BIT Numerical Mathematics (2006)46:000-000
DOI:10.1007/s10543-000-0000-x

Artificial time integration

U. Ascher [*] H. Huang[†] K. van den Doel[‡]

September 5, 2006

Abstract.

Many recent algorithmic approaches involve the construction of a differential equation model for computational purposes, typically by introducing an *artificial time* variable. The actual computational model involves a discretization of the now time-dependent differential system, usually employing forward Euler. The resulting dynamics of such an algorithm is then a discrete dynamics, and it is expected to be "close enough" to the dynamics of the continuous system (which is typically easier to analyze) provided that small – hence many – time steps, or iterations, are taken. Indeed, recent papers in inverse problems and image processing routinely report results requiring thousands of iterations to converge. This makes one wonder if and how the computational modeling process can be improved to better reflect the actual properties sought.

In this article we elaborate on several problem instances that illustrate the above observations. Algorithms may often lend themselves to a dual interpretation, in terms of a simply discretized differential equation with artificial time and in terms of a simple optimization algorithm; such a dual interpretation can be advantageous. We show how a broader computational modeling approach may possibly lead to algorithms with improved efficiency.

1 Introduction

The idea of adding an artificial time variable to a differential problem in order to obtain a solution for the latter upon discretizing the former is probably as old as modern computing itself. Indeed, steady state solutions are often the objective when simulating time-dependent PDE systems. A steady state solution, however, obeys a reduced system without a time variable, hence typical time-following algorithms for such problems involve an "unnecessary", or even "artificial", time variable. In fact, integration to steady state is a particular continuation method, as further elaborated in Section 2.2.

Sincovec & Madsen [70] described as early as 1975 a general-purpose package for solving PDEs, where the solution of a nonlinear elliptic PDE system is obtained by embedding it in a time-dependent one in an obvious fashion, followed by approximately solving the latter. However, doubts regarding the suitability (mainly for reasons of efficiency) of such a general recipe also arose during that era: it is not difficult to construct examples where this embedding inspires a very long, expensive time integration to achieve a decent approximation for the steady state solution.

In the early 1980's, Achi Brandt unsettled many a numerical analyst by insisting that "numerical algorithms must not stall": if there is no slow mechanism in the physical problem being approximated then there should not be one in the numerical algorithm either. One instance on his list of algorithmic approaches that may be unnecessarily slow has been the time embedding for steady state problems,[1] although most attention was focussed on algorithms competing more directly with multigrid methods

[*] Department of Computer Science, University of British Columbia, Vancouver, BC, V6T 1Z4, Canada. (ascher@cs.ubc.ca). This research was supported in part under NSERC Discovery Grant 84306.

[†] Institute of Applied Mathematics, University of British Columbia, Vancouver, BC, V6T 1Z2, Canada (hhzhiyan@math.ubc.ca).

[‡] Department of Computer Science, University of British Columbia, Vancouver, BC, V6T 1Z4, Canada. (kvdoel@cs.ubc.ca).

[1] Private communication.

[15]. If this is taken too simplistically then one may wonder if the "ban on stalling" is always essential, because it does not take into account human development time, especially for complex algorithms such as those in [77]. However, we prefer to understand such statements in a broader context, namely, that by insisting on an efficient algorithm one also learns more about the problem being solved, including aspects that the source of "numerical stalling" helps in hiding. Indeed, such an example occurred in our own work, when we came across an iterative method used in practice for Maxwell's equations in low frequency [58] which we managed to speed up by several orders of magnitude upon taking into account the existence of a casimir (null-space) in the leading term of the differential operator [42].

Many successful approaches that involve artificial time embedding have been reported in recent years, including some of the most innovative methods for scientific computation involving level sets (see Sethian [69], Osher & Fedkiw [62] and Burger & Osher [17]), PDEs in image preocessing and scale spaces (see, e.g., Weickert [81] and Scherzer [68]). Often an artificial physical process is invented in such context as part of the mathematical modeling effort. However, several of the resulting algorithms, although certainly not all, also tend to be unreasonably slow, as noted e.g. in Vogel [79] and elsewhere.

Therefore, we embark upon a more careful examination of the issues at hand. Since any method invoking an artificial time-dependent differential system invariably involves a time discretization, our purpose is to ask whether the continuous computational model that has been so discretized is *practically useful*, with Brandt's admonition in mind.

The paper is organized as follows. In Section 2 we briefly consider various situations where artificial time, and more generally continuation embeddings, have been proposed and discussed in the literature. The very purpose of such embeddings, however, calls for coarse discretization in the artificial time or continuation parameter, and this may in turn necessitate a discrete rather than continuous dynamics analysis.

Sections 3 and 4 both consider inverse problems involving recovering a distributed parameter function in several space dimensions. In Section 3 we consider time embedding as a regularization method and compare it to the Tikhonov regularization approach. The regularization effect is obtained by integrating only up to a finite time, a welcome relief on practical grounds as well. Both continuous and discrete variants are obtained for a simple case, the discrete version corresponding also to a steepest descent regularization. In Section 4 we see similar issues arising in the context of level set regularization for shape optimization.

In Section 5 we consider the problem of denoising, mainly of surface meshes in 3D. Anisotropic diffusion methods have provided high quality reconstructions for these problems, but they are not mandatory, nor are they always the fastest or best methods. We demonstrate this and urge keeping an open mind. A short conclusions section seals the paper.

2 From discrete to continuous dynamics

2.1 Iterative methods and forward Euler

Nonlinear algebraic systems, as well as many large linear ones, are typically solved by iterative methods. A memory-less iterative method can be written as

$$m_{n+1} = m_n + \tau_n g(m_n), \quad n = 0, 1, 2, \ldots, \tag{2.1}$$

where m_n is the solution at the nth iteration, m_0 a starting point, and τ_n is the step size. This can be viewed as the forward Euler discretization of the initial value problem

$$\frac{dm}{dt} = g(m), \quad m(0) = m_0, \tag{2.2}$$

and thus a new initial value differential problem has been born. See [39, 1, 12, 71, 47, 37, 55], as well as the review paper [25] and the feedback control interpretation in [10, 11].

An example is the dynamics of Newton's method. Starting from a given system of nonlinear equations

$$f(m) = 0, \tag{2.3}$$

with the Jacobian matrix $J = \frac{\partial f}{\partial m}$, the *damped Newton method* reads like (2.1) with

$$g(m) = -J(m)^{-1} f(m), \tag{2.4}$$

$0 < \tau \leq 1$. The corresponding differential system (2.2) and its dynamics have been studied by several of the above references, but we are not aware of any practically useful result pertaining to Newton's method or its many variants that has resulted from this approach. Indeed, the step size $\tau = 1$ has a special meaning in the discrete dynamical system (2.1), (2.4), producing a quadratic rate of convergence, hence this is the step size to take under some general conditions that amount to being close enough to steady state under certain regularity assumptions. Of course, *given* a time step for a forward Euler discretization of (2.2) one can easily re-scale the ODE to make $\tau = 1$, but the significance of this value for the rapid convergence to steady state is not depicted by the continuous dynamical system intent on sending τ to 0.

2.2 Continuation methods

A general approach, of which artificial time is a special case, is that of *continuation.* Here a given system (2.3) that is difficult to solve is embedded in a family of problems depending on a parameter s

$$\phi(m; s) = 0, \tag{2.5}$$

where solving (2.5) for $s = 0$ has the same effect as solving the given (2.3), whereas solving (2.5) for $s = 1$, say, is easy and may be considered done. The solution of (2.5), $m(s)$, forms a *homotopy path* for $0 \leq s \leq 1$. One then solves a sequence of problems for values of s decreasing from 1 to 0, where for each such problem a good initial guess from previous steps is at hand. This idea, like all others reviewed in this section, has been around for a while: see Watson et al [80] for a general package, Nocedal & Wright [59] for a discussion in the context of optimization, and Ascher, Mattheij & Russell [5] for boundary value ODEs.

