Accelerated residual methods for the iterative solution of systems of equations

N. C. Nguyen*, P. Fernandez †, R. M. Freund‡, and J. Peraire§

Abstract

We develop accelerated residual methods for solving linear and nonlinear systems of equations, which leverage both past and recent developments in accelerated gradient methods in convex optimization. First, we propose a modification of Nesterov’s accelerated gradient method to obtain an accelerated residual scheme that can be applied to systems of equations. We show that the scheme can be viewed as a finite difference approximation (FDA) of a second-order ordinary differential equation (ODE) system, which turns out to coincide with the ODE system derived from Nesterov’s method. In practice, our scheme converges faster than Nesterov’s method even though it requires an additional residual evaluation per iteration. Next, we discuss stability properties of our scheme and Nesterov’s method for solving linear systems of equations. We then propose an adaptive restarting and a judicious selection of the acceleration parameter to further improve the (empirical) convergence rate of our scheme. Last of all, we generalize the scheme to encompass a family of accelerated residual methods, thereby providing an opportunity to devise improved methods in future work. We demonstrate the usefulness of our scheme on systems of equations resulting from the finite element approximation of linear and nonlinear partial differential equations (PDEs). On a variety of test cases, our numerical results show that the proposed method outperforms pseudo-time marching method, Nesterov’s method, and Newton-Krylov methods.

*MIT Department of Aeronautics and Astronautics, 77 Massachusetts Ave., Cambridge, MA 02139, USA, email: cuongng@mit.edu
†MIT Department of Aeronautics and Astronautics, 77 Massachusetts Ave., Cambridge, MA 02139, USA, email: pablof@mit.edu
‡MIT Sloan School of Management, 77 Massachusetts Ave., Cambridge, MA 02139, USA, email: rfreund@mit.edu
§MIT Department of Aeronautics and Astronautics, 77 Massachusetts Ave., Cambridge, MA 02139, USA, email: peraire@mit.edu
1 Introduction

In this paper, we are concerned with the iterative solution of a general square system of equations of the form:

$$f(u^*) = 0.$$  (1)

Here $f = (f_1(u), \ldots, f_N(u)) \in \mathbb{R}^N$ is a vector-valued function of the vector $u = (u_1, \ldots, u_N) \in \mathbb{R}^N$, where $N$ denotes the dimension of the system. The system (1) often arises when numerically solving partial differential equations or unconstrained optimization problems (by solving $f(u^*) := \nabla c(u^*) = 0$ for some objective function $c(\cdot)$). The development of iterative methods for solving the system of equations (1) is a very important part of applied mathematics because it has a wide range of applications in engineering and science. Numerous methods have been developed and studied in the literature.

When the vector-valued function has the form $f(u) = Au - b$, where $A \in \mathbb{R}^{N \times N}$ is a matrix and $b \in \mathbb{R}^N$ is a vector, Eq. (1) becomes a linear system of equations. Classical iterative methods such as Jacobi, Richardson, and Gauss-Seidel methods have been used for solving linear systems with some success. A generalization of Gauss-Seidel method led to the successive over-relaxation (SOR) method devised by Young and Frankel [19]. An alternative to classical iterative methods are Krylov subspace methods. One such scheme is the conjugate gradient (CG) method, developed by Hestenes and Stiefel [6]. The CG method is particularly suited for linear systems whose matrix is symmetric and positive-definite. Other Krylov methods for linear systems include CGS [15], BiCGSTAB [18], MINRES [12], GMRES [13], and QMR [4], to name a few.

The most well-known method for solving nonlinear systems of equations is Newton’s method. Newton’s method generates a sequence of linear systems, and Krylov methods are often used to solve them, resulting in the so-called Newton-Krylov methods. For many problems, computing the exact Jacobian, as required by Newton’s method, can be computationally expensive. Instead of constructing the exact Jacobian matrix at every iteration, quasi-Newton methods such as the BFGS method [2, 3, 5, 14] approximate it using low-rank updates. Fixed-point methods such as the Richardson iteration can also be used to solve nonlinear systems of equations without having to compute the Jacobian matrix. The Richardson method is very simple to implement and the cost of each iteration is very low. However, a major drawback of the Richardson method is that it may not converge, or if it does, it may suffer from slow convergence. Therefore, it is of considerable interest to devise iterative methods that have the same cost per iteration and converge significantly faster than the Richardson method.

In optimization theory, the optimality condition of an unconstrained minimization of a differentiable objective function results in the problem (1). Iterative methods mentioned in the preceding paragraph can be used to solve unconstrained optimization problems as well. Newton’s method and BFGS method are second-order methods because they use both the first and second derivatives of the objective function, whereas gradient descent methods are first-order methods. Second-order methods converge in much fewer iterations than first-order methods, but the computational cost per iteration of the former is significantly higher than
that of the latter. It is known that when a gradient descent method is applied to convex optimization problems, it has a worst-case convergence rate of $O(1/k)$ [10].

In a seminal paper published in 1983 [7], Nesterov proposed an accelerated gradient method that exhibits the worst-case convergence rate of $O(1/k^2)$ for minimizing smooth convex functions. Nesterov’s method is an optimal gradient method for convex optimization problems because no method based solely on first-order information can achieve faster convergence rate than $O(1/k^2)$ [10]. Since the introduction of Nesterov’s method, there has been much work on the development of first-order accelerated methods, see [8, 9, 10] for theoretical developments and [17] for a unified analysis of these ideas. One drawback of Nesterov’s method is that it does not converge smoothly, but exhibits oscillatory behavior that can severely slow its convergence. To remedy this problem, O’Donoghue and Candès [11] propose a simple restarting technique to improve the convergence rate of Nesterov’s method. Recently, Su et al. [16] show that Nesterov’s method can be interpreted as a finite difference approximation of a second-order ordinary different equation, thereby providing a better understanding of Nesterov’s method. More recently, Attouch and Peypouquet [1] extend the ODE theory of the work [16] to show that the convergence rate of Nesterov’s method is actually $o(1/k^2)$, rather than $O(1/k^2)$.

The main contributions of this paper are as follows. First, we extend the ideas of accelerated gradient methods in convex optimization to solve linear and nonlinear systems of equations. These methods will be referred to as accelerated residual methods because, in general, the vector $f(\cdot)$ is not the gradient but the residual of a nonlinear system of equations. Second, we introduce an additional residual evaluation at every iteration with respect to existing accelerated gradient methods. This allows us to define an adaptive acceleration parameter that results in faster convergence. Third, we present an interpretation of existing accelerated methods as finite differences approximations (FDA) to a second-order ODE. This provides a means to devising improved methods by simply maximizing the stability of the FDA. And fourth, we generalize our scheme to encompass a family of accelerated methods that correspond to an FDA of an $n$-th order ODE, thereby providing another opportunity to construct superior methods by maximizing the stability of the underlying FDA.

