ESAIM: M2AN 43 (2009) 689–708
DOI: 10.1051/m2an/2009025

ESAIM: Mathematical Modelling and Numerical Analysis
www.esaim-m2an.org

GRADIENT DESCENT AND FAST ARTIFICIAL TIME INTEGRATION

Uri M. Ascher[1], Kees van den Doel[1], Hui Huang[2] and Benar F. Svaiter[3]

Abstract. The integration to steady state of many initial value ODEs and PDEs using the forward Euler method can alternatively be considered as gradient descent for an associated minimization problem. Greedy algorithms such as steepest descent for determining the step size are as slow to reach steady state as is forward Euler integration with the best uniform step size. But other, much faster methods using bolder step size selection exist. Various alternatives are investigated from both theoretical and practical points of view. The steepest descent method is also known for the regularizing or smoothing effect that the first few steps have for certain inverse problems, amounting to a finite time regularization. We further investigate the retention of this property using the faster gradient descent variants in the context of two applications. When the combination of regularization and accuracy demands more than a dozen or so steepest descent steps, the alternatives offer an advantage, even though (indeed *because*) the absolute stability limit of forward Euler is carefully yet severely violated.

Mathematics Subject Classification. 65F10, 65F50.

Received July 17, 2008.
Published online July 8, 2009.

1. Introduction

The numerical integration of a time dependent partial differential equation (PDE) system in order to approximate its steady state is by no means a new technique; see, *e.g.*, [4,24,28,36] and references therein. In particular, it is a lazy-person's approach for solving possibly nonlinear elliptic PDEs: although well-known to yield slow algorithms, it has the advantages that setting up an explicit integration scheme is straightforward, certain singularities (*e.g.* of Poisson's equation with Neumann boundary conditions) are automatically overcome, and multiple solutions may be found – with luck or insight – upon selecting different initial value functions.

To focus our discussion consider the simple linear PDE

$$u_t = \nabla \cdot (a\ \boldsymbol{\nabla} u) + q, \quad 0 < x, y < 1,\ t \geq 0, \tag{1.1}$$

where $a = a(x, y) \geq a_{\min} > 0$ and $q = q(x, y)$, subject to Dirichlet boundary conditions (BC) that are fixed in time t. Suppose we discretize the unit square by a uniform mesh with mesh width $h = 1/(J + 1)$.

Keywords and phrases. Steady state, artificial time, gradient descent, forward Euler, lagged steepest descent, regularization.

[1] Department of Computer Science, University of British Columbia, Vancouver, Canada. ascher@cs.ubc.ca; kvdoel@cs.ubc.ca

[2] Department of Mathematics, University of British Columbia, Vancouver, Canada. hhzhiyan@math.ubc.ca

[3] Institute of Pure and Applied Mathematics (IMPA), Rio de Janeiro, Brazil. benar@impa.br

Article published by EDP Sciences

Denoting the approximation to $u(t, ih, jh)$ by $v_{i,j}(t)$, a standard finite volume approach yields the semi-discretization

$$\begin{aligned}\frac{\mathrm{d}v_{i,j}}{\mathrm{d}t} &= h^{-2}\left[a_{i+1/2,j}(v_{i+1,j}-v_{i,j}) - a_{i-1/2,j}(v_{i,j}-v_{i-1,j})\right.\\ &\left. + a_{i,j+1/2}(v_{i,j+1}-v_{i,j}) - a_{i,j-1/2}(v_{i,j}-v_{i,j-1})\right] + q_{i,j}, \quad 1 \le i,j \le J, \end{aligned} \tag{1.2a}$$

where $a_{i+1/2,j}$ is defined using local values of a, for instance

$$a_{i+1/2,j} = h\left[\int_{x_i}^{x_{i+1}} a^{-1}(x, y_j)\mathrm{d}x\right]^{-1}. \tag{1.2b}$$

See, *e.g.*, [2]. The ODE system (1.2) is closed by the BC. We can write it as

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = -(A\mathbf{x} - \mathbf{b}), \tag{1.3}$$

where $\mathbf{x}$ contains the unknowns $v_{i,j}$ reshaped as a vector of length $m = J^2$, $\mathbf{b}$ likewise contains contributions of the inhomogeneities from $q_{i,j}$ and the BC, and A is a symmetric positive definite matrix. We next wish to integrate the system (1.3) to steady state $\mathbf{x} = \mathbf{x}(\infty)$ satisfying $A\mathbf{x} = \mathbf{b}$.

Now, denoting the eigenvalues of A by $\lambda_1 > \lambda_2 > \ldots > \lambda_m > 0$ and $\kappa = \mathrm{cond}(A) = \frac{\lambda_1}{\lambda_m}$, we have $\lambda_1 = O(h^{-2})$, $\lambda_m = O(1)$, hence $\kappa = O(m)$. Thus, the ODE system is mildly stiff, and this brings to mind implicit time discretization. However, the systems of algebraic equations obtained at each step are not very different from what we face solving the steady state equations directly, so let us concentrate on explicit schemes. Here the reigning method in applications is forward Euler

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{r}_k, \quad \text{where } \mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k. \tag{1.4}$$

It is the simplest ODE method, and in the present context its low accuracy order is less of an issue since approximating the time-dependent trajectory well is hardly important for the purpose of obtaining the steady state solution.

The next question is how to select the forward Euler step size α_k. A constant step size α must satisfy the absolute stability requirement

$$\alpha \le 2/\lambda_1. \tag{1.5}$$

It is easy to see that in general the number of steps required to reduce the residual norm $\|\mathbf{r} = \mathbf{b} - A\mathbf{x}\|$ by a constant amount using such a step size is at best proportional to $\kappa = O(m)$; see Theorem 2.2 below. Some significantly larger step sizes must occasionally be taken to obtain faster convergence to steady state!

For the latter purpose let us reinterpret the forward Euler scheme (1.4) as a gradient descent method for the strongly quadratic minimization problem

$$\min_{\mathbf{x}} f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{b}^T\mathbf{x}, \tag{1.6}$$

whose necessary and sufficient conditions coincide with the steady state of (1.3). The question now boils down to selecting the step size for the gradient descent method. Usual strategies are steepest descent (SD)

$$\alpha_k^{SD} = \frac{\mathbf{r}_k^T\mathbf{r}_k}{\mathbf{r}_k^T A\mathbf{r}_k}, \tag{1.7a}$$

TABLE 1. Iteration counts for the model Poisson problem using gradient descent with different step size choices.

m	SD	OM	HM	SD/OM	RSDOM	LSD	HLSD	CG
49	167	169	169	46	57	40	59	9
225	702	696	698	88	126	72	67	24
961	2859	2811	2819	276	311	240	142	50
3969	11 517	11 279	11 299	878	682	356	590	100

or Orthomin (OM) (see, *e.g.*, [15])

$$\alpha_k^{OM} = \frac{\mathbf{r}_k^T A \mathbf{r}_k}{\mathbf{r}_k^T A^T A \mathbf{r}_k}. \tag{1.7b}$$

Both of these are greedy algorithms. The first yields the largest decrease in $f(\mathbf{x}_k + \alpha \mathbf{r}_k)$ for $\alpha = \alpha_k^{SD}$, and the second yields the largest decrease in $\|\mathbf{r}\|^2 = \|\mathbf{b} - A(\mathbf{x}_k + \alpha \mathbf{r}_k)\|^2$ for $\alpha = \alpha_k^{OM}$[4]. The OM variant also yields reduction in f at each step. The guaranteed norm decrease yields stability even though the forward Euler constant step size bound (1.5) is occasionally violated.

Another possible step size selection is the harmonic mean (HM) of the preceding two expressions

$$\alpha_k^{HM} = \frac{2}{\frac{\mathbf{r}_k^T A \mathbf{r}_k}{\mathbf{r}_k^T \mathbf{r}_k} + \frac{\mathbf{r}_k^T A^T A \mathbf{r}_k}{\mathbf{r}_k^T A \mathbf{r}_k}}. \tag{1.7c}$$

There are further variations including using (1.7a) for k even and (1.7b) for k odd (SD/OM) defined as

$$\alpha_k^{SD/OM} = \begin{cases} \frac{\mathbf{r}_k^T \mathbf{r}_k}{\mathbf{r}_k^T A \mathbf{r}_k} & k \text{ even} \\ \frac{\mathbf{r}_k^T A \mathbf{r}_k}{\mathbf{r}_k^T A^T A \mathbf{r}_k} & k \text{ odd}, \end{cases} \tag{1.7d}$$

or a random combination of the two

$$\alpha_k^{RSDOM} = c_k \frac{\mathbf{r}_k^T \mathbf{r}_k}{\mathbf{r}_k^T A \mathbf{r}_k} + (1 - c_k) \frac{\mathbf{r}_k^T A \mathbf{r}_k}{\mathbf{r}_k^T A^T A \mathbf{r}_k}, \tag{1.7e}$$

with c_k a scalar drawn randomly from a uniform distribution on $[0, 1]$ at each step k. Common to all these variants is their monotonic decrease in f, *viz.*, $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$, $\forall k$, which guarantees ODE stability despite occasional violations of the absolute stability requirement (1.5).