Differentiation with respect to s yields the *Davidenko* ODE system ([5], §8.3.2),

$$\frac{\partial \phi}{\partial s} + \frac{\partial \phi}{\partial m}\frac{dm}{ds} = 0. \tag{2.6}$$

Discretizing this then yields a way to advance forward in s. The ODE (2.2) can be obtained from (2.6), (2.5), as a special case. An instance of this is given in Section 4. As mentioned earlier, the embedding of an elliptic PDE as the steady state of a parabolic one has long been recognized to be a continuation method, too. Generally, some improved continuation processes in certain cases are possible using (2.6), yet overall no dramatic improvements appear to have been reported in the literature on continuation methods.

Here, as before, there is the ensuing discretization of the continuation variable s, and the relentless question also follows, namely, whether the continuous system (2.6) yields useful practical and/or intuition-enhancing results for an efficient discrete process. It is important to note in this context that *we are not interested in the reconstruction of the homotopy path, only in reaching its end.* This therefore may require a discrete rather than continuous analytical approach.

2.3 Interior point methods for linear programming

An instance highlighting the need for discrete analysis is provided by interior point methods for the famous linear programming (LP) problem; see e.g. [59]. We consider

$$\begin{aligned} \min_{\mathbf{x}} \ & \mathbf{c}^T\mathbf{x} \\ s.t. \ & A\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned} \tag{2.7}$$

where $\mathbf{x}$ is the unknown vector of length l; A, $\mathbf{c}$ and $\mathbf{b}$ are given; and A is a $k \times l$ matrix of rank $k < l$. An equivalent primal-dual formulation is given by

$$\begin{aligned} A\mathbf{x} &= \mathbf{b}, \\ A^T\mathbf{y} + \mathbf{s} &= \mathbf{c}, \\ \mathbf{x} \geq \mathbf{0}, &\quad \mathbf{s} \geq \mathbf{0}, \\ \mathbf{x}^T\mathbf{s} &= 0. \end{aligned} \tag{2.8}$$

Many efforts following the ground breaking work of Karmarkar [50] have yielded the conclusion that the most efficient of interior point methods for LP are infeasible primal-dual methods. These can be viewed as attempting to stay not too far from the center path, $\mathbf{x}(s), \mathbf{y}(s), \mathbf{s}(s)$, defined as the solution of the one-parameter family obtained from (2.8) by replacing the nonlinear complementarity condition with

$$\mathbf{x}^T\mathbf{s} = s,$$

for $s \geq 0$. Thus, one could think of the process as a continuation method along the center path towards $s = 0$.

However, a continuous analysis of the resulting continuation process does not automatically lead to theoretical results about the discrete algorithm, and the definitive book of Wright [82] takes a discrete approach instead to obtain realistic convergence bounds. Neither is it obvious how to get ahead in a practical sense with a time-embedding interpretation for, say, the Mehrotra predictor-corrector continuation method along the center path [56], a method that has become ubiquitous in practice (although the theory in [82] does not quite cover it either).

2.4 Geometric integration

Most advances in the area of numerical integration of ODEs over the past 20 years have been made in the context of geometric integration; see e.g. Hairer, Lubich & Wanner [44] or Leimkuhler & Reich [51]. Here it is recognized that the flow of a given time-dependent differential system may have important global properties, such as *invariants*, and that often the preservation of such properties under numerical discretization is more important than the point-wise recovery of the flow for a particular initial condition. In particular, if the exact solution of the ODE system satisfies such an invariant, how important is it to preserve this property exactly, or approximate it very well, while the numerical solution resulting from our discretization is far less accurate point-wise? In some instances, but not all, this is very important [6, 2]. For the ODE (2.6) an invariant is defined by the nonlinear equations (2.5) which is obviously of more importance, once approximations are introduced, than the ODE itself.

Another example is provided by isospectral flow [21]. The matrix ODE

$$\begin{aligned} \frac{dL}{dt} &= AL - LA, \\ L(0) &= L_0, \end{aligned} \tag{2.9}$$

where $A(L)$ is skew-symmetric yields a solution with the property that the eigenvalues of $L(t)$ are independent of t. Given a set of real numbers $\lambda_1, \ldots, \lambda_k$ and the task of constructing a symmetric

$k \times k$ matrix $\hat{L}$ with some particular properties (e.g. a Toplitz matrix) that has these numbers as eigenvalues, one can try to find a $k \times k$ skew-symmetric A and an initial L_0 which has these eigenvalues such that $\hat{L}$ is the fixed point of (2.9).[2] The ensuing numerical discretization of (2.9) must then be such that the isospectral property is preserved, which occurs if quadratic algebraic invariants are reproduced at the discrete level. This happens, in turn, if collocation at Gaussian points is utilized (see e.g. Chapter 4 of [6]), regardless of the step size. Solving the ODE accurately is of less importance.

The above comments illustrate the point that, simply constructing a dynamical system as part of the computational modeling process and relying on a straightforward discretization to be "close enough" so that one need not bother about the actual algorithm any more, could be short-sighted.

3 Distributed parameter estimation and the exponential filter function

Let us recall the inverse problem of recovering a distributed parameter model. In general, the data are viewed as a nonlinear function of the model,

$$b = F(m) + \epsilon,$$

where F is the forward modeling operator, m is the model to be recovered, b is the data, and ϵ is measurement noise. The data is assumed given in some spatial domain Ω. Classical theory yields that if the problem is ill-posed or the data are noisy then we must add a priori information and isolate noise effects. In a Tikhonov-type approach this regularization leads to the optimization problem

$$\min_m \frac{1}{2}\|F(m) - b\|^2 + \beta R(m), \tag{3.1}$$

where $R(m)$ is a regularization operator, to be discussed later, and $\beta \geq 0$ is a famous parameter. See e.g. Engl et al [35] or the historical Tikhonov & Arsenin [75]. Throughout this paper we use the least squares norm, unless otherwise specified.

The necessary conditions are

$$J^T(F(m) - b) + \beta R'(m) = 0, \tag{3.2}$$

where R' is the gradient of R with respect to m and

$$J(m) = \frac{\partial F}{\partial m}$$

is the *sensitivity matrix*.

As in Section 2.1 we can view (3.2) as the steady-state equations of a time-dependent differential equation [4, 78],

$$\begin{aligned} M(m)\frac{\partial m}{\partial t} &= -[J^T(F(m) - b) + \beta R'(m)], \\ m(0) &= m_0, \end{aligned} \tag{3.3}$$

where $t \geq 0$ is the artificial time variable and the preconditioner $M = M(m(t))$ is symmetric positive definite. Below we consider the special case of (3.3) where $M = I$ and $F(m) = Jm$, with J a constant matrix.

[2] A. Iserles, private communication.

The continuous time system

The wrinkle here, as compared to Sections 1 and 2, is that typically one no longer intends to carry out the integration to steady state, not even in principle. (An interesting exception is [54, 62].) Indeed, there are really two regularization processes here, one caused by the Tikhonov term $\beta R(m)$, the other caused by integrating the ODE (3.3) to a *finite time*.

