# The chaotic nature of faster gradient descent methods

Kees van den Doel and Uri Ascher *

July 19, 2011

**Abstract**

The steepest descent method for large linear systems is well-known to often converge very slowly, with the number of iterations required being about the same as that obtained by utilizing a gradient descent method with the best constant step size and growing proportionally to the condition number. Faster gradient descent methods must occasionally resort to significantly larger step sizes, which in turn yields a rather non-monotone decrease pattern in the residual vector norm.

We show that such faster gradient descent methods in fact generate chaotic dynamical systems for the normalized residual vectors. Very little is required to generate chaos here: simply damping steepest descent by a constant factor close to 1 will do.

Several variants of the family of faster gradient descent methods are investigated, both experimentally and analytically. The fastest practical methods of this family in general appear to be the known, chaotic, two-step ones. Our results also highlight the need of better theory for existing faster gradient descent methods.

## 1 Faster gradient descent methods

Many efforts have been devoted in the two decades that have passed since the pioneering paper of Barzilai & Borwein [4] to the design, analysis, extension and application of *faster gradient descent methods* for function minimization; see, e.g., [13, 26, 6, 12] and references therein. These are methods that converge significantly faster than the method of steepest descent although, unlike the conjugate gradients (CG) method, they confine their search directions to the gradient vector at each iteration.

*Department of Computer Science, University of British Columbia, Canada, (kvdoel/ascher@cs.ubc.ca), supported in part by NSERC Discovery Grant 84306.

To be specific, consider the classical linear algebra problem

$$A\mathbf{x} = \mathbf{b}, \tag{1}$$

where $A$ is a given real $m \times m$ symmetric positive definite matrix, $\mathbf{b}$ is a given $m$-vector and $\mathbf{x}$ is an $m$-vector to be found. This is of course equivalent to minimizing the convex quadratic function

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

We consider iterative methods and define for any vector $\mathbf{x}_k$ the residual

$$\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k. \tag{3}$$

The gradient descent family of methods is defined by

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{r}_k, \quad (\alpha_k \geq 0), \quad k = 0, 1, \ldots. \tag{4}$$

The steepest descent (SD) method for determining the step size $\alpha_k$ minimizes

$$\psi(\alpha) = f(\mathbf{x}_k + \alpha\mathbf{r}_k), \tag{5}$$

which yields

$$\alpha_k = \alpha_k^{SD} = \frac{\mathbf{r}_k^T\mathbf{r}_k}{\mathbf{r}_k^T A\mathbf{r}_k}. \tag{6a}$$

This is a slow method requiring $O(\kappa)$ iterations to reduce the residual by a fixed amount, where $\kappa = \kappa(A)$ is the condition number [1]. The lagged steepest descent (LSD) method [4] sets instead

$$\alpha_k = \alpha_{k-1}^{SD} = \frac{\mathbf{r}_{k-1}^T\mathbf{r}_{k-1}}{\mathbf{r}_{k-1}^T A\mathbf{r}_{k-1}}, \tag{6b}$$

while the half lagged steepest descent (HLSD) method [26] simply updates the step size $\alpha$ only every second step, reading

$$\alpha_{2j} = \alpha_{2j+1} = \alpha_{2j}^{SD}, \quad j = 0, 1, 2, \ldots. \tag{6c}$$

To be clear, no one we know expects any gradient descent variant ever to perform better than CG for the solution of (1), provided that matrix-vector multiplications $A\mathbf{v}$ for any given vector $\mathbf{v}$ are carried out accurately. Moreover, preconditioning in general applies to CG as well as to gradient descent methods, so the latter remain relatively inferior. Nonetheless, the search for better faster gradient descent methods has not subsided, and more method variants for the basic problem have been proposed in [13, 8, 5], to name but a few references offering interesting possibilities.

There are several reasons for the continued interest in faster gradient descent algorithms:

- The linear CG algorithm does lose some of its luster when applied to more general problems than (1) which arise in specific applications, such as in box-constrained optimization [7, 11] and certain nonlinear problems arising in image processing [2, 19]. If matrix-vector multiplications are performed only approximately then CG can lose efficiency more rapidly than gradient descent, as Experiment 1 below shows; see also [15, 18].

- Gradient descent methods enjoy a natural interpretation as artificial time integration methods with different step size strategies [2, 3, 19, 22]. This is particularly useful for certain inverse problems, e.g., for image deblurring.

- The state of theory for faster gradient descent methods is currently unsatisfactory. There exist essential convergence theorems [13, 26, 10], but they are at best as strong as the simple convergence theorems available for SD, and they do not explain why the rate of convergence of these faster gradient descent methods is more like that of CG than SD, see Table 1 in Experiment 2 below.

**Experiment 1** *Let us first define what we will consistently refer to below as the* model Poisson problem. *The PDE*

$$-\Delta u = q, \quad 0 < x, y < 1, \tag{7}$$

*with $q(x, y)$ known and subject to homogeneous Dirichlet boundary conditions, is discretized using the standard 5-point difference scheme. Utilizing a uniform mesh width $h = 1/(J+1)$, and denoting by $\mathbf{b}$ the reshaped mesh function of $q(ih, jh)$, $1 \le i, j \le J$, and also letting $\mathbf{x}$ be likewise composed of solution mesh values, we have a problem (1) with $m = J^2$ unknowns and a sparse, large, symmetric positive definite matrix $A$. Note that the condition number $\kappa(A)$ is proportional to $m$.*

*The CG method is certainly better than LSD for this problem, for any positive integer $m$, never yielding a larger value of $f(\mathbf{x}_n)$ for any number of iterations $n$.*

*Next, we slightly perturb (7) by considering the nonlinear PDE*

$$-\Delta(u + 0.005/(1 + u^2)) = q, \quad 0 < x, y < 1,$$

*with $\|u\|$ known not to be small. We select $q$ as specified in Experiment 2 below: this is not central here. This PDE is discretized as above, and thus we proceed to solve a perturbed version of the model Poisson problem in which for any given vector $\mathbf{v}$ the product $A\mathbf{v}$ is replaced by the matrix-vector product $A(\mathbf{v} + 0.005/(1 + \mathbf{v}^2))$, where $\mathbf{v}^2$ denotes the component-wise square of $\mathbf{v}$. Now, as can be seen in Fig. 1, the LSD method is better, being more robust than CG against small perturbations to the problem (1).*[1]

[1] The vector $\ell_2$ norm is utilized here and elsewhere, unless otherwise specified.

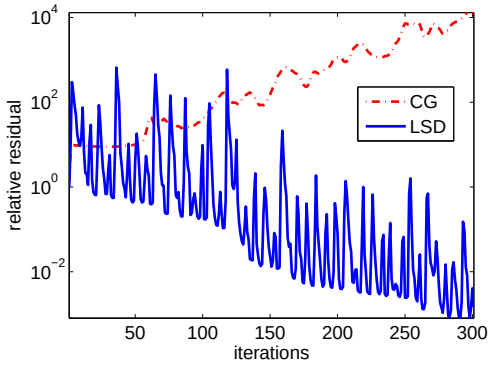



Figure 1: Relative residuals $\|\mathbf{r}_k\|/\|\mathbf{r}_0\|$ for CG and LSD applied to a small perturbation of the model Poisson problem, with $m = 3721$.

In [2] the gradient descent method was interpreted as a forward Euler discretization of the time-dependent ODE

$$\frac{d\mathbf{x}}{dt} = \mathbf{b} - A\mathbf{x}, \tag{8}$$

integrated to steady state. The absolute stability bound

$$\alpha_k \le \frac{2}{\lambda_1}, \tag{9}$$

where $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_m > 0$ are the eigenvalues of $A$,[2] must be obeyed if a uniform step size $\alpha_k = \alpha$ is to be employed. This artificial-time interpretation also suggests that the larger the average step size the smaller the total number of gradient descent iterations (or steps) required.

Further, it is well-known that the SD iteration is quite effective at the beginning, so long as not too many iterations are taken [1, 23, 2, 19]. It slows down later on because the vector of residuals tends, as the iterative algorithm proceeds, towards staying in a two-dimensional plane spanned by the eigenvectors corresponding to the largest and smallest eigenvalues, $\lambda_1$ and $\lambda_m$. This in turn causes the selected step sizes $\alpha_k$ to oscillate inanely, resulting in a method that is not better asymptotically than working throughout with the uniform step size

$$\alpha_k \equiv \alpha = \frac{2}{\lambda_1 + \lambda_m}, \tag{10}$$

see [2]. The faster gradient descent methods must therefore occasionally strongly violate the absolute stability bound (9). Such a large and unstable step is then preceded or followed by small steps to keep the whole process converging.

[2] Let us assume throughout for simplicity that $\lambda_1 > \lambda_2$ and $\lambda_{m-1} > \lambda_m$.

Thus, to better understand the behavior of faster gradient descent methods we should study the behavior of the corresponding dynamical systems, observing the progress of the normalized vector of squared residuals. In this paper we show, by calculating Lyapunov exponents and by numerical experimentation, that these dynamical systems exhibit chaotic behavior. In particular, their produced sequences of iterates are sensitive to the initial data, and the normalized residual vectors behave in a quasi-random fashion, rather than converging to a two-limit cycle as in the SD method. The prospect that the *slow* SD method yields an orderly two-limit cycle whereas the *faster* methods are chaotic is tantalizing in itself. Unfortunately, it also suggests that comparing these faster methods thoroughly could turn out to be a rather involved task, as stopping an oscillating sequence of residual norms by some fixed tolerance is somewhat arbitrary.

