Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Journal of Computational Physics 197 (2004) 491–501

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

A fast solver for the Ornstein–Zernike equations

C.T. Kelley [a,*], B. Montgomery Pettitt [b]

[a] Department of Mathematics, Center for Research in Scientific Computation, North Carolina State University, Raleigh, NC 27695-8205, USA

[b] Department of Chemistry, University of Houston, 136 Fleming Building, Houston, TX 77204-5003, USA

Received 18 September 2003; received in revised form 1 December 2003; accepted 8 December 2003
Available online 22 January 2004

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. The problems chosen are related to simple short ranged fluids with continuous potentials. Speedups over traditional methods for a given accuracy are reported. The new multilevel method is roughly six times faster than Newton-GMRES and 40 times faster than Picard.

1. Introduction

In this paper, we propose a fast multilevel 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 [2,22], 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 range of fluid 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 Percus–Yevick equation [16] and the hyper netted chain equation [21]. The hypernetted chain equation on which we have concentrated is transcendentally nonlinear and includes the Percus–Yevick terms. The OZ and HNC equations then

[*] Corresponding author.
E-mail addresses: tim_kelley@ncsu.edu (C.T. Kelley), pettitt@uh.edu (B.M. Pettitt).

0021-9991/$ - see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2003.12.006

provide essentially two equations and two unknowns and when convenient the HNC equation 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 without 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 a new multilevel approach for solving the OZ equations and apply that approach to two examples. The multilevel method is based on the enhanced version of the Atkinson–Brakhage [1,3] method from [12]. The method of [12] will compute the solution to the accuracy of truncation error in roughly twice the cost of an evaluation of the nonlinear function on the finest mesh.

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

$$h(r) = c(r) + \rho(h * c)(r), \tag{1}$$

where

$$(h * c)(r) = \int c(\|\mathbf{r} - \mathbf{r}'\|)h(\|\mathbf{r}'\|)\,\mathrm{d}\mathbf{r}' \tag{2}$$

and the integral is over R^3. For this example, we take the unknown functions h and c to be radially symmetric, i.e. functions only of the distance $r = \|\mathbf{r}\|$ of $\mathbf{r}$ from the origin, which implies that the convolution $h * c$ is as well.

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 on fluids which provides a connection for this theory to physics.

A naive approach to evaluation of $(h * c)$, a function of the scalar variable r, would be to evaluate the three-dimensional integral in (2). However, the convolution $h * c$ can be computed with only one-dimensional integrals using the spherical-Bessel transform. If h decays sufficiently rapidly, we define

$$\hat{h}(k) = \mathscr{H}(h)(k) = 4\pi \int_0^\infty \frac{\sin(kr)}{kr} h(r) r^2\,\mathrm{d}r$$

and

$$h(r) = \mathscr{H}^{-1}(\hat{h})(r) = \frac{1}{2\pi^2} \int_0^\infty \frac{\sin(kr)}{kr} \hat{h}(k) k^2\,\mathrm{d}k.$$

We compute $h * c$ by discretizing the formula

$$h * c = \mathscr{H}^{-1}(\hat{h}\hat{c}), \tag{3}$$

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.

$$\exp(-\beta u(r) + h(r) - c(r)) - h(r) - 1 = 0 \quad \text{for all } 0 \leqslant r \leqslant \infty. \tag{4}$$

The unknowns are $h, c \in C[0, \infty]$. We truncate the interval to $[0, L]$, for some $L < \infty$, for computational convenience, and consider $h, c \in C[0, L]$ for $L < \infty$. Here, $C[0, L]$ is the space of continuous real-valued functions on the interval $[0, L]$.

In (4), u is the pair potential between particles. Here, we will take the usual Lennard-Jones potential as typical of continuous, short ranged potentials

$$u(r) = 4\epsilon\left(\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right). \tag{5}$$

In (4) and (5), β, ϵ, and σ are parameters. In particular, β is the inverse of the product of absolute temperature and Boltzmann's constant, ϵ is the well depth of the potential, and σ determines the size of the particles, essentially the diameter. Throughout this paper we assume that the potential is continuous. Our convergence results require continuity. We have not considered potentials and correlations with discontinuities and or discontinuous derivatives, such as hard-sphere potentials.