Example 1.1. In Table 1 we record iteration counts required to bring the relative residual norm $\|\mathbf{r}_k\|/\|\mathbf{r}_0\|$ below $tol = 10^{-6}$. We consider (1.1) under homogeneous Dirichlet BC and initial conditions, and set $a \equiv q \equiv 1$. In addition to the methods advertized above we record under LSD iteration counts for the lagged steepest descent formula [5]

$$\alpha_k^{LSD} = \frac{\mathbf{r}_{k-1}^T \mathbf{r}_{k-1}}{\mathbf{r}_{k-1}^T A \mathbf{r}_{k-1}} = \alpha_{k-1}^{SD}. \tag{1.8a}$$

[4]Throughout this article we refer to the l_2 norm, unless specified otherwise.

We also record under HLSD (for "half lagged steepest descent") the number of gradient descent or forward Euler steps required using the prescription

$$\alpha_k^{HLSD} = \begin{cases} \frac{\mathbf{r}_k^T \mathbf{r}_k}{\mathbf{r}_k^T A \mathbf{r}_k} & k \text{ even} \\ \frac{\mathbf{r}_{k-1}^T \mathbf{r}_{k-1}}{\mathbf{r}_{k-1}^T A \mathbf{r}_{k-1}} & k \text{ odd.} \end{cases} \tag{1.8b}$$

Thus, the step size α_k is updated by SD only at even iteration numbers k and is left unchanged for the following odd-k iteration [13,27]. These last two methods do not converge monotonically in either f or $\|\mathbf{r}\|$ (although HLSD converges monotonically in another norm [27]). Also recorded are corresponding counts of conjugate gradient (CG) iterations required to solve the steady state equations starting from a zero guess.

Clearly the greedy step size selections perform relatively poorly, yielding step counts that grow linearly with m. This phenomenon is well-known [1,12,23,26].

The new HM step size tends to a constant written as

$$\lim_{k\to\infty} \alpha_k^{HM} = \frac{2}{\lambda_1 + \lambda_m}. \tag{1.9}$$

This limit value satisfies the Euler stability restriction (1.5): the resulting iteration converges slowly as well.

More surprising perhaps is the superior performance of the alternating selection SD/OM and the random version (1.7e). These perform significantly better than either greedy scheme (1.7a) and (1.7b). The two-step lagged schemes LSD and HLSD perform even better, although not by a large margin. None of these schemes is as good as CG applied directly to the steady state equations, but with the latter gradient descent variants the relative performance as compared to CG is no longer poorer by orders of magnitude.

The results of Example 1.1 are typical for other problems of the form (1.3) or (1.6) as well. Let us also recall that on the continuous level the steady state equation for (1.1), obtained by setting $u_t = 0$, can be reformulated as minimization of the corresponding Ritz functional (see, *e.g.*, [30]).

Generalizing the above exposition consider next an unconstrained optimization problem

$$\min_{\mathbf{x}} f(\mathbf{x}), \tag{1.10a}$$

with $\mathbf{x} \in \mathbb{R}^m$, $f \in C^2$ and $\nabla^2 f$ symmetric positive definite in a sizable neighborhood of the solution we concentrate on. The gradient descent step is

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{r}_k, \quad \mathbf{r} = -\boldsymbol{\nabla} f(\mathbf{x}), \tag{1.10b}$$

and it is equivalent in form to forward Euler for the ODE

$$\mathbf{x}' = \mathbf{r}(\mathbf{x}). \tag{1.10c}$$

Traditionally α_k has been determined by weak line search

$$\alpha_k \approx \text{argmin}_\alpha \; f(\mathbf{x}_k + \alpha \mathbf{r}_k). \tag{1.11}$$

Such a technique ensures descent in f at each step and "global" convergence is obtained under some reasonable conditions; see any text on numerical optimization, *e.g.* [22]. For the special convex quadratic case the exact (1.11) gives (1.7a). Thus the wisdom of using (1.11) for (1.10b) has recently been put under question in view of results such as those of Example 1.1; see for instance [5,8,9,27] and references therein. The purpose of the present article is to critically examine several of the issues involved, with a special eye towards the interpretation of integrating (1.10c).

As demonstrated in Table 1 the well-known slowness of the steepest descent method has to do with the choice of the gradient descent direction, as well as the step size. But the simplicity of this descent direction has a strong appeal, especially when very large, constrained problems are considered. When constraints inhibit taking a full step size in the calculated direction, methods such as CG and Newton-type can lose efficiency significantly [8]. Moreover, the gradient descent method has regularizing properties [4,21,34], so (1.10c) need not be integrated all the way to steady state in certain popular situations. When considering an inexact Newton or Gauss-Newton method for a nonlinear problem requiring regularization, applying a few (possibly preconditioned) gradient descent inner iterations with a better step size than steepest descent can possibly yield a method that may be more efficient than (similarly preconditioned) CG, because the latter loses effectiveness when not used in full [14,16].

In Section 2 we consider the quadratic problem (1.6). We give results regarding the greedy methods (1.7a), (1.7b) and (1.7c) and prove their guaranteed slowness. The faster gradient descent methods discussed above, as well as others discussed in the given references, gain their power essentially by not falling into the trap of producing a cycling step size sequence.

In Section 3 we examine two applications where the role of the gradient descent method is largely to provide regularization, yielding an interpretation as an integration up to a finite time [4]. Some highly efficient methods are introduced and studied. The question is whether the faster gradient descent methods offer advantage in such circumstances.

Conclusions are offered in Section 4.

2. Solving linear positive definite systems

In this section we consider the problem

$$A\mathbf{x} = \mathbf{b}, \tag{2.1}$$

where A is large, symmetric and positive definite. This problem is equivalent to (1.6) and defines the steady state of (1.3). For motivation, assume that not even A but only a routine returning the product $A\mathbf{v}$ for any m-vector $\mathbf{v}$ is available, as is the case for instance in many image processing applications involving the Fourier transform. This makes preconditioning tricky, and eliminates employment of most usual methods, although it leaves CG as a possible and better alternative method to gradient descent, a fact which we proceed to ignore in this section.

2.1. Properties of the one-step algorithms

For both choices (1.7a) and (1.7b) there is a monotonic descent in some norm. For (1.7a) of course there is descent in f, because α_k is the minimizer in the descent direction. Specifically

$$\begin{aligned} f_{k+1} &= \frac{1}{2}(\mathbf{x}_k + \alpha_k \mathbf{r}_k)^T A(\mathbf{x}_k + \alpha_k \mathbf{r}_k) - \mathbf{b}^T(\mathbf{x}_k + \alpha_k \mathbf{r}_k) = f_k - \alpha_k \mathbf{r}_k^T \mathbf{r}_k + \frac{\alpha_k^2}{2}\mathbf{r}_k^T A \mathbf{r}_k \\ &= f_k - \frac{\alpha_k}{2}\mathbf{r}_k^T \mathbf{r}_k < f_k, \end{aligned}$$

where the special form of the step size gets used in the last equality. It also holds that

$$\mathbf{r}_{k+1}^T \mathbf{r}_k = 0.$$

Let us define the energy product and corresponding energy norm with respect to an $m \times m$ symmetric positive definite matrix P by

$$(\mathbf{v}, \mathbf{w})_P = \mathbf{v}^T P \mathbf{w}, \quad \|\mathbf{v}\|_P^2 = (\mathbf{v}, \mathbf{v})_P.$$

It is easy to see that minimizing $f(\mathbf{x})$ is equivalent to minimizing $\|\mathbf{r}\|_{A^{-1}}^2$.

For (1.7b) we obtain monotonic descent in the residual least squares norm $\|\mathbf{r}\|$, because the exact line search is done with respect to this function squared. In fact

$$\begin{aligned}\|\mathbf{r}_{k+1}\|^2 &= \mathbf{r}_k^T(I-\alpha_k A)^2\mathbf{r}_k = \mathbf{r}_k^T\mathbf{r}_k - 2\alpha_k\mathbf{r}_k^T A\mathbf{r}_k + \alpha_k^2\mathbf{r}_k^T A^T A\mathbf{r}_k \\ &= \|\mathbf{r}_k\|^2 - \alpha_k\mathbf{r}_k^T A\mathbf{r}_k < \|\mathbf{r}_k\|^2.\end{aligned}$$

Again, the special choice for α_k enters only in the last equality. Also

$$\mathbf{r}_{k+1}^T A\mathbf{r}_k = 0.$$

Interestingly, we also have for (1.7b)

$$\begin{aligned}f_{k+1} &= f_k - \alpha_k\mathbf{r}_k^T\mathbf{r}_k + \frac{1}{2}\alpha_k^2\mathbf{r}_k^T A\mathbf{r}_k \\ &= f_k - \frac{\alpha_k}{2}\left[\mathbf{r}_k^T\mathbf{r}_k + \mathbf{r}_{k+1}^T\mathbf{r}_{k+1}\right].\end{aligned}$$

Thus, with Orthomin we actually have a monotonic descent in both $\|\mathbf{r}\|$ and f, whereas with steepest descent there is only guaranteed descent at each step in f. This implies a monotonic descent in f (and thus stability and convergence) also for the harmonic mean HM, alternating SD/OM and random weighting RSDOM variants, because f decreases at every step in each of them. This yields

Theorem 2.1. *For the symmetric positive definite linear system* (2.1)*,* i.e.*, the convex quadratic problem* (1.6)*, all gradient descent variants* (1.7)*, including SD/OM and RSDOM, converge Q-linearly and monotonically in* f*.*

Indeed it can be easily shown (*e.g.*, [15]) that for some error measurement the average decrease factor per iteration is at least as large as

$$\nu = \frac{\kappa-1}{\kappa+1} = \frac{\lambda_1-\lambda_m}{\lambda_1+\lambda_m}. \tag{2.2}$$

Unfortunately, however, this bound admits methods that take $O(\kappa)$ steps to reduce the relevant error measure by a constant factor such as 10. For methods such as SD/OM and RSDOM the observed performance is better, although no sharper theory is known to us.

