A FAST SOLVER FOR THE ORNSTEIN-ZERNIKE EQUATIONS *

C. T. KELLEY† AND B. MONTGOMERY PETTITT‡

Abstract. In this paper we report on the design and analysis of a multilevel method for the solution of the Ornstein-Zernike Equations and related systems of integro-algebraic equations. Our approach is based on an extension of the Atkinson-Brakhage method, with Newton-GMRES used as the coarse mesh solver. We report on several numerical experiments to illustrate the effectiveness of the method.

Key words. Multilevel method, Newton-GMRES, Ornstein-Zernike equations, nonlinear equations

AMS subject classifications. 65H10, 65N55, 65R20, 65T50, 92E10,

1. Introduction. In this paper we propose a fast multi-level method for the solution of a class of integral equations called the Ornstein-Zernike (OZ) equations, which are useful in calculating probability distributions of matter (atoms) in fluid states [10]. Our approach is faster than a Newton-Krylov approach, such as the one proposed in [2], because the linear solver is only used on a coarse mesh problem.

The OZ equations were initially designed to model density fluctuations near the critical point via the equilibrium theory of liquids [8,15]. Since then the range of validity and usefulness has been extended to include the entire fluid range of states. This set of nonlinear coupled integral equations has been derived from the full partition function for atomic systems [20] and while essentially never solved without some approximations has proved a useful tool for understanding liquids at the atomic level for over 50 years. While the OZ equation has two unknowns it is usually closed with another often algebraic relation between the two unknown functions. Two useful approximate closure relations are the Perkus-Yevick equation [16] and the hyper netted chain equation [21]. These equations then provide essentially two equations and two unknowns and when convenient may be substituted into the OZ equation to provide a single nonlinear integral equation for the unknown probability distribution function.

The equations, when the physical parameters are reasonably adjusted, have solutions which can be achieved by a variety of techniques [10]. Those methods include Picard iteration with or with out relaxation and basis set (variational) methods. In cases where the physical parameters make the equations stiff, iterative solutions are particularly tedious.

The objectives of this paper are to describe an multilevel approach for solving the OZ equations and apply that new approach to two examples. The multilevel method is based on the enhanced version of the Atkinson-Brakhage [1, 3] method from [13]. We show that we can compute the solution to the accuracy of truncation error in $O(1)$ fine mesh function evaluations.

In their simplest isotropic form the OZ equations are a system consisting of an integral equa-

*Version of May 24, 2002.

† North Carolina State University, Center for Research in Scientific Computation and Department of Mathematics, Box 8205, Raleigh, N. C. 27695-8205, USA (Tim_Kelley@ncsu.edu). The research of this author was supported in part by National Science Foundation grant DMS-0070641. Computing activity was partially supported by an allocation from the North Carolina Supercomputing Center.

‡ University of Houston, Department of Chemistry, 136 Fleming Building, Houston, TX 77204-5003, USA (pettitt@uh.edu). This author's work was supported in part by the NIH and in part by the Robert A. Welch Foundation

tion

$$h(r) = c(r) + \rho(h * c)(r) \tag{1.1}$$

where

$$(h * c)(r) = \int c(\|\mathbf{r} - \mathbf{r}'\|)h(\|\mathbf{r}'\|)d\mathbf{r}' \tag{1.2}$$

and the integral is over R^3. The unknown functions h and c are radially symmetric, *i. e.* functions of $r = \|\mathbf{r}\|$ only. ρ is the total number density usually expressed in particles per volume such as atoms per cubic angstrom; h is the radial pair correlation function and c is the so called direct correlation function and may be taken to be defined by this equation. The total radial correlation function, h, is an experimental observable from x-ray or neutron diffraction experiments which provides a connection for this theory to physics.

The convolution $h*c$ can be computed with only one-dimensional integrals using the spherical-Bessel transform. If h decays sufficiently rapidly we define

$$\hat{h}(k) = \mathcal{H}(h)(k) = 4\pi \int_0^\infty \frac{sin(kr)}{kr} h(r) r^2 dr$$

and

$$h(r) = \mathcal{H}^{-1}(\hat{h})(r) = \frac{1}{2\pi^2} \int_0^\infty \frac{sin(kr)}{kr} \hat{h}(k) k^2 dk.$$

We compute $h * c$ by discretizing the formula

$$h * c = \mathcal{H}^{-1}(\hat{h}\hat{c}), \tag{1.3}$$

where $\hat{h}\hat{c}$ is the pointwise product of functions.