In this paper, we formulate (1)–(4) as a compact fixed point problem for c. By this we mean an equation of the form

$$u = \mathscr{K}(u), \tag{6}$$

where $\mathscr{K}'(u)$, the Fréchet derivative of $\mathscr{K}$ is a compact operator on $X = C[0, L])$. Here, $\mathscr{K}$ is computed in stages. Given c we compute h by solving

$$h - \rho c * h = c. \tag{7}$$

We can solve (7) efficiently using the spherical-Bessel transform and (3) by

$$h = \mathscr{H}^{-1}\left(\frac{\hat{c}}{1 - \rho\hat{c}}\right). \tag{8}$$

Then, we use (4) to define

$$\mathscr{K}(c) = \exp(-\beta u - (h - c)) - (h - c) - 1. \tag{9}$$

Depending on the context, we will express (6) as a fixed point problem or a nonlinear equation

$$\mathscr{F}(u) = u - \mathscr{K}(u) = 0. \tag{10}$$

Eqs. (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. Our approach to discretizaions and intergrid transfers will, in principal, solve these larger systems. We will explore this in future work. In this paper, we focus entirely on the simple system (1)–(4).

2. Algorithms

We begin with a discussion of the discretization and intergrid transfers. We then discuss three solution approaches, Picard iteration, Newton-GMRES, and the new multilevel method. Each of the three solvers, when implemented efficiently, will use at least the coarse-to-fine intergrid transfer.

2.1. Discretization

The notation is complicated by the need to refer to the mesh size for both the approximate solutions and the discretized nonlinear equations.

We will approximate c by piecewise linear functions. The approximating space is V_δ 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. Since the approximation to the spherical-Bessel transform (13) is second-order accurate, and the discrete inverse is the inverse of the discrete transform, the approximation of the convolution integral, as we will define it in (14) is second-order accurate.

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

$$(P_\delta u)(r) = \int_0^L p_\delta(r,r')u(r')\,\mathrm{d}r' = \sum_{i,j=1}^N \phi_i(r) l_{ij} \int_0^L u(r')\phi_j(r')\,\mathrm{d}r', \tag{11}$$

where $\{\phi_i\}$ are the nodal PL basis functions and the coefficients $\{l_{ij}\}$ are defined through the equations

$$\delta_{ij} = \sum_{k=1}^N l_{ik} \int_0^L \phi_k(r')\phi_j(r')\,\mathrm{d}r', \tag{12}$$

where δ_{ij} is the Kronecker delta. Because of (11) and (12), $P_\delta* = P_\delta^2 = P_\delta$, making P_δ the projection onto V_δ. 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 $\delta_k = \pi/L$ (so $\delta_k\delta = \pi/(N-1)$). We define, for $2 \leqslant j \leqslant N-1$

$$\begin{aligned}\hat{v}_j = \mathscr{H}(v)(k_j) &= \frac{4\pi\delta^2}{(j-1)\delta_k} \sum_{i=2}^{N-1} (i-1)v_i \sin((i-1)(j-1)\delta_k\delta) \\ &= \frac{4\pi\delta^3(N-1)}{j-1} \sum_{i=2}^{N-1} (i-1)v_i \sin((i-1)(j-1)\pi/(N-1)),\end{aligned} \tag{13}$$

where $v_i = v(r_i)$. Then, for $2 \leqslant i \leqslant N-1$,

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

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

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

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

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

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

$$Q_\delta(u,v)(r) = \sum_{i=1}^N \phi_i(r)(u*v)_i, \tag{14}$$

where $u_i = u(r_i^\delta)$, $v_i = v(r_i^\delta)$. The sum in (14) can be evaluated with a fast Fourier transform in $O(N \log N)$ floating operations. We define $\mathscr{K}_\delta$ in terms of its values at the nodes. For $c \in C([0, L])$ and $\delta > 0$, we let $c_i = c(r_i^\delta)$ and solve for the nodal values $\{h_i\}$ of $h \in V^\delta$. The fully discrete equation for the nodal values is

$$h_i - \rho(c * h)_i = c_i, \quad 1 \leqslant i \leqslant N, \tag{15}$$

which, analogous to (7) and (8), we can do with the transformed equation

$$h_i = \mathscr{H}^{-1}\left(\frac{(\hat{c})_i}{1 - \rho(\hat{c})_i}\right). \tag{16}$$

