efficient method will be iterative methods using multilevel preconditioners. This iteration will be refereed as the *inner iteration* and a dynamic tolerance can be used for each outer iteration.

Algorithm: $e = \mathrm{inexact}(J, r, B, \theta)$
$e^0 = 0;\ r_{\mathrm{in}} = r;$
while $\|r_{in}\| > \theta\|r\|$ **do**
  $e^i = e^{i-1} + Br;$
  $r_{in} = r - Je^i;$
end

Algorithm 2: Inner iteration of inexact Newton method

In the inner iteration, the Jacobian equation is not solved exactly. It could stop at a tolerance depending on the outer Newton iteration through the parameter θ. In the first few steps of Newton iteration, there is no advantage to solve the Jacobian equation exactly. A simple sequence of θ could be $10^{-2}, 10^{-3}, \ldots, 10^{-6}$ decreasing to the smallest tolerance. We refer to [5] for detailed discussion.

2.3. **Jacobian-free Newton method.** In Krylov methods for solving linear system, there is no need to form the Jacobian matrix explicitly (i.e. store each nonzero entry of the whole matrix). Instead only the action Jv is needed for a given vector v. We can approximate the matrix-vector product using a first order finite difference scheme

$$J_u v \approx [F(u + \epsilon v) - F(u)]/\epsilon = Av - [f(u + \epsilon v) - f(u)]/\epsilon.$$

The approximation error is $\mathcal{O}(\epsilon)$. The choice of ϵ is sensitive to scaling, given u and v. If ϵ is too large, the derivative is poorly approximated and if it is too small the result of the finite difference is contaminated by floating-point roundoff error – we are dividing by a small number. A simple choice of ϵ is

$$\epsilon = \sqrt{\epsilon_{\text{mach}}} \left(\frac{1}{N\|v\|_{l_2}} \sum_{i=1}^{N} |u_i| + 1 \right),$$

where ϵ_{mach} is the machine roundoff error. We refer to [6] for more discussion.

We summarize these variants into the following algorithm. For Jacobian-free Newton, replace the matrix J by a subroutine $J(u, v)$.

Algorithm: $u = \text{NewtonMethod}(u_0, tol)$
$u = u_0; r_0 = f(u_0) - Au_0; r = r_0;$
while $\|r\|/\|r_0\| > \text{tol}$ **do**
  $r = f(u) - Au;$
  $J = Au - f'(u);$
  $e = \text{inexact}(J, r, B, \theta);$
  $u_0 = u;$
  $u = \text{update}(u_0, e);$
  $\theta = \theta/10;$
end

2.4. **Other variants of Newton's method.** Add a diagonal matrix to regularize the Jacobian matrix which is like an implicit method for solving the gradient flow equation. Add more from [9].

Quasi-Newton method. Belong to a class of optimization method.

3. Two-grid Methods

The basic idea of two-grid methods developed by Xu [13] is to get a rough approximation on a coarse space and use it as an initial guess on the fine grid. The nonlinearity is resolved in a relative coarse grid and on the fine grid only one linear iteration is needed. Here we follow Xu [13] to present the methods and refer to [8, 11, 12] for other applications.

Let $\mathbb{V}_h$ and $\mathbb{V}_H$ be two finite element spaces associated to quasi-uniform meshes $\mathcal{T}_h$ and $\mathcal{T}_H$ with mesh size $H > h$, respectively. Suppose we have computed the non-linear Garlerkin approximation $u_H \in \mathbb{V}_H$, i.e.,

$$Au_H = f(u_H) \quad \text{in } \mathbb{V}_H',$$

by some methods (gradient flow or Newton's method). As we shall show later, the dimension of V_H can be very small and therefore the computation cost of u_H is negligible. We try to compute an approximation of the non-linear Garlerkin approximation $u_h \in \mathbb{V}_h$