We apply the proposed method as well as a number of other iterative methods to solve systems of equations resulting from the finite element approximation of linear and nonlinear partial differential equations (PDEs). The following empirical conclusions can be drawn from the results of our numerical experiments. First, accelerated schemes converge much faster and thus require significantly less iterations than the standard gradient (forward Euler) scheme. Second, it is found out that restarting is key to improving the convergence rate and robustness of accelerated schemes. For problems in which the systems are nonlinear and nonconvex, they often do not converge without restarting. Third, the scheme we develop in this paper appears more robust than Nesterov’s method since it succeeds in a number of cases where Nesterov’s method fails. Fourth, our choice of the acceleration parameter performs better than the one proposed by Nesterov since it reduces the number of iterations required for convergence to the same error tolerance. And fifth, our scheme is very competitive with Newton-Krylov methods for solving nonlinear systems. Finally, we emphasize that while the
main focus of this paper rests on systems of equations arising from PDEs, our approach can be applied to unconstrained optimization problems as well.

The paper is organized as follows. In Section 2, we give an overview of the development of accelerated gradient methods. In Section 3, we propose an accelerated residual method for solving systems of equations and generalize it to encompass a family of methods. In Section 4, we present numerical results to demonstrate the method on both linear and nonlinear systems resulting from the finite element approximation of PDEs. Finally, in section 5, we end the paper with several remarks on future research.

2 Overview of Accelerated Gradient Methods

2.1 Nesterov’s Method

Before discussing Nesterov’s method, we recall the Richardson iteration for solving the system of equations (1):

\[ M u_{k+1} = M u_k - \alpha_k f(u_k), \quad k \geq 0, \]

where \( M \) is a given preconditioning matrix and \( \alpha_k \) is a stability parameter. The iteration is stopped if the following criterion is met

\[ \|f(u_k)\| < \epsilon, \]

where \( \epsilon \) is a given error tolerance. We easily see that the Richardson iteration is nothing but the forward Euler method applied to the following ODE system:

\[ M \dot{u} + f(u) = 0, \quad u(t = 0) = u_0, \]

in which the time-step size is set to the stability parameter \( \alpha_k \). When the system (1) results from a numerical discretization of PDEs, the stability parameter \( \alpha_k \) is restricted by the Courant-Friedrichs-Levy (CFL) condition to ensure stability and convergence of this method. The CFL condition depends on the underlying PDE, the numerical discretization, and the preconditioning matrix. Although the method is very simple and cost-effective, it can suffer from extremely slow convergence because of the severe restriction on the stability parameter. On the other hand, when solving an unconstrained optimization problem, the iteration (2) is a gradient descent scheme.

In recent years, there has been a considerable interest in accelerated first-order methods driven primarily by their optimal performance for solving convex optimization problems. These methods were first developed by Nesterov in the context of convex optimization and subsequently extended by many other researchers. Nesterov’s method for solving the system (1) takes the following form: Starting with \( u_0 \) and \( u_1 = u_0 - \alpha_0 M^{-1} f(u_0) \), we perform the following iteration

\[ v_k = u_k + \beta_k (u_k - u_{k-1}), \]

\[ M u_{k+1} = M v_k - \alpha_k f(v_k). \]
The first step (5a) is an extrapolation step in which \( v_k \) is determined from the two previous iterates \( u_{k-1} \) and \( u_k \). The second step (5b) is a solution update in which the next iterate \( u_{k+1} \) is computed using \( v_k \) instead of \( u_k \). The parameter \( \alpha_k \) is typically chosen the same as the one in the iteration (2).

The acceleration parameter \( \beta_k \) is crucial because it has a significant impact on the convergence of Nesterov’s method. In the original paper [7], Nesterov suggested to set \( \beta_k = k/(k+3) \). It is recommended to choose \( \alpha_k \leq \alpha \equiv 1/L \), where the Lipschitz constant \( L \) of \( f(\cdot) \) is defined as follows

\[
\| f(x) - f(y) \| \leq L \| x - y \|, \quad \forall x, y \in \mathbb{R}^N,
\]  

assuming that \( f(\cdot) \) is Lipschitz continuous. When being applied to solve smooth convex optimization problems, Nesterov’s method exhibits the worst-case convergence rate [7]

\[
c(u_k) - c(u^*) \leq \frac{2\| u_0 - u^* \|^2}{\alpha(k + 1)^2}.
\]  

Here, \( u^* \) is a minimizer of the convex function \( c(u) \), which is also a solution of the system (1). It is known that the convergence rate \( O(1/k^2) \) is optimal for convex optimization among all methods using only information about the gradient [7]. This is in contrast to the gradient descent method (2), which has the same computational cost but can only achieve the worst-case convergence rate of \( O(1/k) \). The optimal convergence of Nesterov’s method is due to the introduction of the extrapolation step (5a) and the choice of the acceleration parameter \( \beta_k = k/(k+3) \).

In a recent paper [16], Su, Boyd, and Candès, show that Nesterov’s method can be viewed as a finite difference approximation of the following second-order ODE system in the limit of an infinitesimal step size:

\[
M \dddot{u} + \frac{3}{t} M \dot{u} + f(u) = 0,
\]  

with the initial conditions

\[
u(t = 0) = u_0, \quad \dot{u}(t = 0) = 0.
\]  

Indeed, discretizing this ODE system by finite difference yields

\[
M \frac{u(t + \Delta t) - 2u(t) + u(t - \Delta t)}{\Delta t^2} + \frac{3}{t} M \frac{u(t) - u(t - \Delta t)}{\Delta t} + f(v(t)) = 0,
\]  

where

\[
v(t) = u(t) + \left( 1 - \frac{3\Delta t}{t} \right) (u(t) - u(t - \Delta t)).
\]  

This equation is equivalent to

\[
M u(t + \Delta t) = M \left( u(t) + \left( 1 - \frac{3\Delta t}{t} \right) (u(t) - u(t - \Delta t)) \right) - \Delta t^2 f(v(t)).
\]
By setting $u(t + \Delta t) := u_{k+1}$, $u(t) := u_k$, $u(t - \Delta t) := u_{k-1}$, $v(t) := v_k$, and noting $\Delta t/t \approx 1/(k+3)$ in the limit $\Delta t \to 0$, we arrive at

$$Mu_{k+1} = M \left( u_k + \left(1 - \frac{3}{k+3}\right)(u_k - u_{k-1})\right) - \Delta t^2 f(v_k).$$

This is exactly the iteration of Nesterov’s method with $\alpha_k = \Delta t^2$ and $\beta_k = \frac{k}{k+3}$.

