9

Efficient Iterative Methods for Fast Solution of Integral Operators Related Problems

Ke Chen

University of Liverpool, UK; k.chen@liverpool.ac.uk

9.1 Abstract

Discretization of integral operators related problems inevitably leads to some kind of linear systems involving dense matrices. Such systems in a large scale can be prohibitively expensive to solve.

In this paper, we shall first review various works that aimed to solve such systems effectively. We start from the solution of boundary integral equations for the exterior Helmholtz problem with smooth boundaries in low and medium wavenumbers, solved by conjugate gradients and multigrid methods. We discuss the importance of effective preconditioning in the contexts of fast multipole methods and wavelet methods.

Then we present some recent work on restoring images in the framework of inverse deconvolution, where the integral operator induced dense matrix, though structured, can only be generated but cannot be computed due to extremely large sizes. No optimal solvers exist for this problem if the nonlinear total-variation semi-norm based regulariser is used. An effective optimisation based multilevel method, using the idea of fast multipole like methods, is developed and presented here. Various numerical experiments are also reported. Finally a brief discussion of open challenges is given.

9.2 Fast iterative methods for the Helmholtz equation

The Helmholtz equation, with Neumann's boundary condition $\frac{\partial \phi}{\partial \mathbf{n}} = g$,

$$\nabla^2 \phi + k^2 \phi = 0, \quad p \in \Omega \tag{9.1}$$

in an infinite domain (i.e. the domain Ω exterior to the surface $S = \partial\Omega$ of some interior domain $\Omega^- \subset \mathbb{R}^3$) is typically solved by a boundary integral equation reformulation into

$$\begin{aligned} L\phi(p) = -\frac{1}{2}\phi(p) + \int_{\partial\Omega} \phi(q) \left(\frac{\partial G_k}{\partial \mathbf{n}_q}(p,q) + \alpha \frac{\partial^2 G_k}{\partial \mathbf{n}_p \mathbf{n}_q} \right) dS_q = \\ \frac{\alpha}{2} g(p) + \int_{\partial\Omega} g(q) \left(G_k(p,q) + \alpha \frac{\partial G_k}{\partial \mathbf{n}_p} \right) dS_q \end{aligned} \tag{9.2}$$

where $\mathbf{n}_q$ is the unit normal exterior to the boundary $\partial\Omega$ away from Ω with

$$G_k(p,q) = \frac{e^{ik|p-q|}}{4\pi|p-q|},$$

and k is the wavenumber; see [AHW92]. Discretization using boundary elements leads to the $n \times n$ linear system [AHW92, Chen05]

$$Au = f \tag{9.3}$$

where $u = \boldsymbol{\phi}$; for collocation method $A_{i,j} = L(p_i)\psi_j$ and for Galerkin method $A_{i,j} = (\psi_i, L(p_i)\psi_j)$.

9.2.1 Order $O(n^2)$ iterative algorithms

With the traditional method of using piecewise polynomial basis functions $\{\psi_j\}$, the above matrix A is dense so matrix-vector multiplications $A\mathbf{x}$ costs $O(n^2)$ operations. In this context, commonly used methods using these multiplications are the following two [Chen05].

Conjugate gradients methods (CGM)

Assume that A is unsymmetric; if symmetric, simpler variants can be used. *Normal equation approach.* The idea is to consider instead of (9.3)

$$AA^T y = f, \quad u = A^T y,$$

since AA^T is symmetric positive definite. Given an initial guess $u^{(0)}$ with residual $r^{(0)} = f - AA^T y^{(0)} = f - Au^{(0)}$, we obtain $\alpha = (r^{(k)})^T r^{(k)} / (p^{(k)})^T AA^T p^{(k)}$ from solving

$$\min_y \|r\|_{AA^T} = \|f - AA^T y\|_{AA^T}$$

in the form $y = y^{(k+1)} = y^{(k)} + \alpha p^{(k)}$ with $p^{(k)}$ a new search direction. Here at $k = 0$, we take $p^{(0)} = r^{(0)}$ and for $k \geq 1$, with $r^{(k)} = f - AA^T y^{(k)} = f - Au^{(k)}$, we use the conjugate gradient direction $p^{(k)} = r^{(k)} + \beta p^{(k-1)}$ with $(p^{(k)})^T AA^T p^{(k-1)} = 0$ i.e.