Next we can view the closure equation as an algebraic constraint. Here we chose the HNC equation which may be derived as an approximation from the partition function for the system.

$$\exp(-\beta u(r) + h(r) - c(r)) - h(r) - 1 = 0, \text{ for all } 0 \le r \le \infty. \tag{1.4}$$

The unknowns are $h, c \in C[0, \infty]$. We truncate the interval for computational purposes and consider $h, c \in C[0, L]$ for $L < \infty$.

In (1.4), u is the pair potential between particles. Here we will take the usual Lennard-Jones potential as typical

$$u(r) = 4\epsilon \left(\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^{6} \right). \tag{1.5}$$

In (1.5), β, ϵ, and σ are parameters. In particular β is the inverse of the product of absolute temperature and Boltzmanns constant, ϵ is the well depth of the potential, and σ determines the size of the particles.

Equations (1.1)–(1.4) are representative of more general systems of equations [7, 18] in which the ρ may be unknown and/or the unknowns may be matrix-valued. We apply the algorithm proposed in this paper to such problems in § 3.

The system (1.1)-(1.4) and the more complex systems from [7, 18] can be expressed in the general form

$$\mathcal{F}(U) = M(U) - K(U) = 0. \tag{1.6}$$

In (1.6) the unknown is an vector-valued function $U \in X = C([0, L], R^P)$.

$M(U)$ a substitution operator. $K(U)$ is a completely continuous nonlinear map of the form

$$K(U)(r) = \Phi\left(\int B(U(\|\mathbf{r} - \mathbf{r}'\|), U(\mathbf{r}'))\, d\mathbf{r}'\right), \tag{1.7}$$

In (1.7) $\Phi \in C(R^P, R^P)$, $B \in C(R^P \times R^P, R^P)$ and B is linear in its first argument. This means that evaluations of $K(U)$ and $K'(U)W$ require only convolutions.

We will assume throughout this paper that the standard assumptions [14] for nonlinear equations hold for (1.6). These are that there is a solution $U^* \in X$, that $\mathcal{F}$ is Fréchet differentiable and Lipschitz continuous in a neighborhood of U^*, and that $\mathcal{F}'(U^*)$ is nonsingular. These assumptions imply that $M'(U^*)$ is a nonsingular $P \times P$ matrix-valued function of r.

In the simple example consisting of equations (1.1) and (1.4), $P = 2$, Φ is the identity, $U = (h, c)^T$,

$$B(U, U') = B\left(\begin{pmatrix} h \\ c \end{pmatrix}, \begin{pmatrix} h' \\ c' \end{pmatrix}\right) = \begin{pmatrix} 0 \\ \rho c h' \end{pmatrix},$$

and

$$\begin{aligned} M(U)(r) &= \begin{pmatrix} h(r) - c(r) \\ \exp(-\beta u(r) + h(r) - c(r)) - h(r) - 1 \end{pmatrix}, \text{ and} \\ K(U)(r) &= -\begin{pmatrix} 0 \\ \rho(c * h)(r) \end{pmatrix}. \end{aligned} \tag{1.8}$$

2. Algorithms. In this section we focus entirely on the simple system (1.1)-(1.4). The more general systems can be solved using exactly the same intergrid transfers and discretizations. The notation is complicated by the need to refer to the mesh size for both the approximate solutions and the discretized nonlinear equations.

2.1. Discretization. We will approximate (h, c) by piecewise linear functions. The approximating space is

$$X_\delta = V_\delta^2$$

where V_δ is the space of piecewise linear functions with nodes on

$$\Omega_\delta = \{r_i^\delta\}_{i=1}^N$$

where $\delta = L/(N-1)$ is the mesh width and $r_i^\delta = (i-1)\delta$. We will approximate the integral operators with the trapezoid rule.

Our fine-to-coarse mesh intergrid transfer will be based on the usual L^2 projection onto V_δ. For $u \in C[0, L]$ let

$$(P_\delta u)(r) = \int_0^L p_\delta(r, r') u(r')\, dr' = \sum_{i,j=1}^N \phi_i(r) l_{ij} \int_0^L u(r') \phi_j(r')\, dr'$$

where $\{\phi_i\}$ are the nodal PL basis functions and

$$\delta_{ij} = \sum_{k=1}^N l_{ik} \int_0^L \phi_k(r') \phi_j(r')\, dr'.$$

P_δ is an integral operator with kernel

$$p_\delta(r,r') = \sum_{i,j=1}^{N} l_{ij}\phi_i(r)\phi_j(r').$$

For $u, v \in V_\delta$ we compute the discrete convolution indirectly using a discrete spherical-Bessel transform. We begin by discretizing frequency in a way that allows us to use the fast fourier transform to evaluate the convolution. Let $k_j = (j-1)\delta_k$ where $\delta_k = \pi\delta/(N-1)$. We define, for $2 \le j \le N-1$