Then $\mathscr{K}_\delta(c) \in V^\delta$ is defined using the algebraic constraint

$$\exp(-\beta u_i + h_i - c_i) - h_i - 1 = 0, \quad 1 \leqslant i \leqslant N, \tag{17}$$

as

$$\mathscr{K}_\delta(c) = \exp(-\beta u^\delta - (h - c)) - (h - c) - 1, \tag{18}$$

where u^δ is the piecewise linear interpolant of u.

We approximate the solution of (1)–(4) with the function $c^\delta \in V_\delta$ that satisfies

$$c^\delta - \mathscr{K}_\delta(c^\delta) = \mathscr{F}_\delta(c^\delta) = 0. \tag{19}$$

Note that (19) is equivalent to a fully discrete system (15)–(17) for the values of $c^\delta \in V_\delta$ at the nodes, which is the finite-dimensional system that one solves numerically.

The Kantorovich theorem [11], the uniform Lipschitz continuity of $\mathscr{F}'_\delta$, and the fact that the discretized convolution is second-order accurate, imply that if $c*$ is sufficiently smooth and $\mathscr{F}'(c*)$ is nonsingular, then (19) has a solution c^δ for all δ sufficiently small, and that

- $c^\delta = c * + O(\delta^2)$,
- $\mathscr{F}'_\delta(c^\delta)$ is nonsingular for δ sufficiently small.

Moreover, for all $c \in C([0, 1])$, $\mathscr{K}'_\delta(c) \to \mathscr{K}'(c)$ in the operator norm on $C([0, 1])$. Therefore, $\mathscr{K}'(c^\delta) \to \mathscr{K}'(c*)$ 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 V_{δ_s} and target V_{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. We identify $C([0, 1])$ 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 $\mathscr{F}(c) = 0$ on a sequence of meshes, terminating with a solution at a target, finest mesh.

The nonlinear solve inside the loop of Algorithm **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 each mesh is computed. The coarse mesh has width Δ and the mesh is refined by halving the width. The ultimate mesh width is $\delta = 2^{-l_{\max}}\Delta$.

Algorithm 1. Generic nested iteration.

```
  nest_generic($c, \mathscr{F}, \Delta, l_{\max}$)
  $l \leftarrow 0$; $\delta \leftarrow \Delta$; $c_0^{\Delta} \leftarrow c$
  Solve $\mathscr{F}_{\delta}(c^{\Delta}) = 0$ to high accuracy with $c_0^{\Delta}$ as the initial iterate.
  while $l < l_{\max}$ do
    $\delta \leftarrow \delta/2$; $l \leftarrow l + 1$
    Interpolate $c^{2\delta}$ to $\Omega_{\delta}$ to obtain $c_0^{\delta}$
    Solve $\mathscr{F}_{\delta}^{0}(c^{\delta}) = 0$ to reasonable accuracy with $c_0^{\delta}$ 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 10, for example, on the subsequent, finer grids. This took a single nonlinear iteration in the computations reported in Section 3.

2.3. Nested Picard iteration

The most direct approach, and most common in the literature, is successive substitution or Picard iteration. The iteration is

$$c_{n+1} = \mathscr{K}(c_n). \tag{20}$$

Picard iteration can be rapidly convergent. In fact, the fast multilevel method in Section 2.5 can the thought of as a type of Picard iteration, but not for the original map.

For the problems considered in this paper, Picard iteration is slow. A nested iteration will improve its performance, and we include a nested Picard iteration in the comparisons in Section 3.

2.4. 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 c_c,

$$\mathscr{F}'(c_c)s = -\mathscr{F}(c_c), \tag{21}$$

with a GMRES [19] iteration. The termination criterion for the linear iteration is the standard [9,13] inexact Newton condition

$$\|\mathscr{F}'(c_c)s + \mathscr{F}(c_c)\| \leqslant \eta\|\mathscr{F}(c_c)\|, \tag{22}$$

where η is a parameter, which we set to $1/10$ in all of the computations reported in Section 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 $\mathscr{K}'(c^{\delta})$ to $\mathscr{K}'(c*)$ in the operator norm [5,6]. This means that then the number of GMRES iterations needed to satisfy (22) from an initial iterate of $s = 0$ is independent of the level δ of discretization.

Algorithm 2. Newton-GMRES