$$Au_h = f(u_h) \quad \text{in } \mathbb{V}_h'. \tag{12}$$

Note that u_h and u_H satisfy the error estimate in (3). In particular

$$\|u - u_H\|_p \lesssim H^2 \|u\|_{2,p} \tag{13}$$

$$\|u - u_h\|_1 \lesssim h\|u\|_2. \tag{14}$$

Algorithm 3.1 (Two-Grid + One Step Fixed Point Iteration)**.** *Compute $u^h \in \mathbb{V}_h$ such that*

$$Au^h = f(u_H) \quad \text{in } \mathbb{V}_h'. \tag{15}$$

From (12) and (15), we obtain the error equation

$$A(u_h - u^h) = f(u_h) - f(u_H) \quad \text{in } \mathbb{V}_h'. \tag{16}$$

Assume $f(u)$ is Lipschitz continuous with respect to u in L^2-norm. By the ellipticity of the operator A and the error estimate (3), we have

$$\|u_h - u^h\|_1 = \|f(u_h) - f(u_H)\|_{-1} \leq L\|u_h - u_H\| \leq CH^2,$$

and

$$\|u - u^h\|_1 \leq \|u - u_h\|_1 + \|u_h - u^h\|_1 \leq C(h + H^2).$$

Therefore to obtain optimal approximation in $\mathbb{V}_h$, it suffices to choose $H = h^{1/2}$ which is much bigger than h. For example, in two dimensions, for $h = 1/2^{10}$, $\dim \mathbb{V}_h \approx 2^{20} = 1,048,576$ while $H = 1/2^5, \dim \mathbb{V}_H \approx 2^{10} = 1,024$.

It can be expected that if we use Newton's iteration in the fine grid the result can be further improved.

Algorithm 3.2 (Two-Grid + One Step Newton's method)**.** *Compute $e^h \in \mathbb{V}_h$ such that*

$$J_H e^h = f(u_H) - Au_H \quad \text{in } \mathbb{V}_h', \tag{17}$$

and set

$$u^h = u_H + e^h.$$

To analyze the convergence, we write

$$Au^h = f(u_H) + f'(u_H)(u^h - u_H),$$
$$Au_h = f(u_H) + f'(u_H)(u_h - u_H) + \frac{1}{2}f''(\xi)(u_h - u_H)^2.$$

Therefore we get the error equation

$$J_H(u_h - u^h) = \frac{1}{2}f''(\xi)(u_h - u_H)^2 \quad \text{in } \mathbb{V}_h'. \tag{18}$$

Assume $J_H : H_0^1(\Omega) \to H^{-1}(\Omega)$ is an isomorphism and f'' is uniformly bounded. Using the error estimate (13) and the analysis for the Newton's method, we then obtain

$$\|u_h - u^h\|_1 \leq C\|u_h - u_H\|^2_{L^{12/5}} \lesssim H^4,$$

and by the triangle inequality and error estimate (14)

$$\|u - u^h\|_1 \leq C(h + H^4).$$

To obtain optimal approximation in $\mathbb{V}_h$, it suffices to choose $H = h^{1/4} \gg h$. for $h = 1/2^{10}$, $\dim \mathbb{V}_h \approx 2^{20} = 1,048,576$ while $H = 1/6, \dim \mathbb{V}_H \approx 36$ is enough!

Algorithm 3.3 (Two-Grid + One Step Newton's method in $\mathbb{V}_h$ + One Step Newton's method in $\mathbb{V}_H$)**.** *Let $u^h \in \mathbb{V}_h$ be computed using previous two grid method. Compute $e^H \in \mathbb{V}_H$ such that*

$$J_H e^H = \frac{1}{2}f''(u_H)(u^h - u_H)^2 \quad \text{in } \mathbb{V}_H', \tag{19}$$

and set

$$u_*^h = u^h + e^H.$$

By (18) and (19), we obtain the error equation

$$J_H(u_h - u_*^h) = g \quad \text{in } \mathbb{V}_H', \tag{20}$$

where

$$g := \frac{1}{2}f''(u_H)[(u_h - u_H)^2 - (u^h - u_H)^2] + \frac{1}{6}f'''(\xi)(u_h - u_H)^3.$$