Note that the effective step size $\Delta t$ of Nesterov’s method (5) is the square root of the stability parameter $\alpha_k$. Moreover, Su et al. [16] show that the solution of the ODE system (8) satisfies

$$c(u(t)) - c(u^*) \leq \frac{2\|u_0 - u^*\|^2}{t^2}.$$ 

Since $t = k\Delta t = k\sqrt{\alpha}$, this equation yields the convergence rate (7). Furthermore, the number 2 in the numerator of the error bound cannot be replaced with a smaller number [16].

More recently, Attouch and Peypouquet [1] prove that if $\beta_k = k/(k + \gamma)$ for $\gamma > 3$ then the convergence rate of Nesterov’s method is actually $o(1/k^2)$, rather than $O(1/k^2)$. It should be noted that the theoretical results discussed in this section are restricted to convex optimization only.

### 2.2 Restarting strategies for Nesterov’s Method

Unlike the gradient descent method, Nesterov’s method is not guaranteed to be monotone in the objective value. Even for strongly convex optimization problems, Nesterov’s method exhibits periodic oscillatory behavior that can degrade its convergence rate [11, 16]. This behavior can be explained by examining the ODE system (8). The second-order time derivative of this ODE system represents acceleration, while the first-order time derivative represents damping. When $t$ is small, the ODE system is overdamped because the ratio $3/t$ is large. This results in a smooth decrease in the objective value. However, as $t$ increases, the solution of the ODE system becomes more oscillatory because the acceleration effect dominates the damping effect. This leads to oscillatory convergence in the objective value as illustrated in Figure 1. The results in Figure 1 are obtained by applying Nesterov’s method to minimize a convex function $c(u) = 0.02u_1^2 + 0.005u_2^2$ with different values of $\alpha_k$.

To suppress the oscillatory behavior, we can re-solve the ODE system (8) with a new initial condition $u(t = 0) = u_0$, $\dot{u}(t = 0) = 0$, where $u_0$ can be set to the latest solution before the oscillatory behavior appears. This amounts to restarting Nesterov’s method whenever the convergence stagnates or gets worse. Figure 2 illustrates how restarting Nesterov’s method can significantly improve its convergence rate. We see that restarting Nesterov’s method amounts to solving a sequence of ODE systems with different initial states. Whenever the ODE system is restarted, the damping effect becomes dominant again and ensures the smooth convergence of Nesterov’s method.

The restarted version of Nesterov’s method for solving the system (1) is presented in Algorithm 1. The algorithm needs a convergence criterion to stop the iteration as well as a
Figure 1: Convergence trajectory of Nesterov’s method when minimizing the convex function $c(u) = 0.02u_1^2 + 0.005u_2^2$: (a) trajectory of the iterates, (b) zoom in near the optimal solution, and (c) convergence of the error in the objective value. This example is taken from [16].

Figure 2: Demonstration of the oscillatory convergence behavior of Nesterov’s method and the suppression of this behavior by a restarting technique.
Algorithm 1 Nesterov’s method with restarting

0. Start with \( \{\alpha_k, \beta_k\}, u_0 \) and \( k = 0 \)

1. Repeat

2. \( v_k = \begin{cases} u_k, & k = 0 \\ u_k + \beta_k(u_k - u_{k-1}), & k > 0 \end{cases} \)

3. \( u_{k+1} = v_k - \alpha_k M^{-1} f(v_k) \)

4. \( k = k + 1 \)

5. If a convergence criterion is met then exit loop

6. If a restarting condition is met then reset \( u_0 = u_k \) and \( k = 0 \)

7. End Repeat

restarting condition to restart the iteration. One may stop the iteration if \( |c(u_{k+1}) - c(u_k)| \) or \( ||u_{k+1} - u_k|| \) is less than a specified tolerance. A more rigorous convergence criterion would be \( \|f(u_k)\| \leq \epsilon \). However, because this criterion requires an additional evaluation of the residual/gradient vector at every iteration, one may want to use a heuristic convergence criterion instead, such as \( \|f(v_k)\| \leq \epsilon \).

Several restarting strategies were proposed in the literature. In a recent paper [11], O’Donoghue and Candès propose two restarting strategies to improve the convergence rate of Nesterov’s method: (1) the function restarting restarts whenever \( c(u_{k+1}) > c(u_k) \) and (2) the gradient restarting restarts whenever \( f(v_k)^T(u_{k+1} - u_k) > 0 \). The gradient restarting has an advantage over the function restarting in that no extra computation is required since all quantities in the gradient restarting are already calculated. Recently, Su et al. [16] proposed the so-called speed restarting that restarts whenever \( ||u_{k+1} - u_k|| > ||u_k - u_{k-1}|| \). It is shown in [16] that speed restarting can ensure linear convergence of the resulting scheme when minimizing strongly convex functions, albeit at a rate that is usually associated with non-accelerated methods. Figure 3 shows the results obtained by using speed restarting and gradient restarting. We see that both restarting techniques are very effective at suppressing the oscillatory convergence of Nesterov’s method. In summary, while restarting Nesterov’s method does not increase the computational cost, it can significantly improve the actual convergence rate.

3 Accelerated Residual Methods

3.1 The basic scheme

We propose a modification of Nesterov’s method by replacing the extrapolation step \( v_k = u_k + \beta_k(u_k - u_{k-1}) \) with the following one:

\[
\tilde{u}_k = u_k - \alpha_k M^{-1} f(u_k),
\]

\[
v_k = \tilde{u}_k + \beta_k(\tilde{u}_k - u_{k-1}).
\]
Figure 3: Convergence trajectory of Nesterov’s method and restarted Nesterov’s method when minimizing the convex function $c(u) = 0.02u_1^2 + 0.005u_2^2$. Trajectory of the iterates on the left, and convergence of the error in the objective value on the right.

Substituting the first equation into the second one, we get

$$v_k = u_k + \beta_k(u_k - u_{k-1}) - \alpha_k(1 + \beta_k)M^{-1}f(u_k).$$

The update step remains the same as Nesterov’s method. Hence, the proposed scheme has the following iterative structure

$$v_k = u_k + \beta_k(u_k - u_{k-1}) - \alpha_k(1 + \beta_k)M^{-1}f(u_k), \quad (13a)$$

$$u_{k+1} = v_k - \alpha_kM^{-1}f(v_k). \quad (13b)$$

The proposed scheme resembles Nesterov’s method (5) except that its extrapolation step has an additional term $\alpha_k(1 + \beta_k)M^{-1}f(u_k)$.