For the SD, OM and HM variants (1.7a)–(1.7c) we next show that there is an asymptotic lower bound for the number of iterations required.

2.2. **Slowness of the greedy algorithms**

Let us first consider the choice of constant step size α. Thus we have the recursion

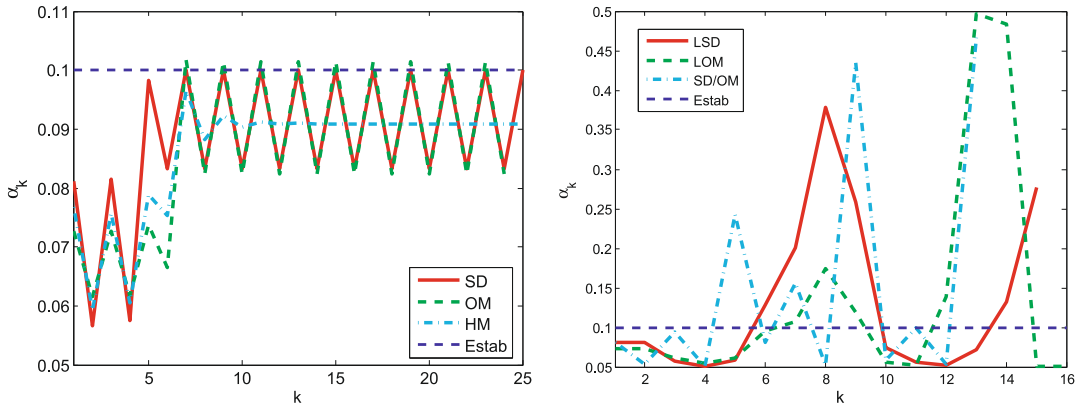
$$\mathbf{r}_k = (I-\alpha A)\mathbf{r}_{k-1} = \ldots = (I-\alpha A)^k\mathbf{r}_0.$$

The iteration is stationary, and the rate of convergence is determined by the spectral radius

$$\rho(I-\alpha A) = \max_i |1-\alpha\lambda_i| = \max[1-\alpha\lambda_m, -(1-\alpha\lambda_1)].$$

The best choice of constant step size α minimizes this spectral radius, which happens when $1-\alpha\lambda_m = -(1-\alpha\lambda_1)$, *i.e.*, for $\alpha^* = 2/(\lambda_1+\lambda_m)$. Let Q be the orthogonal similarity transformation such that

$$Q^T AQ = \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_m).$$



(a) Greedy step sizes (b) Wilder step sizes

FIGURE 1. Step size dynamics for $A = \mathrm{diag}(20, 10, 2, 1)$. (a) The greedy strategies produce orderly dynamical systems which have a steady state or are two-periodic. Forward Euler absolute stability limit, depicted here as 'Estab', therefore decrees slow convergence. (b) When this pattern is broken a faster convergence to steady state results.

Then using the best constant step size, $Q^T\mathbf{r}_{k+1} = (I - \alpha^*\Lambda)Q^T\mathbf{r}_k$. The last component of $Q^T\mathbf{r}_k$ reduces most slowly, by the factor $1 - \frac{2\lambda_m}{\lambda_1+\lambda_m} = \nu$ defined by (2.2). We have thus reconstructed the following well-known result.

Theorem 2.2. *For the ODE (1.3), barring special initial conditions, the fastest reduction in residual norm towards steady state using forward Euler with a constant step size is obtained using the step size*

$$\alpha^* = \frac{2}{\lambda_1 + \lambda_m}.$$

For this step size the number of iterations required to reduce the residual error by a constant amount is proportional to the condition number κ.

Next we consider the more promising algorithms where α_k is allowed to vary. A greedy algorithm would choose α_k so that a local property such as descent in f over the step is optimized. The essential theory is in [1], see also [12,23]. Let $\mathbf{x}^{(i)}$ be the ith eigenvector, so $A\mathbf{x}^{(i)} = \lambda_i\mathbf{x}^{(i)}$. Assuming away certain special initial guesses, Akaike showed for SD that asymptotically the error $\mathbf{e}_k = \mathbf{x}^* - \mathbf{x}_k$ tends to be in the space spanned by the first and last eigenvectors $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(m)}$. This implies a similar result about the residual $\mathbf{r}_k = A\mathbf{e}_k$. Now, since for SD each two consecutive residuals are orthogonal, clearly in the asymptotic regime $\mathbf{r}_k$ and $\mathbf{r}_{k+2}$, being in the same 2D plane and both orthogonal to $\mathbf{r}_{k+1}$, must be parallel, *i.e.* there is a constant γ such that $\mathbf{r}_{k+2} \approx \gamma\mathbf{r}_k$. The same can be shown for OM. This constant cancels out in (1.7a) and (1.7b), hence $\alpha_{k+2} \approx \alpha_k$. Figure 1(a) depicts this behavior for a simple example.

Now expand

$$\mathbf{r}_k = \sum_{i=1}^{m} r_i^{(k)}\mathbf{x}^{(i)}.$$

We may take $\mathbf{x}^{(i)}$ to be the ith unit vector, *i.e.* $r_i^{(k)}$ is the ith element of $\mathbf{r}_k$. Then

$$\mathbf{r}_{k+1} = \sum_{k=1}^{m} r_i^{(k)}(1 - \alpha_k\lambda_i)\mathbf{x}^{(i)},$$

i.e. $r_i^{(k+1)} = (1 - \alpha_k \lambda_i) r_i^{(k)}$. For the SD step (1.7a) we have

$$\alpha_k^{SD} = \frac{\sum_j (r_j^{(k)})^2}{\sum_j \lambda_j (r_j^{(k)})^2}.$$

For the OM step (1.7b) we have likewise

$$\alpha_k^{OM} = \frac{\sum_j \lambda_j (r_j^{(k)})^2}{\sum_j \lambda_j^2 (r_j^{(k)})^2}.$$

In both cases we can see how the forward Euler absolute stability restriction may be legally violated. If r_i are all approximately equal then α_k is roughly proportional to the reciprocal of the larger eigenvalues. But then the ensuing iteration is more effective in reducing the magnitude of the components r_i that correspond to the larger eigenvalues than those corresponding to the smaller ones. Hence in the next iteration the step size α_{k+1} may be larger, cutting down the magnitudes corresponding to the smaller eigenvalues while actually increasing those of the larger ones, but not so much as to have an overall divergent effect.

Theorem 2.3. *Consider the method* (1.4) *for the problem* (2.1). *Assume that the initial value* $\mathbf{x}_0$ *is not special in the sense that* $\mathbf{r}_0$ *is not an eigenvector and that* $\mathbf{e}_0$ *has a nonzero component in the direction of the first and last eigenvectors. For simplicity assume also that the eigenvalues of* A *are distinct. Then the following holds:*

1. *For SD* (1.7a) *there are constants* β_0 *and* β_1 *satisfying*

$$\frac{2}{\beta_0^{-1} + \beta_1^{-1}} = \frac{2}{\lambda_1 + \lambda_m}, \tag{2.3}$$

   such that as $j \to \infty$, $\alpha_{2j} \to \beta_0$ *and* $\alpha_{2j+1} \to \beta_1$.
2. *For OM* (1.7b) *there are constants* β_0 *and* β_1, *not necessarily the same as those for SD but also satisfying* (2.3) *and the alternating step size property described above.*
3. *For HM* (1.7c), *assume further that the sequence of residual vectors tends to satisfy*

$$\mathbf{r}_{k+1} = \zeta D_k \mathbf{r}_k, \quad k \to \infty, \tag{2.4}$$

   where ζ *is a positive scalar and* D_k *is a diagonal matrix with diagonal elements* ± 1. *(See remark following the proof of this theorem and note the harmless abuse in the simplification of asymptotic notation that yields the equality of* (2.4).*)*
   Then the limit (1.9) *holds and* ζ *equals* ν *given by* (2.2).
4. *For all three variants the number of steps necessary to decrease the error by a constant amount grows linearly in the condition number* κ.