We estimate $\|J_H(u_h - u_*^h)\|_{-1}$ as follows

$$(J_H(u_h - u_*^h), v) = (J_H(u_h - u_*^h), v - v_H) + (g, v_H) = I_1 + I_2.$$

We choose $v_H \in \mathbb{V}_H$ such that

$$\|v - v_H\| \lesssim H\|v\|_1, \text{ and } \|v_H\|_1 \leq \|v\|_1.$$

Then

$$|I_1| \lesssim H\|J_H(u_h - u_*^h)\|\|v\|_1 \leq H\left(\|J_H(u_h - u^h)\| + \|J_H e^H\|\right)\|v\|_1.$$

By (18),

$$\|J_H(u_h - u^h)\| \lesssim \|(u_h - u_H)^2\| \leq \|u_h - u_H\|_{L^4}^2 \lesssim H^4.$$

By (19),

$$\|J_H e^H\| \lesssim \|(u^h - u_H)^2\| \leq \|u^h - u_H\|_{L^4}^2 \leq \|u_h - u^h\|_1^2 + \|u_h - u_H\|_{L^4}^2 \lesssim H^4.$$

Therefore

$$|I_1| \lesssim H^5\|v\|_1.$$

We then estimate I_2 by dividing it into two parts:

$$\begin{aligned}((u_h - u_H)^2 - (u^h - u_H)^2, v_H) &= ((u_h - u^h)(u_h - u_H + u^h - u_H), v_H)\\ &\leq \|u_h - u^h\|_{L^3}\|u_h - u_H + u^h - u_H\|_{L^2}\|v_H\|_{L^6}\\ &\leq \|u_h - u^h\|_1\left(\|u_h - u_H\| + \|u^h - u_H\|\right)\|v_H\|_1\\ &\leq H^6\|v\|_1.\end{aligned}$$

And

$$((u_h - u_H)^3, v_H) \leq \|(u_h - u_H)^3\|_{L^{6/5}}\|v_H\|_{L^6} \lesssim \|u_h - u_H\|_{L^4}^3\|v\|_1.$$

Therefore

$$|I_2| \lesssim H^6\|v\|_1.$$

We then obtain

$$\|u_h - u_*^h\|_1 \lesssim \|J_H(u_h - u_*^h)\|_{-1} \lesssim H^5,$$

and

$$\|u - u_*^h\|_1 \lesssim C(h + H^5).$$

With one more Newton's iteration in the coarse space (which needs very little extra work), we can choose $H = h^{1/5}$! For $h = 1/2^{10}$, $\dim \mathbb{V}_h \approx 2^{20} = 1,048,576$ while $H = 1/2^2, \dim \mathbb{V}_H \approx 16$.

Remark 3.4. The constraint of the two-grid method is that essentially the nonlinearity should be captured by the coarse grid. From this point of view, H cannot be too large.

4. FAS: NONLINEAR MULTIGRID

In this section we introduce the nonlinear multigrid method FAS developed by Brandt for solving the nonlinear equation

$$\mathcal{L}_h(u_h) = 0.$$

Recall that the success of multigrid methods relies on two ingredients:

- High frequency will be damped by smoothers.
- Low frequency can be approximated well on a coarse grid.

We shall first introduce an effective smoother for nonlinear problem: nonlinear Gauss-Seidel iteration and then discuss the most subtle part of nonlinear multigrid method: a perturbated equation to be solved for a coarse grid correction.