$$\begin{aligned} \hat{v}_j &= \mathcal{H}(v)(k_j) \\ &= \tfrac{4\pi\delta^2}{(j-1)\delta_k}\textstyle\sum_{i=2}^{N-1}(i-1)v_i \sin((i-1)(j-1)\delta_k\delta) \qquad (2.1) \\ &= \tfrac{4\pi\delta^3(N-1)}{j-1}\textstyle\sum_{i=2}^{N-1}(i-1)v_i \sin((i-1)(j-1)\pi/(N-1)). \end{aligned}$$

Then, for $2 \le i \le N-1$,

$$\mathcal{H}^{-1}(\hat{v})_i = \frac{1}{2(i-1)\pi\delta^3}\sum_{j=2}^{N-1} k\hat{v}_j \sin((i-1)(j-1)\pi/(N-1)).$$

Finally, define for $2 \le i \le N-1$,

$$(u*v)_i = \mathcal{H}^{-1}(\hat{u}\hat{v})$$

where $\hat{u}\hat{v}$ denotes the component-wise product. We set $(u*v)_N = 0$ and define $(u*v)_1$ by linear interpolation

$$(u*v)_1 = 2(u*v)_2 - (u*v)_3.$$

This discretization of the convolution is second order accurate in δ for $u, v \in C^2$.

Our discrete convolution operator is the piecewise linear interpolant of $\{(u*v)_i\}_{i=1}^N$ at the nodes $\{r_i\}_{i=1}^N$.

$$Q_\delta(u,v)(r) = \sum_{i=1}^{N}\phi_i(r)(u*v)_i \qquad (2.2)$$

where $u_i = u(r_i^\delta)$, $v_i = v(r_i^\delta)$. The sum in (2.2) can be evaluated with a fast Fourier transform in $O(N\log N)$ floating operations. We approximate K by

$$K_\delta(U) = \begin{pmatrix} 0 \\ \rho Q_\delta(P_\delta h, P_\delta c) \end{pmatrix}.$$

We approximate the solution of (1.1)-(1.4) with the function $U^\delta \in X_\delta$ that satisfies

$$M(U^\delta)(r_i^\delta) - K_\delta(U^\delta)(r_i^\delta) = 0, \qquad (2.3)$$

for $1 \le i \le N$. We will express this equation as

$$\mathcal{F}_\delta(U^\delta) = M_\delta(U^\delta) - K_\delta(U^\delta) = 0.$$

Note that (2.3) is equivalent to a fully discrete system for the values of h and c at the nodes

$$\begin{aligned} h_i - c_i - \rho Q_\delta(h,c)(r_i^\delta) &= 0 \\ \exp(-\beta u_i + h_i - c_i) - h_i - 1 &= 0, \end{aligned}$$

which is the finite-dimensional system that one solves numerically.

The Kantorovich theorem [11] and the uniform Lipschitz continuity of $\mathcal{F}'_\delta$ imply that if U^* is sufficiently smooth and $\mathcal{F}'(U^*)$ is nonsingular, then (2.3) has a solution U^δ for all δ sufficiently small,

- $U^\delta = U^* + O(\delta^2)$, and
- $\mathcal{F}'_\delta(U^\delta)$ is nonsingular.

Moreover, for all $U \in X$, $K'_\delta(U) \to K'(U)$ in the operator norm on X. Therefore, $K'(U^\delta) \to K'(U^*)$ in the operator norm.

2.2. Nested Iteration. Both of the algorithms in this paper use multiple meshes and require intergrid transfers. Following standard notation [4], we let $I_{\delta_s}^{\delta_t}$ be the intergrid transfer between a source X_{δ_s} and target X_{δ_t}. Nested iteration requires only a coarse-to-fine transfer, in which $\delta_s > \delta_t$ and we do this transfer at the fully discrete level with piecewise linear interpolation. Since

$$X_\delta \subset X_{\delta/2}$$

the coarse-to-fine intergrid transfer is simply the identity if the grids are nested. We identify X with X_0 and use the notation I_δ^0 for the map that interpolates a vector in X_δ to produce piecewise linear functions h and c.

A nested iteration approximately solves $\mathcal{F}(U) = 0$ on a sequence of meshes, terminating with a solution at a target, finest mesh.