Proof.

1. The alternating property of the step sizes follows from the analysis in [1]. Using these results [23] showed for SD that in the asymptotic limit

$$\beta_0 = \frac{1 + c^2}{\lambda_m (1 + c^2 \kappa)}, \quad \beta_1 = \frac{1 + c^2}{\lambda_m (c^2 + \kappa)},$$

   where

$$c = \lim_{j \to \infty} \frac{r_1^{(2j)}}{r_m^{(2j)}} = -\lim_{j \to \infty} \frac{r_m^{(2j+1)}}{r_1^{(2j+1)}}.$$

The constant c depends only on the initial guess $\mathbf{x}^{(0)}$ and the eigenpairs of A. It is easy to verify that (2.3) holds for these expressions.

2. Clearly $\|\mathbf{r}\|^2 = \|A^{1/2}\mathbf{r}\|^2_{A^{-1}}$. Hence OM is equivalent to SD for the problem

$$\min_{\mathbf{y}} \|\tilde{\mathbf{b}} - A\mathbf{y}\|_{A^{-1}}, \quad \mathbf{y} = A^{1/2}\mathbf{x}, \ \ \tilde{\mathbf{b}} = A^{1/2}\mathbf{b}.$$

But SD for the transformed problem can be directly verified to yield

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \alpha_k^{OM} A^{1/2}\mathbf{r}.$$

The analysis in [1] applies to this, and multiplying the recursion by $A^{-1/2}$ recovers gradient descent with (1.7b).

3. For the harmonic mean variant HM, assume without loss of generality that A is diagonal. (Otherwise consider $Q^T\mathbf{r}_k$.) Then using (2.4) we can write for each component of $\mathbf{r}_k$, as $k \to \infty$, the equality

$$\pm\zeta r_i^{(k)} = (1 - \alpha_k^{HM}\lambda_i) r_i^{(k)}, \quad i = 1, \ldots, m.$$

It can be easily seen that these m equalities can hold only if there are two indices p and l, $1 \le l < p \le m$, such that

$$r_i^{(k)} = 0, \quad i \ne l, \ i \ne p.$$

Further, the remaining two components of $\mathbf{r}_k$ are nonzero. Canceling these out and solving the two linear equations for ζ and α yields

$$\alpha = \frac{2}{\lambda_l + \lambda_p}, \quad \zeta = \frac{\lambda_l - \lambda_p}{\lambda_l + \lambda_p}.$$

Moreover, we next show that the limit residual satisfying (2.4) is stable under perturbation iff $l = 1$, $p = m$, thereby obtaining the claimed result.

Assume to the contrary that $l > 1$ and consider the perturbed limit residual $\mathbf{r}_k = (\varepsilon, 0, \ldots, 0, r_l, 0, \ldots, 0, r_p, 0, \ldots, 0)^T$. Then $\alpha_k^{HM} = \alpha + \delta$, where $\delta = \delta(\varepsilon) \to 0$ as $\varepsilon \to 0$. We can write

$$\left| \frac{r_l^{(k+1)}}{r_l^{(k)}} \right| = |1 - \lambda_l(\alpha + \delta)| \le \zeta + \lambda_l|\delta|,$$

with a similar result holding for $\left| \frac{r_p^{(k+1)}}{r_p^{(k)}} \right|$. But at the same time

$$\left| \frac{r_1^{(k+1)}}{r_1^{(k)}} \right| = |1 - \lambda_1(\alpha + \delta)| \ge (-1 + \lambda_1\alpha) - \lambda_1\delta = \zeta + (\lambda_1 - \lambda_l)\alpha + O(|\delta|).$$

Thus, there is a constant $K > 1$ such that for $|\varepsilon|$ small enough

$$\left| \frac{r_1^{(k+1)}}{r_1^{(k)}} \right| \ge K \max\left[\left| \frac{r_l^{(k+1)}}{r_l^{(k)}} \right|, \left| \frac{r_p^{(k+1)}}{r_p^{(k)}} \right| \right].$$

This shows that such a perturbation grows faster than the only coefficients of the residual $\mathbf{r}$ that are supposed to stay nonzero. Therefore, we must have $l = 1$.

Similar arguments show that $p = m$ for $\mathbf{r}$ of (2.4) to be stable. Moreover, if $l = 1$ and $p = m$ then a similar argument shows a stable configuration, proving our claim.

Some additional, straightforward algebra yields that in the limit the normalized residual satisfies

$$r_1 = \pm \left(\frac{-\lambda_m + \sqrt{\lambda_1 \lambda_m}}{\lambda_1 - \lambda_m} \right)^{1/2}, \quad r_m = \pm \left(\frac{\lambda_1 - \sqrt{\lambda_1 \lambda_m}}{\lambda_1 - \lambda_m} \right)^{1/2}.$$

4. Next, the slow linear convergence of HM follows directly from the constancy of the asymptotic step size $\alpha_k = 2/(\lambda_1 + \lambda_m)$, as in Theorem 2.2. Indeed, if A is diagonal then

$$r_m^{(k)} = (1 - \alpha_k \lambda_m)^k r_m^{(0)} = \nu^k r_m^{(0)}.$$

Regarding SD (and likewise OM) we have for each component of $\mathbf{r}$ that in the asymptotic limit

$$r_i^{(k+2)} = (1 - \beta_1 \lambda_i)(1 - \beta_0 \lambda_i) r_i^{(k)}.$$

The amplification factor over two steps is fixed, hence stability requires $|R(\lambda)| \leq 1$ for $\lambda_m \leq \lambda \leq \lambda_1$, where $R(\lambda) = (1 - \beta_1 \lambda)(1 - \beta_0 \lambda)$. Note that $R'(\lambda) = -(\beta_0 + \beta_1) + 2\beta_0 \beta_1 \lambda$. The second derivative is positive, hence the minimum is at

$$\lambda^* = \frac{\beta_0 + \beta_1}{2\beta_0 \beta_1},$$

where we must therefore require

$$R(\lambda^*) = 1 - \frac{(\beta_0 + \beta_1)^2}{4\beta_0 \beta_1} \geq -1.$$

This yields the condition

$$(\beta_1 - \beta_0)^2 \leq 4\beta_0 \beta_1, \tag{2.5}$$

and we also must have $R(\lambda_1) \leq 1$ and $R(\lambda_m) \leq 1$. Assuming, without loss of generality, that $\beta_0 \leq \beta_1$, the condition (2.5) implies that

$$\beta_1 \leq (3 + 2\sqrt{2})\beta_0 \approx 5.8\, \beta_0,$$

which means that the two step sizes must have the same order of magnitude. This in turn implies that they both must have the same order of magnitude as $1/\lambda_1$ (say, within a factor of 10). Thus, to achieve a constant reduction in the magnitude of r_m corresponding to low frequencies (smallest eigenvalue), we require a number of steps that is proportional to $\kappa = \text{cond}(A) = \lambda_1/\lambda_m$. □

Remark 2.1. Assumption (2.4) is a conjecture that was observed to hold in all our experiments. However, it may well be as hard to prove directly as are its conclusions. We believe that the proof of Part 3 of Theorem 2.3 is illuminating; still, a more complete treatment comparable to that available for Part 1 remains an open problem.

Note that the results above hold more generally for energy norms $P = A^l$, with l any integer. Specifically, determining the step size $\alpha_k = \alpha_k^{(l)}$ as the minimizer of $\|\mathbf{b} - A(\mathbf{x}_k + \alpha \mathbf{r}_k)\|_{A^l}^2$, thus obtaining

$$\alpha_k^{(l)} = \frac{\mathbf{r}_k^T A^{l+1} \mathbf{r}_k}{\mathbf{r}_k^T A^{l+2} \mathbf{r}_k},$$

yields a step sequence that behaves essentially like that for SD. Moreover, a harmonic mean of the expressions for $\alpha_k^{(p)}$ and $\alpha_k^{(l)}$, $p \neq l$, is observed to yield a step sequence that behaves essentially like that for HM. What is special about α_k^{SD} $(l = -1)$ and α_k^{OM} $(l = 0)$ is that their computation can be arranged to cost only two vector

TABLE 2. Maximum violation of monotonic decline in $\|\mathbf{r}\|$ and f.

m	Method	$\max_k \frac{\|\mathbf{r}_{k+1}\|}{\|\mathbf{r}_k\|}$	$\max_k \frac{f_{k+1}-f^*}{f_k-f^*}$	Method	$\max_k \frac{\|\mathbf{r}_{k+1}\|}{\|\mathbf{r}_k\|}$	$\max_k \frac{f_{k+1}-f^*}{f_k-f^*}$
49	SD	9.67 e–1	8.54 e–1	OM	9.24 e–1	8.62 e–1
	HM	9.24 e–1	8.54 e–1	CG	6.64 e–1	9.96 e–1
	SD/OM	1.89	8.53 e–1	RSDOM	1.93	9.0 e–1
	LSD	9.33	8.18	HLSD	1.42 e+1	1.24 e+2
225	SD	9.96 e–1	9.63 e–1	OM	9.81 e–1	9.65 e–1
	HM	9.81 e–1	9.63 e–1	CG	9.30 e–1	1.28
	SD/OM	3.19	9.70 e–1	RSDOM	4.95	9.77 e–1
	LSD	6.96 e+1	1.19 e+2	HLSD	5.15 e+1	6.77 e+2
961	SD	1.0	9.92 e–1	OM	9.95 e–1	9.92 e–1
	HM	9.95 e–1	9.92 e–1	CG	1.08	1.79
	SD/OM	7.09	9.93 e–1	RSDOM	9.94	9.95 e–1
	LSD	2.07 e+2	8.93 e+3	HLSD	3.93 e+2	1.39 e+5
3969	SD	1.00	1.00	OM	9.99 e–1	1.00
	HM	9.99 e–1	1.00	CG	2.14	1.22
	SD/OM	1.64 e+1	9.98 e–1	RSDOM	1.99 e+1	9.99 e–1
	LSD	1.63 e+3	1.13 e+6	HLSD	9.75 e+2	5.46 e+5

inner products, because the vector $\mathbf{s}_k = A\mathbf{r}_k$ is required anyway to update $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{s}_k$, *cf.* (1.4), whereas other values of l yield additional, usually more costly multiplications by A.