4.1. **Nonlinear Gauss-Seidel iteration.** Since the iteration is defined on a fixed space, we will skip the subscript h in this subsection. Recall that the Gauss-Seidel iteration for a linear equation $Au = b$ can be interpret as a subspace correction method based on the canonical decomposition $\mathbb{V} = \sum_{i=1}^{N} \operatorname{span}\{\phi_i\}$. That is we solve the original equation only in the 1-D subspace spanned by one basis function $\mathbb{V}_i = \operatorname{span}\{\phi_i\}$. For non-linear problems, suppose v is the current approximation of the exact solution u, we are looking for correction in subspace $\mathbb{V}_i$ only, i.e., we solve the 1-D nonlinear problem: find $\alpha \in \mathbb{R}$ such that

$$\mathcal{L}(v + \alpha\phi_i) = 0. \tag{21}$$

The nonlinear problem (21) can be solved by, e.g., Newton's method. Given the current approximation α_k, one Newton iteration is obtained by linearization at α_k

$$\mathcal{L}(v + \alpha_k\phi_i) + \mathcal{L}'(v + \alpha_k\phi_i)(\alpha - \alpha_k)\phi_i = 0. \tag{22}$$

Again the weak formulation of (22) is in the form

$$(\mathcal{L}'(v + \alpha_k\phi_i)e_\alpha\phi_i, \phi_i) = -(\mathcal{L}(v + \alpha_k\phi_i), \phi_i). \tag{23}$$

One non-linear Gauss-Seidel (NGS) consists of a loop of all unknowns, i.e., $i = 1 : N$ and solve N 1-D non-linear problems. Note that we need to update residual and Jacobian when moving from i to $i + 1$ but that update is usually local.

When the nonlinear equation $\mathcal{L}(u) = 0$ is the first order equation of an optimization problem $\min_u E(u)$, the 1-D nonlinear Gauss-Seidel iteration in $\mathbb{V}_i$ is equivalent to the optimization problem restricted to one dimensional subspace, i.e.

$$\min_{\alpha\in\mathbb{R}} E(v + \alpha\phi_i).$$

Convergence analysis for convex energy E can be found in [10].

4.2. **Coarse grid problem.** We continue with the optimization problem and introduce a coarse space $\mathbb{V}_H$. Denoted by Pro_H the prolongation from $\mathbb{V}_H$ to $\mathbb{V}_h$. The transpose is the restriction $\mathrm{Res}_h : \mathbb{V}_h' \to \mathbb{V}_H'$ which will be used to restrict the residual to the coarse space $r_H = \mathrm{Res}_h r_h$. Introduce a projection $I_H : \mathbb{V}_h \to \mathbb{V}_H$. A typical choice is the nodal interpolation. Define $v_H = I_H v_h$ as a good approximation of v_h.

For optimization problem, $\mathcal{L}(u) = \nabla E(u)$ and the residual at the current approximation v_h is $r_h = -\nabla E(v_h)$. So $r_H = \mathrm{Res}_h r_h$ is a projection of the negative gradient to the coarse space. We correct the direction r_H by the Hessian of an energy E_H which is a good approximation of E

$$\nabla^2 E_H(v_H)e_H = r_H \tag{24}$$

and update

$$u_h = v_h + \mathrm{Pro}_H e_H = v_h + \mathrm{Pro}_H \left[(\nabla^2 E_H(v_H))^{-1} \mathrm{Res}_h(-\nabla E(v_h))\right].$$

Compare with Newton's method, the Hessian $\nabla^2 E(v_h) \approx \nabla^2 E_H(v_H)$ and the cost to compute the inverse is reduced.

We now write one Newton step (24) as a nonlinear equation problem: find $u_H \in V_H$ s.t.

$$\nabla E_H(u_H) - \nabla E_H(v_H) = r_H. \tag{25}$$

After that set

$$e_H = u_H - v_H.$$

The corresponding optimization problem is

$$\min_{w \in \mathbb{V}_H} E_H(w) - (w, \tau_H), \tag{26}$$

where $\tau_H = \nabla E_H(v_H) + r_H$. In view of nonlinear function $\mathcal{L}$, the coarse grid problem is

$$\mathcal{L}_H(u_H) = \tau_H := \mathcal{L}_H(I_H v_h) - \mathrm{Res}_h \mathcal{L}(v_h). \tag{27}$$

When $\mathcal{L}$ and $\mathcal{L}_H$ are linear, (27) becomes

$$A_H e_H = \mathrm{Res}_h r_h. \tag{28}$$

It is called the *full approximation scheme* (FAS) because the problem in the coarse grid is solved for the full approximation u_H rather than the correction for the linear problem (28). The correction will be obtained through $e_H = u_H - v_H$ and $e_h = I_H e_H$.