In Section 2 we introduce additional faster gradient descent methods and then investigate numerically their sensitivity to changes in the initial data. A comparison of various gradient descent variants for different matrices $A$ and different initial values is given in Section 3. In Section 4 we then define the corresponding dynamical systems and measure their Lyapunov exponents, observing chaos for the faster methods.

The simplest methods we consider are relaxed SD, where at each iteration $k$ we calculate $\alpha_k^{SD}$ by (6a) and set

$$\alpha_k = \omega \alpha_k^{SD}, \tag{11}$$

with $0 < \omega < 2$ a fixed constant [25]. We refer to this as SD($\omega$). It turns out that there is a range of parameter values $\omega$ contained in the interval $(0, 1)$ which yield fast, chaotic gradient descent methods. In Section 5 we further investigate this surprising damped steepest descent family.

The SD method is special in that its iteration residuals tend to oscillate in a two-dimensional plane [1], a trap which the faster, chaotic gradient descent methods avoid. But the behavior of the latter near planar sub-cycles is important nonetheless, and this is investigated in Sections 6 and 7. Conclusions and further discussion are offered in Section 8.

## 2 Sensitivity to Initial Conditions

Let us introduce several additional faster gradient descent variants that have been considered by others. We can denote the methods considered in [8] by HLSD(s), where each steepest descent step size is kept fixed for $s$ consecutive iterations. Thus, HLSD(2) = HLSD, and more generally they are defined by

$$\alpha_{sj} = \alpha_{sj+1} = \ldots = \alpha_{s(j+1)-1} = \alpha_{sj}^{SD}, \quad j = 0, 1, 2, \ldots. \tag{12}$$

The method we shall denote by LSD(s) sets

$$\alpha_k = \alpha_{k-s}^{SD}, \tag{13}$$

so LSD(1) = LSD. A variant of this, denoted here by RLSD(s), chooses the step size $\alpha_k$ randomly from $\alpha_k^{SD}, \alpha_{k-1}^{SD}, \ldots, \alpha_{k-s}^{SD}$. See [13]. Another method mixes the step sizes $\alpha_k^{SD}$ and

$$\alpha_k^{OM} = \frac{\mathbf{r}_k^T A \mathbf{r}_k}{(A\mathbf{r}_k)^T A \mathbf{r}_k}$$

either regularly alternating [9] (SDOM) or by a uniformly weighted random average of the two [2] (RSDOM). Finally we consider a relaxation method which sets the step size as in SD($\omega$) but with $\omega \in (0, 1]$ selected at each step as uniformly random (RSD). This was considered in [26].

All these methods significantly improve the speed of convergence of the SD method, for some choices of the parameters involved.

**Experiment 2** *Consider the unperturbed model Poisson problem described in Experiment 1. Table 1 records iteration counts required by different methods to bring the relative residual norm* $\|\mathbf{r}_k\|/\|\mathbf{r}_0\|$ *below a tolerance of* $10^{-12}$*, for a right hand side* $\mathbf{b} = \mathbf{1}$ *of all ones.*

| $m$ | $\mathbf{x}_0$ | CG | SD | LSD | HLSD | LSD(2) | SD(0.8) | RSD | RLSD(4) |
|---|---|---|---|---|---|---|---|---|---|
| 49 | (a) | 10 | 341 | 71 | 69 | 87 | 113 | 145 | 82 |
| | (b) | 10 | 341 | 77 | 62 | 85 | 120 | 170 | 77 |
| 225 | (a) | 33 | 1414 | 141 | 179 | 152 | 279 | 302 | 127 |
| | (b) | 32 | 1414 | 215 | 151 | 143 | 290 | 393 | 166 |
| 961 | (a) | 71 | 5721 | 412 | 279 | 313 | 585 | 717 | 311 |
| | (b) | 70 | 5721 | 441 | 417 | 377 | 535 | 1049 | 319 |
| 3969 | (a) | 143 | 22979 | 797 | 712 | 732 | 1331 | 1313 | 692 |
| | (b) | 143 | 22979 | 976 | 567 | 828 | 1459 | 1901 | 585 |

Table 1: Iteration counts for the model Poisson problem using gradient descent with different step size choices for initial vectors (a) $\mathbf{x}_0 = \mathbf{0}$ and (b) $\mathbf{x}_0 = 10^{-3} \cdot \mathbf{1}$.

*We use two starting guesses as specified in the Table's caption. They are equally smooth and differ from each other by* $10^{-3}$ *in the maximum norm.*

*The results in Table 1 exhibit the usual traits of the faster gradient descent methods observed, e.g., in [2]. Thus, (i) the iteration counts increase much slower than* $\kappa = O(m)$*, behaving more like those of CG in trend, (ii) none of these gradient descent method variants is consistently better than the others, and (iii) the progress of the iteration counts as a function of* $m$ *is less consistent than that of CG. The latter observation relates directly to the fact that the quantities* $\|\mathbf{r}_k\|$ *oscillate wildly as a function of* $k$*; see Fig. 2 of [2], where the behavior of the resulting step size sequence is also depicted.*

*Furthermore, we observe here the sensitivity of the total number of iterations required to achieve a fixed accuracy to small changes in the initial vector* $\mathbf{x}_0$*. This*

*is in marked contrast to the behavior of the CG iteration, or the SD iteration, and it suggests a chaotic behavior of the iterative process for the faster gradient descent methods.*

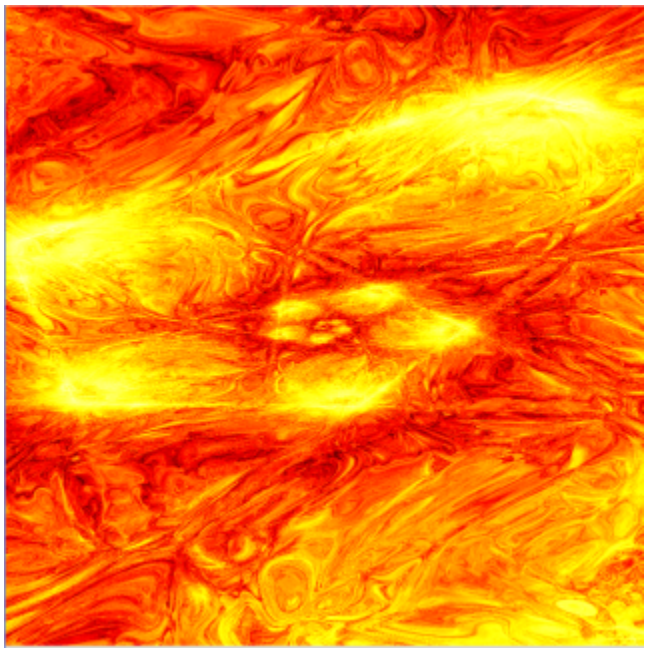


Figure 2: Iteration counts for LSD with a perturbation in a plane of extent $10^{-3}$. Brighter color corresponds to higher iteration number.

To further illustrate the point, Fig. 2 depicts iteration counts, applying LSD to the model Poisson problem for $m = 121$, where now the initial guess is perturbed to

$$\mathbf{x}_0 = a_1\mathbf{e}_1 + a_2\mathbf{e}_2.$$

Here $-10^{-3} \leq a_i \leq 10^{-3}$, and $\mathbf{e}_1$ and $\mathbf{e}_2$ are two randomly chosen orthogonal vectors. The iteration counts for a tolerance of $10^{-8}$ on the relative residual norm are plotted as a function of $a_1$ on the horizontal axis and $a_2$ on the vertical axis. These counts varied between 96 (black) and 127 (white). The intricate structure is characteristic of chaotic dynamical systems.

# 3 Comparison of Various Methods

In this section we compare statistically the performance of the faster gradient descent algorithms by averaging over 100 different initial vectors (or equivalently over different right hand sides).

In order to make sure we are not seeing artifacts from the highly symmetrical Poisson problem on a uniform mesh, we consider here also three other matrices.

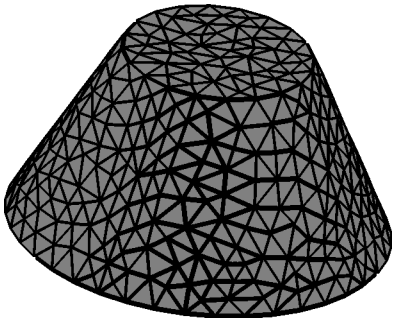


Figure 3: A generalized Poisson equation with variable conductivity $\sigma(\mathbf{x})$ was discretized using linear finite elements, resulting in an $m \times m$ matrix with $m = 5970$ and $\kappa(A) = 3800$.

1. First, consider what is commonly referred to as the generalized Poisson equation

$$-\nabla(\sigma(\mathbf{x})\nabla u) = q, \tag{14}$$

subject to homogeneous Dirichlet boundary conditions, on a 3D cone geometry depicted in Fig. 3. The (positive) conductivity $\sigma(\mathbf{x})$ was taken to vary linearly in a direction orthogonal to the symmetry axis of the cone. The resulting system was discretized using a nodal finite element method with second order elements, resulting in a matrix $A$ of dimension $m = 5970$. We refer to this below as the FEM model.