The nonlinear solve inside the loop of **nest_generic** is often only a single step of an approximate Newton iteration. We advocate two such nested iteration algorithms here. They differ only in how the approximate Newton step on approximate on each mesh is computed. The coarse mesh has width δ_0 and the mesh is refined by halving the width. The ultimate mesh width is $\delta = 2^{-lmax}\delta_0$.

Algorithm 1 Generic nested iteration.

nest_generic$(U, \mathcal{F}, \delta_0, l_{max})$
  $l \leftarrow 0$; $\delta \leftarrow \delta_0$; $U_0^{\delta_0} \leftarrow U$
  Solve $\mathcal{F}_\delta(U^{\delta_0}) = 0$ to high accuracy with $U_0^{\delta_0}$ as the initial iterate.
  while $l < l_{max}$ **do**
    $\delta \leftarrow \delta/2$; $l \leftarrow l + 1$
    Interpolate $U^{2\delta}$ to Ω_δ to obtain U_0^δ
    Solve $\mathcal{F}_\delta(U^\delta) = 0$ to reasonable accuracy with U_0^δ as the initial iterate.
  end while

For the computations in this paper, we used nested grids with the mesh size reduced by a factor of two at each level. The design of the algorithms allows for non-nested grids. Second order accuracy implies that the truncation error should be reduced by a factor of four at each level. To that end we solve the equation on the coarsest mesh of high precision, driving the nonlinear residual to a very small value, and then ask that the nonlinear solver reduce the nonlinear residual by a factor of ten on the subsequent, finer, grids. This took a single nonlinear iteration in the computations reported in § 3.

2.3. Nested Newton-GMRES. The Newton-GMRES method is an inexact Newton method which approximates the solution of the linear equation for the Newton step from U_c,

$$\mathcal{F}'(U_c)s = -\mathcal{F}(U_c), \tag{2.4}$$

with a GMRES [19] iteration. The termination criterion for the linear iteration is the standard [9,14] inexact Newton condition:

$$\|\mathcal{F}'(U_c)s + \mathcal{F}(U_c)\| \le \eta\|\mathcal{F}(U_c)\|. \tag{2.5}$$

η is a parameter, which we set to $1/10$ in all of the computations reported in § 3.

The Newton-GMRES code NITSOL [17] was applied to the OZ equations in [2]. The algorithm performed well, which is not surprising in view of the mesh independence results for GMRES when applied to integral equations.

Mesh-independence of the GMRES iteration will follow from the convergence of $K'(U^\delta)$ to $K'(U^*)$ in the operator norm [5, 6]. This means that then the number of GMRES iterations needed to satisfy (2.5) from an initial iterate of $s = 0$ is independent of the level δ of discretization.

Algorithm 2 Newton-GMRES

newton-gmres$(U, \mathcal{F}, \tau_a, \tau_r, \eta)$
  Evaluate $\mathcal{F}(U)$; $\tau \leftarrow \tau_r\|\mathcal{F}(U)\| + \tau_a$.
  while $\|\mathcal{F}(U)\| > \tau$ **do**
    Solve the linear system $\mathcal{F}'(U)s = -\mathcal{F}(U)$
    with GMRES and terminate when (2.5) holds.
    $U \leftarrow U + s$
    Evaluate $\mathcal{F}(U)$.
  end while

Our implementation of nested Newton-GMRES uses **newton-gmres** to solve the coarse mesh equation to high precision. On the finer meshes we ask that the size of the residual be reduced by a factor of ten. This nesting is a step beyond the method in [2] and substantially improves performance, because the most of the matrix-vector products are done on coarse grids.

The theory in [6] implies that the number of GMRES iterations at each level is bounded independently of δ. Since each GMRES iteration requires a function evaluation for the forward difference approximation to the Jacobian-vector product and only one nonlinear iteration per level will be needed if the coarse mesh solution is sufficiently accurate, the number of calls to the function at each level is bounded. Let $C_F(\delta)$ denote the the cost of a function evaluation on Ω_δ. For the examples considered here,

$$C_F(\delta) = O(\log(1/\delta)/\delta).$$

Assume that no more than C_G GMRES iterations are needed at each level. The the cost of the solve can be bounded by

$$(C_G + 1)\sum_{l=0}^{lmax} C_F(2^l/\delta_0 \log(2^l/\delta_0) \le 2(C_G + 1)C_F(\delta_{lmax}). \tag{2.6}$$

Hence, if the coarse mesh solution is sufficiently accurate, a solution accurate to truncation error can be obtained at a cost proportional to that of a fine-mesh function evaluation. The proportionality constant is related to the number of GMRES iterations needed for each nonlinear iteration.