A two level FAS scheme is described as follows.

Full Approximation Scheme (FAS): $v \to u$**.**

- Pre-smoothing: $v = \mathrm{NGS}(v, m)$;
- Compute the residual: $r = -\mathcal{L}(v)$;
- Restriction: $r_H = \mathrm{Res}_H r$ and $v_H = I_H v$.
- Solve the coarse grid problem: $\mathcal{L}_H(u_H) = \mathcal{L}_H(v_H) + r_H$.
- Compute the coarse grid correction: $e_H = u_H - v_H$.
- Interpolate the correction and update: $u = v + \mathrm{Pro}_H e_H$.
- Post-smoothing: $u = \mathrm{NGS}(u, m)$.

The nonlinear problem $\mathcal{L}_H(u_H) = \mathcal{L}_H(v_H) + r_H$ on the coarse grid can be solved by recursion and a standard V-cycle is obtained consequently. Also in NGS, there is no need to solve the local nonlinear problem accurately. One gradient or Newton iteration is enough. For convergence analysis, we refer to [3].

Remark 4.1. We end up with comparison with the two-grid method and emphasize the importance of solving a perturbation of the original nonlinear equation $\mathcal{L}_H(u_H) = \tau_H$. If we solve $\mathcal{L}_H(u_H) = 0$ and use u_H as a good initial guess in the fine space, that is the two-grid method in Section 3 which requires $\|u - u_H\|$ is small; see Remark 3.4. Notice that the true solution u may be oscillate or contains high frequency in the fine space so that it may not be well approximated by the coarse grid solution. After the NGS smoother, the correction $e = u - v$ becomes smoother and can be well approximated by the coarse space.

REFERENCES

[1] N. Bićanić and K. H. Johnson. Who was "Raphson"? *Internat. J. Numer. Methods Engrg.*, 14(1):148–152, 1979. 4

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

[3] L. Chen, X. Hu, and S. Wise. Convergence Analysis of the Fast Subspace Descent Methods for Convex Optimization Problems. *Mathematics of Computation*. April 7, 2020. 10

[4] C. Kelley. *Iterative methods for linear and nonlinear equations*. Society for Industrial Mathematics, 1995. 5

[5] P. Knabner and L. Angermann. *Numerical Methods for Elliptic and Parabolic Partial Differential Equations*. Springer, 2003. 4, 5

[6] D. Knoll and D. Keyes. Jacobian-free Newton–Krylov methods: a survey of approaches and applications. *J. Comput. Phys.*, 193(2):357–397, 2004. 6

[7] C. Liu and S. Atluri. A novel time integration method for solving a large system of non-linear algebraic equations. *CMES: Computer Modeling in Engineering and Sciences*, 31:71–83, 2008. 3

[8] M. Marion and J. Xu. Error estimates on a new nonlinear galerkin method based on two-grid finite elements. *SIAM J. Numer. Anal.*, 32(4):1170–1184, 1995. 6

[9] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, New York, 1999. 3, 5, 6

[10] X. Tai and J. Xu. Global convergence of subspace correction methods for convex optimization problems. *Math. Comp.*, 71(237):105–124, 2002. 9

[11] J. Xu. A new class of iterative methods for nonselfadjoint or indefinite problems. *SIAM J. Numer. Anal.*, 29:303–319, 1992. 6

[12] J. Xu. A novel two-grid method for semilinear elliptic equations. *SIAM J. Sci. Comput.*, 15(1):231–237, 1994. 6

[13] J. Xu. Two-grid discretization techniques for linear and nonlinear PDEs. *SIAM J. Numer. Anal.*, 33(5):1759–1777, 1996. 1, 6