Compared to Nesterov’s method, our scheme requires one additional evaluation of the residual vector $f(u_k)$ at every iteration. Nevertheless, our scheme gains some important advantages owing to the additional residual evaluation. First, as demonstrated next, the effective time-step size of our scheme is larger than that of Nesterov’s method. Second, we exploit the availability of $f(u_k)$ to define a rigorous convergence criterion and a new restarting strategy. Third, it allows for an adaptive selection of the acceleration parameter $\beta_k$ that can improve the convergence rate of our scheme. Last but not least, we generalize the scheme to obtain a rich family of accelerated residual descent methods, which provides us a valuable tool for devising improved methods.

### 3.2 ODE interpretation of the basic scheme

Let us derive the ODE system for the proposed scheme (13). To this end, we write (13b) as follows

$$u_{k+1} = (u_k + \beta_k(u_k - u_{k-1})) - \alpha_k(1 + \beta_k)M^{-1}f(u_k) - \alpha_kM^{-1}f(v_k).$$
Dividing both sides by $\alpha_k(2 + \beta_k)$ and rearranging the terms, we obtain
\[
\frac{u_{k+1} - 2u_k + u_{k-1}}{\alpha_k(2 + \beta_k)} + \frac{1 - \beta_k}{\sqrt{\alpha_k(2 + \beta_k)}} \frac{(u_k - u_{k-1})}{\sqrt{\alpha_k(2 + \beta_k)}} + M^{-1} \left( \frac{1 + \beta_k}{2 + \beta_k} f(u_k) + \frac{1}{2 + \beta_k} f(v_k) \right) = 0.
\]

Now suppose that we choose $\beta_k$ by the following equation
\[
\beta_k = 1 - \frac{\gamma}{k + 3}
\]
for some $\gamma \geq 0$. Letting $\Delta t_k = \sqrt{\alpha_k(2 + \beta_k)}$, we get
\[
\frac{u_{k+1} - 2u_k + u_{k-1}}{\Delta t_k^2} + \frac{\gamma}{(k + 3)\Delta t_k} \frac{(u_k - u_{k-1})}{\Delta t_k} + M^{-1} \left( \frac{1 + \beta_k}{2 + \beta_k} f(u_k) + \frac{1}{2 + \beta_k} f(v_k) \right) = 0.
\]

In the limit $\Delta t_k \to 0$ the above equation is nothing but a finite difference approximation of the following second-order ODE system
\[
M \ddot{u} + \frac{\gamma}{t} M \dot{u} + f(u) = 0.
\]

We see that this $\gamma$-parametrized ODE system is exactly the same as the ODE system (8) with $\gamma = 3$. This is not surprising because our scheme is constructed based on Nesterov’s method. In the particular case $\gamma = 3$, we have from (14) that $\beta_k = k/(k + 3)$, which is exactly the same value proposed by Nesterov [7]. Choosing a different value for $\gamma$ results in a different value for $\beta_k$ at each iteration. In [16], it is shown that $\gamma = 3$ is the smallest constant by which the optimal convergence rate (11) still holds.

We observe that the effective time-step size of the proposed scheme is $\Delta t_k = \sqrt{\alpha_k(2 + \beta_k)}$, which is $\sqrt{2 + \beta_k}$ larger than the effective time-step size $\sqrt{\alpha_k}$ of Nesterov’s method when the same value of $\alpha_k$ is used for both methods. Our scheme converges faster than Nesterov’s method in terms of the number of iterations, but it requires twice as residual evaluations per iteration as Nesterov’s method. When measuring in terms of the number of residual evaluations, the worst-case convergence rate of our scheme is actually slower than that of Nesterov’s method by a factor of $2/\sqrt{2 + \beta_k}$. However, we empirically observe through the numerical experiments presented in Section 4 that our scheme converges faster than Nesterov’s method in terms of the number of residual evaluations. A rationale for this observation will be presented in Section 4.

### 3.3 Linear stability analysis

As shown in the previous section, both Nesterov’s method and the accelerated residual method can be interpreted as different finite difference approximations of the same ODE, Eq. (16). In order to accelerate convergence, the stability of the finite difference scheme can be maximized so that larger time steps can be used. In this section, we perform a linear stability analysis to determine the stability limit of both schemes in the case of a linear system $f(u) = Au - b = 0$, $A \in \mathbb{R}^{N \times N}$. 
Symmetric positive definite case for large $k$

First, we analyze the asymptotic behavior of the schemes for large $k$ and $A \succ 0$ symmetric positive definite (SPD). Upon diagonalization $A$, the difference equation associated with the accelerated residual method (13) is given by

$$u_{i,k+1} - \left((1 + \beta_k) - \frac{1}{2 + \beta_k} \Delta t^2 \lambda_i \right) u_{i,k} + \beta_k \left(1 - \frac{1}{2 + \beta_k} \Delta t^2 \lambda_i \right) u_{i,k-1} = 0.$$  \hspace{1cm} (17)

Here, $\lambda_i > 0$, $i = 1, ..., N$ denote the eigenvalues of $A$ and $u_{i,k}$ is the $i$-th component of $u_k$ in the basis of eigenvectors. For large $k$, $\beta_k \to 1$ according to (14) and the above difference equation reduces to

$$u_{k+1} - \left(2 - \frac{4}{3} \eta + \frac{2}{9} \eta^2 \right) u_k + \left(1 - \frac{1}{3} \eta \right) u_{k-1} = 0,$$  \hspace{1cm} (18)

where we dropped the subscript $i$ and introduced $\eta := \Delta t^2 \lambda$ to simplify the notation. The characteristic polynomial of this difference equation is then given by

$$P_{\text{ARM}}(\xi) = \xi^2 - \left(2 - \frac{4}{3} \eta + \frac{2}{9} \eta^2 \right) \xi + \left(1 - \frac{1}{3} \eta \right).$$  \hspace{1cm} (19)

The accelerated residual method is stable if all the roots of the characteristic equation $P_{\text{ARM}}(\xi^*) = 0$ lie within the unit circle of the complex plane, i.e., $|\xi^*| \leq 1$. This leads to the following condition on which the maximum time-step size must be satisfied for the accelerated residual method to be stable:

$$(\Delta t_{\text{ARM}}^{\text{max}})^2 \lambda_{\text{max}} \leq \eta_{\text{ARM}}^{\text{max}},$$  \hspace{1cm} (20)

where $\eta_{\text{ARM}}^{\text{max}} = 9/2$ is determined from the requirement that all the roots of the characteristic equation $P_{\text{ARM}}(\xi^*) = 0$ lie within the unit circle of the complex plane.