$$\beta = -\frac{(r^{(k)})^T AA^T p^{(k-1)}}{(p^{(k-1)})^T AA^T p^{(k-1)}} = -\frac{(r^{(k)})^T r^{(k)}}{(r^{(k-1)})^T r^{(k-1)}}.$$

Equivalently let $p^{(k)}$ denote $A^T p^{(k)}$ for the purpose of eliminating the intermediate variable y. Then $u = u^{(k)} + \alpha p^{(k)}$, $p^{(k)} = A^T r^{(k)} + \beta p^{(k-1)}$ with $(p^{(k)}) p^{(k-1)} = 0$ and

$$\alpha = (r^{(k)})^T r^{(k)} / (p^{(k)})^T AA^T p^{(k)} = (r^{(k)})^T r^{(k)} / (p^{(k)})^T p^{(k)}.$$

Algorithm 1 (CGN algorithm)

(CGN Algorithm) *given $x = x_0$, $r = b - Ax_0$ and set initially $p = A^T r$, and $r_{new} = r^T r$*	*(Naive CGN)* *given $y = y_0$, $r = b - AA^T y_0$ and set initially $p = r$, and $r_{new} = r^T r$*
(1) $q = Ap$ *(2)* $\alpha_k = r_{new}/(p^T p)$ *(3) Update the solution* $x = x + \alpha_k p$ *(4)* $r = b - Ax = r - \alpha_k q$ *and set* $r_{old} = r_{new}$ *(5) Compute* $r_{new} = r^T r$ *(exit if* r_{new} *is small enough)* *(6)* $\beta_k = r_{new}/r_{old}$ *(7) Update the search direction* $p = A^T r + \beta_k p$ *and continue with step (1) for* $k = k + 1$.	*(1)* $q = AA^T p, \quad (p^T q = \|A^T p\|_2^2)$ *(2)* $\alpha_k = r_{new}/(p^T q)$ *(3) Update* $y = y + \alpha_k p$ *(4)* $r = b - Ax = r - \alpha_k q$ *and set* $r_{old} = r_{new}$ *(5) Compute* $r_{new} = r^T r$ *(exit* $x = A^T y$ *if* r_{new} *is small)* *(6)* $\beta_k = r_{new}/r_{old}$ *(7) Update the search direction* $p = r + \beta_k p$ *and continue with step (1) for* $k = k + 1$.

Generalized minimal residual approach. The generalized minimal residual method by [Saad96] with m steps of restart aims to solve

$$\min_{u \in V_m} \|r\|_2 = \|f - Au\|_2$$

where $V_m = \text{span}(q_1, q_2, \ldots, q_m)$; here q_j's are columns of an orthogonal matrix Q_m from an Arnoldi iteration

$$AQ_m = Q_m H_m + h q_{m+1} e_m^T,$$

where $e_m \in \mathbb{R}^n$ is the m-th unit vector. Further the above minimization is reduced to the simple least squares' problem

$$H_{m+1} y = \|r^{(0)}\|_2 \hat{e}_1,$$

where $y \in \mathbb{R}^m$ and $\hat{e}_1 \in \mathbb{R}^m$ is the first unit vector. After finding y, we obtain the next iterate $u^{(m)} = u^{(0)} + Q_m y$.

Multigrid method

A multigrid method (MGM) utilizes a series of grids $\partial\Omega_\ell$ with

$$L_\ell u_\ell = f_\ell, \quad \ell = 1, 2, \cdots, L$$

such that $\partial\Omega_1$ is the finest grid where we desire to solve and the other coarser grids are here to speed up the computation. See [Trottenberg2001, Chen05].