2.4. Multilevel Iteration. In this section we describe an approximate Newton method that uses an extension of the method in [13] to approximate the Newton step. The idea is to use a coarse mesh approximate inverse of $\mathcal{F}'(U)$ and base the computation of the approximate Newton step on that approximate inverse.

The approach in [13], which we follow in this section, is a degenerate kernel approach for solving second kind Fredholm integral equations of the form

$$(I-K)u(r) = u(r) - \int_0^L k(r,r')u(r')\,dr' = f(r). \tag{2.7}$$

The approach is to build an approximate inverse To solve the discretization of (2.7) on Ω_δ, we approximate K by K_Δ, where

$$(K_\Delta u)(r) = \int_0^L k_\Delta(r,r')u(r')\,dr' \approx (Ku)(r) = \int_0^L k(r,r')u(r')\,dr',$$

where

$$k_\Delta(r,r') = \sum_{i,j} k(r_i^\Delta, r_j^\Delta)\phi_i(r)\phi_j(r').$$

and ϕ_j is the piecewise linear "hat function" centered at r_j^Δ. The operators $I-K_\Delta$ converge in the operator norm to $I-K$ and, therefore, $(I-K_\Delta)^{-1}$ is an approximate inverse of $I-K$ and can be used as a preconditioner for a Richardson iteration to solve the discrete problem on Ω_δ for any $\delta < \Delta$.

The preconditioned Richardson iteration for the discretization of (2.7) is

$$u_+ = u_c - (I-K_\Delta)^{-1}(f-(I-K)u_c).$$

The implementation requires one fine-mesh operator-function product to compute the residual $w = f-(I-K)u_c$. After the computation of r, one solves

$$s - K_\Delta s = w. \tag{2.8}$$

Let P^Δ be the L^2 projection onto the piecewise linear functions with nodes on Ω_Δ. To solve (2.8) we first find $s^\Delta = P^\Delta s$. as follows. Since s^Δ is uniquely determined by its values at the coarse mesh odes, we can solve a finite dimensional system for

$$s_i^{FD} = s^\Delta(r_i^\Delta), \text{ for } i = 1,\ldots,N_\Delta,$$

where N_Δ is the number of points in the coarse mesh. The fully discrete coarse-mesh equations for s^{FD} are

$$s_i^{FD} - \sum_j k(r_i^\Delta, r_j^\Delta)s_j^{FD} = (P^\Delta w)(r_i^\Delta), \text{ for } i = 1,\ldots,N_\Delta,$$

which one can solve with GMRES, for example. Since

$$K_\Delta s = K_\Delta P^\Delta s,$$

one can recover s at the fine mesh points with the Nyström interpolation

$$s(r) = w(r) + (K_\Delta P^\Delta s)(r).$$

Algorithm 3 Nested Richardson Iteration for $u - Ku = f$

nest_richardson(u, K, Δ, l_{max})

$l \leftarrow 0$; $\delta \leftarrow \Delta$
Solve $u^\Delta - K_\Delta u^\Delta = f$ to high accuracy.
while $l < l_{max}$ **do**
$\delta \leftarrow \delta/2$; $l \leftarrow l + 1$
Interpolate $U^{2\delta}$ to Ω_δ to obtain U_0^δ.
$w \leftarrow f - (I - K_\delta)u_0^\delta$.
$s \leftarrow (I - K_\Delta)^{-1} w$.
$u^\delta \leftarrow u_0^\delta - s$
end while

The nested iteration form of this algorithm is

We apply this algorithm to the OZ equations by using it to solve the equation for the Newton step

$$(M'(U) - K'(U))s = \mathcal{F}'(U)s = -\mathcal{F}(U)$$

rewritten as

$$I - M'(U)^{-1}K'(U)s = M'(U)^{-1}\mathcal{F}'(U)s = -M'(U)^{-1}\mathcal{F}(U). \tag{2.9}$$

We implement the multilevel solver as an approximate inverse for $\mathcal{F}'$, rather than one for $I - M'^{-1}K'$. We define the fine-to-coarse intergrid transfer $I_\delta^\Delta = J^\Delta$ and the coarse-to-fine transfer I_Δ^δ as piecewise linear interpolation. Given a current iterate $U_c^\delta \in X_\delta$, and coarse mesh solution U^Δ, we obtain

$$\begin{aligned} J_\Delta^{-1} &= (I - I_\Delta^\delta I_\delta^\Delta)M_\delta'(I_\Delta^\delta U^\Delta)^{-1} \\ &+ I_\Delta^\delta \mathcal{F}_\Delta'(U^\Delta)^{-1} M_\Delta'(U^\Delta)^{-1} I_\delta^\Delta M_\delta'(I_\Delta^\delta U^\Delta)^{-1} \end{aligned} \tag{2.10}$$

