# A Fast Continuation Method for the Ornstein-Zernike Equations

C. T. Kelley

Joint work with

Monte Pettitt, Yannis Kevrekidis, Liang Qiao

Department of Mathematics

Center for Research in Scientific Computation

North Carolina State University

Raleigh, North Carolina, USA

NCSU

November 2, 2004

Supported by NSF, ARO.

# Outline

- The Ornstein-Zernike (OZ) Equations
- Fast solvers for compact fixed point problems
  Application to OZ + uniqueness problems
- Path following: introduction
  Nonlinear solvers
  Pseudo-arclength continuation
- Multilevel method.
- Results

# OZ Equations: O-Z, 1914

Used to calculate probability distributions of atoms in fluid states. Unknowns are $h, c \in C[0,L]$.

- $h$: radial pair correlation function, observable
- $c$: direct correlation function, defined by IE

Integral Equation:

$$h(r) - c(r) - \rho(h * c)(r)$$

where

$$(h * c)(r) = \int_{R^3} c(\|\mathbf{r} - \mathbf{r}'\|) h(\|\mathbf{r}'\|) d\mathbf{r}'.$$

# Algebraic Closure Constraint

$$\exp(-\beta U(r) + h(r) - c(r)) - h(r) - 1 = 0.$$

where $u$ is the Lennard-Jones potential.

$$U(r) = 4\varepsilon \left( \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^{6} \right).$$

# Parameters

Data are parameters

- $\rho$: number density, sometimes unknown
- $\beta = 1/($absolute temperature $\times$ Boltzmann's constant$)$
- $\varepsilon$: well depth of the potential
- $\sigma$: determines size of the particles

# Discretization

- Uniform grid on $[0, L]$
- Trapezoid rule for integration
- Discrete Hankel transform for evaluation of integrals

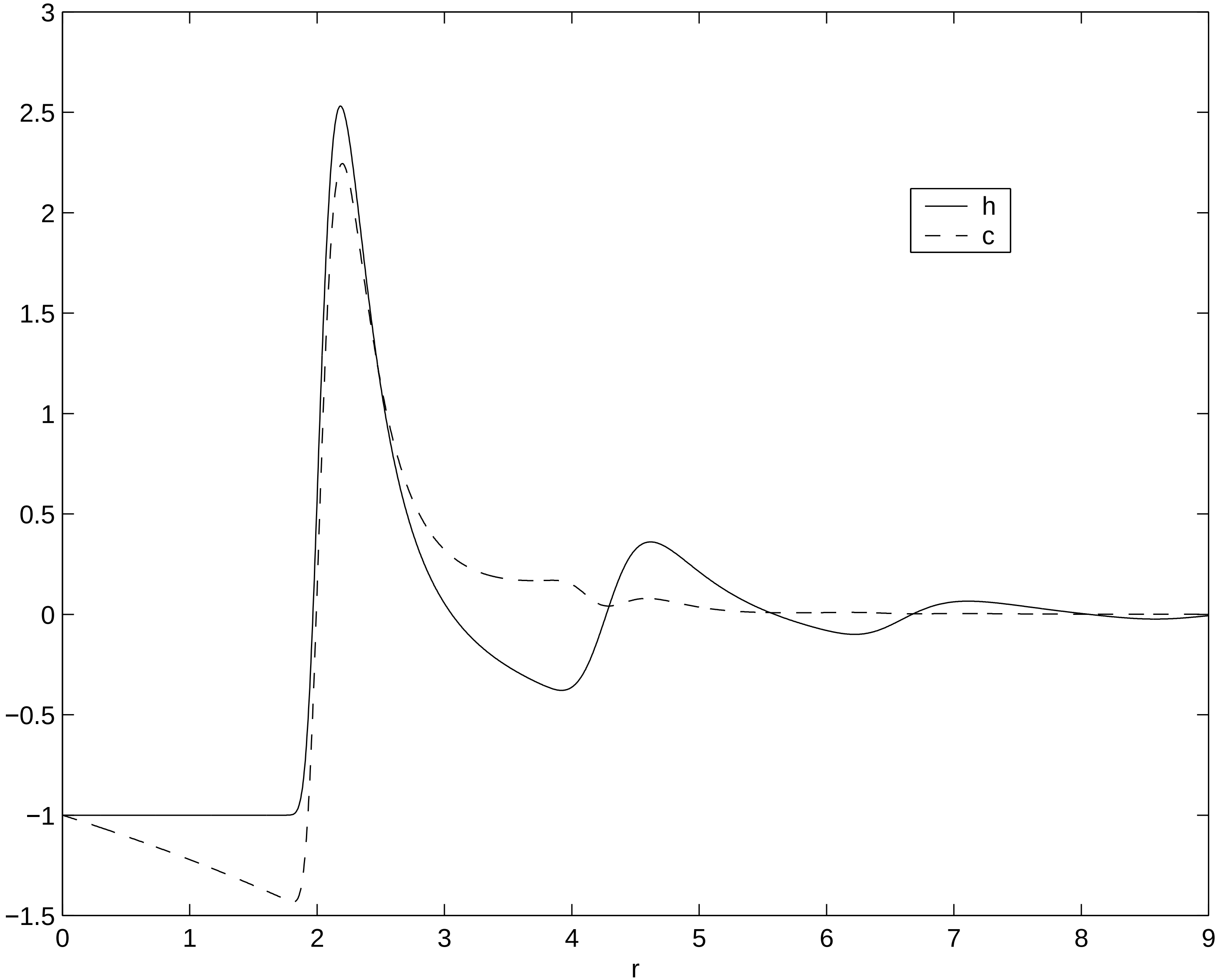
$$\mathscr{H}(h)(k) = 4\pi \int_0^\infty \frac{sin(kr)}{kr} h(r) r^2 dr$$