The idea of a MGM relies on the residual correction principle. If $\tilde{u}_1$ is a known approximation of u_1 with a nonzero residual $r_1 = f_1 - L_1\tilde{u}_1$. Then solving the residual equation $L_2 v_2 = R_1 r_1$
or for a nonlinear case $L_2 w_2 = L_2 R_1 \tilde{u}_1 + R_1 r_1, \quad v_2 = w_2 - R_1\tilde{u}_1$
will result in an improved approximation $\bar{u}_1 = \tilde{u}_1 + P_1 v_2$ if the original approximation $\tilde{u}_1$ is smooth (not required to be close to u_1). Here R_1, P_1 are respectively the restriction and interpolation operators. This is for a two grid between $\partial\Omega_1, \partial\Omega_2$. Repeated use of this idea will lead to a MGM.

The commonly used V-cycling MGM can be simply stated as $MGM(u_1, f_1, 1)$

Algorithm 2 (MGM algorithm)

```
    $MGM(u_k, f_k, k)$:
(1) Pre-smoothing over $\partial\Omega_k$:    $L_k u_k = f_k$
(2) Computation of the residual $\partial\Omega_k$:    $r_k = f_k - L_k u_k$
(3) Set$k = k + 1$
(4) Restriction to coarse grid $\partial\Omega_k$: $u_k = R_{k-1}u_{k-1}$ and $r_k = R_{k-1}r_{k-1}$
(5) If $k = L$ (coarsest grid), solve $L_k v_k = r_k$ 'accurately'; otherwise
    call the MGM step again: $MGM(u_k, f_k, k)$.
(6) Set $k = k - 1$
(7) Interpolation of the correction $v_k = P_k v_{k+1}$
(8) Update the fine grid solution $u_k = u_k + v_k$
(9) Return to continue
```

The most expensive part of the above two methods CGM and MGM is in matrix-vector products, which may be speeded up.

9.2.2 Order $O(n)$ iterative algorithms

Two excellent ideas of speeding up matrix-vector products shown below lead to fundamentally new and fast methods.

Fast Multi-pole methods (FMM)

The FMM makes to the top 10 algorithms [Cipra00]. For boundary integral equations, the decay properties of the integral kernel can be utilized analytically to design hierarchical expansions that may be arranged to give the FMM approximation for computing each row of matrix L_1 multiplying a vector quickly. Similar algorithms to the FMM can be derived using a function-free and H-matrix approach. See [BR97, Fong09, AP99, BH08].

We remark that while the FMM offers a fast solution per iteration for an iterative method, the overall number of iterations required is dependent on the conditioning of an underlying problem. To speed up such iterations effective preconditioning is necessary. However, computation for a preconditioner is usually restricted by the inaccessibility of matrix elements from far field.

Wavelets methods

Wavelets methods [Dahmen97] offer a revolutionary idea of using bi-orthogonal basis functions, instead of the traditional piecewise polynomials, for discretization of a boundary integral equation. The near orthogonality has two related consequences on the resulting matrix: first it can be easily preconditioned by its diagonal matrix and second the matrix is almost sparse; in fact a full orthogonality would imply a strictly diagonal matrix in some cases. The drawback of a wavelet method is that the wavelets are not easily constructed, though the complication will be worthwhile.

Some recent work by [Mazya07] proposed to work with non-convergent wavelets for solving operator equations. The main idea is that the construction of approximate wavelets is much easier and it remains to test the efficiency of such methods for practical applications.

Combining the traditional boundary elements with the wavelet transforms for the purpose of near optimal preconditioning leads an effective method; this can be done explicitly [CC02] or implicitly [HC05].

9.3 Fast iterative methods for an image deblurring model

A rich class of problems involving an integral operator arise from high resolution image processing [Vogel2002, CS05]. Among others, one example is the image deblurring problem (as shown in Figure 9.1) of reconstructing image u from

$$z(x,y) = (Ku)(x.y) + \eta(x,y)$$

given a noisy and blurred image z in $\Omega = [0,1]^2 \subset \mathbb{R}^2$, where η is unknown but assumed to be a Gaussian white noise with 0 mean and estimable standard deviation σ [Vogel2002].