Akaike also proposed in [1] an acceleration algorithm which we have simplified and implemented as follows. In the asymptotic regime $\mathbf{x}^* - \mathbf{x}_k \approx \gamma(\mathbf{x}^* - \mathbf{x}_{k-2})$. Hence convergence acceleration can be obtained by setting

$$\mathbf{x}_{k+1} = \frac{\mathbf{x}_k - \gamma \mathbf{x}_{k-2}}{1-\gamma}, \quad \gamma = \frac{\|\mathbf{r}_k\|}{\|\mathbf{r}_{k-2}\|}. \tag{2.6}$$

We have implemented this and tested over several examples, applying the correction (2.6) whenever

$$|\alpha_k - \alpha_{k-2}| \leq tola\ (\alpha_k + \alpha_{k-2}).$$

Indeed a rather significant improvement of SD or OM is obtained using $tola = 10^{-6}$, say. However, the results are not as good as those obtained for LSD or HLSD because the value for $tola$ required to enable an effective use of (2.6) was found to be quite strict.

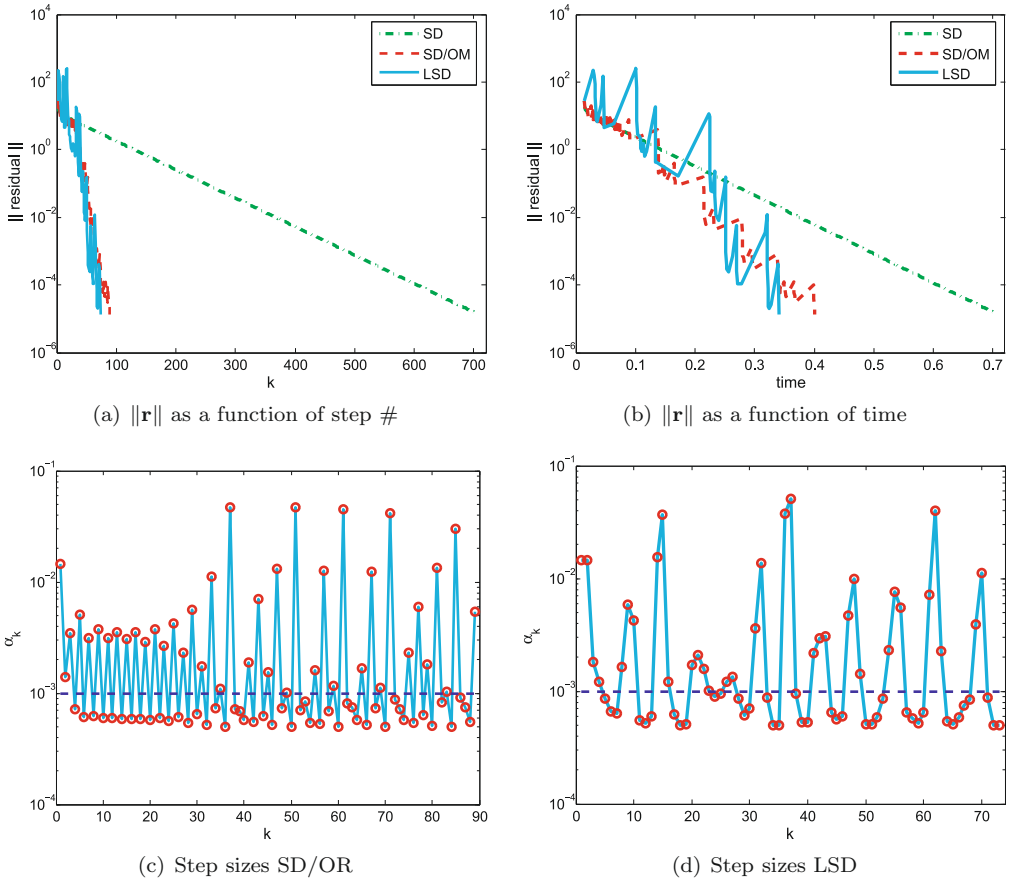
2.3. **Two-step gradient descent variants**

The lagged steepest descent (LSD) has a similar analogue for the Orthomin variant (1.7b), *viz.* $\alpha_k^{LOM} = \alpha_{k-1}^{OM}$, see [5]. The performance of LOM is similar to that of LSD.

The lagged methods and HLSD exhibit generally comparable practical performances and all have been shown to converge at least R-linearly. However, there are steps k where either f or $\|\mathbf{r}\|$ (or both) may actually grow [5,13,27]. This is not really good news in terms of iteration control, but it is theoretically interesting.

Example 2.1. In Table 2 we record the maximum violation of monotonicity in $f(\mathbf{x}_k) - f(\mathbf{x}^*) \equiv f_k - f^*$ and $\|\mathbf{r}_k\|$ (where $\mathbf{x}^* = A^{-1}\mathbf{b}$ is the exact steady state) over $k > 0$ for the runs of Example 1.1.

Clearly there are monotonicity violations in $\|\mathbf{r}\|$ for all of the faster methods. For the one-step step selection variants there is a monotonic descent in f as theory predicts. Moreover, the monotonicity violations in the two-step methods are much more serious and worrisome than those in the one-step methods.



(a) $\|\mathbf{r}\|$ as a function of step #

(b) $\|\mathbf{r}\|$ as a function of time

(c) Step sizes SD/OR

(d) Step sizes LSD

FIGURE 2. Residual norms and step size sequences on a log scale for the Poisson problem of Examples 1.1 and 2.1 with $m = 225$.

Figure 2 displays progress of the residual norms as a function of step number and time $t_k = \sum_{j=0}^{k-1} \alpha_j$. These plots visualize some of the behavior pattern depicted in Table 2. Also depicted are corresponding step size sequences.

We have run similar tests also for other problems, obtaining qualitatively similar results. A related general note is that with the faster gradient descent methods the number of iterations required to satisfy a given tolerance varies significantly (up to a factor of two, say) with the choice of the right hand side $\mathbf{b}$ in (1.6), or homogeneity q in the PDE (1.1). A Java applet that may help illustrate this point is available at `http://www.cs.ubc.ca/$\sim$kvdoel/lag/`.

The LSD variant has been successfully implemented in general purpose codes for minimizing quadratic objective functions subject to box constraints [8] and applied to compressed sensing and other image processing problems [11,32]. In this context it often gives better results than either the steepest descent or the conjugate gradient alternatives.

Figure 1(b), like Figures 2(c) and 2(d), depicts typical step size behavior for the non-greedy variants. The general pattern where occasionally a large step size is taken is common to all variants that produce faster convergence than the greedy schemes. In cases where the condition number gets large, the largest step size taken by LSD or HLSD can be much larger than the bound (1.5) suggests. For instance, in the last row of Table 1 the largest step size taken by both LSD and HLSD was about 1000 times larger than $2/\lambda_1$.

The LSD method was prescribed for the more general problem (1.10) as the least squares solution of

$$\alpha_k^{-1}(\mathbf{x}_k - \mathbf{x}_{k-1}) \approx -(\mathbf{r}_k - \mathbf{r}_{k-1})$$

yielding

$$\alpha_k = \frac{\|\mathbf{x}_k - \mathbf{x}_{k-1}\|^2}{(\mathbf{r}_{k-1} - \mathbf{r}_k, \mathbf{x}_k - \mathbf{x}_{k-1})}.$$

The motivation was an extension of the scalar secant method. However, the formula (1.7a) can also be obtained as a secant approximation for the same derivative required in Newton's method. Just postulate

$$\alpha_k^{-1}(\mathbf{x}_{k+1} - \mathbf{x}_k) \approx -(\mathbf{r}_{k+1} - \mathbf{r}_k).$$

For the quadratic case, and using least squares, we obtain

$$\alpha_k = \frac{\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2}{(\mathbf{r}_k - \mathbf{r}_{k+1}, \mathbf{x}_{k+1} - \mathbf{x}_k)} = \frac{\mathbf{r}_k^T \mathbf{r}_k}{\mathbf{r}_k^T A \mathbf{r}_k},$$

where α_k^2 cancels in numerator and denominator.