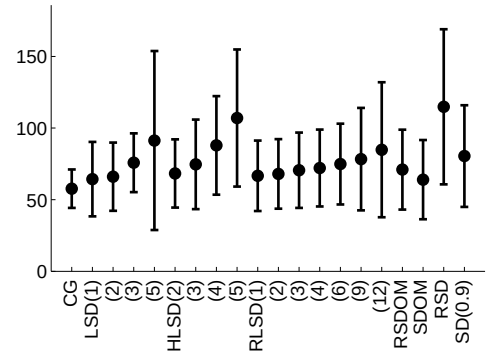
2. Next, denoting by $L$ the matrix corresponding to the model Poisson problem for (7), we consider the matrix

$$A = L^{-2} + 10^{-3}L, \tag{15}$$

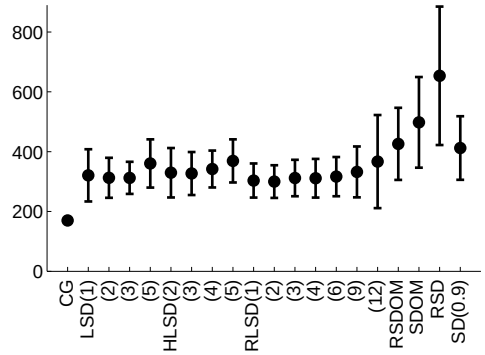
which is typical in systems that arise in regularized inverse problems involving PDEs [30]. Note that to form matrix-vector products $A\mathbf{v}$, an inner linear problem $L\mathbf{y} = \mathbf{w}$ must be solved twice. In our experiments this is done accurately using a direct method.

3. Finally, we consider a diagonal matrix $A$ with random positive eigenvalues constrained to have a specified condition number $\kappa = \lambda_1/\lambda_m$.

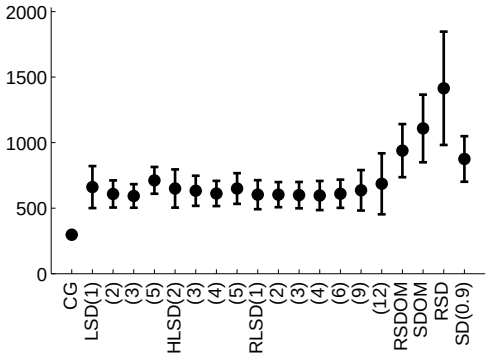
**Experiment 3** *Fig. 4 shows the average number of iterations required to converge to various tolerances for the FEM model. Results with tol=*$10^{-6}$ *for a model Poisson system of comparable size are also displayed for comparison. We show the average number of iterations for* 100 *randomly chosen smooth initial vectors* $\mathbf{x}_0$*. The random vectors were constructed by placing* 10 *Gaussian hat sources of the form* $e^{-||\mathbf{x}-\tilde{\mathbf{x}}||^2/2}$ *at random locations* $\tilde{\mathbf{x}}$ *in the geometry. The interval shown on each datum represents the* 95% *confidence interval.*



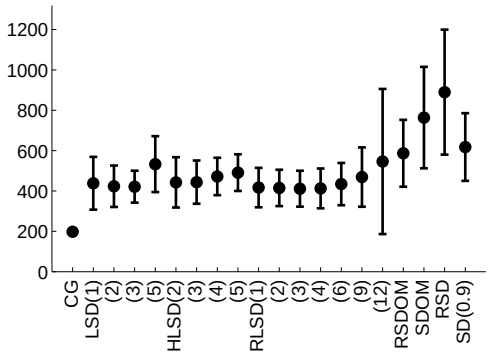

(a) FEM for (14), tol=$10^{-2}$




(b) FEM for (14), tol=$10^{-6}$




(c) FEM for (14), tol=$10^{-12}$




(d) model Poisson problem, tol=$10^{-6}$

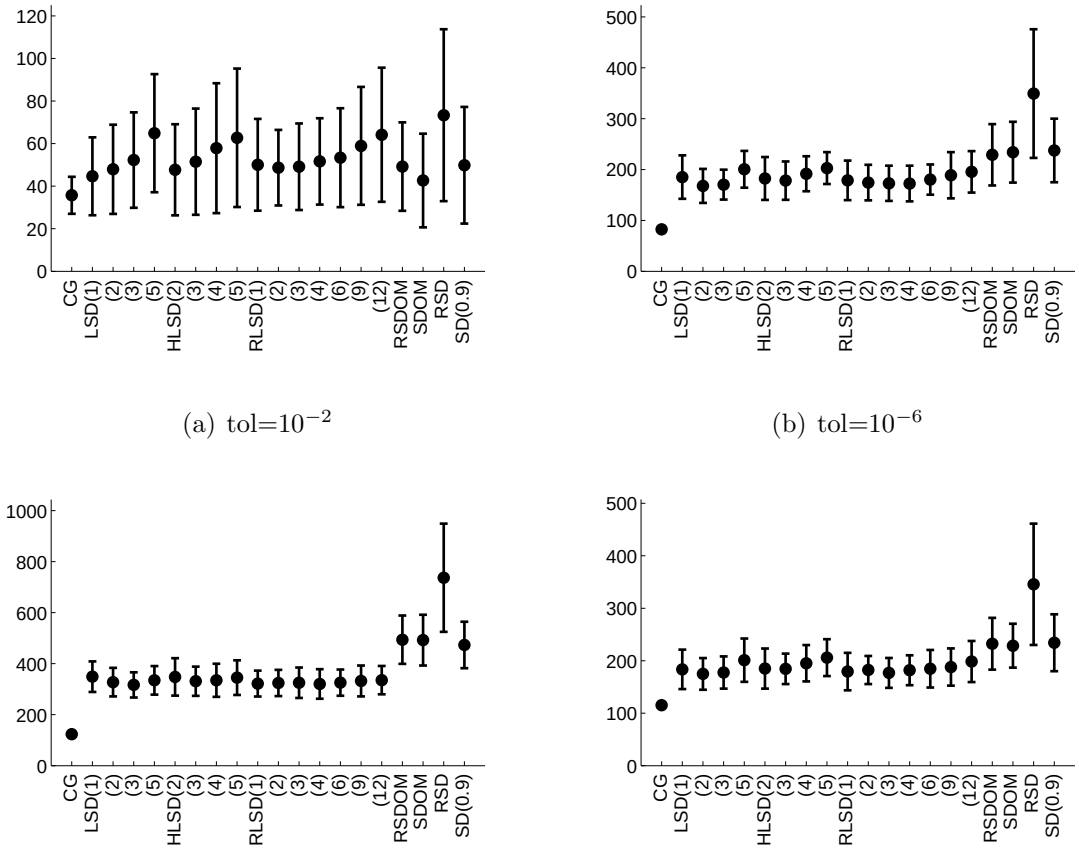
Figure 4: Average iteration counts and 95% confidence intervals for the finite element discretization of (14) with $m = 5970$, $\kappa(A) = 3800$. Also displayed are corresponding results for the model Poisson model problem with $m = 5929$, $\kappa(A) = 3500$.

*Fig. 5 displays the results applied to (15) for various tolerances. For comparison we also show results with tol=$10^{-6}$ for a diagonal positive definite random matrix of the same size and condition number.*

*These experiments yield the following observations:*

1. *The methods' performance for the finite element discretization of the generalized Poisson system is similar in general trend to their performance for the model finite difference discretization of the Poisson equation on a square mesh.*

2. *The methods' performance for the inverse problem matrix (15) is similar in trend to that for a diagonal positive definite random matrix having the same condition number. Together with the previous observation, it appears that we can*




(a) tol=$10^{-2}$

(b) tol=$10^{-6}$

(c) tol=$10^{-12}$

(d) Random, tol=$10^{-6}$

Figure 5: Average iteration counts and 95% confidence intervals for the inverse problem equations (15) with $m = 961$, $\kappa(A) = 400$. Results for a diagonal positive definite random matrix of the same size and condition number are also displayed.

*analyze these fast gradient descent methods with synthetic (and not so realistic) matrices.*

3. *For a very coarse tolerance of $10^{-2}$, the variants LSD, HLSD, SDOM, and RLSD(1) (which use information from at most one previous step) are clearly superior to the others. The methods with higher lag tend to oscillate in a wilder fashion and with longer periods.*

4. *RSD considered in [26] performs much worse than SD($\omega$) with a constant under-relaxation parameter $\omega = 0.9$.*

5. *At tighter tolerances than $10^{-2}$, the one-step methods (R)SDOM, RSD, and SD($\omega$) are inferior to other faster gradient descent variants.*

6. *In the families of LSD, HLSD, and RLSD, a longer lag offers at most an impractically small advantage at tighter tolerances.*

Digesting these observations we conclude that LSD, HLSD, and RLSD(1) are experimentally indistinguishable, and are the methods that perform best under a wide variety of circumstances.

Next we turn to examine the behavior of the residual vector for some of the faster methods on one example.