```
  newton-gmres($c, \mathscr{F}, \tau_a, \tau_r, \eta$)
   Evaluate $\mathscr{F}(c)$; $\tau \leftarrow \tau_r\|\mathscr{F}(c)\| + \tau_a$.
   while $\|\mathscr{F}(c)\| > \tau$ do
     Solve the linear system $\mathscr{F}'(c)s = -\mathscr{F}(c)$
     with GMRES and terminate when (22) holds.
     $c \leftarrow c + s$
     Evaluate $\mathscr{F}(c)$.
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 10. 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 cost of a function evaluation on Ω_δ. For the examples considered here

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

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

$$(C_G + 1)\sum_{l=0}^{l_{\max}} C_F(2^l/\Delta \log(2^l/\Delta) \leqslant 2(C_G + 1)C_F(\delta_{l_{\max}}). \tag{23}$$

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.

2.5. Multilevel iteration

In this section, we describe an approximate Newton method that uses an extension of the method in [12] to approximate the Newton step. The idea is to use a coarse mesh approximate inverse of $\mathscr{F}'(c)$ and base the computation of the approximate Newton step on that approximate inverse.

The approach in [12], which we follow in this section, is a degenerate kernel approach for solving linear second kind Fredholm integral equations of the form

$$(I - K)u(r) = u(r) - \int_0^L k(r, r')u(r')\,\mathrm{d}r' = f(r). \tag{24}$$

The approach is to build an approximate inverse of $I - K$ and use that to solve the discretization of (24) on Ω_δ. We approximate $\mathscr{K}$ by $\mathscr{K}_\Delta$, where

$$(K_\Delta u)(r) = \int_0^L k_\Delta(r, r')u(r')\,\mathrm{d}r' \approx (Ku)(r) = \int_0^L k(r, r')u(r')\,\mathrm{d}r',$$

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" (or "rooftop 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 (24) 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 w, one solves

$$s - K_\Delta s = w \tag{25}$$

as follows.

Let P_Δ be the L^2 projection onto V^δ. To solve (25) we first find $s^\Delta = P_\Delta s$ as follows. Since s^Δ is uniquely determined by its values at the coarse mesh nodes, we can solve a finite-dimensional fully discrete system for

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

where the superscript FD indicates that the system is fully discrete, N_Δ is the number of points in the coarse mesh, and s_i^{FD} denotes the ith component of $s^{\mathrm{FD}} \in R^{N_\Delta}$. The fully discrete coarse mesh equations for s^{FD} are

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

which one can solve with GMRES, for example. Since

$$K_\Delta s = \mathscr{K}_\Delta P_\Delta s,$$

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

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

The nested iteration form of this algorithm is Algorithm **nest_richardson**.

For nonlinear fixed point problems, we apply the preconditioned Richardson iteration to the linear equation for the Newton step. Table 1 indicates that the number of coarse mesh iterations remains low, as predicted the theory in [12]. More importantly, for Δ sufficiently small, a single nonlinear iteration will suffice for each δ and one can refine the mesh after each Newton iteration [12]. This means that only one fine-mesh function evaluation is needed at each level.