To see the latter, note that the solution of (3.3) with $m(0) = 0$ and $\beta = 0$ is

$$m(t) = (I - \exp(-tJ^TJ))J^\dagger b. \tag{3.4}$$

For notational simplicity assume further that J is nonsingular and consider its SVD,

$$J = U(diag\{s_i\})V^T. \tag{3.5}$$

(See e.g. the introductory chapter in [79].) The exact solution without noise is

$$m = J^{-1}b = V diag\{s_i^{-1}\}U^Tb.$$

The effect of regularization is to modify s_i^{-1} for those singular values which are very small, replacing s_i^{-1} by

$$\omega(s_i^2)s_i^{-1}.$$

In (3.4) we get

$$m(t) = V diag\{(1 - e^{-ts_i^2})s_i^{-1}\}U^Tb,$$

hence

$$\omega(s) = 1 - e^{-ts}. \tag{3.6}$$

Thus, the effect of small singular values gets dampened while large ones remain almost intact for an appropriate *finite* value of t.

For the more general case where $\beta \geq 0$, assume further that $R(m) = \frac{1}{2}\|m\|^2$. Then (3.3) reads

$$\frac{\partial m}{\partial t} = -[J^TJ + \beta I]m + J^Tb. \tag{3.7}$$

It is easy to see that

$$\omega(s) = \frac{1 - e^{-t(s+\beta)}}{s + \beta}s. \tag{3.8}$$

For the special case $t \to \infty$ we have the Tikhonov filter

$$\omega(s) = \frac{s}{s + \beta}, \tag{3.9}$$

whereas $\beta = 0$ yields the exponential filter (3.6).

We have compared the two filter functions appearing in (3.6) and (3.9), see also [20]. Of course they both tend to 0 as $s \to 0$ and to 1 as $s \to \infty$. But even for mid-range values of s there is a similarity in the general shape. It appears that the choice $t \approx \frac{1}{2\beta}$ makes these filters particularly close under certain conditions, and that the exponential filter switches from 0 to 1 more sharply; see Fig. 3.1.

Calvetti & Reichel [19] give further reasons for preferring (3.6) in the linear case. They approximate the exponential of a matrix in (3.4) by a Lanczos process, see also [18]. In fact, they never refer to an ODE giving rise to this filter function.

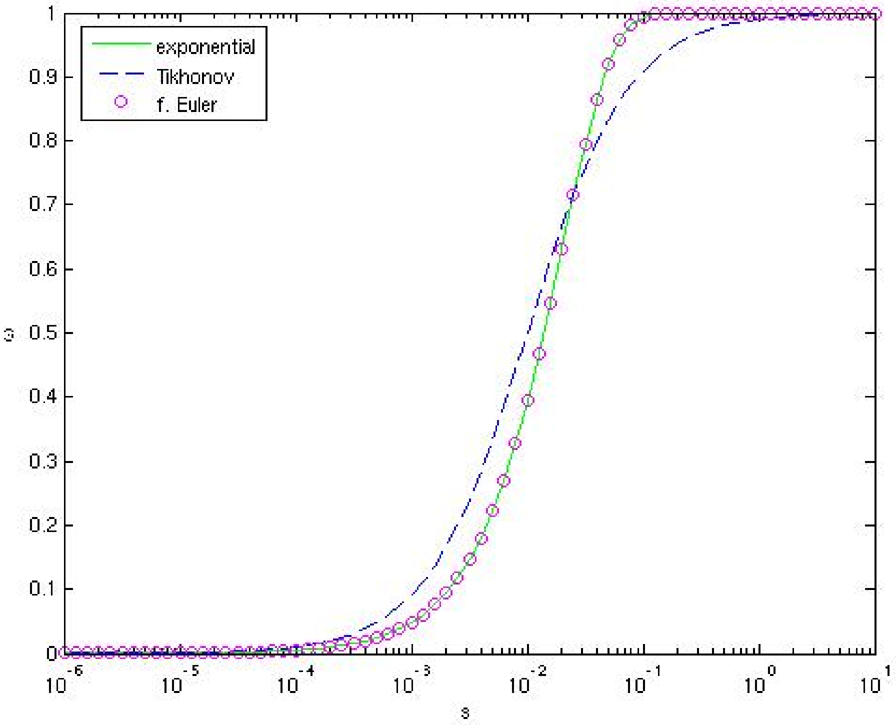


Figure 3.1: Exponential filter (3.6), ODE forward Euler discretization (3.13), and Tikhonov filter (3.9) for $t\beta = 1/2$, with $\tau = .1, \beta = .01$.

The discrete time system

Next, consider the usual discretization of the ODE (3.7). Applying forward Euler with a time step τ we have at the nth time step

$$\begin{aligned} m_{n+1} &= m_n - \tau[J^T J + \beta]m_n + \tau J^T b \\ &= G m_n + \tau J^T b, \quad G = I - \tau(J^T J + \beta I). \end{aligned} \tag{3.10}$$

Note that ODE stability holds if

$$0 < \tau \leq 1/\|J^T J + \beta I\|^2. \tag{3.11}$$

Moreover, the step size τ need not be constant. Further below in this section we discuss its choice.

Due to the simplicity of F and R we can unravel the recursion in (3.10) in the usual manner. Moreover, the SVD of G is inherited from that of J. Assuming $m_0 \equiv 0$ we obtain

$$\begin{aligned} m_{n+1} &= \sum_{j=0}^{n} G^j \tau J^T b \\ &= \tau V \left(diag\{\sum_{j=0}^{n} (1 - \tau(s_i^2 + \beta))^j s_i\} \right) U^T b. \end{aligned}$$

Since

$$\sum_{j=0}^{n}(1-\tau(s^2+\beta))^j = \frac{1-(1-\tau(s^2+\beta))^n}{\tau(s^2+\beta)},$$

we have

$$\omega(s) = \frac{1-(1-\tau(s+\beta))^n}{s+\beta}s. \tag{3.12}$$

Setting $\beta = 0$ in (3.12) yields the Landweber iteration [35]. For a finite time there is a regularization effect since

$$\omega(s) = 1-(1-\tau s)^n, \tag{3.13}$$

see Fig. 3.1 again. As $n \to \infty$ we get $\omega \to 1$, so there is no regularization. For $n = 1$ we get heavy regularization (damping) with $\omega(s) = \tau s$. The smaller s is, the stronger the damping, which is good. The question becomes that of choosing n for the right amount of regularization. The usual method is to stop the integration when the iteration starts to diverge. See [35] and many other references for this.

Letting $t = n\tau$ be fixed as τ decreases and n increases in (3.12) we obtain the continuous system formulae discussed earlier in this section. However, we obviously do not wish to take τ very small in practice. So, even though the limit process yields a viable regularization through (3.6) or (3.8), and the discrete process yields a viable regularization through (3.13) or (3.12), respectively, it is not clear a priori that the latter should be best considered a discretization of the former! Typically, the continuous model is easier to analyze, and such an analysis can be used to motivate a coarse discretization. Such a motivation is gratified in case that the actual method turns out to work well in practice, as in Fig. 3.1. However, the continuous analysis is neither strictly necessary for practice nor sufficient for a complete theory.

Steepest descent regularization

The above considerations highlight not only the potential advantage in using artificial time-stepping but also the limitations inherent in taking this too inflexibly. On the other hand, if we consider (3.10) as a *steepest descent* step for minimizing the relevant special case of (3.1) [60, 37, 10] then, because the objective function is quadratic and convex, the step size $\tau = \tau_n$ can be determined precisely based on an exact line search. This exact line search aims at finding m_{n+1} of (3.10) which is closest to the steady state of (3.7):

$$\tau = \frac{[J^T(Jm-b)+\beta m]^T[J^T(Jm-b)+\beta m]}{[J^T(Jm-b)+\beta m]^T(J^TJ+\beta I)[J^T(Jm-b)+\beta m]}. \tag{3.14}$$

This formula and its extensions for the genreal case are used routinely by clever practitioners utilizing the Landweber method [3]. They set $\beta = 0$ and stop the iteration when it no longer improves much.

Recall that in the present context, for the case $\beta = 0$ the problem (3.1) is singular or at best highly ill-conditioned. A corollary of the above analysis is therefore that applying a finite number of steepest descent iterations has a regularization effect! See also [57] for more evidence on the steepest descent revival.