**Experiment 4** *Consider the FEM model and apply LSD, LSD(2), and HLSD. In Fig. 6(a) we plot the relative residual norm over a small window. For each of the methods we also display the behavior of* $||\mathbf{r}_k||/||\mathbf{r}_{k-1}||$ *in the frequency domain, using the Morlet wavelet transform [17]. The following observations are in order:*

1. *LSD seems quasi-periodic, as does LSD(2); HLSD is less so.*
2. *The period of LSD is shorter and the amplitude of oscillation much smaller than for LSD(2).*
3. *The dominant frequencies of LSD and LSD(2), about* 0.16 *and* 0.1 *(corresponding to periods of about* 6 *and* 10 *iterations, respectively), appear to be universal for the methods; we have observed the same values for a variety of matrices and right hand sides.*
4. *A large amount of pseudo-randomness can be observed.*

The numerical experiments above clearly suggest chaotic behavior in the faster gradient descent methods.

# 4 Chaos

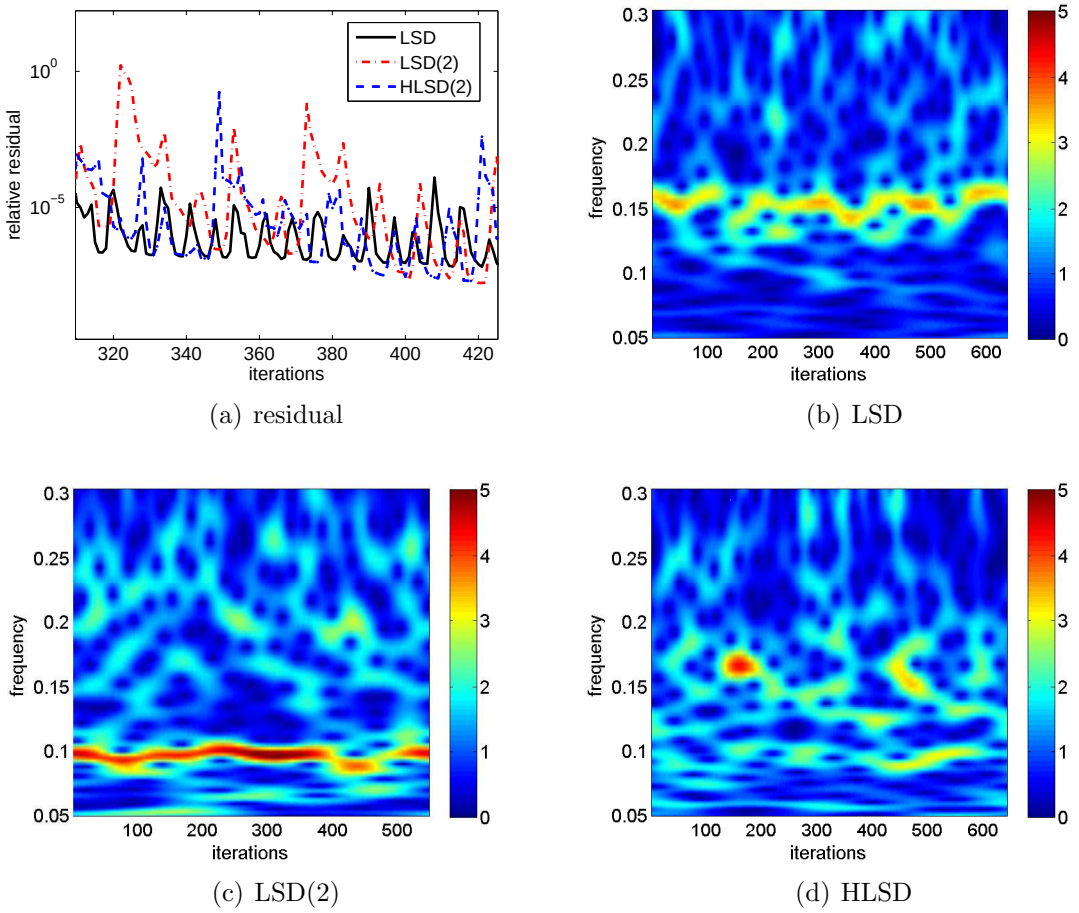
The gradient descent family of methods (4) is completely characterized by the residual evolution, written as

$$\mathbf{r}_{k+1} = (1 - \alpha_k A)\mathbf{r}_k, \quad k = 0, 1, \ldots. \tag{16}$$

Furthermore, if we write $A = U\Lambda U^T$ with $U$ orthogonal and $\Lambda$ the diagonal matrix of eigenvalues of $A$, and let $\hat{\mathbf{r}} = U^T\mathbf{r}$, then (16) becomes

$$\hat{\mathbf{r}}_{k+1} = (1 - \alpha_k \Lambda)\hat{\mathbf{r}}_k, \tag{17}$$

with $||\hat{\mathbf{r}_k}|| = ||\mathbf{r}_k||$. Thus, (17) has precisely the same convergence behavior as (16), and hence we may consider without loss of generality a diagonal $A$ for analysis purposes. Note that now, if $\alpha_k = \lambda_i^{-1}$ for some $i$, $1 \leq i \leq m$, then the $i$th component




(a) residual (b) LSD

(c) LSD(2) (d) HLSD

Figure 6: Performance of LSD, LSD(2), and HLSD: (a) residual versus iteration number on a limited interval of iterations; (b-d) individual frequency domain representations (i.e., magnitudes of the Morlet wavelet transform coefficients) of $||\mathbf{r}_k||/||\mathbf{r}_{k-1}||$ over the entire iteration range. Results are for the FEM model with $m = 5970$.

of the next residual vanishes: $r_i^{(k+1)} = 0$. If $m$ is large, though, then even if we knew the eigenvalues we would not want to use them in this way in practice.

Below we will often omit the iteration index $k$ where no confusion can possibly arise. An alternative notation to (16), for instance, is

$$\mathbf{r} \leftarrow (1 - \alpha A)\mathbf{r}. \tag{18}$$

To study the behavior of these residual vectors, we associate with $\mathbf{r}$ the Akaike probability $\mathbf{p}$, see [1], which is the component-wise square of the normalized residual, given by

$$p_i = (r_i)^2/||\mathbf{r}||^2, \quad i = 1, \ldots, m. \tag{19}$$

Note that $\mathbf{p}$ is formally a probability distribution as its values are non-negative and they sum to 1, but it is not really the probability of anything of specific interest as such.

Under the gradient descent family, the Akaike probability evolves according to

$$p_i \leftarrow (\lambda_i - \gamma)^2 p_i / M, \tag{20}$$

where $\boldsymbol{\lambda}$ is the vector of the eigenvalues of $A$, $\gamma = 1/\alpha$ is the inverse step size, and $M$ is a generic normalizing factor.

Below we use the notation

$$\langle \mathbf{v} \rangle_{\mathbf{p}} \equiv \sum_{i=1}^{m} v_i p_i \tag{21}$$

for the mean value of $\mathbf{v}$ under the probability distribution $\mathbf{p}$. We will further omit the subscript $\mathbf{p}$ in (21) if no ambiguity can arise. With this notation we have

$$M = \langle (\boldsymbol{\lambda} - \gamma)^2 \rangle_{\mathbf{p}},$$

and the relative reduction in residual norm

$$\zeta_k = ||\mathbf{r}_{k+1}||^2 / ||\mathbf{r}_k||^2$$

can be written (using (18) and (19)) as

$$\zeta = \langle (\boldsymbol{\lambda} - \gamma)^2 \rangle / \gamma^2. \tag{22}$$

For the steepest descent method we have $\gamma^{SD} = \langle \boldsymbol{\lambda} \rangle = \sum_{i=1}^{m} \lambda_i p_i$ and

$$\zeta^{SD} = \langle (\boldsymbol{\lambda} - \langle \boldsymbol{\lambda} \rangle)^2 \rangle / \langle \boldsymbol{\lambda} \rangle^2, \tag{23}$$

which is the relative variance of $\boldsymbol{\lambda}$. So if we could get the variance to be small, we would get a large reduction in residual norm.

In [1] it was shown that the iteration (20), viewed as a non-linear dynamical system, has a two-cycle attractor. In a nutshell, this paper showed by explicit computation that the variance of $\boldsymbol{\lambda}$ under $\mathbf{p}$ in (20) is non-decreasing, from which it follows that it must tend to a constant. From this it follows that in the limit $\mathbf{p}$ must have only two nonzero values, and it oscillates between them under (20). A stability argument then shows that these are the components associated with the largest and smallest eigenvalues of $A$. In particular, this means that $\zeta^{SD}$ will never become small, unless the oscillatory values are close to 0 and 1.

The proof in [1] does not apply for any of the other members of the gradient descent family considered here. To analyze these other, faster methods and establish their chaotic behavior we compute the Lyapunov spectrum [21] associated with

the dynamical system (20) numerically. If the maximum eigenvalue $\mu$, called the Lyapunov exponent, is positive, the dynamical system is chaotic.

We first write (20) in first order form. For the LSD(s) family the inverse step size is given by

$$\gamma = a_0 \langle \boldsymbol{\lambda} \rangle_{\mathbf{p}} + \sum_{j=1}^{s} a_j \langle \boldsymbol{\lambda} \rangle_{\mathbf{p}_{-j}},$$

with $\mathbf{p}_{-j} = \mathbf{p}_{k-j}$ where $\mathbf{p} = \mathbf{p}_k$. Various choices of the coefficients $a_j$ result in different such methods, including the one-step SD($\omega$) if we set $a_0 = 1/\omega$. The iteration (20) can now be written as

$$\begin{aligned} p_i &\leftarrow \left( \lambda_i - a_0 \langle \boldsymbol{\lambda} \rangle_{\mathbf{p}} - \sum_{j=1}^{s} a_j z_j \right)^2 p_i / M \\ z_1 &\leftarrow \langle \boldsymbol{\lambda} \rangle_{\mathbf{p}} \\ z_2 &\leftarrow z_1 \\ &\cdots \\ z_s &\leftarrow z_{s-1}. \end{aligned} \tag{24}$$

For the HLSD(s) family we can collect the $s$ iterations with the same step size into a single step, and the corresponding dynamical system reads

$$p_i \leftarrow (\lambda_i - \langle \boldsymbol{\lambda} \rangle_{\mathbf{p}})^{2s} \, p_i / M. \tag{25}$$

Each step of (25) corresponds to $s$ steps of gradient descent (or an s-stage, 1st order accurate Runge-Kutta method for (8)).