Although more recent models exist, the most well-known model for the above problem is due to Rudin-Osher-Fatemi [ROF92]

$$\min_u \left\{ E(u) \equiv \alpha \int_\Omega |\nabla u| dxdy + \int_\Omega |Ku - z|^2 \, dxdy \right\}, \tag{9.4}$$

where the first term is the total-variation (TV) semi-norm regularizer with $|\nabla u| = \sqrt{u_x^2 + u_y^2}$ and the integral operator K is assumed to have a spatially-invariant kernel i.e.

$$(Ku)(x,y) = \int_\Omega k(x-x', y-y')u(x',y')dx'dy'.$$

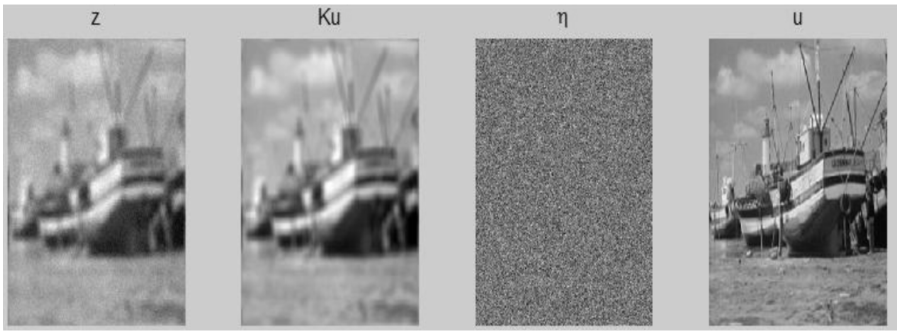

Fig. 9.1. An image deblurring problem.

The Euler-Lagrange partial differential equation of (9.4) is the following

$$-\alpha\nabla\cdot\frac{\nabla u}{|\nabla u|_\beta}+K^*Ku=K^*z \tag{9.5}$$

where $|\nabla u|_\beta=\sqrt{u_x^2+u_y^2+\beta}$ (with $\beta>0$ a small regularization parameter) and K^* is the adjoint operator of K.

Owning to the usually large dimension of a discrete image z (e.g. $n\times n=1024\times 1024\approx 1$ million), the operator K (when discretized as a dense but structured matrix) cannot be directly formed. If u is assumed to have a zero Dirichlet boundary condition, then K will be a block Toeplitz matrix with Toeplitz blocks (BTTB) or if u is assumed to have a periodic boundary condition, then K will be a block circulant matrix with circulant blocks (BCCB). For either case [Vogel2002], the use of fast Fourier transforms (FFT) can ensure the matrix vector product Ku to be efficient; however this restricts many possible solvers to be developed.

Two kinds of efficient methods for solving (9.4) are discussed below.

9.3.1 The methods of intermediate variables

The first method [HNW08] introduces an intermediate variable v for (9.4)

$$\min_{u,v}\ E_1(u,v)\equiv\int_\Omega\left\{\alpha|\nabla u|+\frac{\gamma}{2}(u-v)^2+\frac{1}{2}|Kv-z|^2\right\}dxdy. \tag{9.6}$$

The second method [WYYZ08] introduces an intermediate variable w for (9.4)

$$\min_{u,\omega}\ E_2(u,\omega)\equiv\int_\Omega\left\{\alpha|\omega|+\frac{\gamma}{2}|\omega-\nabla u|^2+\frac{1}{2}|Ku-z|^2\right\}dxdy. \tag{9.7}$$

For either method, alternating minimization leads to simple solutions: For (9.6) solving for u is a simple denoising problem and solving for v can be through FFT because the regularizer now adds a constant diagonal to K^*K. For (9.7) solving for ω can be done analytically while the solution of u involves a simpler semi-norm to enable fast solvers.

Although elegant, both methods involve many (up to 20) iterations between two sub-problems and hence are non-optimal. Besides, practically, only a nearby problem is solved as γ cannot be taken too large.