[3] H. Engl, private communication.

Backward Euler

Let us return to the more general differential system (3.3). Many algorithms in recent literature can be viewed as a forward Euler discretization of this equation with appropriate choices of M and R. A few researchers, however, have preferred to consider a backward Euler discretization for $M = I$. This can lead to much fewer iterations for achieving a satisfactory solution in case that (3.3) is very stiff, which in turn may result if (3.2) involves a highly nonuniform spatial discretization (e.g. [30, 52]).

Of course a very large backward Euler step for (3.3) amounts essentially to solving the steady state, Tikhonov-type equations (3.2).

Assume further that

$$R'(m) = \hat{R} \cdot m,$$

where $\hat{R}(m)$ is a nonsingular matrix, and let $m_0 = 0$. Then (3.2) can be written as one backward Euler step,

$$\frac{m - m_0}{\tau} = -\lambda \hat{R}^{-1} J^T (F(m) - b),$$

for $\tau = 1$, $\lambda = \beta^{-1}$. This can be replaced by N backward Euler steps of size $\tau = 1/N$ each,

$$m_n - m_{n-1} = -\frac{\lambda}{N} \hat{R}^{-1} J^T (F(m_n) - b), \quad n = 1, \ldots, N.$$

The effect is that of replacing (3.2) by the iteration

$$J^T(m_n)[F(m_n) - b] + [\beta N]\hat{R}(m_n)[m_n - m_{n-1}] = 0, \quad n = 1, \ldots, N. \tag{3.15}$$

The effective size of the regularization parameter is thus increased by a factor of N, which may make solving the system for each n easier, but at a price.[4] The precise total cost depends on what further Krylov-space approximations are utilized for the solution of these systems.

Some such ideas are interesting and encouraging, although the gains seem modest. In any case the obtained discrete dynamics is clearly different from the continuous dynamics of (3.3). It can be argued that a similar situation holds whenever one approximates a stiff ODE system: only the obtained solution is close to the exact one, while stiff modes are not well-approximated. The difference, however, is that in a typical stiff ODE case the continuous system is *given*, whereas here it is our own invention. If we are not intent on following the dynamics of the continuous system (3.3) then what is it there for in the first place?

4 Level sets for shape optimization

Let us continue with inverse problems involving the recovery of a 2D or 3D distributed parameter function. We use the notation in (3.3), (3.1) and (3.2) introduced in the previous section. The forward modeling operator $F(m)$ now involves the inversion of an elliptic PDE system with the resulting field sampled e.g. at the domain's boundary to match the given data. This makes the inverse problem highly ill-posed. Applications include linear potential problems [45], DC resistivity [72], magnetotelluric inversion [63], diffraction tomography [31], oil reservoir simulation [34], aquifer calibration [40], electrical impedance tomography (EIT) [13, 24, 14] and low frequency electromagnetic data inversion [43].

In addition to the forward operator it is assumed that the solution $m(\mathbf{x})$ can take only one of two values at each $\mathbf{x}$. This then becomes a problem of shape optimization. For such problems, when $m(\mathbf{x})$ is known to contain discontinuities it is generally impossible to obtain good, *reliable* recovery results without explicitly imposing the piecewise constant property of the solution (cf. [3, 4]). An approach considered by many for this purpose is to use level sets, and we proceed to investigate this further in the present context, following [78].

[4] E. Haber, private communication.

A level set approach

Thus, in place of (3.3) we consider

$$M(\psi)\frac{\partial \psi}{\partial t} = -[\chi' J^T(F(m(\psi)) - b) + \beta R'(\psi)], \qquad \psi(0) = \psi_0. \tag{4.1}$$

The function $\psi(\mathbf{x};t)$ is a level set function, and it is spatially smoother than m. The two are related by

$$m = \chi(\psi), \tag{4.2a}$$

where the function χ is a grid-smoothing of the characteristic function based on the values that m may take. We choose e.g.

$$\chi(s) = \frac{m_1 - m_2}{2}\tanh(s/h) + \frac{m_1 + m_2}{2}, \tag{4.2b}$$

where h is a given grid spacing. (Strictly speaking, ψ is one derivative smoother than m in the limit $h \to 0$.) Note that

$$\lim_{h\to 0}\tanh(s/h) = \begin{cases} -1 & s < 0 \\ 1 & s > 0 \end{cases}.$$

Thus, for each t sharpening happens across the 0- level set of $\psi(\mathbf{x};t)$; see Fig. 4.1.

Variants of this have been considered in [33, 38, 23, 26] and elsewhere. We caution, however, that in several such papers many iterations, or time steps, are required before convergence is deemed to have been achieved, often by unclear criteria.

Instead of discretizing (3.3) we may consider the corresponding generalization of (3.1), (3.2), namely

$$\min_{\psi} \frac{1}{2}\|F(m(\psi)) - b\|^2 + \beta R(\psi), \tag{4.3a}$$

$$\chi' J^T(F(m(\psi)) - b) + \beta R'(\psi) = 0, \tag{4.3b}$$

where m is the characteristic function of ψ as before. In short, we are headed directly towards the steady state of (4.1), although we may stop well short of getting there. This approach was initiated by Santosa [66] and subsequently refined and used e.g. by Haber [41] and Leitao & Scherzer [53]. Applying *damped Gauss-Newton* it is often possible to obtain in two dozen iterations results of comparable quality to those obtained using orders of magnitude more work to follow the dynamical system (4.1) to its steady state.

A *dynamic regularization* method has proved to work particularly well in practice. Starting from an initial guess ψ_0, the iterations read

$$(\chi' J^T J \chi' + \beta_0 X)\delta\psi = -\chi' J^T(F(m(\psi_n)) - b), \tag{4.4a}$$

$$\psi_{n+1} = \psi_n + \tau\delta\psi, \tag{4.4b}$$

for $n = 0, 1, 2, \ldots$, where $\beta_0 > 0$ is a small constant and $0 < \tau \le 1$ is a step size. The matrices in (4.4a) are all evaluated at ψ_n. We choose $X = R''$, the Hessian of a regularization operator $R(\psi)$. For the latter, the choice

$$R(\psi) = \frac{1}{2}\left[\int_\Omega (1 - |\nabla\psi|^2)d\mathbf{x}\right]^2 \tag{4.5a}$$

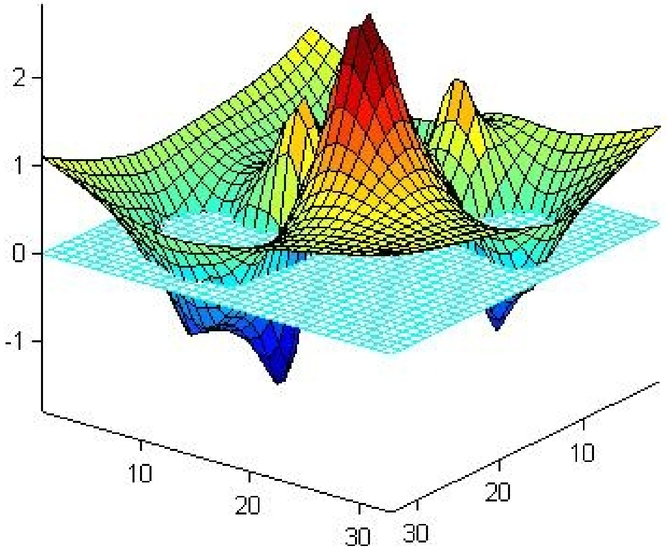


Figure 4.1: Level set function.