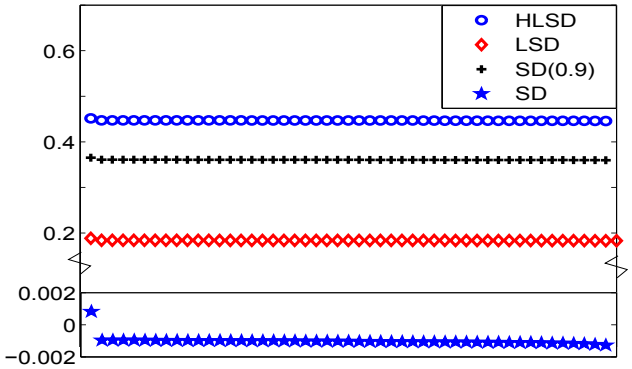



Figure 7: Lyapunov spectra for various methods applied to the model Poisson problem with $m = 49$, for $K = 10000$ iterations. The small positive $\mu$ for the SD method tends to 0 for $K \to \infty$.

The analytic formulas of the Jacobians $J_k$ of the maps (24) and (25) can be easily computed. We then perform a large number $K$ of iterations and compute the product

of all the iteration Jacobians

$$J(K) = J_{K-1} \cdots J_1 J_0.$$

Finally, we compute the spectral radius of $J(K)$, denoted $\nu_K$, and then estimate the Lyapunov exponent by

$$\mu \approx \mu_K = \frac{1}{K} \log \nu_K.$$

To prevent overflow and misalignment of the directions of maximal expansion we perform a QR decomposition of $J(K)$ every 10 iterations, rescale R to have maximum matrix element 1, and keep track of the logarithm of the (rapidly growing) scale factor.

For a stable system, $\nu_K \approx e^{\mu K}$ for some negative $\mu$, whereas for a chaotic system, $\mu > 0$ and nearby orbits separate exponentially, cf. [27, 28].

**Experiment 5** *Fig. 7 shows the Lyapunov spectrum for various methods applied to the model Poisson problem with $m = 49$. For all the fast methods we have $\mu > 0$, whereas SD is marginally stable: further experiments with larger $K$ suggest that $\mu \to 0$ for $K \to \infty$.*

*The over-relaxed SD($\omega$) variant with $\omega = 1.1$ (not shown) has $\mu < -.2 < 0$ and is stable. Thus, the over-relaxed SD($\omega$) joins the list of slow methods. See Experiment 6 for further experimentation and discussion of this case.*

# 5 Relaxed steepest descent

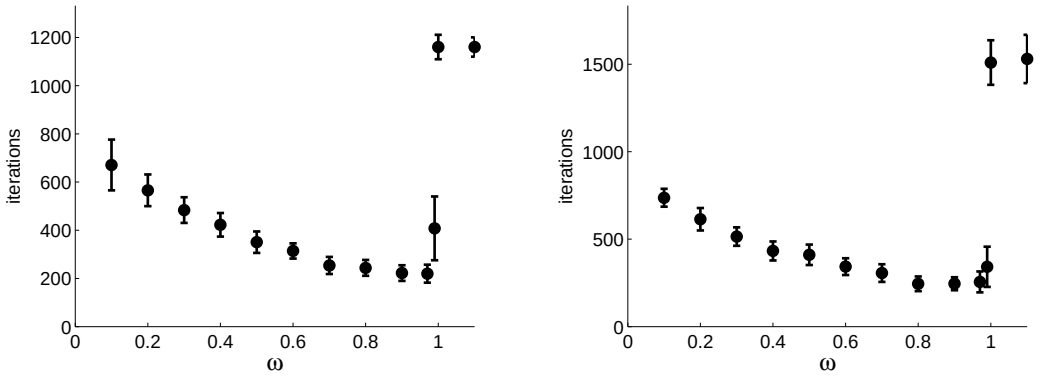
The under-relaxed steepest descent method SD($\omega$) is not quite as fast as LSD, HLSD or RLSD. But it is close, as Figs. 4 and 5 show, and this in itself may be considered surprising. Note that here is a "clean", memoryless one-step method. Furthermore, there are no random parameters and no switches in the step size selection strategy (such artifacts could be considered as external to a dynamical system).

Moreover, since $\psi(\alpha) = \psi(\omega\alpha_k^{SD})$ defined in (5) is a quadratic function that obtains its minimum at $\omega = 1$, and $\psi(2) = \psi(0)$, we have $\psi(\alpha) < \psi(0)$ for $0 < \omega < 2$ (unless we already are at the solution $\mathbf{x}$). Standard arguments (e.g., [26, 24]) then imply that the method yields monotonic decrease in $f(\mathbf{x})$ of (2) and converges Q-linearly.

Finally, although this method with $\omega < 1$ takes at each iteration a fraction of the SD step size, its average step size is much larger than that of SD! Here then is one of the simplest and cleanest instances of both a chaotic system and the peril of greed (in numerical algorithms at least).

**Experiment 6** *Fig. 8 displays average iteration counts as a function of the relaxation parameter $\omega$ for the model Poisson problem with $m = 225$, and a small version of the FEM model with $m = 224$. A tolerance of $10^{-10}$ was employed.*

*Fig. 9(a) further shows for the model Poisson problem with $m = 225$ and various values of $\omega$ the progress of the relative residual as a function of iteration.*




(a) Model Poisson problem (b) FEM model

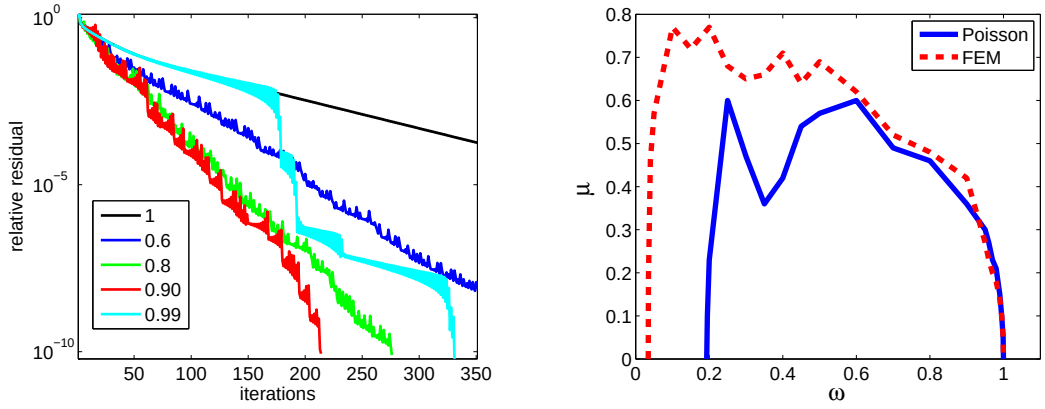
Figure 8: Under-relaxed (or damped) steepest descent iteration counts satisfying a tolerance of $10^{-10}$ for the model Poisson problem and for the FEM model.

*In Fig. 9(b) the Lyapunov exponent is displayed where it is positive. Apart from the absolute number of iterations, the results for the model Poisson problem are almost independent of $m$. In particular, we found the Lyapunov exponent to vary less than 1% between $m = 49$ and $m = 961$.*

*We make the following observations:*

1. *Almost any $.5 < \omega < 1$ improves SD dramatically.*
2. *There is no chaos for $\omega \geq 1$, and using $\omega > 1$ is worse than SD (which uses $\omega = 1$).*
3. *The transition to chaos at $\omega = 1$ is quite sudden, reflected in a very sharp increase of $\mu$.*
4. *Depending on the problem (1), there is a minimum $\omega_c$ below which there is again no chaos. For the model Poisson problem we estimate $\omega_c \approx 0.2$ whereas for the finite element matrix $\omega_c \approx 0.03$.*
5. *In all cases we tried (including many that are not shown here) a value of $\omega \approx 0.9$ performs very well.*
6. *In the examples shown in Figs. 8 and 9 the value $\omega = 0.99$ also performs well, but as we see from the residual plot of Fig. 9(a) it takes about 200 iterations before chaos kicks in, a reflection of the small Lyapunov exponent. As a result this choice of $\omega$ yields a very non-monotone behavior.*

Let us consider SD($\omega$) near the critical value $\omega = 1$. As can be seen in Fig. 9(a) the effect of $\omega$ is very slow to show up and in the beginning the behavior is like SD.




(a) Model Poisson problem for various $\omega$ values

(b) Lyapunov exponent $\mu$

Figure 9: Residual decrease and Lyapunov exponent (where positive), Experiment 6.