9.3.2 Optimization based multilevel methods

Below we consider how to solve (9.4) directly after discretizing it. Given $z \in \mathbb{R}^{n\times n}$, the above model [ROF92] rewritten as

$$\min_u E(u), \qquad E(u) = \int_\Omega \left(\overline{\alpha}\sqrt{u_x^2 + u_y^2} + \frac{1}{2}(Ku - z)^2\right) dxdy,$$

can be discretized to give rise to the discrete optimization problem

$$\min_{u\in\mathbb{R}^{n\times n}} E(u), \tag{9.8}$$

$$\begin{aligned} E(u) = \alpha \sum_{i=1}^{n-1}\sum_{j=1}^{n-1} &\sqrt{(u_{i,j} - u_{i,j+1})^2 + (u_{i,j} - u_{i+1,j})^2} \\ + \frac{1}{2}&\sum_{i=1}^{n}\sum_{j=1}^{n}\Big(\sum_{\ell=1}^{n}\sum_{m=1}^{n} K_{i,j;\ell,m}u_{\ell,m} - z_{i,j}\Big)^2, \end{aligned}$$

with $\alpha = \overline{\alpha}/h$ and $h = 1/(n-1)$. Here we shall assume that $K = (K_{i,j;\ell,m})$ is a block circulant matrix with circulant blocks (BCCB). This is the case if we adopt the periodic boundary condition [NB03, Vogel2002, CS05].

Now solve (9.8) by the coordinate descent method on the finest level 1:

$$\begin{cases} \text{Given } u^{(0)} = (u_{i,j}^{(0)}) = (z_{i,j}) \text{ with } l = 0, \\ \text{Solve } \; u_{i,j}^{(l)} = \text{argmin}_{u_{i,j}\in\mathbb{R}} E^{\text{loc}}(u_{i,j}) \text{ for } i,j = 1,2,\ldots,n \\ \text{Set} \quad u^{(l+1)} = (u_{i,j}^{(l)}) \text{ and repeat the above step with } l = l+1 \\ \qquad\quad \text{until a prescribed stopping step on } l, \end{cases} \tag{9.9}$$

where

$$\begin{aligned} E^{\text{loc}}(u_{i,j}) = \frac{1}{2}\|Ku - z\|^2 + \alpha \Bigg[&\sqrt{(u_{i,j} - u_{i+1,j}^{(l)})^2 + (u_{i,j} - u_{i,j+1}^{(l)})^2} \\ &+\sqrt{(u_{i,j} - u_{i-1,j}^{(l)})^2 + (u_{i-1,j}^{(l)} - u_{i-1,j+1}^{(l)})^2} \\ &+\sqrt{(u_{i,j} - u_{i,j-1}^{(l)})^2 + (u_{i,j-1}^{(l)} - u_{i+1,j-1}^{(l)})^2}\,\Bigg]. \end{aligned} \tag{9.10}$$

This iterative method can be applied over a general level k

$$\min_{c_{i,j} \in \mathbb{R}} J(\widetilde{u} + P_k c_{i,j})$$

where P_k is an interpolation operator, distributing a single constant over an index block (i, j) on level k and then padding zeros over the rest of the entire grid of level 1 [CC10].

Although each subproblem in (9.9) is only one dimensional, we see that it has an $O(n^2)$ complexity because the fitting term involves vectors of length n^2 and in particular $(Ku)_{i,j} = u_{i,j} w_t \ + \widetilde{w}_t$, where $t = (j-1)n + i$, $w_t \in \mathbb{R}^{n^2}$ is the tth column of K and $\widetilde{w}_t$ is a vector not involving $u_{i,j}$ (i.e. a weighted sum of all columns of K except t).