(or, more precisely, a discretization of this on the grid of m) leads to

$$X = 2(\psi^T L \psi - |\Omega|)\ L + 4(L\psi)(L\psi)^T, \tag{4.5b}$$

i.e. X is a rank-1 modification of a solution-dependent multiple of L, which in turn is a positive definite matrix based on the standard 5-point discrete Laplacian. The simpler choice

$$R(\psi) = \frac{1}{2}\int_\Omega |\nabla \psi|^2 d\mathbf{x}, \tag{4.6}$$

which leads to $X = L$, occasionally performs worse in practice.

The matrix $\chi' J^T J \chi'$ can be further thresholded, reflecting the fact that it "lives" essentially only on and near the interface, and this leads to fast algorithms for carrying out the iteration. We refer to [78] for this and other implementation considerations, as well as numerical experiments that demonstrate that this method works very well, often delivering good reconstructions in less than 10 iterations. Here we concentrate just on deriving the method (4.4) from different points of view.

1. The dynamic regularization iteration can be viewed as a forward Euler discretization of (4.1) with $\beta = 0$ and

   $$M = \chi' J^T J \chi' + \beta_0 X. \tag{4.7}$$

   Thus, the regularization chore has shifted from R to M in (4.1), see also [38]. Although the choice of time step τ is not obvious here, and the choice of M is not intuitive, this is an important interpretation, as some significant differences observed in [78] between dynamic regularization and the Tikhonov approach no longer seem major in the present setting.

2. Taking the optimization point of view, we set $\beta = 0$ in (4.3). The Gauss-Newton method then involves a singular matrix, so it is regularized Levenberg-Marquardt fashion, but with the twist that the regularization matrix is $X = R''$, not simply the identity.

   The step size τ is determined by a weak line search. However, the choice $X = R''$ is far from obvious in this interpretation. Burger [16] chose a different X. Interestingly, he evolved the level set geometrically, using integration of the Hamilton-Jacobi equations, but we and others have not found this practically necessary. This once again highlights the difference between simulating a *given* mathematical model, as distinct from using or constructing a mathematical model for the purpose of carrying out a given simulation task.

3. The above two interpretations reveal yet again that it may be easier to interpret a given algorithm as an instance of an artificial time or an optimization method than to use such an interpretation as an insight to derive the best variant of the algorithm. Here is a third interpretation, more specifically connecting dynamic regularization to the Tikhonov regularization (4.3b). We have seen an inverse relationship between β and the artificial time t already in Section 3. Let us therefore set $t = 1/\beta$ and differentiate (4.3b) with respect to t. After rearranging terms this gives

   $$(4.8) \qquad (\chi' J^T J \chi' + \beta R'') \frac{d\psi}{dt} = -\beta \chi' (F(m(\psi)) - b).$$

   Setting $\beta = \beta_0$ and applying forward Euler with time step τ/β yields the dynamic regularization iteration (4.4).

   The choice $X = R''$ is now more natural. Moreover, the two often conflicting roles of β in the Tikhonov setting, namely, ensuring a sufficiently nonsingular iteration matrix and balancing terms in (4.3a) depending on the noise level, are seen separated here! The step τ is still best determined by the weak line search described before, though.

In summary, different interpretations are each seen to shed some light on the same iteration process. They are all partially useful, yet none renders the others unnecessary.

5 Smoothing surface meshes in 3D

In the previous section we show for a highly ill-posed inverse problem that some advantage can be gained by considering the discrete dynamics that an optimization method yields. Next we consider the use of anisotropic diffusion and its alternatives in denoising algorithms.

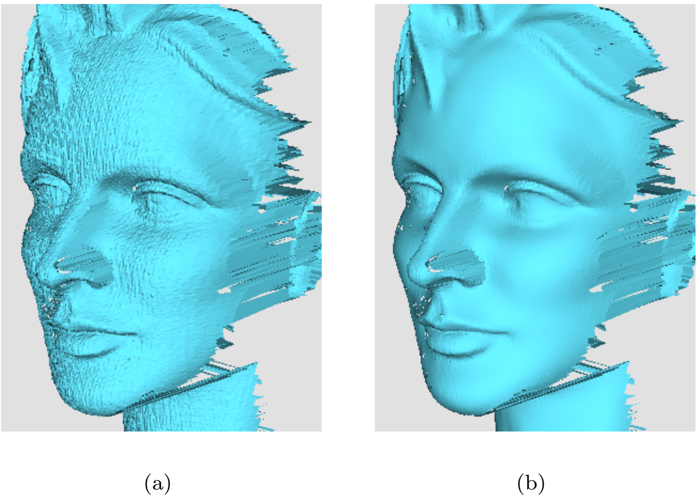
Use of PDEs for the computational modeling of image processing tasks has become ubiquitous in recent years; see Sapiro [67] as well as [79, 68, 81, 22]. Particular attention has been devoted to the simplest problem in its class, that of image denoising [64, 65, 9, 81, 73, 61]. Here numerical analysis is easier: in the notation of Section 3 we have $F(m) = m$, $J = I$, so the necessary conditions for a Tikhonov-type regularization, (3.2), become

$$(5.1) \qquad \lambda(m - b) + R'(m) = 0,$$

where $\lambda = 1/\beta \geq 0$. The limit problem of $\beta \to 0$ is thus well-conditioned, the Jacobian matrix is sparse if R'' is, and its eigenvalues are those of R'' translated by λ. As before, a gradient descent method for (3.1) followed by sending the step size τ to 0 yields the artificial time-dependent PDE

$$(5.2) \qquad \frac{\partial m}{\partial t} = R'(m) + \lambda(m - b),$$

and we may next let $\lambda = 0$ as well.



(a)

(b)

Figure 5.1: Fast denoising without losing significant features: (a) scanned Lady face model (41K Vertices) contains unknown noise; (b) smoothed model by (5.3)-(5.6) (3 iterations, 3.0 secs). All runs reported in this section were carried out on a laptop with a 2.0 GHz Pentium M CPU.

Consider further a typical regularization operator $R(m)$ which is a spatial discretization of

$$R(m) = \int_\Omega \rho(|\nabla m|).$$

Setting $\rho(s) = \frac{1}{2}s^2$ yields (ignoring spatial discretization for a moment) $R'(m) = -\nabla \cdot \nabla m$, so (5.2) is a diffusion equation, well-known to produce a smoothing effect in physical (and multigrid) applications. This, however, also smears intensity height discontinuities, or sharp features. Hence one selects an *edge-stopping function* ρ, obtaining an anisotropic diffusion process where diffusion, or smoothing, is applied only in directions that do not cross edges [64]. The total variation (TV) choice in (5.1), $\rho(s) = s$, yields $R'(m) = -\nabla \cdot \left(\frac{1}{|\nabla m|}\nabla m\right)$ [65, 62]. It is well known, though, that this choice must be modified when $|\nabla m| \to 0$ [4]. The TV choice is especially attractive for analysis because R is convex, however other alternatives also perform well in practice [67, 9]. More complex processes involving tensor anisotropic diffusion have been proposed and demonstrated as well [81].

Applications of such anisotropic diffusion operators in image processing are not limited to just simple denoising. Here, then, is a class of important problems that can be solved well by methods of the type under consideration in the present article. Remember, however, that the discussion in Section 3 about efficient discretization and simulation of (5.2) still applies. Here, in addition, a mildly stiff problem has been created, hence small time steps are required due to stability restrictions if an explicit discretization is applied in time. In practical situations there often exist cheaper alternatives to anisotropic diffusion for image processing tasks.

To highlight these points in a hopefully more interesting setting let us concentrate further on the related problem of denoising triangle meshes describing surface manifolds in 3D [29, 27, 7, 74, 8, 46, 28, 48]. Such meshes, consisting of vertices $\mathbf{x}_i$ and edges $\mathbf{e}_{i,k}$ in 3D, are ubiquitous in computer graphics and geometric design. Scanning and acquisition technology used for obtaining them often yields high frequency noise in the position of the vertices, so a mesh smoothing algorithm is required to rapidly remove noise while preserving real artifacts in the acquired data, see Fig. 5.1.