Similarly, we obtain the difference equation associated with Nesterov’s method as

$$u_{k+1} - \left(2 - 2 \eta \right) u_k + \left(1 - \eta \right) u_{k-1} = 0,$$  \hspace{1cm} (21)

when $\beta_k$ asymptotes to 1. As a result, the maximum time-step size for the stability of Nesterov’s method satisfies the following condition:

$$(\Delta t_{\text{NM}}^{\text{max}})^2 \lambda_{\text{max}} \leq \eta_{\text{NM}}^{\text{max}},$$  \hspace{1cm} (22)

where $\eta_{\text{NM}}^{\text{max}} = 4/3$ is found by requiring that all the roots of the characteristic equation of Eq. (21) lie within the unit circle of the complex plane. Hence, the maximum time-step size of the accelerated residual method can be taken $\sqrt{27/8} \approx 1.837$ times larger than that of Nesterov’s method.

Not only the stability analysis gives us insight into how accelerated first-order schemes work, but also it may offer us means to constructing more stable and faster methods. This is the subject of ongoing research.

11
General case

Next, we generalize the results in the previous section to arbitrary $\beta_k \in [0,1]$ and non-defective $A \in \mathbb{R}^{N \times N}$. Figure 4 shows the stability regions of Nesterov’s method and the accelerated residual method for various values of $\beta_k$. A scheme is stable provided all $\eta_i = \Delta t^2 \lambda_i$, $i = 1, \ldots, N$ lie inside the region of stability. From this figure, both schemes perform better for non-symmetric systems, i.e. when $A$ has complex eigenvalues, if $\beta_k$ is small. This provides insight to devise efficient methods for non-convex optimization and system of equations arising from the discretization of convection-diffusion operators. Also, Nesterov’s method and the accelerated residual method reduce to the Richardson iteration when $\beta_k = 0$. As such, the stability region of the forward Euler method is recovered and $\eta$ is identified with $\eta := \lambda \Delta t$ in this case. We note that the two Richardson steps per ARM iteration are responsible for the factor of two increase in the size of the stability region in ARM with respect to that of forward Euler.

![Figure 4: Stability regions of Nesterov’s method and the accelerated residual method for various values of $\beta_k$.](image)

3.4 Adaptive restarting and stopping criterion

Next, we introduce a restarting strategy and a convergence criterion into our basic scheme. The goal is to ensure that the magnitude of the residual is reduced in every iteration and that if it is smaller than a specified tolerance then the iteration stops. By making use of the already-computed residual vector $f(u_k)$, we can achieve this goal without additional computations. In particular, the iteration stops successfully if $\|f(u_k)\| < \epsilon$ and restarts if

$$\|f(u_k)\| > \|f(u_{k-n})\|, \quad (23)$$

For $\beta_k = 0$, Nesterov’s method and the accelerated residual method are consistent to the first-order ODE system $\dot{u} + f(u) = 0$. 

---

1For $\beta_k = 0$, Nesterov’s method and the accelerated residual method are consistent to the first-order ODE system $\dot{u} + f(u) = 0$. 

12
for some integer $n \geq 1$. The convergence criterion and restarting strategy do not require extra computation since all the quantities involved are already calculated in our scheme. The resulting iteration is summarized in Algorithm 2. We note that the restarting strategy (23) ensures the residual norm decreases monotonically throughout the iteration. For this reason, we will refer to it as Accelerated Residual Descent Method (ARDM).

**Algorithm 2** Accelerated Residual Descent Method

0. Start with $\{\alpha_k, \beta_k\}, u_0, \epsilon, n$, and $k = 0$

1. Repeat

2. $\tilde{u}_k = u_k - \alpha_k M^{-1} f(u_k)$

3. $v_k = \begin{cases} \tilde{u}_k, & k = 0 \\ u_k - \beta_k (\tilde{u}_k - u_{k-1}), & k > 0 \end{cases}$

4. $u_{k+1} = v_k - \alpha_k M^{-1} f(v_k)$

5. $k = k + 1$

6. **If** $\|f(u_k)\| < \epsilon$ **then** exit loop

7. **If** $\|f(u_k)\| > \|f(u_{k-n})\|$ **then** reset $u_0 = u_{k-1}$ and $k = 0$

8. End Repeat

It remains to choose $\alpha_k$ and $\beta_k$ in Algorithm 2. These parameters are crucial to the convergence rate of the method and should be related to mathematical properties of the vector-valued function $f(u)$. On the one hand, $\alpha_k$ must be chosen to satisfy the CFL stability condition of the forward Euler method when numerically solving PDEs. As such, $\alpha_k$ depends on both the underlying PDE and the particular numerical discretization. On the other hand, when solving convex optimization problems, $\alpha_k$ must be chosen such that $\alpha_k \leq \alpha \equiv 1/L$, where the Lipschitz continuity constant $L$ is defined in Eq. (6). The Lipschitz constant $L$ can be estimated by using backtracking line search. Typically, $\alpha_k$ is set to $1/L$. The selection of $\beta_k$ will be discussed next.

### 3.5 Adaptive selection of the acceleration parameter

When minimizing a strongly convex function $c(u)$, it is known that the following choice of the acceleration parameter

$$\beta_k = \frac{1 - \sqrt{\frac{\mu}{L}}}{1 + \sqrt{\frac{\mu}{L}}}$$

(24)

guarantees the linear convergence rate of Nesterov’s method [10, 11], that is,

$$c(u) - c(u^*) \leq L \left(1 - \sqrt{\frac{\mu}{L}}\right)^k \|u_0 - u^*\|^2.$$  

(25)

Here $\mu$ is the strong convexity parameter, defined as the maximum value that satisfies

$$c(u) \geq c(u^*) + \frac{\mu}{2} \|u - u^*\|^2, \quad \forall u \in \mathbb{R}^N.$$  

(26)
However, this linear convergence is only guaranteed when the function parameters $\mu$ and $L$ are known in advance. Estimating $\mu$ is much more challenging than estimating $L$. It is observed that slightly over or underestimating the optimal value of $\mu$ can have a detrimental effect on the rate of convergence of Nesterov’s method [11]. In the absence of a good estimate of the strong convexity parameter $\mu$, $\beta_k$ can be set to $\frac{k}{k+3}$. It is important to point out that the formula (24) is only applicable to systems of equations which arise from strongly convex optimization problems.

Instead of $\beta_k = \frac{k}{k+3}$, we propose the following formula for $\beta_k$:

$$
\beta_k = \frac{\|f(u_k)\|}{\|f(u_{k-1})\|}. 
$$

This formula is motivated by our intuition regarding the ODE system (16). In particular, whenever the convergence is relatively slow we want $\beta_k$ close to 1 so as to speed up the convergence. When the convergence becomes faster, we would like $\beta_k$ small in order to maintain a stable convergence. Also, setting $\beta_k$ to the square of the proposed expression can be interpreted as an attempt to make the new search direction $(\nabla^2 c)$-conjugate to the previous one in the context of convex optimization. It is indeed not coincidental the similarity with the orthogonalization parameter in the conjugate gradient method. We note that the formula (27) together with our restarting strategy ensures $\beta_k \leq 1$ with $n = 1$. It is important to point out that if we use the formula (27) for Nesterov’s method, it would require an additional residual evaluation per iteration and thus increase the computational cost of Nesterov’s method by a factor of 2. For the proposed method, it does not require extra computation since the residual norms are already calculated in our scheme.