To take an approximate Newton step

$$U_+ = U_c - J_\Delta^{-1}\mathcal{F}_\delta(U_c)$$

on X_δ, we apply the approximate inverse to $\mathcal{F}_\delta(U_c)$. This requires only coarse mesh solves and multiplications by M'^{-1}. The latter is trivial, since M' is a 2×2 matrix-value function of r.

In fact, if we set

$$N(h, c)(r) = \exp(-\beta u(r) + h(r) - c(r)), \tag{2.11}$$

then for each $r \in [0, L]$,

$$M'(U) = \begin{pmatrix} \partial M_1/\partial h & \partial M_1/\partial c \\ \partial M_2/\partial h & \partial M_2/\partial c \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ N-1 & -N \end{pmatrix}. \tag{2.12}$$

Therefore the discrete operator M' is a block diagonal matrix with 2×2 blocks.

For Δ sufficiently small, a single nonlinear iteration will suffice for each δ and one can refine the mesh after each Newton iteration [13].

The multilevel algorithm is

Algorithm 4 Nested Multilevel Algorithm

nest_generic$(U, \mathcal{F}, \Delta, l_{max})$
  $l \leftarrow 0$; $\delta \leftarrow \Delta$. $U_0^{\delta_0} \leftarrow U$
  Solve $\mathcal{F}_\delta(U^\Delta) = 0$ to high accuracy
  with U_0^Δ as the initial iterate.
  while $l < l_{max}$ **do**
    $\delta \leftarrow \delta/2$; $l \rightarrow l + 1$
    $U_0^\delta \leftarrow I_{2\delta}^\delta U^{2\delta}$.
    $U^\delta \leftarrow U^\delta - J_\Delta^{-1} \mathcal{F}_\delta(U_0^\delta)$
  end while

The cost of the computation differs from that of the nested Newton-GMRES algorithm only in that there are no fine-mesh GMRES iterations. Hence, neglecting all coarse mesh work, the bound on the cost of a solve to truncation error is

$$2C_F(\delta_{lmax}).$$

3. Numerical Results. In this section we consider three examples and compare the performance of the nested Newton-GMRES method with the multilevel method proposed in § 2

The computations were done in MATLAB 5.3, using the Newton-GMRES code from [12] for the nested Newton-GMRES results and the coarse mesh computations for the multilevel results.

A nested iteration, if working properly, will decrease the error at each level in a way consistent with theory. To illustrate this, we tabulate the scaled l^2 norm of initial nonlinear residual at each mesh. In the language of Algorithm **nest_generic**, we tabulate

$$R_\delta = \|\mathcal{F}_\delta(U_0^\delta)\|, \text{ where } U_0^\delta = I_{2\delta}^\delta U^{2\delta},$$

at each level. The scaled l^2 norm of a vector in $w \in R^N$ is

$$\|w\| = \left(\frac{1}{\sqrt{N}} \sum_{j=1}^N w_i^2 \right)^{1/2}.$$

Because both the discretization and the coarse-to-fine intergrid transfer are second order accurate, one would expect these residual norms to decrease by factors of four as $\delta \rightarrow \delta/2$. We also tabulate the number of GMRES iterations i_G for each nonlinear iteration. This refers to fine-mesh iterations (*i. e.* on Ω_δ) for Newton-GMRES and coarse mesh (*i. e.* on Ω_Δ) for the multilevel method.

In each example the multilevel iteration is significantly less costly than the nested Newton-GMRES. The multilevel iteration requires only one fine mesh function evaluation for each iteration, while the Newton-GMRES requires at least 5 for the two examples.