The same complexity problem persists on level k, where $\min_{c_{i,j}} J(\widetilde{u} + P_k c_{i,j})$ leads to minimization of the local subproblem

$$J^{\rm loc}(c_{i,j}) = \alpha T(c_{i,j}) + \frac{1}{2}\|c_{i,j} w_t + K\widetilde{u} - z\|^2 = \alpha T(c_{i,j}) + \frac{1}{2}\|c_{i,j} w_t - \widetilde{z}\|^2, \quad (9.11)$$

where $\widetilde{z} = z - K\widetilde{u}$ is known, the vector $w_t \in \mathbb{R}^{n^2}$ denotes the summation of all columns of K corresponding to the entries inside the (i, j) block on level k, and the TV related term $T(c_{i,j})$ is defined by

$$\begin{aligned} T(c_{i,j}) = {} & \sum_{\ell=\ell_1}^{\ell_2} \sqrt{(c_{i,j} - h_{k_1-1,\ell})^2 + v_{k_1-1,\ell}^2} \ + \sum_{m=k_1}^{k_2-1} \sqrt{(c_{i,j} - v_{m,\ell_2})^2 + h_{m,\ell_2}^2} \ + \\ & \sum_{\ell=\ell_1}^{\ell_2-1} \sqrt{(c_{i,j} - h_{k_2,\ell})^2 + v_{k_2,\ell}^2} \ + \sum_{m=k_1}^{k_2} \sqrt{(c_{i,j} - v_{m,\ell_1-1})^2 + v_{m,\ell_1-1}^2} \ + \\ & \quad \sqrt{2}\sqrt{(c_{i,j} - \overline{v}_{k_2,\ell_2})^2 + \overline{h}_{k_2,\ell_2}^2}, \end{aligned}$$

$$\begin{cases} v_{m,\ell} = \widetilde{u}_{m,\ell+1} - \widetilde{u}_{k,\ell}, & h_{m,\ell} = \widetilde{u}_{m+1,\ell} - \widetilde{u}_{m,\ell}, \\ \overline{v}_{k_2,\ell_2} = \dfrac{v_{k_2,\ell_2} + h_{k_2,\ell_2}}{2}, & \overline{h}_{k_2,\ell_2} = \dfrac{v_{k_2,\ell_2} - h_{k_2,\ell_2}}{2}. \end{cases} \quad (9.12)$$

Clearly since each iteration would take $O(n^2)$ per block on any level, the overall algorithm will have $O(n^4)$ at least and is hence not optimal.

A new breakthrough on the issue was made in [CC10], based on the reorganizing the solution of the above coarse level subproblems. We first observe that the first order condition of (9.11) takes the form

$$\alpha T'(c_{i,j}) + w_t^T w_t c_{i,j} = w_t^T \widetilde{z}, \quad (9.13)$$

where $w_t^T w_t$, $w_t^T \widetilde{z}$ (for all w_t recursively as one deals with partial sums in the fast multi-pole method [BR97]) can be worked out, though of complexity $O(n^2)$, once only. Then we anticipate that after all such quantities are computed and stored first before each multilevel cycle, the local solvers will not be expensive to proceed.

Finally our new multilevel method for the combined denoising and deblurring problem for solving (9.8) is the following:

Algorithm 3 *Given z and an initial guess $\widetilde{u} = z$, with $L+1$ levels,*

```
Pre-calculation.
1. Compute all root matrices $T_k$ and $w_t^T w_t = \|T_k\|_F^2$ for partial sum matrices
   on level $k = 1, 2, \ldots, L+1$.
Multilevel Iterations.
2. Iteration starts with $u_{old} = \widetilde{u}$ ($\widetilde{u}$ contains the initial guess before the first
   iteration and the updated solution at all later iterations).
for $\nu$ times on each level $k = 1, 2, 3, \ldots, L+1$:
  3. Compute $\widetilde{z} = z - K\widetilde{u}$ and form $K^T\widetilde{z}$ via the FFT.
  for each block on level $k$,
    4. form each $w_t^T\widetilde{z}$ from $K^T\widetilde{z}$ and compute the minimizer $c$ of (9.13).
  end block.
  5. add all the corrections (from all blocks on level $k$), $\widetilde{u} = \widetilde{u} + P_k c$, where
     $P_k$ is the interpolation operator distributing $c_{i,j}$ to the corresponding
     $b \times b$ block on level $k$.
end level $k$.
6. On level $k = 1$, check the possible patch size for each position $(i, j)$:

       $patch = \{(i_\ell, j_\ell) \ : \ |u_{i_\ell, j_\ell} - u_{i,j}| < \varepsilon\}$

   for some small $\varepsilon$.
   First compute the partial sum vector $w_t$ related the detected columns.
   Then implement the piecewise constant update as with Steps $3-5$.
7. If $\|\widetilde{u} - u_{old}\|_2$ is small enough, exit with $u = \widetilde{u}$ or return to Step 2 and
   continue with the next multilevel cycle.
```

Two illustrating examples solved by Algorithm 3 are shown in Figures 9.2-9.3.