We present in Figure 5 the results for minimizing $c(u) = 0.02u_1^2 + 0.005u_2^2$. We see that our method converges faster than both Nesterov’s method and its restarted variant. The convergence rate is further improved when switching from $\beta_k = k/(k+3)$ to $\beta_k = \|f(u_k)\|/\|f(u_{k-1})\|$. This convergence improvement mainly comes from the modification of Nesterov’s scheme (13) and the choice of the acceleration parameter (27).

### 3.6 Generalization

Here we generalize these ideas somewhat. All the methods discussed in this paper belong to the following family of methods:

\[ v_k = \sum_{i=1}^{n} (a_{ik}u_{k+1-i}) + b_{ik}M^{-1}f(u_{k+1-i}), \]

\[ u_{k+1} = v_k - \alpha_k M^{-1}f(v_k). \]  

In particular, we have $n = 1$, $a_{1k} = 1$ and $b_{1k} = 0$ for the forward Euler method. We also have $n = 2$, $a_{1k} = (1 + \beta_k)$, $a_{2k} = -\beta_k$, and $b_{1k} = b_{2k} = 0$ for Nesterov’s method. And we have $n = 2$, $a_{1k} = (1 + \beta_k)$, $a_{2k} = -\beta_k$, $b_{1k} = \alpha_k(1 + \beta_k)$, and $b_{2k} = 0$ for the accelerated residual method. Since the extrapolation step (28a) can be interpreted as an attempt to
accelerate the iterative process using the two subspaces \{u_{k+1-i}\}_{i=1}^n and \{f(u_{k+1-i})\}_{i=1}^n, the n-step extrapolation — as opposed to the one-step extrapolation in the case of Nesterov’s method and accelerated residual method — could yield improved performance.

Note that the forward Euler method is a finite difference approximation (FDA) of the first-order ODE system (4), while the accelerated residual method and Nesterov’s method are FDAs of the second-order ODE system (8). Therefore, the generalized scheme (28) may be viewed as a FDA of an n-order ODE system. Obviously, the parameters \(a_{ik}\) and \(b_{ik}\) play a crucial role in the convergence and stability of the resulting method. We believe that there are some other choices of these parameters that may lead to superior methods. In future work, we will focus on devising new methods along this line of ideas.

4 Numerical Experiments

In this section, we demonstrate and compare the performance of various iterative methods on a wide variety of problems arising from the continuous Galerkin finite element approximation of Poisson equation, linear convection-diffusion equation, nonlinear elliptic equation, and nonlinear convection-diffusion equation. For simplicity of exposition, we will take \(M\) to be an identity matrix in all the examples. Future work will investigate how to determine the preconditioning matrix \(M\) for improving the convergence rate. Furthermore, we will set \(\beta_k = k/(k+3)\) for Nesterov’s method and its gradient restarting variant, as well as \(\beta_k = \|f(u_k)\|/\|f(u_{k-1})\|\) for the accelerated residual descent method. We will also set \(\alpha_k\) to a fixed value \(\alpha\).
In all the examples, we will evaluate the performance of all methods based on the Euclidean norm of the residual vector as a function of the number of residual evaluations. We will set the error tolerance as $\epsilon = 10^{-8}$. A method is said to perform better than the other if the former requires a smaller number of residual evaluations than the latter to converge to the prescribed tolerance.

4.1 Poisson equation

We consider the following Poisson problem

$$-\Delta u = 2\pi^2 \sin(\pi x) \sin(\pi y), \quad \text{in } \Omega = (0, 1) \times (0, 1),$$  \hspace{1cm} (29)

with a Dirichlet boundary condition $u = 0$ on $\partial\Omega$. The finite element discretization leads us to solve the following linear system

$$Au^* = b,$$  \hspace{1cm} (30)

where the stiffness matrix $A \in \mathbb{R}^{N \times N}$ is a symmetric positive-definite (SPD) matrix. The solution of the above linear system is the minimizer of the following convex minimization problem

$$\min_{v \in \mathbb{R}^N} \frac{1}{2} v^T A v - v^T b.$$  \hspace{1cm} (31)

The gradient vector of the objective function is thus given by

$$g(v) = Av - b,$$  \hspace{1cm} (32)

which is the same as the residual vector $f(u) = Au - b$. Note that our finite element approximation employs $n_{\text{elem}}$ uniform triangular elements of the polynomial degree $p$. Hence, the problem size $N$ depends on the polynomial degree $p$ and the number of elements $n_{\text{elem}}$.

Let us now assess the methods described herein for the solution of this problem and compare their performance with that of the conjugate gradient (CG) method. Figure 6 shows the performance of all the methods for different values of $(p, n_{\text{elem}}, \alpha)$. There are a number of interesting observations. We see that the number of residual evaluations required for convergence to a given tolerance increases with the polynomial degree and the number of elements due to the increase in the condition number $L/\mu = \lambda_{\text{max}}/\lambda_{\text{min}}$. Although Nesterov’s method exhibits the periodic behavior discussed earlier, it converges much faster than the forward Euler method. The gradient restarting Nesterov’s method yields a significant improvement in the convergence rate. Our method outperforms Nesterov’s method and its gradient restarting variant in all cases. Also, we see that the Nesterov’s method and its gradient restarting variant diverge when $p = 3$ and $\alpha = 0.1$, whereas our method converges in those cases. This is consistent with the improved stability of our scheme, as shown in Section 3.3. Finally, the CG method has the best performance for this particular problem since it is the method of choice for solving linear systems whose matrix is symmetric positive definite. This is not surprising since all the methods generate iterates in the Krylov subspace $\mathcal{K}_k(A, b)$, and the CG method takes at every iteration the minimizer of the $A$-norm of the error in $\mathcal{K}_k(A, b)$. 
4.2 Linear convection-diffusion problem

In the second example, we consider a linear convection-diffusion problem of the form

$$-\Delta u + c_x \frac{\partial u}{\partial x} + c_y \frac{\partial u}{\partial y} = 10, \quad \text{in } \Omega = (0, 1) \times (0, 1),$$

(33)

with a Dirichlet boundary condition $u = 0$ on $\partial \Omega$, where $(c_x, c_y)$ is the convective velocity field. We use the continuous Galerkin finite element method to discretize the above equation on a uniform mesh of $n_{\text{elem}} = 800$ triangular elements and polynomial degree $p = 3$. This spatial discretization results in the following linear system

$$Au^* = b,$$

(34)

where $A \in \mathbb{R}^{N \times N}$ is a nonsymmetric matrix whenever the convective velocity is nonzero. Figure 7 depicts the numerical solution for $(c_x, c_y) = (1, 1)$ and $(c_x, c_y) = (50, 50)$; which correspond to a diffusion-dominated case and a convection-dominated case, respectively.