Recall that for SD the Akaike probabilities $\mathbf{p}$ tend to having just $p_1$ and $p_m$ nonzero, with all the other components decaying to zero. If $p_2, \ldots, p_{m-1}$ are negligibly small then (20) reduces to a one-dimensional dynamical system, since we have $p_m = 1 - p_1$. For $\omega = 1$ we obtain

$$p_1 \leftarrow 1 - p_1$$

which yields a two-cycle with $p_1$ and $p_m$ exchanging values. These values themselves are indeterminate, which is reflected in the single zero eigenvalue in the Lyapunov spectrum of SD (see Fig. 7). Let us now examine in detail the dynamics in this case, which corresponds to the behavior for the planar case $m = 2$.

For generic $\omega$, still assuming only $p_1$ and $p_m$ nonzero, we parameterize $p_1 = c/(1+c)$ and $p_m = 1/(1+c)$ with $c \geq 0$. Substitution in (20) yields after some manipulations

$$c \leftarrow F(c) = \left( \frac{1 + \kappa\eta(1+c)}{(1-\eta)c - \eta} \right)^2 c, \tag{26}$$

where $\eta = (\omega - 1)/(\kappa - 1)$, with the condition number $\kappa = \lambda_1/\lambda_m$.

The dynamical system (26) depends non-trivially on both $\kappa$ and $\omega$, as depicted in Fig. 10. It can be seen from the figure that the system is chaotic for most parameters $\omega < 1$, but not everywhere.

The one-cycles of (26) can be found by solving $F(c) = c$ for $c \geq 0$, and they will be stable if and only if $|F'(c)| < 0$ at the solution. There is a fixed point at $c = 0$ with $F'(0) = (\kappa\omega - 1)^2/(\omega - 1)^2$, which is stable only for $0 < \omega \leq 2/(1 + \kappa)$, i.e., for very small $\omega$. The second fixed point is at

$$c^* = -\frac{1 + \kappa\eta}{\kappa\eta} = \frac{\omega(\kappa + 1) - 2}{2\kappa - (\kappa + 1)\omega},$$

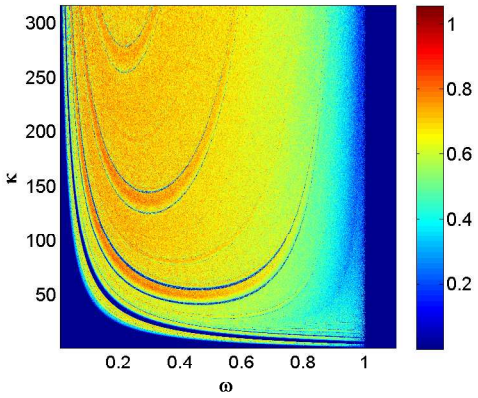



Figure 10: Lyapunov exponent where positive for relaxed steepest descent as a function of $\omega$ and $\kappa$ for the two-dimensional case.

and exists for

$$2/(1+\kappa) \le \omega \le 2\kappa/(1+\kappa),$$

i.e., almost everywhere for large $\kappa$. Calculation of $|F'(c)|$ reveals that the fixed point is unstable for $\omega < 2/(1+\kappa)$ or $\omega > 2\kappa/(1+\kappa)$, which are close to 0 and 2 and, more interestingly, for $4\kappa/(1+\kappa)^2 < \omega < 1$. The latter covers most of the region of interest for reasonably large values of $\kappa$.

For $\omega$ a bit greater than 1, i.e., slightly over-relaxed, we have a stable cycle near $c = 1$, corresponding to $p_1 = p_m = 1/2$ and a uniform step size. This is the neutrally stable SD two-cycle, which has now become a one-cycle with specific values of $p_1, p_m$.

Fig. 10 suggests that the unstable fixed point does not settle into a stable higher order cycle, but decays into chaos, except for some sparse special combinations of $\omega$ and $\kappa$. While not being able to perform a complete analysis we can demonstrate chaos for any specific numerical values of $\omega$ and $\kappa$ by showing that there exists a three-cycle, i.e., a non-negative solution of

$$F(F(F(c))) = c. \tag{27}$$

In that case Sharkovsky's Theorem applies, since $F(c)$ is continuous for $0 < \omega < 1$, and there must be (unstable) cycles of any period as well as chaotic cycles [20]. For example, with $\kappa = 10$, $\omega = 0.5$, the only (real) solutions of (27) are 0 and 7/29 which are however also one-cycles. But for $\kappa = 10$, $\omega = 0.9$, we find 4 nontrivial three-cycles, by numerically solving (27). Fig. 11 depicts the behavior of $p_1$ for $\omega = 0.9$ and $\omega = 1.1$, for a diagonal matrix $A$ with $\kappa = 100$. Note that for the unstable one the oscillations in $p_1$ grow wider and wider until they are thrown back to approximately the middle, and another cycle starts. This is significant since (23), approximately valid for $\omega \approx 1$, indicates that the residual norm reduction is large for $p_1$ near 0 or 1,

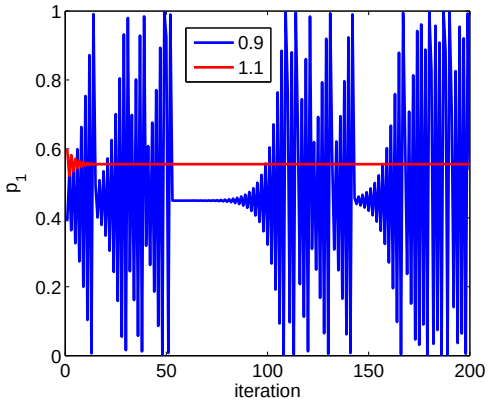



Figure 11: Evolution of $p_1$ under (26) for $\omega = 0.9$ and $\omega = 1.1$. The condition number was $\kappa = 100$, and $p_1$ was initialized to 0.5.

as this minimizes the variance. Specifically, (23) becomes

$$\zeta = p_1(1-p_1)/(p_1 + \lambda_m/(\lambda_1 - \lambda_m))^2. \tag{28}$$

# 6 LSD and HLSD in the planar case

The analysis for just $p_1$ and $p_m$ nonzero is easier for LSD and HLSD. Thus we set $m = 2$ in this short section, preparing for the next one. For LSD we obtain

$$p_1^{(k+1)} = \frac{p_1^{(k)}\left(p_m^{(k-1)}\right)^2}{p_1^{(k)}\left(p_m^{(k-1)}\right)^2 + p_m^{(k)}\left(p_1^{(k-1)}\right)^2}, \tag{29}$$

and similarly for $p_m^{(k+1)}$. Parameterizing as before by $p_1 = c/(1+c)$ and $p_m = 1/(1+c)$ we obtain

$$c_{k+1} = c_k/(c_{k-1})^2. \tag{30}$$

The system (30) has an unstable fixed point at $c = 1$ (i.e., $p_1 = p_m = 1/2$) and rapidly diverges. Fig. 12 displays $\mathrm{loglog}(c) = \mathrm{sign}(c)\log_{10}(|\log_{10}(c)|)$ starting at $c_0 = 3/2$ and $c_1 = 2/3$. The values alternate between very large (already $10^{10^7}$ after 50 iterations) and very small ones. The large and small values come in groups of two or three with no discernible pattern. In this case (22) becomes

$$\zeta_{k+1} = \langle(\boldsymbol{\lambda} - \langle\boldsymbol{\lambda}\rangle_{\mathbf{p}_{k-1}})^2\rangle_{\mathbf{p}_k}/\langle\boldsymbol{\lambda}\rangle^2_{\mathbf{p}_{k-1}},$$

and if both $\mathbf{p}_k$ and $\mathbf{p}_{k-1}$ are close to $(1,0)^T$ or $(0,1)^T$, which does happen as we can see in Fig. 12, then this becomes very small in magnitude, resulting in a very fast

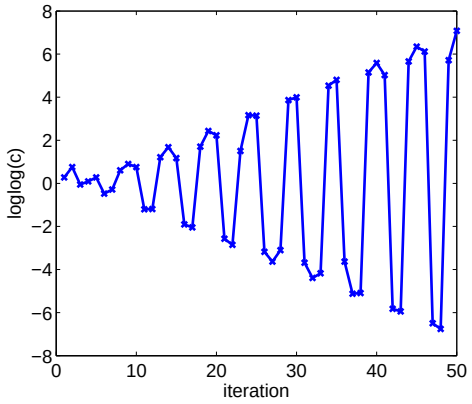



Figure 12: Evolution of $c$ under (29) on a doubly logarithmic scale for a problem with $m = 2,\ \kappa = 100$.

reduction of the residual norm: significantly faster not only than SD but also than SD(0.9) for $\kappa \gg m = 2$.

For HLSD the analogue of (29) reads

$$p_1^{(k+1)} = \frac{\left(p_m^{(k)}\right)^4}{p_1^{(k)}\left(p_m^{(k)}\right)^4 + p_m^{(k)}\left(p_1^{(k)}\right)^4}, \tag{31}$$

which becomes in terms of $c$

$$c_{k+1} = 1/(c_k)^3. \tag{32}$$

This recursion has the explicit solution

$$c_k = c_0^{(-3)^k},$$

which is again doubly exponentially divergent and approaches an alternation between $\mathbf{p}_k = (1, 0)^T$ and $\mathbf{p}_k = (0, 1)^T$. Since in (22) $\gamma = \gamma_k$ is held fixed over two consecutive iterations, we again get a doubly exponential reduction of the residual norm. The corresponding iteration count for a given tolerance is thus comparable to LSD and significantly better than SD($\omega$) for any $\omega$.