9.4 Open problems and challenges

In this short paper, we discussed two problems solved by integral methods. Both problems may be studied further.

Firstly for the Helmholtz equation, an optimal way of combining the FMM and wavelets with suitable preconditioning is still to be found out. The more recent interests of many researchers have turned to high frequency modeling (where the wavenumber k is large) and inverse problems where the boundary information (either boundary conditions or the boundary itself) is missing while some measurement of the solution is known.

Secondly for the image deblurring problem, it remains to develop effective iterative methods for cases where a more sophisticated choice of regularisers (e.g. the mean curvature [BC10]) is used or a spatially variant blur kernel is used. A more practical problem is the blind deblurring where the kernel k is not known and must be estimated together with u. See [CW98, MK08].

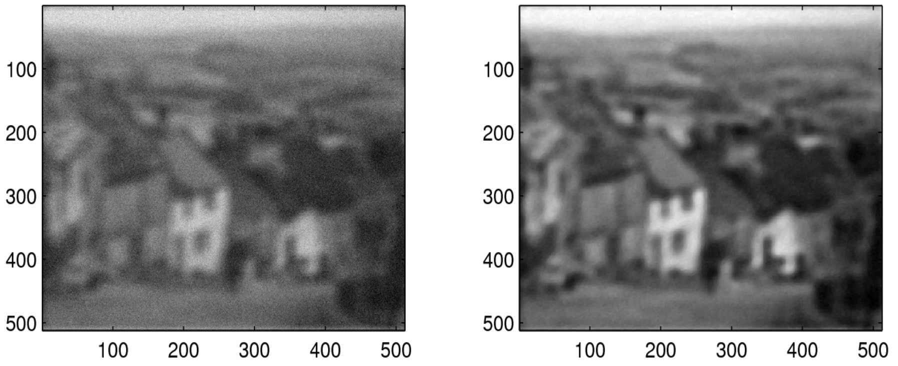

Fig. 9.2. Image deblurring example 1 by Algorithm 3: left z and right u.

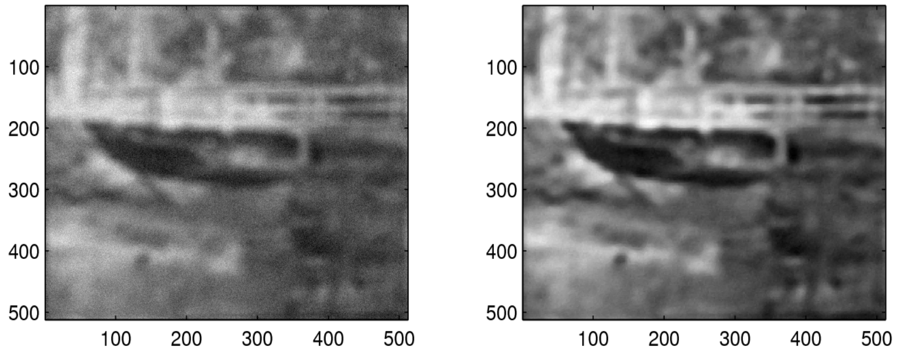

Fig. 9.3. Image deblurring example 2 by Algorithm 3.

Acknowledgements. Most of the results reviewed in this paper were achieved through joint work with collaborators including Paul J. Harris (University of Brighton), Siamiak Amini (University of Salford), Tony F. Chan (University of California at LA) and Raymond H. Chan (Chinese University of Hong Kong).