Of course, the solution of the above linear system satisfies the following minimization problem

$$\min_{v \in \mathbb{R}^N} \frac{1}{2}(Av - b)^T(Av - b).$$

(35)

The gradient vector of the objective function is given by

$$g(v) = A^TAv - A^Tb,$$

(36)
which differs from the residual vector $f(u) = Au - b$. Note that Nesterov’s method is intended to solve the normal equation $g(u^*) = 0$ instead of the residual equation $f(u^*) = 0$. When applied to the normal equation $g(u^*) = 0$, the maximum step size for stability reduces dramatically due to the squaring of the Lipschitz constant and Nesterov’s method becomes impractical. When applied to the residual equation $f(u^*) = 0$, however, there is no theoretical guarantee on how the method will perform. Also, since the matrix $A$ is not SPD, the CG method is not suited to solve this problem. The conjugate gradient method on the normal equations (CGN) and the generalized minimum residual method (GMRES), however, are well-established methods for nonsymmetric linear systems of equations.

Figure 8 and Figure 9 show the performance of all the methods for the diffusion-dominated case $c_x = c_y = 1$ and the convection-dominated case $c_x = 50$, respectively. In these figures, Nesterov’s method is applied to the residual equation. (When applied to the normal equation, Nesterov’s method rapidly diverges for all the values of $\alpha$ considered.) We see that our method converges, while Nesterov’s method and its gradient restarting variant diverge when $\alpha$ is set to 0.1. When $\alpha$ is reduced to 0.05, Nesterov’s method and its gradient restarting variant are able to converge in the diffusion-dominated case. Our method is less sensitive to the variation of the stability parameter $\alpha$ and more robust than Nesterov’s method. Furthermore, it is interesting to note that the performance of our method is better in the convection-dominated case than in the diffusion-dominated case. In particular, our method requires roughly 400 residual evaluations to converge to the error tolerance $10^{-8}$ in the convection-dominated case, whereas it requires more than 800 residual evaluations in the diffusion-dominated case. This is in contrast to Nesterov’s method, which appears to perform better in the diffusion-dominated regime.

GMRES has the best performance among all the methods. We emphasize that our method is capable of solving both linear and nonlinear systems, whereas GMRES is de-
Figure 8: Comparison among the iterative methods for solving for the convection-diffusion equation with $c_x = c_y = 1$.

Figure 9: Comparison among the iterative methods for solving for the convection-diffusion equation with $c_x = c_y = 50$. 
signed to solve linear systems. In order to solve nonlinear systems, GMRES is coupled with
the Newton’s method, giving rise to the so-called Newton-GMRES method. In the next
two examples, we will compare our method against the Newton-GMRES and Newton-CGN
methods for solving nonlinear PDEs.

## 4.3 Nonlinear elliptic problem

In the third example, we consider a nonlinear elliptic problem of the form

$$-\nabla \cdot ((1 + u^2) \nabla u) + u = 4\pi^2, \quad \text{in } \Omega = (0, 1) \times (0, 1),$$

(37)

with a Dirichlet boundary condition $u = 0$ on $\partial\Omega$. The weak formulation of the finite element
method is to find $u_h \in V_h$ such that

$$\int_{\Omega} \left( (1 + u_h^2) \nabla u_h \cdot \nabla v + u_h v \right) \, dx \, dy - 4\pi^2 \int_{\Omega} v \, dx \, dy = 0, \quad \forall v \in V_h,$$

(38)

where $V_h = \{ v \in C_0(\Omega) : v|_K \in P^3(K), \forall K \in T_h, \text{ and } v = 0 \text{ on } \partial\Omega \}$ is the finite element
space. Here $T_h$ is a finite element mesh of 200 triangular elements and $P^3(K)$ is the space of
polynomials of degree at most 3 on $K \in T_h$. Solving the weak formulation (38) corresponds
to finding the solution of the following nonlinear system

$$f(u^*) = 0,$$

(39)

where $u^* \in \mathbb{R}^N$ is the vector of degrees of freedom of $u_h$, and $f \in \mathbb{R}^N$ is the residual
vector whose entries $f_k$ are obtained by setting $v$ in the weak formulation (38) to the basis
functions of the space $V_h$. The initial vector $u_0$ is set to the numerical solution of a linear
elliptic problem in which the nonlinear term $u^2$ is removed from the equation (37).

Figure 10 shows the convergence of various schemes for the stability parameter $\alpha = 0.015$.
We observe that the accelerated residual descent method converges slightly better than the
gradient restarting Nesterov’s method, and that the two methods perform much better than
the forward Euler method and Nesterov’s method. They take a little more than 600 residual
evaluations to converge to a tolerance of $\epsilon = 10^{-8}$, while forward Euler method and Nes-
terov’s method take many thousands of residual evaluations. Table 1 shows the convergence
of the Newton-CGN and Newton-GMRES methods. In each Newton iteration, CGN and
GMRES are used to solve a linear system resulting from the linearization of the weak for-
mulation (37). We note that the CGN method needs to be used instead of the CG method due
to the presence of the nonlinear term. The number of CGN and GMRES iterations required
for solving the linear system with a prescribed tolerance of $10^{-8}$ is also tabulated in Table 1.
We observe that Newton-CGN fails to converge at the sixth Newton iteration. In particular,
the condition number of the normal equation is too large to converge the nonlinear problem
to the prescribed tolerance in double precision. Newton-GMRES, however, converges to a
residual norm of $2.1354 \times 10^{-12}$ after six Newton iterations. Note that the total number
of GMRES iterations / matrix-vector products is 692. In general, matrix-vector product is
more computationally expensive than the residual vector evaluation. Furthermore, GMRES requires the orthogonalization of the Krylov subspace, which adds significantly to the overall computational cost of the Newton-GMRES method. In this example, our method outperforms the Newton-CGN and Newton-GMRES methods. However, Newton-GMRES is highly sensitive to the prescribed GMRES tolerance and the initial guess. Using a higher GMRES tolerance and a better initial guess could make Newton-GMRES perform better than our method.

Table 1: Convergence of the Newton-CGN and Newton-GMRES methods for the nonlinear elliptic problem.