3. Gradient descent smoothing in applications

In this section we describe two experiments investigating the regularizing properties of the faster gradient descent variants, comparing specifically SD against LSD. In both experiments we know that LSD would reach steady state much faster than SD, but we are not interested in that limit. Rather, we carry out only a few such steps, which corresponds to finite time regularization [4]. Integration to steady state would lead in the first example below to a blank image, and in the second to an attempt to solve a singular system. And yet applying a few integration steps, stopping well short of the disastrous end, produces highly efficient algorithms. For discussions of criteria to stop the iteration see, *e.g.*, [10,33,35].

3.1. **Image denoising in 2D**

The problem considered here is to clean a given noisy 2D image $b(x, y)$. Let us denote the result $w(x, y)$. Of course both b and w are in reality given or found as discrete function values over pixels, but it is convenient to develop notation and understanding using a continuous analogue.

In [3] the authors proposed to consider the usual Tikhonov-type formulation

$$(w - b) + \beta R_w = 0,$$

with $R_w = \boldsymbol{\nabla} R(w)$[5], a discretization with mesh width h of

$$R_w(w) = -\nabla \cdot (g(|\boldsymbol{\nabla} w|)\boldsymbol{\nabla} w), \quad \text{where} \quad g(\tau) = \frac{\rho'(\tau)}{\tau},$$

and ρ is chosen as the Huber function

$$\begin{aligned} \rho(\tau) &= \begin{cases} \tau, & |\tau| \geq \gamma, \\ \tau^2/(2\gamma) + \gamma/2, & |\tau| < \gamma, \end{cases} \\ R_w(w) &\leftarrow \nabla \cdot \left(\min\left\{ \frac{1}{\gamma}, \frac{1}{|\boldsymbol{\nabla} w|} \right\} \boldsymbol{\nabla} w \right). \end{aligned}$$

[5]The penalty functional $R(w)$ itself is defined below (3.1); it is never used directly in our algorithms.

The innovation was in the adaptive choice of the switching parameter

$$\gamma = \frac{h}{|\Omega|}\int_\Omega |\boldsymbol{\nabla} w|, \tag{3.1}$$

where the integral is discretized on the same mesh as

$$R(w) = \int_\Omega \rho(|\boldsymbol{\nabla} w|).$$

The above is referred to as the implicit scheme because a nonlinear system of algebraic equations for w must be solved, a task for which lagged diffusivity [35] was applied in [3].

Consider instead the explicit algorithm

$$w_0 = b, \tag{3.2a}$$
$$w_{k+1} = w_k - \alpha_k R_w(w_k), \quad k = 0, 1, 2, \ldots \tag{3.2b}$$

This is a forward Euler discretization for (1.1), where now the function a depends on the solution when γ is small enough, in which case the process may be viewed as anisotropic diffusion. For $\gamma \to \infty$ we have the least squares (LS) case in ρ, for which the discretized PDE is similar to that considered in Example 1.1 with $q \equiv 0$ and corresponds to isotropic diffusion.

The alternative view of (3.2), as before, is as a gradient descent method for minimizing $R(w)$, *i.e.* with $\beta \to \infty$ above, starting from the data. Of course the actual minimizer of R, like the steady state of the homogeneous (1.1), is a bleak and boring zero array. But we hope that just a few well-aimed gradient descent or forward Euler steps would significantly reduce only the residuals corresponding to the large eigenvalues, *i.e.* the high frequencies which are more pronounced in noise than in a clean, piecewise smooth surface function [25].

For step size let us consider the two choices

$$\alpha_k^{SD} = \frac{(R_w(w_k))^T (R_w(w_k))}{(R_w(w_k))^T \hat{R}(w_k)(R_w(w_k))}, \tag{3.3a}$$
$$\alpha_k^{LSD} = \frac{(R_w(w_{k-1}))^T (R_w(w_{k-1}))}{(R_w(w_{k-1}))^T \hat{R}(w_{k-1})(R_w(w_{k-1}))}. \tag{3.3b}$$

Here $R_w(w) = \hat{R}(w) \cdot w$. Strictly speaking, these would be steepest descent and lagged steepest descent only if $\hat{R}$ were constant. But we proceed to freeze $\hat{R}$ for this purpose anyway, which amounts to a lagged diffusivity approach [35].

Figure 3 depicts the discrete dynamics for a denoising test problem, where we synthesize data by adding 20% randomly distributed Gaussian white noise to the clean image `Cameraman` (256×256) depicted in Figure 4(a). With LS regularization, *i.e.*, $\gamma \to \infty$, SD step sizes (3.3a) (red dots in Fig. 3(a)) regularly oscillate about the forward Euler stability restriction (blue dash). Such step size cycling ties directly to the slowness of the resulting gradient descent method, see the behavior of relative error indicator

$$\hat{e}_k = \|w_{k+1} - w_k\| / \|w_{k+1}\|, \tag{3.4a}$$

depicted by red dots in Figure 3(c).

Figure 3(b) depicts the behavior of step sizes when we employ the "Huber regularization" with our choice of switching parameter (3.1) for the same denoising problem. An almost constant blue dashed line records the forward Euler stability bound derived by freezing $\hat{R}$ locally. Even though step sizes obtained by (3.3a) clearly violate this upper bound at the beginning of integration, α_k decreases monotonically here and quickly

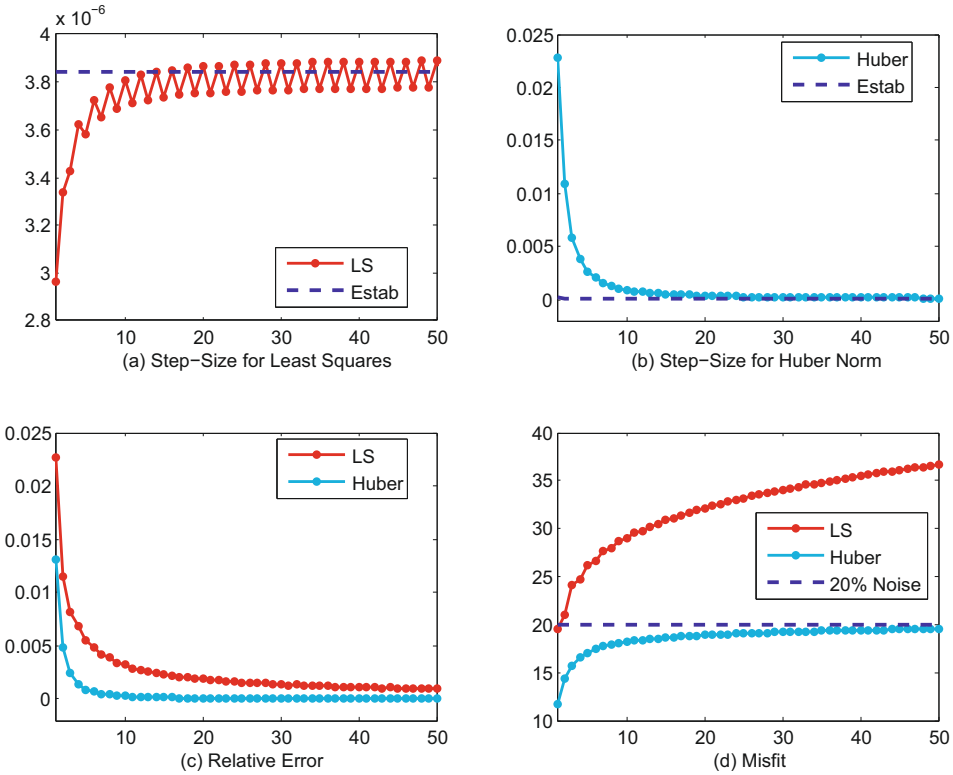


FIGURE 3. Discrete dynamics for the test problem `Cameraman` in Figure 4(a) with 20% white noise added. Two regularizations, LS (γ large) and Huber switching (γ by (3.1)), are compared. The horizontal axis displays iteration numbers.

approaches its bound. The overall process is perfectly stable, and it behaves very similarly in other test problems tried. This results in fast convergence as can be seen in both relative error indicators and misfits

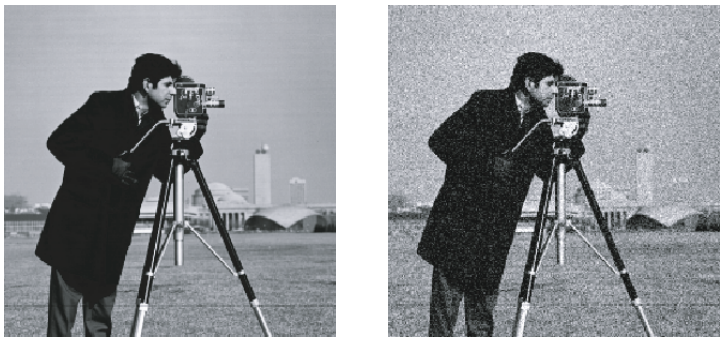
$$\eta_k = \|w_k - b\|/256, \tag{3.4b}$$

see sky blue dots in Figures 3(c) and 3(d).