References

[AP99] Amini S. and Profit, A. J., *Analysis of a diagonal form of the fast multipole algorithm for scattering theory*, BIT, 39(4):585-602, (1999).

[AHW92] Amini S., Harris P. J. and Wilton D. T.: *Coupled Boundary And Finite Element Methods For The Solution Of The Dynamic Fluid-Structure Interaction Problem*, Lecture Note in Engineering, 77, ed. C. A. Brebbia and S. A. Orszag, Springer-Verlag, London (1992).

[BH08] Banjai, L. and Hackbusch W., *Hierarchical matrix techniques for low and high frequency Helmholtz equation*, IMA J. Numer. Anal., 28(1):46-79, (2008).

[BR97] Beatson R. and Greengard L., *A short course on fast multipole methods*, in: `Wavelets, Multilevel Methods and Elliptic PDEs`, Oxford University Press, pp.1-37, (1997). See also
http://math.nyu.edu/faculty/greengar/shortcourse_fmm.pdf

[BC10] Brito C. and Chen K., *Multigrid algorithm for high order denoising*, SIAM Journal on Imaging Sciences, 3 (3):363-389, (2010).

[CC10] Chan R. H. and Chen K., *A multilevel algorithm for simultaneously denoising and deblurring images*, SIAM Journal on Scientific Computing, 32(2):1043-1063, (2010).

[CC02] Chan T. F. and Chen K., *On two variants of an algebraic wavelet preconditioner*, SIAM Journal on Scientific Computing, 24(1):260-283, (2002)

[CS05] Chan T. F. and Shen J. H., *Image Processing and Analysis – Variational, PDE, Wavelet, and Stochastic Methods*, SIAM Publications, USA (2005).

[CW98] Chan T. F. and Wong C. K., *Total variation blind deconvolution*, IEEE Trans. Image Process., 7(3):370-375, (1998).

[Chen05] Chen K., *Matrix Preconditioning Techniques and Applications*, Series: Cambridge Monographs on Applied and Computational Mathematics (No. 19), Cambridge University Press, (2005).

[Cipra00] Cipra B. A., *The Best of the 20th Century: Editors Name Top 10 Algorithms*, SIAM News, 33 (4), (2000).

[Dahmen97] Dahmen W., *Wavelet and multiscale methods for operator equations*, Acta Numerica, 1997:55-228, (1997).

[Fong09] Fong, W. and Darve E., *The black-box fast multipole method*, J. Comp. Phys., 228 (23):8712-8725, (2009).

[HC05] Hawkins S. and Chen K., *An implicit wavelet approximate inverse preconditioner*, SIAM Journal on Scientific Computing, 27(2):667-686, (2005).

[HNW08] Huang Y., Ng M. and Wen Y., *A fast total variation minimization method for image restoration*, SIAM Journal on Multiscale Modeling and Simulation, 7:774-795, (2008).

[Mazya07] Mazya V. and Schmidt G., *Approximate Approximations*, American Mathematical Society, (2007).

[MK08] Money J. H. and Kang S. H., *Total variation minimizing blind deconvolution with shock filter*, Image Vision Comput., 26(2):302-314, (2008).

[NB03] Ng M. and Bose N. K., *Mathematical analysis of super-resolution methodology*, IEEE Signal Proc. Magazine, 20(3):62-74, (2003).

[ROF92] Rudin L. I., Osher S. and Fatemi E., *Nonlinear total variation based noise removal algorithms*, Physica D, 60:259-268, (1992).

[Saad96] Saad Y. and Schultz M. H., *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7:856-869, (1986).

[Trottenberg2001] Trottenberg U., Oosterlee C. and Schuller A., *Multigrid*, Academic Press, (2001).

[Vogel2002] Vogel C. R., *Computational Methods for Inverse Problems*, SIAM Publications, Philadelphia, USA (2002).

[WYYZ08] Wang Y., Yang J., Yin W. and Zhang Y., *A new alternating minimization algorithm for total variation image reconstruction*, SIAM Journal on Imaging Sciences, 1(3):248-272, (2008).