3.1. Scalar Equations. In this section we report on computational experiments with the simple system (1.1)-(1.4). The parameters in the equation, similar to those used in [8] are

$$\epsilon = .1, \sigma = 2, \rho = .2, \beta = 10, \text{ and } L = 9. \tag{3.1}$$

The Newton-GMRES iteration performed somewhat better with a different formulation of the equations. We solve the equivalent system

$$
\begin{aligned}
&h(r) + 1 - \exp(-\beta u(r) + h(r) - c(r)) = 0 \\
&c(r) + 1 - \exp(-\beta u(r) + h(r) - c(r)) + \rho(c * h)(r) = 0.
\end{aligned}
\tag{3.2}
$$

So, letting N be defined by (2.11),

$$
\begin{aligned}
M(U)(r) &= \begin{pmatrix} h(r) + 1 - N(h,c)(r) \\ c(r) + 1 - N(h,c)(r) \end{pmatrix}, \text{ and} \\
K(U)(r) &= -\begin{pmatrix} 0 \\ \rho(c * h)(r) \end{pmatrix}.
\end{aligned}
\tag{3.3}
$$

Since the Fréchet derivative of M is simply a multiplication operator, we can compute it for each $r \in [0, L]$, *i. e.*

$$M'(U)V(r) = M'(U)(r)V(r)$$

for all $U, V \in X$. Here

$$M'(U)(r) = \begin{pmatrix} 1 - N(h,c)(r) & N(h,c)(r) \\ -N(h,c)(r) & 1 + N(h,c)(r). \end{pmatrix}.$$

The reformulation has the additional advantage that $det(M'(U))(r) = 1$ for all $r \in [0, L]$. Therefore, it is easy to analytically use $M'(U)^{-1}$ in the multilevel iteration (2.10).

$$M'(U)^{-1}(r) = \begin{pmatrix} 1 + N(h,c)(r) & -N(h,c)(r) \\ N(h,c)(r) & 1 - N(h,c)(r) \end{pmatrix}.$$

In Table 3.1 we show the results for both the Newton-GMRES and multilevel solvers.

TABLE 3.1
Iteration statistics: example 1

	Newton-GMRES		Multilevel	
N	R_δ	i_G	R_δ	i_G
33	5.96e–01	183	5.96e–01	183
65	2.74e–01	16	2.74e–01	27
129	9.79e–02	21	6.95e–02	23
257	2.49e–02	6	1.64e–02	26
513	5.62e–03	9	3.74e–03	25
1025	1.24e–03	16	8.58e–04	24
2049	3.05e–04	17	1.98e–04	26
4097	6.62e–05	17	4.93e–05	27

3.2. Two Integral Equations and One Constraint. In this example ρ is also unknown. The system is

$$\begin{aligned} h(r) &= \exp\left(-\beta u(r) + h(r) - c(r)\right) - 1 \\ h(r) &= c(r) + (c * (h\rho))(r) \\ \rho(r) &= A_1 \exp\left(-\beta u^{(1)}(r) + (c * \rho)(r)\right), \end{aligned} \tag{3.4}$$

where

$$u^{(1)}(r) = A_2 \cos(dr),$$

and u is defined by (1.5).

So, in the language of § 1, $P = 3$.

As in § 3.1, we rearrange the equations and solve the system as

$$\begin{aligned} h(r) &= \exp\left(-\beta u(r) + h(r) - c(r)\right) - 1 \\ c(r) &= \exp\left(-\beta u(r) + h(r) - c(r)\right) - 1 - (c * (h\rho))(r) \\ \rho(r) &= A_1 \exp\left(-\beta u^{(1)}(r) + (c * \rho)(r)\right) \end{aligned} \tag{3.5}$$

Hence, for $U = (h, c, \rho)$, and $N(h, c)$ defined by (2.11)

$$M(U)(r) = \begin{pmatrix} h(r) + 1 - N(h,c)(r) \\ \\ c(r) + 1 - N(h,c)(r) \\ \\ \rho(r) \end{pmatrix},$$

and

$$K(U) = -\begin{pmatrix} 0 \\ (c * (\rho h))(r) \\ \\ A_1 \exp\left(-\beta u^{(1)}(r) + (\rho * c)(r)\right) \end{pmatrix}$$

As in §3.1 M' and M' inverse are easy to compute.

$$M'(U) = \begin{pmatrix} 1 - N(h,c)(r) & N(h,c)(r) & 0 \\ -N(h,c)(r) & 1 + N(h,c)(r) & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

which has determinant 1 for all r. Hence

$$M'(U)^{-1} = \begin{pmatrix} 1 + N(h,c)(r) & -N(h,c)(r) & 0 \\ N(h,c)(r) & 1 - N(h,c)(r) & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

and the multiplications needed in (2.10) can be done very rapidly.

In the computations, we use the parameters

$$\beta = 10, \epsilon = .1, \sigma = 2, A_1 = .1, A_2 = .05; d = .2, \text{ and } L = 100.$$

This is a more difficult problem than the one in § 3.1, both because $L = 100$, which will require more mesh points, and because ρ is also an unknown. A coarse mesh of 65 points was needed for Newton-GMRES and one of 257 for the multilevel method. While these imply larger values of δ than the ones needed in § 3.1, the number of unknowns is much larger. In Table 3.2 we compare the two methods. The first three rows are the same because the multilevel method is only applied when $N \geq 513$.