# 7 The role of planar sub-cycles in general

The analysis provided in Section 5 is relevant also for the general case of $m > 2$. In this case none of the faster methods ever settle into a sustained pattern where $p_2, \ldots, p_{m-1}$ are negligibly small, but numerical evidence presented in this section shows they do so briefly. The mechanisms from Sections 5 and 6 then cause a fast

reduction of the residual norm until $\lambda_1 p_1$ no longer dominates $\lambda_i p_i$, $i > 1$, and a more complicated dynamics involving more than just $p_1$ and $p_m$ sets in. In effect the chaos acts as a pump that occasionally moves the significant residual components into just the first and last components. Then the planar fast error reduction mechanism discussed above kicks in and the residual is again spread over all its components. In the remainder of this section we present numerical evidence that this pattern is actually happening.

## Under-relaxed steepest descent

Let us now consider $p_2, \ldots, p_{m-1}$ very small but positive. For SD this is a stable situation, and $p_2, \ldots, p_{m-1}$ get smaller and smaller [1]. Let us write (20) as

$$p_i \leftarrow \xi_i p_i,$$

with $\xi_i$ the growth (or reduction) factor. Neglecting $p_2, \ldots, p_{m-1}$ and writing $(p_1, p_m) = (1-p, p)$ we have for $i = 2, \ldots, m-1$

$$\xi_i = \frac{p}{1-p}(1-\theta_i/p)^2, \quad \text{with } \theta_i = (\lambda_1 - \lambda_i)/(\lambda_1 - \lambda_m),$$

so $0 < \theta_i < 1$. Now, if each iteration interchanges $p_1$ and $p_m$ as in SD, the product of the growth factors after two iterations is

$$\xi_i^{(k)} \xi_i^{(k+1)} = [(1-\theta_i/p)(1-\theta_i/(1-p))]^2 = \left(1 + \frac{\theta_i(\theta_i - 1)}{p(1-p)}\right)^2 < 1, \tag{33}$$

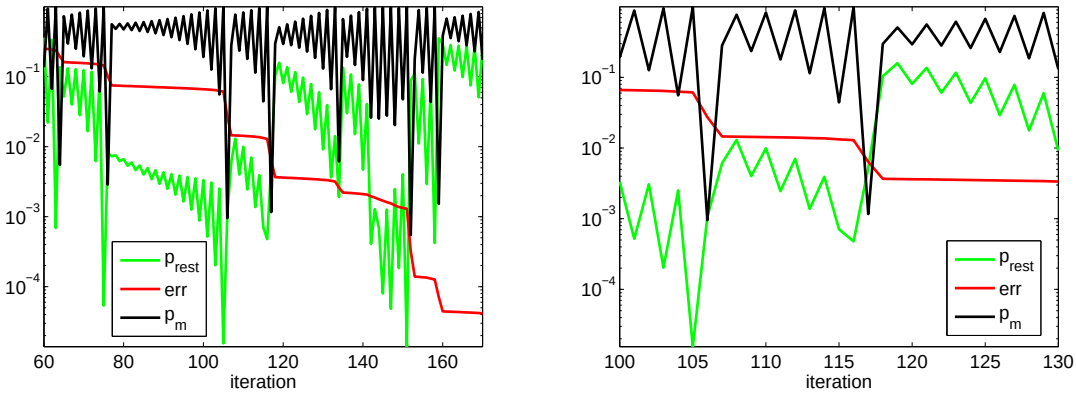
hence stability follows. (This is reflected in the negative Lyapunov eigenvalues depicted in Fig. 7.) If the two-cycle is not stable, however, as is the case in damped steepest descent, then this stability no longer holds and the probabilities $p_2, \ldots, p_{m-1}$ can grow.

**Experiment 7** *Fig. 13 shows part of the iterations applying SD(0.99) to the model Poisson problem for $m = 961$. We have diagonalized $A$, and plot the probability component $p_m$, the error $\sqrt{\mathbf{r}^T A^{-1} \mathbf{r}}$ (which is monotonically decreasing and therefore somewhat clearer than the $\ell_2$ residual norm), and*

$$p_{rest} = \sum_{i=2}^{m-1} p_i,$$

*which measures how much action is going on outside the extremal planar subspace. The following observations are in order:*

1. *$p_{rest}$ remains fairly small.*




(a) Iterations (b) A zoom

Figure 13: Error, $p_m$, and $p_{rest}$ for SD(0.99) applied to the model Poisson problem.

2. *The diverging oscillation patterns observed in Fig. 11 occur here too.*

3. *When the oscillations in $p_m$ grow slowly, (33) is approximately valid, hence $p_{rest}$ gets reduced in an oscillatory manner as in SD.*

4. *The error reduces in jumps which occur when the $p_m$ oscillation terminates and a new cycle starts. The reason for the sharp reduction in error is that $(p_1, p_m)$ hit their extremal values, resulting in a small $\zeta$ in (28).*

5. *$p_{rest}$ usually increases sharply after each jump in the error, because (33) is no longer approximately valid.*

## LSD and HLSD

We next consider a numerical experiment using the faster methods LSD and HLSD, which mimics the one above for damped SD, in an attempt to see what is different for these methods.

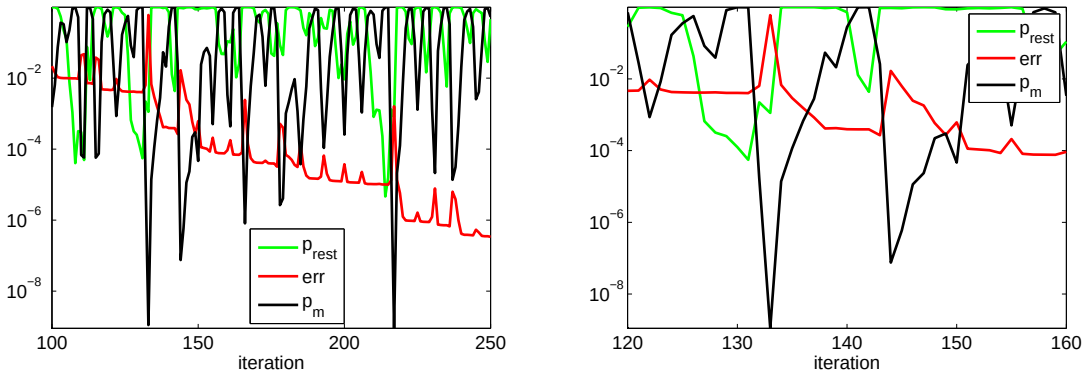
**Experiment 8** *Figs. 14 and 15 show comparable plots to those in Fig. 13 for LSD and HLSD, respectively.*

*The following observations are in order:*

1. *$p_{rest}$ remains significantly large most of the time.*

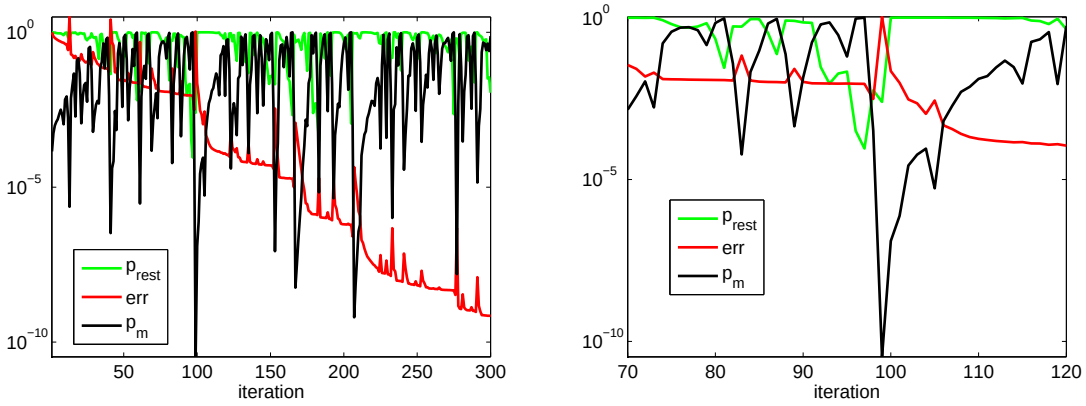
2. *Each significant decrease in error is preceded by a sharp drop in $p_{rest}$. This indicates that the planar subspace dynamics is responsible for the reduction in error.*

3. *Each significant decrease in error is preceded by a large oscillation in $p_m$.*




(a) Iterations (b) A zoom

Figure 14: Error, $p_m$, and $p_{rest}$ for LSD applied to the model Poisson problem.




(a) Iterations (b) A zoom

Figure 15: Error, $p_m$, and $p_{rest}$ for HLSD applied to the model Poisson problem.

# 8 Conclusions and further discussion

It is obvious that a faster gradient descent method must occasionally employ step sizes that are, strictly speaking, unstable [2]. We have further shown that such a method exhibits chaotic behavior, having in particular a positive Lyapunov exponent. Our numerical experiments clearly exhibit chaotic traits such as performance sensitivity to initial data.

The relaxed SD($\omega$) method was analyzed. For the under-relaxed, or damped case, this is a surprisingly fast, chaotic one-step method free of switches and random choices which can significantly outperform SD.