The problem is different from, and more complicated than, the image denoising problem described

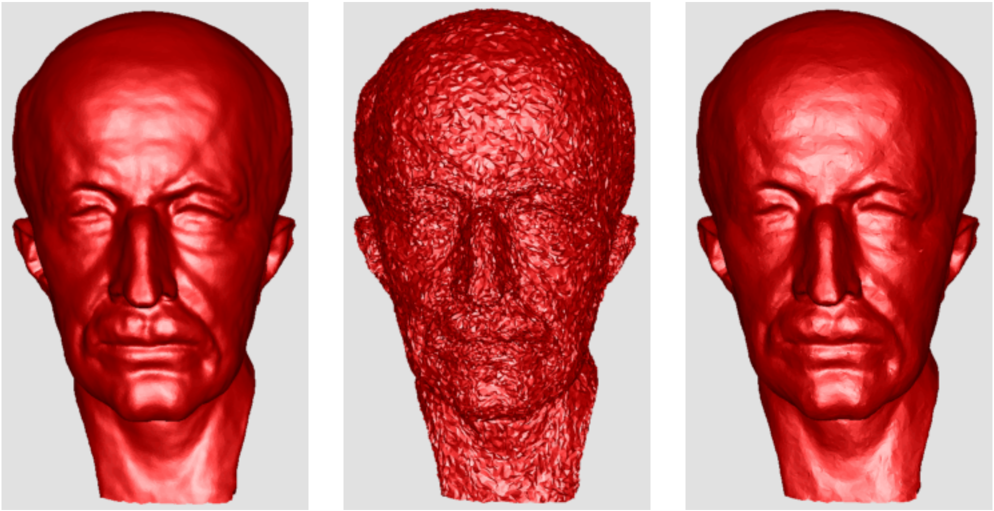


Figure 5.2: Rapidly remove heavy Gaussian noise from a corrupted model: (a) original Max Planck model (50K vertices); (b) model is corrupted by 20% Gaussian noise in normal direction; (c) smoothed by AL (4 iterations, 5.1 secs).

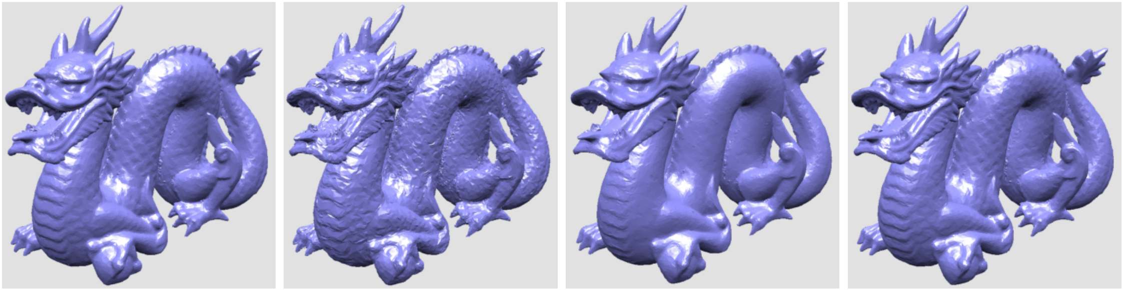
earlier. The noise is not only in intensity heights at given pixel locations: distinguishing at a given vertex $\mathbf{x}_i$ the normal direction $\mathbf{n}_i$ and the tangent plane to the manifold at that point, there is noise in both normal height and location. Moreover, nonuniform spatial meshes are common and there is a volume associated with the body described by the mesh that a careless denoising application may shrink. Nonetheless, typical algorithms often “lift up” techniques from image processing, albeit in a nontrivial way. Anisotropic diffusion has taken center stage here, too [29, 27, 28, 46]. The more recent references describe rather complex diffusion processes that yield impressive reconstructions, especially in the presence of sharp corners and absence of high scale texture. However, employing forward Euler yields unwanted stability restrictions here, too, and implicit schemes require the solution of large, nontrivial systems of equations at each time step, for which some Krylov-space method is employed [46, 28]. With meshes occasionally containing as many as 100,000 vertices and 200,000 faces, these methods can therefore become expensive. Are there cheaper alternatives?

The method of bilateral filtering [76, 36, 49] offers such a cheap alternative for some models. But in fact, even a simpler method will do, and we now specify one such, following [48]. First some notation is required. A manifold $\mathcal{M}$ is discretized by a triangular surface mesh S with its set of vertices $V(S) = \{\mathbf{x}_i, i = 1, \ldots, N\}$ and set of directed edges $E(S)$. If two distinct vertices $\mathbf{x}_i$ and $\mathbf{x}_k$ are linked by an edge $\mathbf{e}_{i,k} = \mathbf{x}_k - \mathbf{x}_i$ then we denote $k \in \mathcal{N}(i)$. The given, noisy data is denoted $\mathcal{M}_0 = \mathcal{M}(\mathbf{v}_1, \ldots, \mathbf{v}_N)$.

For each vertex $\mathbf{x}_i$ of the given data mesh we estimate the corresponding normal $\mathbf{n}_i$ as the mean of the adjacent face normals

$$\mathbf{n}_i = normalize \left(\sum_{k \in \mathcal{N}(i)} \frac{\mathbf{e}_{i,k} \times \mathbf{e}_{i,k+1}}{|\mathbf{e}_{i,k} \times \mathbf{e}_{i,k+1}|} \right). \tag{5.3}$$

In case that there is a lot of noise we use an additional mollification procedure [48]. These normals are then considered part of the input.



(a) (b) (c) (d)

Figure 5.3: Smoothing a dragon while retaining its scales: (a) original Dragon model (50K vertices); (b) corrupted model by about 10% Gaussian noise in both tangential and normal directions; (c) smoothed model by our AL variant with $\lambda = 0$ (3 iterations, 3.8 secs); (d) smoothed model by MSAL (3 iterations, K = 1/5, 3.8 secs).

Next, define the local height intensity $h_{i,k} = \mathbf{e}_{i,k}^T \mathbf{n}_i$, the projection of $\mathbf{e}_{i,k}$ on the normal $\mathbf{n}_i$ at the vertex. A corresponding discrete anisotropic Laplacian (AL) operator is then given by

$$\Delta \mathbf{x}_i = \frac{1}{C_i}\left(\sum_{k\in\mathcal{N}(i)} g(h_{i,k})h_{i,k}\right)\mathbf{n}_i, \tag{5.4}$$

where $C_i = \sum_{k\in\mathcal{N}(i)} g(h_{i,k})$.

Further, we employ a Gaussian filter in this context, which is more robust than TV and clearly reduces the influence of neighbors that contain large discontinuities in normal space:

$$g(h_{i,k}) = exp(-\frac{h_{i,k}^2}{2\sigma_i^2}), \tag{5.5a}$$

$$\sigma_i = 2\,\mathtt{mean}(|\mathbf{h}_i - \mathtt{mean}(\mathbf{h}_i)|), \tag{5.5b}$$

where $\mathbf{h}_i = \{h_{i,k}\}_{k\in\mathcal{N}(i)}$. Thus, the *robust scale* σ_i is automatically estimated by the mean absolute deviation scaling.

The iteration reads

$$\mathbf{x}_i \leftarrow \mathbf{x}_i + \Delta\mathbf{x}_i + \lambda_i(\mathbf{v}_i - \mathbf{x}_i), \quad i = 1, \dots, N, \tag{5.6}$$

using (5.4), (5.5). Three iterations (5.6), starting with $\mathbf{x}_i = \mathbf{v}_i$ and maintaining $\lambda_i \equiv 0$ throughout, are sufficient to produce the result displayed in Fig. 5.1. Another experiment, requiring four simple iterations, is depicted in Fig. 5.2.