TABLE 3.2
Iteration statistics: example 2

	Newton-GMRES		Multilevel	
N	R_δ	i_G	R_δ	i_G
65	1.59e–01	83	1.59e–01	83
129	2.53e–01	53	2.53e–01	53
257	1.11e–01	24	1.11e–01	24
513	6.36e–02	40	6.36e–02	56
1025	2.50e–02	27	1.63e–02	55
2049	6.92e–03	10	5.61e–03	55
4097	1.70e–03	19	1.30e–03	61
8193	3.67e–04	19	3.81e–04	53

REFERENCES

[1] K. E. ATKINSON, *Iterative variants of the Nyström method for the numerical solution of integral equations*, Numer. Math., 22 (1973), pp. 17–31.

[2] M. J. BOOTH, A. G. SCHLIPER, L. E. SCALES, AND A. D. J. HAYMET, *Efficient solution of liquid state integral equations using the newton-gmres algorithm*, Computer Physics Communications, 119 (1999), pp. 122–134.

[3] H. BRAKHAGE, *Über die numerische Behandlung von Integralgleichungen nach der Quadraturformelmethode*, Numer. Math., 2 (1960), pp. 183–196.

[4] W. L. BRIGGS, V. E. HENSON, AND S. MCCORMICK, *A Multigrid Tutorial, 2nd edition*, Society for Industrial and Applied Mathematics, Philadelphia, 2000.

[5] S. L. CAMPBELL, I. C. F. IPSEN, C. T. KELLEY, AND C. D. MEYER, *GMRES and the minimal polynomial*, BIT, 36 (1996), pp. 664–675.

[6] S. L. CAMPBELL, I. C. F. IPSEN, C. T. KELLEY, C. D. MEYER, AND Z. Q. XUE, *Convergence estimates for solution of integral equations with GMRES*, Journal of Integral Equations and Applications, 8 (1996), pp. 19–34.

[7] Z.-M. CHEN, O. A. KARIM, AND B. M. PETTITT, *A theory of the interionic structure of graphite intercalation synthetic metals: Variations with respect to interactions and state*, J. Chem. Phys, 89 (1988), pp. 1042–1048.

[8] Z.-M. CHEN AND B. M. PETTITT, *Non-isotropic solution of an OZ equation: matrix methods for integral equations*, Computer Physics Communications, 85 (1995), pp. 239–250.

[9] R. DEMBO, S. EISENSTAT, AND T. STEIHAUG, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[10] J. P. HANSEN AND I. R. MCDONALD, *Theory of Simple Liquids*, Academic Press Inc., London, second ed., 1986.

[11] L. KANTOROVICH AND G. AKILOV, *Functional Analysis*, Pergamon Press, New York, second ed., 1982.

[12] C. T. KELLEY, *Solving nonlinear equations with Newton's method*. Unpublished book manuscript.
[13] ——, *A fast multilevel algorithm for integral equations*, SIAM J. Numer. Anal., 32 (1995), pp. 501–513.
[14] ——, *Iterative Methods for Linear and Nonlinear Equations*, no. 16 in Frontiers in Applied Mathematics, SIAM, Philadelphia, 1995.
[15] L. S. ORNSTEIN AND F. ZERNIKE, *Accidental deviations of density and opalescence at the critical point of a single substance*, Proc. K. Ned. Akad. Wet., 17 (1914), pp. 793–806.
[16] J. PERKUS AND G. YEVICK, *Approximate integral equations*, Phys. Rev., 110 (1958), pp. 1–15.
[17] M. PERNICE AND H. F. WALKER, *NITSOL: A Newton iterative solver for nonlinear systems*, SIAM J. Sci. Comput., 19 (1998), pp. 302–318.
[18] B. M. PETTITT AND M. KARPLUS, *The potential mean force between polyatomic molecules in polar solvents*, J. Chem. Phys, 83 (1985), pp. 781–789.
[19] Y. SAAD AND M. SCHULTZ, *GMRES a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comp., 7 (1986), pp. 856–869.
[20] G. STELL, *Integral equations*, Physica, 29 (1963), pp. 517–527.
[21] J. VAN LEEUWEN, J. GOENEVELD, AND J. D. BOER, *Hypernetted chain equation*, Physica, 25 (1959), pp. 792–804.