Considered over a wide range of problems (1) and initial guesses, the LSD and HLSD variants are as good as any other practical gradient descent method and better

than most.

The inverse step size $\gamma_k = 1/\alpha_k$ using any of the methods SD, LSD and HLSD can be written as

$$\gamma_k = \frac{\mathbf{q}_k^T A \mathbf{q}_k}{\mathbf{q}_k^T \mathbf{q}_k}, \tag{34}$$

for appropriate $\mathbf{q}_k$ defined at the $k$th iteration in terms of the residuals $\mathbf{r}_k$ and $\mathbf{r}_{k-1}$. Further assuming that $A$ is diagonal and defining for each iteration the probabilities

$$p_i = (q_i)^2/\|\mathbf{q}\|^2, \quad i = 1, \ldots, m,$$

we have

$$\gamma_k = \sum_{i=1}^{m} \lambda_i p_i.$$

Starting the iterative process with roughly equal probabilities $p_i$, the terms $\lambda_i p_i$ corresponding to large eigenvalues dominate the expression for $\gamma_k$, and hence $\alpha_k$ is small, being proportional to $\lambda_1^{-1}$. The resulting iteration is a *smoother*, reducing the large-eigenvalue residuals far more effectively than the small-eigenvalue ones [29, 2]. The difference between SD and the faster methods is that in SD, $\lambda_1 p_1$ remains dominant for all iterations $k$, hence the method is just a smoother and its efficiency is comparable to one with the best uniform step size. With LSD or HLSD, on the other hand, the large-eigenvalue probabilities reduce further so that eventually other $\lambda_i p_i$ dominate, yielding a smaller $\gamma$ hence a larger step size $\alpha = \alpha_k$. An iteration with such a large step size is not a smoother in itself, but it is more effective in reducing other, smaller-eigenvalue residuals [2]. Its instabilities increase the large-eigenvalue probabilities which then return to domination, and so on. The analysis and experimentation offered in this article further explain and expand on this broad-brush description.

A gradient descent iteration can be written as

$$\mathbf{r}_{k-1} = c(A - \gamma_{k-1} I)^{-1} \mathbf{r}_k,$$

where $c$ is an irrelevant proportionality constant. This is an inverse power iteration for $A$ using the shift $\gamma_{k-1}$ (e.g., [16]). Therefore, $\mathbf{r}_{k-1}$ better approximates the eigenvector associated with the eigenvalue of $A$ that is closest to $\gamma_{k-1}$ than $\mathbf{r}_k$ does. Now, for LSD $\mathbf{q}_k = \mathbf{r}_{k-1}$, hence $\gamma_k$ is closer than $\gamma_{k-1}$ to the eigenvalue that $\gamma_{k-1}$ approximates. The same effect occurs for HLSD every second iteration. The ensuing iteration with both methods is particularly effective in decreasing the corresponding residual component. This effect is strongest for the planar case $m = 2$, where choosing $\gamma_0 = \lambda_1$ and $\gamma_1 = \lambda_2$ would yield exact convergence in two iterations. Sections 6 and 7 further elaborate upon, analyze and numerically demonstrate these issues, clarifying why amongst the chaotic methods the two-step ones are faster. See also [14].

For any fixed number of iterations $n$, the optimal gradient descent method in the sense of minimizing $f(\mathbf{x}_n)$ is CG. This is because the CG and gradient descent iterates are in the same Krylov subspace, over which CG minimizes $f$. With CG, denoting $\mathbf{r}_n = p_n(A)\mathbf{r}_0$ for an appropriate polynomial $p_n$ of degree $n$ satisfying $p(0) = 1$, we can further write

$$p_n(A) = (I - \alpha_n^{CG} A) \cdots (I - \alpha_1^{CG} A),$$

where $\alpha_1^{CG}, \ldots, \alpha_n^{CG}$ are the (positive) roots of $p_n$. Thus we obtain CG as a gradient descent method using these $\alpha_k^{CG}$ as step sizes.

Of course, these optimal step sizes are not known without carrying out the CG method, thus solving the problem at the preprocessing stage. Moreover, the expression for $p_n(A)$ does not imply any particular ordering of the step sizes. Hence a method that attempts to find the step sizes $\alpha_k^{CG}$ approximately one at a time would not necessarily produce a monotone step size sequence nor a monotonically decreasing sequence of residual norms. It is also possible to obtain a faster gradient descent method by restarting CG every few iterations. Such a method could be strictly inferior to CG, though. Potentially related methods may be designed by using a look-ahead approach, cf. [5]. We leave further thoughts on this for future work.

Finally, let us emphasize that although we have provided analytical as well as numerical indications regarding the performance of our favorite faster gradient descent methods, a firm hold on the rate of convergence of these methods, such as is available for CG, remains elusive.

**Acknowledgment**

The second author thanks IMPA, Rio de Janeiro, for support and hospitality during the first few months of 2010 when this work was completed.

# 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*, 11:1–16, 1959.

[2] U. Ascher, K. van den Doel, H. Huang, and B. Svaiter. Gradient descent and fast artificial time integration. *M2AN*, 43:689–708, 2009.

[3] U. Ascher, H. Huang, and K. van den Doel. Artificial time integration. *BIT*, 47:3–25, 2007.

[4] J. Barzilai and J. Borwein. Two point step size gradient methods. *IMA J. Num. Anal.*, 8:141–148, 1988.

[5] A. Bhaya, P.-A. Bliman, and F. Pazos. Control-theoretic design of iterative methods for symmetric linear systems of equations. In *48th IEEE Conf. on Decision and Control*, pages 2347–2380. IEEE, 2009.

[6] Y. Dai and R. Fletcher. On the asymptotic behaviour of some new gradient methods. *Math. Programming*, 103:541–559, 2005.

[7] Y. Dai and R. Fletcher. Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming. *Numerische. Math.*, 100:21–47, 2005.

[8] Y. Dai, W. Hager, K. Schittkowsky, and H. Zhang. A cyclic Barzilai-Borwein method for unconstrained optimization. *IMA J. Num. Anal.*, 26:604–627, 2006.

[9] Y. Dai and Y. Yuan. Alternate minimization gradient method. *IMA Journal of Numerical Analysis*, 23:377–393, 2003.

[10] Y. H. Dai and L. Z. Liao. R-linear convergence of the barzilai and borwein gradient method. *IMA Numer. Anal.*, 22:1–10, 2002.

[11] M. Figueiredo, R. Nowak, and S. Wright. Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems. *IEEE J. Special Topics on Signal Processing*, 1:586–598, 2007.

[12] R. Fletcher. On the barzilai-borwein method. In *Optimization and Control with Applications, Eds. L. Qi, K. Teo and X. Yang*, volume 96, pages 235–256. Kluwer Series in Applied Optimization, 2005.

[13] A. Friedlander, J. Martinez, B. Molina, and M. Raydan. Gradient method with retard and generalizations. *SIAM J. Num. Anal.*, 36:275–289, 1999.

[14] W. Glunt, T. L. Hayden, and M. Raydan. Molecular conformations from distance matrices. *J. Comput. Chem.*, 14:114–120, 1993.

[15] G. Golub and Q. Ye. Inexact preconditioned conjugate gradient method with inner-outer iteration. *SIAM J. Scient. Comp.*, 21:1305–1320, 2000.

[16] G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins University Press, 1988.

[17] P. Goupillaud, A. Grossmann, and H. Morlet. Cycle-octave and related transforms in seismic signal analysis. *Geoexploration*, 23:85–102, 1984.

[18] E. Haber and U. Ascher. Preconditioned all-at-one methods for large, sparse parameter estimation problems. *Inverse Problems*, 17:1847–1864, 2001.

[19] H. Huang and U. Ascher. Faster gradient descent and the efficient recovery of images. *Math. Programming*, 2011. to appear.

[20] T. Y. Li and J. A. Yorke. Period three implies chaos. *Amer. Math. Monthly*, 82:985–992, 1975.

[21] A. M. Lyapunov. *The general problem of the stability of motion*. Tayor and Francis, 1992.

[22] J. Nagy and K. Palmer. Steepest descent, CG and iterative regularization of ill-posed problems. *BIT*, 43:1003–1017, 2003.

[23] J. Nocedal, A. Sartenaer, and C. Zhu. On the behavior of the gradient norm in the steepest descent method. *Comp. Optimization Applic.*, 22:5–35, 2002.

[24] J. Nocedal and S. Wright. *Numerical Optimization*. New York: Springer, 1999.

[25] L. Pronzato, H. Wynn, and A. Zhigljavsky. *Dynamical Search: Applications of Dynamical Systems in Search and Optimization*. Chapman & Hall/CRC, Boca Raton, 2000.

[26] M. Raydan and B. Svaiter. Relaxed steepest descent and Cauchy-Barzilai-Borwein method. *Comp. Optimization Applic.*, 21:155–167, 2002.

[27] I. Shimada and T. Nagashima. A Numerical Approach to Ergodic Problems of Disipative Dynamical Systems. *Progress of Theoretical Physics*, 61(6):1605–1616, 1979.

[28] A. M. Stuart and A. R. Humphries. *Dynamical systems and numerical analysis*. Cambridge University Press, Cambridge, England, 1996.

[29] U. Trottenberg, C. Oosterlee, and A. Schuller. *Multigrid*. Academic Press, 2001.

[30] C. Vogel. *Computational methods for inverse problem*. SIAM, Philadelphia, 2002.