The iteration (5.6) can be viewed as a forward Euler discretization with step size $\tau = 1$ of a PDE such as (5.2) on a manifold (see [32] for PDEs on manifolds). However, this view does not explain, or even provide intuition for, the fact that 3 or 4 such explicit "time steps" are sufficient!

Bilateral filtering, as well as our AL, are also known to deliver less than stellar reconstructions for models containing significant high scale features (see e.g. the first figure in [49]). We refer to this as *intrinsic texture*. Note, though, that the quality of AL can be improved without significantly modifying its price, as we show in [48]. This is done by using different data fidelity weights λ in a multiscale fashion (MSAL). The basic multiscale idea vaguely follows [73], but everything else is different here. Since the surface scale is local and the mesh is generally nonuniform, we choose λ_i at each such iteration depending on the spatial location i. We use (5.5b) divided by $\max \sigma_i$ so that $\lambda_i \leq 1, \forall i$. The importance of the entire function $\lambda(\mathbf{x})$ obtained this way is magnified in the next iteration step by damping out the anisotropic diffusion operator AL of (5.4), to recapture a higher frequency band. Thus, in the jth iteration we calculate λ and set

$$\mathbf{x}_i \leftarrow \mathbf{x}_i + K^j \Delta \mathbf{x}_i + \lambda_i(\mathbf{v}_i - \mathbf{x}_i), \quad i = 1, \ldots, N, \tag{5.7}$$

where $0 < K < 1$ is an input parameter. See Fig. 5.3 for the resulting effect and [48] for details.

The MSAL iteration (5.7) can be viewed, like other algorithms throughout this article, as a forward Euler discretization with unit time step of the ODE system

$$\frac{d\mathbf{x}}{dt} = K^t \Delta \mathbf{x} + \lambda(\mathbf{v} - \mathbf{x}), \quad t \geq 0. \tag{5.8}$$

From this ODE it is immediate that as $t \to \infty$ the original, noisy data is recovered. Our taking only 3 or 4 steps corresponds again, as in Section 3, to the smoothing effect of integrating to a finite time. However, again it is also clear that the slick continuous view does not really account for all of the essential behavior of this algorithm.

6 Conclusions

Artificial time embedding for inverse problems and for image processing problems has been found useful and beneficial by many researchers. Good intuition leading to theory as well as to practical algorithms has been gained, as noted in the introductory Section 1. Artificial time embedding has also proved a tool of convenience over several decades, allowing the extension of algorithms and software for solving a larger class of problems, albeit occasionally at the expense of reduced efficiency. Indeed, more general continuation methods have often provided the only ready made tool for solving tough nonlinear problems in a variety of applications. We have highlighted the positive features of this approach for several general problem instances.

However, it is also important to realize that the life of an apparatus depends on its utility. Inventing computational models that are then used primarily for analysis rather than for computation may prove to be an impediment for the goal of constructing useful, efficient computational algorithms for the actual given problem to be solved.

The discretization of a continuous artificial time system, which is what gets implemented, is the actual computational model. The dynamics of the *discrete* process is therefore crucial. If the dynamics of the continuous computational model is significantly different from that of an efficient discrete system, or if a close discretization of the continuous model leads to many costly iterations, then the continuous model may not be the right one to look at in the first place. A case in point is provided in Section 5.

REFERENCES

1. Y.I. Alber. Continuous processes of the newton type. *Differential Equations*, 7:1461–1471, 1971.
2. U. Ascher. DAEs that should not be solved. In *IMA Proceedings Dynamics of Algorithms*, volume 118, pages 55–68. Springer, 1999.
3. U. Ascher and E. Haber. Computational methods for large distributed parameter estimation problems with possible discontinuities. *Proc. Symp. Inverse Problems, Design & Optimization*, pages 201–208, 2004. M. Colaco, H. Orlande and G. Dulikravich (eds.).
4. U. Ascher, E. Haber, and H. Huang. On effective methods for implicit piecewise smooth surface recovery. *SIAM J. Scient. Comput.*, 28:339–358, 2006.

5. U. Ascher, R. Mattheij, and R. Russell. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*. SIAM, Philadelphia, 1995.

6. U. Ascher and L. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, Philadelphia, PA, 1998.

7. C. Bajaj and G. Xu. Adaptive surface fairing by geometric diffusion. In *Symp. Computer Aided Geometric Design*, pages 731–737. IEEE Computer Society, 2001.

8. C. Bajaj and G. Xu. Anisotropic diffusion on surfaces and functions on surfaces. *ACM Trans. Graphics (SIGGRAPH)*, 22(1):4–32, 2003.

9. D. Barash. A fundamental relationship between bilateral filtering, adaptive smoothing and the nonlinear diffusion equation. *IEEE Trans. PAMI*, 24:844–847, 2002.

10. A. Bhaya and E. Kaszkurewicz. Iterative methods as dynamical systems with feedback control. In *42nd IEEE Conf. on Decision and Control*, pages 2347–2380. IEEE, 2003.

11. A. Bhaya and E. Kaszkurewicz. *Control Perspectives on Numerical Algorithms and Matrix Problems*. SIAM, 2006.

12. P. Boggs and J.E. Dennis. A stability analysis for perturbed nonlinear analysis methods. *Math. Comp.*, 30:199–215, 1976.

13. L. Borcea, J. G. Berryman, and G. C. Papanicolaou. High-contrast impedance tomography. *Inverse Problems*, 12:835–858, 1996.

14. L. Borcea, G. Gray, and Y. Zhang. Variationally constrained numerical solution of electrical impedance tomography. *Inverse Problems*, 19:1159–1184, 2003.

15. A. Brandt. Multigrid techniques: 1984 Guide with applications to fluid Dynamics. The Weizmann Institute of Science, Rehovot, Israel, 1984.

16. M. Burger. Levenberg-marquardt level set methods for inverse obstacle problems. *Inverse problems*, 20:259–282, 2004.

17. M. Burger and S. J. Osher. A survey on level set methods for inverse problems and optimal design. *European J. Appl. Math.*, 16:263–301, 2005.

18. D. Calvetti, B. Lewis, and L. Reichel. Smooth or abrupt: a comparison of regularization methods. In *Advanced Signal Processing Algorithms, Architectures and Implementations VIII, ed. F.T. Luk*, volume 3461, pages 286–295. SPIE, 1998.

19. D. Calvetti and L. Reichel. Lanczos-based exponential filtering for discrete ill-posed problems. *Numer. Algorithms*, 29:45–65, 2002.

20. D. Calvetti, L. Reichel, and Q. Zhang. Iterative exponential filtering for large discrete ill-posed problems. *Numer. Math.*, 83:535–556, 1999.

21. M.P. Calvo, A. Iserles, and A. Zanna. Numerical solution of isospectral flows. *Math. Comp.*, 66:1461–1486, 1997.

22. T. Chan and J. Shen. *Image Processing and Analysis: Variational, PDE, Wavelet and Stochastic Methods*. SIAM, 2005.

23. T. Chan and X. Tai. Level set and total variation regularization for elliptic inverse problems with discontinuous coefficients. *J. Comp. Phys.*, 193:40–66, 2003.

24. M. Cheney, D. Isaacson, and J.C. Newell. Electrical impedance tomography. *SIAM Review*, 41:85–101, 1999.

25. M.T. Chu. On the continuous realization of iterative processes. *SIAM Review*, 30:375–387, 1988.

26. E. Chung, T. Chan, and X. Tai. Electrical impedance tomography using level set representations and total variation regularization. *J. Comp. Phys.*, 205:357–372, 2005.