<table>
<thead>
<tr>
<th>Newton iteration</th>
<th>Residual norm</th>
<th># CGN iterations</th>
<th># GMRES iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.9017 × 10^9</td>
<td>841</td>
<td>130</td>
</tr>
<tr>
<td>2</td>
<td>9.7774 × 10^-1</td>
<td>841</td>
<td>113</td>
</tr>
<tr>
<td>3</td>
<td>1.3988 × 10^-1</td>
<td>712</td>
<td>107</td>
</tr>
<tr>
<td>4</td>
<td>4.1785 × 10^-3</td>
<td>676</td>
<td>107</td>
</tr>
<tr>
<td>5</td>
<td>3.6006 × 10^-6</td>
<td>457</td>
<td>106</td>
</tr>
<tr>
<td>6</td>
<td>2.1354 × 10^-12</td>
<td>no convergence</td>
<td>129</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>–</td>
<td>692</td>
</tr>
</tbody>
</table>
4.4 Nonlinear convection-diffusion problem

In the last example, we consider solving a Burgers problem

\[-\nabla \cdot (\kappa \nabla u) + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = s, \quad \text{in } \Omega = (0, 1) \times (0, 1), \tag{40}\]

with \(u = 0\) on \(\partial \Omega\) and \(\kappa = 0.02\). The source term \(s\) is specified such that the problem has the following exact solution

\[u = xy \tanh \left( \frac{1-x}{\kappa} \right) \tanh \left( \frac{1-y}{\kappa} \right). \tag{41}\]

The weak formulation of the finite element method is to find \(u_h \in V_h\) such that

\[\int_\Omega \left( \kappa \nabla u_h \cdot \nabla v + u_h \frac{\partial u_h}{\partial x} v + u_h \frac{\partial u_h}{\partial y} v \right) \, dx dy - \int_\Omega s v \, dx dy = 0, \quad \forall \, v \in V_h, \tag{42}\]

where \(V_h = \{ v \in C_0(\Omega) : v|_K \in P^3(K), \forall K \in T_h, \text{and } v = 0 \text{ on } \partial \Omega \}\) with a basis set \(\{ \phi_k \}_{n=1}^{N=7921}\) and \(T_h\) is the finite element mesh of 1800 triangular elements. In this particular example, the initial solution \(u_0\) is set to the solution of a linear elliptic problem in which the nonlinear convective terms are removed from the equation (40).

![Figure 11: Performance of the four different schemes for the Burgers problem.](image)

Figure 11 shows the convergence of the four different schemes for two different values of the stability parameter \(\alpha\). We observe that the accelerated residual descent method needs 700 residual evaluations to converge to a tolerance of \(\epsilon = 10^{-8}\) for \(\alpha = 5\), while both Nesterov’s method and its gradient restarting variant do not converge. When \(\alpha\) is reduced from 5 to 2.5, the gradient restarting Nesterov’s method converges to the said tolerance with more than 1000 residual evaluations, while Nesterov’s method still does not converge. Table 2 shows the convergence of the Newton-CGN and Newton-GMRES methods, together with the number...
of CGN and GMRES iterations required for solving the linear system with a prescribed tolerance of $10^{-8}$. Again, Newton-CGN fails to converge at the fifth Newton iteration. The Newton-GMRES method, however, converges to a residual norm of $4.0558 \times 10^{-10}$ after six Newton iterations, requiring a total of 1461 matrix-vector products. This is roughly 2 times more than the number of residual evaluations of the accelerated residual descent method. In this example, our method outperforms the gradient restarting Nesterov’s method, the Newton-CGN method, and the Newton-GMRES method.

Table 2: Convergence of the Newton-CGN and Newton-GMRES methods for the Burgers problem.

<table>
<thead>
<tr>
<th>Newton iteration</th>
<th>Residual norm</th>
<th># CGN iterations</th>
<th># GMRES iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$3.8045 \times 10^{-2}$</td>
<td>3135</td>
<td>229</td>
</tr>
<tr>
<td>2</td>
<td>$8.8212 \times 10^{-3}$</td>
<td>3862</td>
<td>233</td>
</tr>
<tr>
<td>3</td>
<td>$1.5580 \times 10^{-3}$</td>
<td>3529</td>
<td>234</td>
</tr>
<tr>
<td>4</td>
<td>$1.2478 \times 10^{-4}$</td>
<td>3278</td>
<td>249</td>
</tr>
<tr>
<td>5</td>
<td>$1.6119 \times 10^{-6}$</td>
<td>no convergence</td>
<td>253</td>
</tr>
<tr>
<td>6</td>
<td>$4.0558 \times 10^{-10}$</td>
<td>–</td>
<td>263</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>–</td>
<td>1461</td>
</tr>
</tbody>
</table>

5 Conclusions

We have presented accelerated residual methods for solving linear and nonlinear systems of equations. The methods are based upon an extension of Nesterov’s method by introducing an additional residual term to the extrapolation step, and adaptively choosing the acceleration parameter. Just like Nesterov’s method, the proposed scheme can be cast as a finite difference approximation of the same second-order ODE system. However, it has a larger step size and thus a faster convergence than Nesterov’s method. Numerical experiments presented in this paper demonstrate that our method outperforms Nesterov’s method and its gradient restarting variant on many different examples. Furthermore, the results show that our method is very competitive with Newton-Krylov methods for solving nonlinear systems.

We have focused on solving systems of equations arising from the numerical discretization of PDEs. We believe that the method can be used to solve systems of equations in other contexts. For instance, just like Nesterov’s method, the proposed method can be used to solve systems of equations arising from the optimality condition of unconstrained convex optimization problems. This is a subject of future research.

We also proposed a generalization of our scheme to encompass a family of accelerated residual methods. It will be interesting to find new powerful schemes by maximizing their stable timestep sizes based on linear stability analysis. Future work will investigate acceler-
ated residual methods that discretize the ODE system

\[ M\ddot{u} + \gamma t C\dot{u} + f(u) = 0, \]  

instead of Eq. (16). When applied to linear systems of equations, the finite difference approximation of this ODE system would generate iterates outside the Krylov subspace, thereby bringing an opportunity to go beyond Krylov methods such as CG and GMRES. Another line of research is how to determine an easy-to-invert matrices, \( M \) and \( C \), so as to improve the convergence rate. The matrix \( M \) was taken to be an identity matrix in all the examples. This worked well because we considered the discretization of scalar PDEs on isotropic uniform meshes. How do we develop better methods? How would we choose \( M \) and \( C \) if the finite element meshes are stretched and anisotropic or if the material properties are anisotropic? These questions are interesting and worthy of investigation.

Acknowledgements

The authors would like to acknowledge AFOSR Grants No. FA9550-15-1-0276 and FA9550-16-1-0214 for supporting this work. The second author’s research is partially supported by “la Caixa” Foundation Graduate Studies Fellowship. The third author also acknowledges support from the MIT-Belgium Université Catholique de Louvain Fund. Finally, we thank Mohamed Aziz Bhouri for his useful comments and suggestions during the review process.

References