Algorithm 3. Nested Richardson iteration for $u - Ku = f$

```
  nest_richardson($u, K, \Delta, l_{\max}$)
  $l \leftarrow 0$; $\delta \leftarrow \Delta$
  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 $c^{2\delta}$ to $\Omega_\delta$ to obtain $c^\Delta$.
```

Table 1
Iteration statistics: example 1

N	Picard		Newton-GMRES		Multilevel	
	R_δ	i_G^δ	R_δ	i_G^δ	R_δ	i_G^Δ
65	3.5900 e + 00	650	3.5900 e + 00	85	3.5900 e + 00	85
129	1.3696 e – 01	11	1.3696 e – 01	4	1.3696 e – 01	8
257	2.0031 e – 02	3	2.9413 e – 02	5	4.1900 e – 02	7
513	4.8144 e – 03	9	6.9937 e – 03	5	9.4120 e – 03	7
1025	2.3568 e – 03	14	1.5400 e – 03	5	2.0205 e – 03	7
2049	3.6543 e – 04	15	3.5596 e – 04	7	4.6015 e – 04	8
4097	8.2396 e – 05	22	8.4570 e – 05	5	1.0831 e – 04	8
8193	2.2253 e – 05	38	2.0784 e – 05	7	2.6411 e – 05	8
16,385	4.0075 e – 06	48	5.2729 e – 06	8	6.5042 e – 06	8
32,769	9.7738 e – 07	32	1.2263 e – 06	5	1.6132 e – 06	8
65,537	2.3869 e – 07	44	3.0647 e – 07	7	4.0169 e – 07	8

```
    $w \leftarrow f - (I - K_\delta)u^\Delta$.
    $s \leftarrow (I - K_\Delta)^{-1}w$.
    $u^\delta \leftarrow u^\Delta - s$
end while
```

Algorithm **nest_multilevel** is the nonlinear multilevel algorithm for (10).