Most important to note in these results is how well η approximates the variance of the noise for the Huber regularization employing (3.1), and how poorly it does so for LS. Using the former the noisy image is rapidly decomposed into a fairly good approximation of the clean image plus the noise that can now be estimated by $b - w_k$; see, *e.g.*, [31]. The simpler LS regularization does not give this because of the well-known smearing of data that it introduces; see, *e.g.*, [24]. Using this noise estimate for more realistic situations where we don't know the noise level in advance, it is now possible to rapidly improve the image quality further using implicit schemes as in [3,18]. Such techniques will be reported separately elsewhere.

Figure 4 shows denoising results with 20% noise, using the Huber regularization with (3.1) and applying both SD and LSD step selection strategies. The noise level estimate was found to vary smoothly with the number of iterations for both step size selection methods. Probably the cleanest image is obtained by SD in Figure 4(e), but this is a rather slow method. There is a clear advantage in using LSD for rapidly obtaining an albeit imperfectly cleaned image with a good estimate of the noise. This relative advantage of LSD does diminish for coarser tolerances, though.

For a similar experiment on the famous Lena image (256×256, not shown) we obtain that for the relative error tolerance of 10^{-4}, 17 SD steps *vs.* 13 LSD steps are required. The resulting image has similar quality



(a) True image (b) 20% noise

(c) Misfit = 18.94 (d) Misfit = 20.44

(e) Misfit = 20.96 (f) Misfit = 21.04

FIGURE 4. Using algorithm (3.2) with adaptive γ by (3.1). (a) Clean image `Cameraman`; (b) image corrupted by 20% Gaussian white noise; (c) denoised image using SD step size (3.3a) for error tolerance $tol = 10^{-4}$ on $\hat{e}_k$ of (3.4a): iters = 21, $\eta_{21} = 18.94$; (d) denoised image using LSD step size (3.3b) for the same tol: iters = 14, $\eta_{14} = 20.44$; (e) same as (c) with $tol = 10^{-5}$: iters = 382, $\eta_{382} = 20.96$; (f) same as (d) with $tol = 10^{-5}$: iters = 117, $\eta_{117} = 21.04$. The CPU time is roughly proportional to the number of steps required.

for these step size choices and can be further improved by switching to the implicit algorithm. Furthermore, for the relative error tolerance of 10^{-5}, 309 SD steps *vs.* only 140 LSD steps are required. The resulting image with the stricter tolerance is again similar for the two step size choices, and it requires no further improvement, *i.e.* switching to the implicit scheme is not necessary in this particular test.

These results make intuitive sense. The regularizing effect of gradient descent does not seem to depend too strongly on the choice of step size, and the aspect of more rapid progress towards the steady state is achieved faster using the LSD choice (3.3b). Similar observations arise when we apply these algorithms to deblurring problems, see [18].

3.2. **Shape optimization in 3D for EIT and DC resistivity**

The problem and numerical methods considered here are described in [34]. Specifically, a shape $w(x, y, z)$ is sought that arises in electric impedance tomography (EIT) or direct current (DC) resistivity problems; see, *e.g.*, [6,7,20,29] and references therein for background on the applications. A distributed parameter function is sought that at each point (x, y, z) in a domain $\Omega \in \mathbb{R}^3$ takes one of two values ("body" or "background"). The predicted data $F(w)$ consists of values of the field (solution of discretized generalized Poisson PDEs[6]) at locations where observed data b are measured, for several different Neumann boundary conditions (corresponding to different external current injections). The method describes the piecewise constant $w(x, y, z)$ in terms of a differentiable level set function $\psi(x, y, z)$, and a Gauss-Newton method for the *singular* data fitting problem is considered. This outer iteration reads

$$\begin{aligned} (\hat{J}^T \hat{J})\delta\psi &= -\hat{J}^T(F(w(\psi_l)) - b), \\ \psi_{l+1} &= \phi_l + \tau_l \delta\psi, \quad l = 0, 1, 2, \ldots \end{aligned} \tag{3.5}$$

where the Jacobian matrix $\hat{J} = \frac{\partial F}{\partial \psi} = \frac{\partial F}{\partial w}\frac{\partial w}{\partial \psi}$ is evaluated at the current state ψ_l and τ_l is a step size generally obtained by weak line search (although in the sequel always $\tau_l = 1$).

Towards solving the singular linear algebra problem (3.5) a *small fixed* number $iters_{\text{in}}$ of preconditioned SD (PSD) or PCG or PLSD (*viz.*, a preconditioned version of (1.8a)) inner iterations is applied to the outer iteration (3.5), relying on their regularizing power. The preconditioner is a simple (not generalized) Poisson solver, ensuring that $\delta\psi$ is smooth enough. In fact, we apply some steps towards solving (3.5) but carefully avoid attempting to solve it too well (it is singular, to recall). The number of outer iterations $iters_{\text{out}}$ is determined by a data fitting stopping criterion, see [34]. The number of PDE solutions required is therefore $2K * iters_{\text{in}} * iters_{\text{out}}$, where K is the number of experiments and a "PDE solve" refers to an appropriately accurate solution of a discretized generalized Poisson PDE. We can consider a PDE solve as our work unit and gauge the performance of an algorithm by counting the number of such work units.

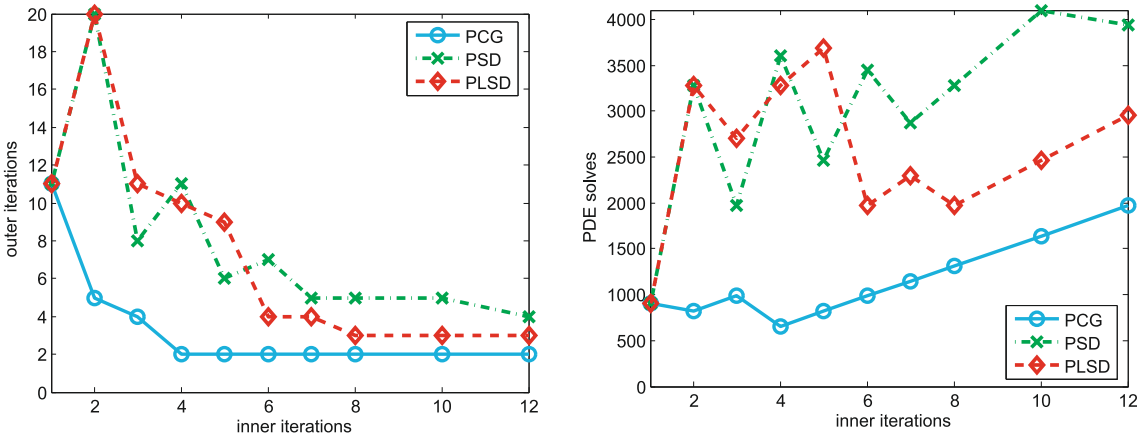
Figure 5 displays the number of outer iterations required for various choices of method and number of inner iterations. The actual experimental setting for the results reported here and corresponding reconstructions are described in [34], Section 5.2, as "Experiment 2". The number of data collection experiments there was $K = 41$. See also [33] for further discussion of criteria for terminating the iteration.

The results in Figure 5(a) show that when sufficiently many inner iterations are used, PLSD requires fewer outer iterations than PSD while PCG requires even fewer. The regularizing effect of gradient descent remains in force using any of the methods for selecting the time step. However, when considering in Figure 5(b) the number of total PDE solves required, aiming to minimize it, PLSD does not add much. The results agree with those of [34], in that PSD with one inner iteration is a valid option that is both competitive and simple, and otherwise a few (up to 5) inner PCG iterations provide good choices. Note that the prescription of 2 PSD/PLSD inner iterations performs poorly in this example, requiring 20 outer iterations, because the residual increases in the second inner step, in line with the discussion in Section 2.3.

To keep a more global perspective let us mention that any of the variants described here that solves such a problem requiring, say, 2400 PDE inversions or less (which is less than 60 PDE solves per data collection experiment) can be orders of magnitude faster than several reconstruction techniques that have appeared in recent years elsewhere in the literature.

[6]Specifically, the field u satisfies for given piecewise constant function w and source function q the PDE

$$-\nabla \cdot (\mathrm{e}^w \boldsymbol{\nabla} u) = q.$$



(a) Outer iterations required (b) Work units

FIGURE 5. Inner iteration counts and number of PDE solves for the shape reconstruction problem of [34]. For $iters_{\text{in}} = 1$ all three methods coincide.