27. U. Clarenz, U. Diewald, and M. Rumpf. Anisotropic geometric diffusion in surface processing. In *Proceedings IEEE Visualization*, pages 397–405, 2000.

28. U. Clarenz, U. Diewald, and M. Rumpf. Processing textured surfaces via anisotropic geometric diffusion. *IEEE Transactions on Image Processing*, 13(2):248–261, 2004.

29. M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Anisotropic feature–preserving denoising of height fields and bivariate data. *Graphics Interface*, pages 145–152, 2000.

30. M. Desbrun, M. Meyer, P. Schroeder, and A. Barr. Implicit fairing of arbitrary meshes using diffusion and curvature flow. In *SIGGraph*. ACM, 1999.

31. A. J. Devaney. The limited-view problem in diffraction tomography. *Inverse Problems*, 5:510–523, 1989.

32. M. P. Do-Carmo. *Riemannian Geometry*. Birkhauser, Boston-Basel-Berlin, 1992.

33. O. Dorn, E.L. Miller, and C.M. Rappaport. A shape reconstruction method for electromagnetic tomography using adjoint fields and level sets. *Inverse Problems*, 16, 2000. 1119-1156.

34. R. Ewing (Ed.). *The mathematics of reservoir simulation*. SIAM, Philadelphia, 1983.

35. H.W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Kluwer, 1996.

36. S. Fleishman, I. Drori, and D. Cohen-Or. Bilateral mesh denoising. *ACM Trans. Graphics (SIGGRAPH)*, 22(3):950–953, 2003.

37. V. Fridman. An iteration process with minimum errors for a nonlinear operator equation. *Dokl. Akad. Nauk. SSSR*, 139:1063–1066, 1961.

38. F Fruhauf, O. Scherzer, and A. Leitao. Analysis of regularization methods for the solution of ill-posed problems involving unbounded operators and a relation to constraint optimization. *SIAM J. Numer. Anal.*, 43:767–786, 2005.

39. M.K. Gavurin. Nonlinear equations and continuous analogs of iterative methods. *Izv. Vyssh. Uchebn. Zaved. (Matematika)*, 5:18–31, 1958. (in Russian).

40. S. Gomez, A. Perez, and R. Alvarez. Multiscale optimization for aquifer parameter identification with noisy data. In *Computational Methods in Water Resources XII, Vol. 2*, 1998.

41. E. Haber. A multilevel, level-set method for optimizing eigenvalues in shape design problems. *J. Comp. Phys.*, 198:518–534, 2004.

42. E. Haber, U. Ascher, D. Aruliah, and D. Oldenburg. Fast simulation of 3D electromagnetic using potentials. *J. Comput. Phys.*, 163:150–171, 2000.

43. E. Haber, U. Ascher, and D. Oldenburg. Inversion of 3D electromagnetic data in frequency and time domain using an inexact all-at-once approach. *Geophysics*, 69:1216–1228, 2004.

44. E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration*. Springer, 2002.

45. F. Hettlich and W. Rundell. Iterative methods for the reconstruction of an inverse potential problem. *Inverse Problems*, 12:251–266, 1996.

46. K. Hildebrandt and K. Polthier. Anisotropic filtering of non-linear surface features. *EUROGRAPHICS*, 23(3):391–400, 2004.

47. M.W. Hirsch and S. Smale. On algorithms for solving f(x) = 0. *Comm. Pure Appl. Math.*, 32:281–312, 1979.

48. H. Huang and U. Ascher. Fast denoising of surface meshes with intrinsic texture. *Preprint*, 2006.

49. T. Jones, F. Durand, and M. Desbrun. Non-iterative, feature preserving mesh smoothing. *ACM Trans. Graphics (SIGGRAPH)*, 22(3):943–949, 2003.

50. N. Karmarkar. A new polynomial time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

51. B. Leimkuhler and S. Reich. *Simulating Hamiltonian Dynamics.* Cambridge University Press, 2004.

52. B. Leimkuhler and R. Skeel. Symplectic integrators in constrained hamiltonian systems. *J. Comp. Phys.*, 112:117–125, 1994.

53. A. Leitao and O. Scherzer. On the relation between constraint regularization, level sets, and shape optimization. *Inverse Problems*, 19:L1–L11, 2003.

54. A. Marquina and S. Osher. Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removel. *SIAM J. Scient. Comput.*, 22:387–405, 2000.

55. S. Maruster. The stability of gradient-like methods. *Appl. Math. Comp.*, 117:103–115, 2001.

56. S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM J. Optimization*, 2:575–601, 1992.

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

58. G.A. Newman and D.L. Alumbaugh. Three-dimensional magnetotelluric inversion using nonlinear conjugate gradients. *Geophysical journal international*, 140:410–418, 2000.

59. J. Nocedal and S. Wright. *Numerical Optimization.* New York: Springer, 1999.

60. J.M. Ortega and W.C. Rheinboldt. *Iterative Solutions of Nonlinear Equations in Several Variables.* Academic Press, 1970.

61. S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation based image restoration. *SIAM J. multiscale model. simul.*, 4:460–489, 2005.

62. S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces.* Springer, 2003.

63. R. L. Parker. *Geophysical Inverse Theory.* Princeton University Press, Princeton NJ, 1994.

64. P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

65. L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

66. F. Santosa. A level-set approach for inverse problems involving obstacles. *ESAIM Controle Optim. Calc. Var.*, 1:17–33, 1996.

67. G. Sapiro. *Geometric Partial Differential Equations and Image Analysis.* Cambridge, 2001.

68. O. Scherzer. Scale-space methods and regularization for denoising and inverse problems. *Advances in Image and Electron Physics*, 128:445–530, 2003.

69. J.A. Sethian. *Level Set Methods and Fast Marching Methods : Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Material Sciences.* Cambridge, 1996.

70. R. Sincovec and N. Madsen. Software for nonlinear partial differential equations. *ACM Trans. Math. Software*, 1:232–260, 1975.

71. S. Smale. A convergent process of price adjustment and global newton methods. *J. Math. Economics*, 3:107–120, 1976.

72. N.C. Smith and K. Vozoff. Two dimensional DC resistivity inversion for dipole dipole data. *IEEE Trans. on geoscience and remote sensing*, GE 22:21–28, 1984.

73. E. Tadmor, S. Nezzar, and L. Vese. A multiscale image representation using heirarchical (bv,l^2) decompositions. *SIAM J. multiscale model. simul.*, 2:554–579, 2004.

74. T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher. Geometric surface smoothing via anisotropic diffusion of normals. In *Proceedings IEEE Visualization*, pages 125–132, 2002.

75. A.N. Tikhonov and V.Ya. Arsenin. *Methods for Solving Ill-posed Problems.* John Wiley and Sons, Inc., 1977.

76. C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *IEEE Intl. Conf. Computer Vision*, pages 839–846, Bombay, India, 1998.

77. U. Trottenberg, C. Oosterlee, and A. Schuller. *Multigrid.* Academic Press, 2001.

78. K. van den Doel and U. Ascher. On level set regularization for highly ill-posed distributed parameter estimation problems. *J. Comp. Phys.*, 216:707–723, 2006.

79. C. Vogel. *Computational methods for inverse problem.* SIAM, Philadelphia, 2002.

80. L. T. Watson, M. Sosonkina, R. C. Melville, A. P. Morgan, and H. F. Walker. 777: HOMPACK90: A suite of FORTRAN 90 codes for globally convergent homotopy algorithms. *ACM Trans. Math. Software*, 23:514–549, 1997.

81. J. Weickert. *Anisotropic Diffusion in Image Processing.* B.G Teubner Stuttgart, 1998.

82. S. Wright. *Primal-Dual Interior Point Methods.* SIAM, Philadelphia, PA, 1997.