Algorithm 4. Nested multilevel algorithm for $\mathscr{F}(c) = c - \mathscr{K}(c) = 0$

```
    nest_multilevel($c, \mathscr{F}, \Delta, l_{\max}$)
    $l \leftarrow 0$; $\delta \leftarrow \Delta \cdot c_0^\Delta \leftarrow c$
    Solve $\mathscr{F}_\delta^0(c^\Delta) = 0$ to high accuracy
    with $c_0^\Delta$ as the initial iterate.
    while $l < l_{\max}$ do
      $\delta \leftarrow \delta/2; l \rightarrow l + 1$
      $c_0^\delta \leftarrow I_{2\delta}^\delta c^{2\delta}$.
      $c^\delta \leftarrow c^\delta - (I - \mathscr{K}'_\Delta(c^\Delta))^{-1}\mathscr{F}_\delta(c_0^\delta)$
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

$$2C_F(\delta_{l_{\max}}).$$

3. Numerical results

In this section, we consider an example and compare the performance of the nested Picard iteration and the nested Newton-GMRES method with the multilevel method proposed in Section 2. The initial iterate on the coarse mesh was $c_0 = \mathscr{K}(0)$ for each method.

The computations were done in MATLAB 6.5, using the Newton-GMRES code from [14] for the nested Newton-GMRES results and the coarse mesh computations for the multilevel results. The GMRES code from [13] was used to solve the coarse mesh linear systems in the multilevel method.

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

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

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 $\delta \to \delta/2$. We also tabulate the number of GMRES iterations i_G for each nonlinear iteration. This refers to fine-mesh iterations i_G^δ (i.e., on Ω_δ) for Newton-GMRES and coarse mesh iterations i_G^Δ (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.

We report on computational experiments with the fixed point formulation (6) of the system (1)–(4). The parameters in the equation, similar to those used in [8] are

$$\epsilon = 0.1 \text{ kcal/mol}, \quad \sigma = 2 \text{ Å}, \quad \rho = 0.2\#/\text{Å}^3, \quad \beta = 1/(kT) = 10, \quad L = 9. \tag{26}$$

In (26), # indicates that ρ is a number density, k is Boltzmann's constant, and T is temperature. This corresponds to a state in reduced units of $\rho* = 0.8$ and $T* = 1.0$. The coarse mesh had $N = 65$ points, making $\Delta = 1/64$. The finest mesh had $N = 8193$ points with a mesh width of 1/8192.

In Table 1, we show the results for the Picard, Newton-GMRES and multilevel solvers. The most significant things in the table are:

- The multilevel method is roughly six times faster than Newton-GMRES and 40 times faster than Picard.
- The performance of Picard iteration degrades as the mesh is refined.
- The accuracy of the discretization appears to be second order, since, for either method, $R_\delta/R_{2\delta} \approx 1/4$ for small δ.

Acknowledgements

The research of CTK was partially supported by National Science Foundation Grants DMS-0070641 and DMS-0209695. The research of BMP was supported in part by the NIH and the Robert A. Welch Foundation. The authors are grateful to Dr. Dimitry A. Tikhonov for his comments on an earlier version of the manuscript.

References

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

[2] M.J. Booth, A.G. Schliper, L.E. Scales, A.D.J. Haymet, Efficient solution of liquid state integral equations using the Newton-GMRES algorithm, Comput. Phys. Commun. 119 (1999) 122–134.

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

[4] W.L. Briggs, V.E. Henson, S. McCormick, A Multigrid Tutorial, second ed., Society for Industrial and Applied Mathematics, Philadelphia, 2000.
[5] S.L. Campbell, I.C.F. Ipsen, C.T. Kelley, C.D. Meyer, GMRES and the minimal polynomial, BIT 36 (1996) 664–675.
[6] S.L. Campbell, I.C.F. Ipsen, C.T. Kelley, C.D. Meyer, Z.Q. Xue, Convergence estimates for solution of integral equations with GMRES, J. Integral Equat. Appl. 8 (1996) 19–34.
[7] Z.-M. Chen, O.A. Karim, 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) 1042–1048.
[8] Z.-M. Chen, B.M. Pettitt, Non-isotropic solution of an OZ equation: matrix methods for integral equations, Comput. Phys. Commun. 85 (1995) 239–250.
[9] R.S. Dembo, S.C. Eisenstat, T. Steihaug, Inexact Newton methods, SIAM J. Numer. Anal. 19 (1982) 400–408.
[10] J.P. Hansen, I.R. McDonald, Theory of Simple Liquids, second ed., Academic Press, London, 1986.
[11] L.V. Kantorovich, G.P. Akilov, Functional Analysis, second ed., Pergamon Press, New York, 1982.
[12] C.T. Kelley, A fast multilevel algorithm for integral equations, SIAM J. Numer. Anal. 32 (1995) 501–513.
[13] C.T. Kelley, Iterative methods for linear and nonlinear equations, Frontiers in Applied Mathematics, vol. 16, SIAM, Philadelphia, 1995.
[14] C.T. Kelley, Solving nonlinear equations with Newton's method, Fundamentals of Algorithms, vol. 1, SIAM, Philadelphia, 2003.
[15] L.S. Ornstein, F. Zernike, Accidental deviations of density and opalescence at the critical point of a single substance, Proc. K. Ned. Akad. Wet. 17 (1914) 793–806.
[16] J.K. Perkus, G.J. Yevick, Approximate integral equations, Phys. Rev. 110 (1958) 1–15.
[17] M. Pernice, H.F. Walker, NITSOL: a Newton iterative solver for nonlinear systems, SIAM J. Sci. Comput. 19 (1998) 302–318.
[18] B.M. Pettitt, M. Karplus, The potential mean force between polyatomic molecules in polar solvents, J. Chem. Phys. 83 (1985) 781–789.
[19] Y. Saad, M.H. Schultz, GMRES a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 7 (1986) 856–869.
[20] G. Stell, Integral equations, Physica 29 (1963) 517–527.
[21] J.M.J. van Leeuwen, J. Goeneveld, J. De Boer, Hypernetted chain equation, Physica 25 (1959) 792–804.
[22] G. Zerah, An efficient Newton's method for the numerical solution of fluid integral equations, J. Compot. Phys. 61 (1985) 280–285.