4. Conclusions

In this article we have investigated greedy and non-greedy one-step and two-step strategies for selecting the step size in gradient descent methods. These methods also admit interpretation as forward Euler discretizations of a related ODE system that is to be integrated to steady state. We have shown using the forward Euler absolute stability restriction that methods which generate step size sequences that tend to oscillate or tend to a uniform value must be particularly slow to converge. Such methods include steepest descent (SD) and Orthomin (OM), as well as their harmonic average, but not necessarily other variations of the two formulas (1.7a) and (1.7b) that therefore may converge faster.

If a constant, or uniform step size is desired then for the linear symmetric positive definite system the best step size (in terms of fast integration to steady state) is given by the harmonic mean of the largest and smallest eigenvalues. This value may be obtained by applying first a few gradient descent steps using α_k^{HM} given by (1.7c), see (1.9), or using the harmonic average of two consecutive step sizes of SD once they get into their asymptotic oscillatory pattern. The performance of the greedy algorithms (1.7a)–(1.7c) is in fact comparable to integrating with the best uniform step size. This is often too slow for comfort.

The lagged methods (1.8) that use information from the previous "time" step perform best from amongst the schemes tried in terms of step count, or number of matrix-vector multiplications, although they are generally in the same class as the faster one-step methods when compared to the greedy algorithms.

One could think of other Runge-Kutta (RK) methods to replace forward Euler. In particular, Runge-Kutta-Chebyshev methods [17,19] come to mind because they ease the stability restriction for real eigenvalues somewhat. For the special problem (1.10c) an s-stage RK method can be written as

$$\mathbf{r}_{k+1} = \left[I - \alpha_k A + \sum_{j=2}^{s} (-\alpha_k)^j d_j A^j \right] \mathbf{r}_k,$$

where the coefficients d_j are expressed in terms of the tableau coefficients of the method and are independent of the step k. Such a step costs s matrix-vector multiplications and has only one adaptive step size parameter to determine, unlike s forward Euler steps which have s such parameters. Even if the resulting absolute stability restriction is relaxed by a factor s^2, this may not be a lot compared to the step sizes required to approximate the inverses of all eigenvalues of A well enough. On the other hand, each two consecutive HLSD iterations can

be viewed as one step of a two-stage, first order RK method with step size $\alpha_k = 2\alpha_k^{SD}$. More generally, keeping α_k^{SD} fixed over s consecutive gradient descent iterations [9,13], the compound iteration can be written as an s-stage, first order accurate RK step with step size $\alpha_k = s\alpha_k^{SD}$.

There are many applications that give rise to nonlinear mathematical models which locally involve positive definite systems. Moreover, the importance of gradient descent or forward Euler for a finite time integration is often in its rapid smoothing or regularization properties. We have examined two such applications in Section 3 and found out that using the faster, more unruly LSD over the more cautious SD does not diminish these regularization properties. The advantage that LSD offers is moderate, however, because by the time the slowness of SD starts to hurt we are better off switching to another method – an implicit scheme in the case of Section 3.1 and a new outer iteration in the case of Section 3.2.

One major goal of this article has been simply to bring some relevant, surprisingly simple, recent developments in numerical optimization to the consciousness level of the community that deals with simulating differential equations.

In doing so, however, we have also found several observations and motivations in the optimization literature that have left us searching for better explanations. Specifically, for unconstrained optimization, all that is required of a gradient descent method to be significantly faster than SD is to "break the spell" of convergence to a step size sequence that cycles often, while still ensuring convergence. Contrary to statements in some of our references the method may be one-step, as (1.7e) proves, and it may be Q-linearly and monotonically converging in a computable norm as all methods (1.7) including SD/OM are. At the same time, significant violations of monotonicity in the computable norms in the case of multistep methods for calculating the step size pose a challenge when contemplating the practical usage of such schemes. Furthermore, the interpretation of LSD as a secant method is less important. Finally, it seems to us that wherever CG (or PCG, or a more general variant) may be used for several steps without interruption then it and not any gradient descent variant should be employed, because the entire search direction and not only a scalar step size is modified to advantage.

Acknowledgements. The first author thanks Dr. M. Friedlander for several fruitful discussions.

References

[1] H. Akaike, On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method. *Ann. Inst. Stat. Math. Tokyo* **11** (1959) 1–16.
[2] U. Ascher, *Numerical Methods for Evolutionary Differential Equations*. SIAM, Philadelphia, USA (2008).
[3] U. Ascher, E. Haber and H. Huang, On effective methods for implicit piecewise smooth surface recovery. *SIAM J. Sci. Comput.* **28** (2006) 339–358.
[4] U. Ascher, H. Huang and K. van den Doel, Artificial time integration. *BIT* **47** (2007) 3–25.
[5] J. Barzilai and J. Borwein, Two point step size gradient methods. *IMA J. Num. Anal.* **8** (1988) 141–148.
[6] M. Cheney, D. Isaacson and J.C. Newell, Electrical impedance tomography. *SIAM Review* **41** (1999) 85–101.
[7] E. Chung, T. Chan and X. Tai, Electrical impedance tomography using level set representations and total variation regularization. *J. Comp. Phys.* **205** (2005) 357–372.
[8] Y. Dai and R. Fletcher, Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming. *Numer. Math.* **100** (2005) 21–47.
[9] Y. Dai, W. Hager, K. Schittkowsky and H. Zhang, A cyclic Barzilai-Borwein method for unconstrained optimization. *IMA J. Num. Anal.* **26** (2006) 604–627.
[10] H.W. Engl, M. Hanke and A. Neubauer, *Regularization of Inverse Problems*. Kluwer (1996).
[11] M. Figueiredo, R. Nowak and S. Wright, Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE J. Sel. Top. Signal Process.* **1** (2007) 586–598.
[12] G.E. Forsythe, On the asymptotic directions of the s-dimensional optimum gradient method. *Numer. Math.* **11** (1968) 57–76.
[13] A. Friedlander, J. Martinez, B. Molina and M. Raydan, Gradient method with retard and generalizations. *SIAM J. Num. Anal.* **36** (1999) 275–289.
[14] G. Golub and Q. Ye, Inexact preconditioned conjugate gradient method with inner-outer iteration. *SIAM J. Sci. Comp.* **21** (2000) 1305–1320.
[15] A. Greenbaum, *Iterative Methods for Solving Linear Systems*. SIAM, Philadelphia, USA (1997).

[16] E. Haber and U. Ascher, Preconditioned all-at-one methods for large, sparse parameter estimation problems. *Inverse Problems* **17** (2001) 1847–1864.
[17] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Second Edition, Springer (1996).
[18] H. Huang, *Efficient Reconstruction of 2D Images and 3D Surfaces*. Ph.D. Thesis, University of BC, Vancouver, Canada (2008).
[19] W. Hundsdorfer and J.G. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer (2003).
[20] Y. Li and D.W. Oldenburg, Inversion of 3-D DC resistivity data using an approximate inverse mapping. *Geophys. J. Int.* **116** (1994) 557–569.
[21] J. Nagy and K. Palmer, Steepest descent, CG and iterative regularization of ill-posed problems. *BIT* **43** (2003) 1003–1017.
[22] J. Nocedal and S. Wright, *Numerical Optimization*. Springer, New York (1999).
[23] J. Nocedal, A. Sartenar and C. Zhu, On the behavior of the gradient norm in the steepest descent method. *Comput. Optim. Appl.* **22** (2002) 5–35.
[24] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. Springer (2003).
[25] P. Perona and J. Malik, Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.* **12** (1990) 629–639.
[26] L. Pronzato, H. Wynn and A. Zhigljavsky, *Dynamical Search: Applications of Dynamical Systems in Search and Optimization*. Chapman & Hall/CRC, Boca Raton (2000).
[27] M. Raydan and B. Svaiter, Relaxed steepest descent and Cauchy-Barzilai-Borwein method. *Comput. Optim. Appl.* **21** (2002) 155–167.
[28] R. Sincovec and N. Madsen, Software for nonlinear partial differential equations. *ACM Trans. Math. Software* **1** (1975) 232–260.
[29] N.C. Smith and K. Vozoff, Two dimensional DC resistivity inversion for dipole dipole data. *IEEE Trans. Geosci. Remote Sens.* **22** (1984) 21–28.
[30] G. Strang and G. Fix, *An Analysis of the Finite Element Method*. Prentice-Hall, Engelwood Cliffs, NJ (1973).
[31] E. Tadmor, S. Nezzar and L. Vese, A multiscale image representation using hierarchical (BV, L^2) decompositions. *SIAM J. Multiscale Model. Simul.* **2** (2004) 554–579.
[32] E. van den Berg and M. Friedlander, Probing the Pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comput.* **31** (2008) 840–912.
[33] K. van den Doel and U. Ascher, On level set regularization for highly ill-posed distributed parameter estimation problems. *J. Comp. Phys.* **216** (2006) 707–723.
[34] K. van den Doel and U. Ascher, Dynamic level set regularization for large distributed parameter estimation problems. *Inverse Problems* **23** (2007) 1271–1288.
[35] C. Vogel, *Computational methods for inverse problem*. SIAM, Philadelphia, USA (2002).
[36] J. Weickert, *Anisotropic Diffusion in Image Processing*. B.G. Teubner, Stuttgart (1998).