  and

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

- Fast evaluation via FFT

# **Solution:** $\rho = .2, \sigma = 2; \varepsilon = .1; \beta = 10; L = 9$



# Reduction to single equation

Let $g = h - c$, then the closure constraint expresses $c$ as a function of $g$.

$$c(r) = c(g(r)) = \exp(-\beta U(r) + g(r)) - g(r) - 1.$$

The integral equation is

$$h - \rho c * h = c.$$

Take Hankel transforms

$$\hat{h} - \rho \hat{h}\hat{c} = \hat{c},$$

and obtain $\hat{h} = \hat{c}/(1 - \rho\hat{c})$.

# $g \longrightarrow c \longrightarrow h$ leads to . . .

$$h = h(c(g)) = c(g) + \mathscr{K}(g).$$

Subtract $c$ and obtain a fixed point problem for $g$.

$$g = h(c(g)) - c(g) = \mathscr{K}(g).$$

$\mathscr{K}$ is a nonlinear integral operator with compact Fréchet derivative.

# Alternative: reduce to single equation in $c$

- $c \longrightarrow h(c)$ via solution of integral equation
- $h(c) - c = \mathscr{G}(c)$, $\mathscr{G}$ compact
- $\mathscr{K}(c) = \exp(-\beta U - \mathscr{G}(c)) - \mathscr{G}(c) - 1$

Compact fixed point problem:

$$c = \mathscr{K}(c)$$

# More General OZ Equations

Unknowns $h, c, \rho, \in C[0, L]$

$$
\begin{aligned}
h(r) &= \exp\left(-\beta U(r) + h(r) - c(r)\right) - 1 \\
h(r) &= c(r) + \int_0^r c(r - r')\rho(r')h(r')\,dr' \\
\rho(r) &= A_1 \exp\left(-\beta U(r) + \int_0^r \rho(r - r')c(r')\,dr'\right).
\end{aligned}
$$

Also matrix-valued unknowns.

# Compact Fixed Point Problems

We're worried about problems like

$$F(u) = u - \mathscr{K}(u) = 0, \text{ on a Banach space } X,$$

where

- $\mathscr{K} \in C^1_{LIP}(X)$.
- $\mathscr{K}' \in Com(X)$.
- Compactness will lead to fast solvers.

# How to exploit compactness

- Discretization

# How to exploit compactness

- Discretization
  - Almost every reasonable scheme works, but

# How to exploit compactness

- Discretization
    - Almost every reasonable scheme works, but
    - some approximations to $\mathscr{K}'$ converge in norm.

# How to exploit compactness

- Discretization
  - Almost every reasonable scheme works, but
  - some approximations to $\mathscr{K}'$ converge in norm.
- Solvers

# How to exploit compactness

- Discretization
  - Almost every reasonable scheme works, but
  - some approximations to $\mathscr{K}'$ converge in norm.
- Solvers
  - Krylov solvers need no preconditioning (in theory).

# How to exploit compactness

- Discretization
  - Almost every reasonable scheme works, but
  - some approximations to $\mathscr{K}'$ converge in norm.
- Solvers
  - Krylov solvers need no preconditioning (in theory).
  - Multilevel methods are easy to design.

# How to exploit compactness

- Discretization
  - Almost every reasonable scheme works, but
  - some approximations to $\mathscr{K}'$ converge in norm.
- Solvers
  - Krylov solvers need no preconditioning (in theory).
  - Multilevel methods are easy to design.
  - No smoothers are needed.

# How to exploit compactness

- Discretization
  - Almost every reasonable scheme works, but
  - some approximations to $\mathscr{K}'$ converge in norm.
- Solvers
  - Krylov solvers need no preconditioning (in theory).
  - Multilevel methods are easy to design.
  - No smoothers are needed.
- Fast evaluation ($O(N \log(N)$) is common.

# How to exploit compactness

- Discretization
  - Almost every reasonable scheme works, but
  - some approximations to $\mathscr{K}'$ converge in norm.
- Solvers
  - Krylov solvers need no preconditioning (in theory).
  - Multilevel methods are easy to design.
  - No smoothers are needed.
- Fast evaluation ($O(N \log(N))$) is common.
- Newton-Krylov, Newton-MG nonlinear solvers work with no surprises (most of the time).

# World's Easiest Example

Linear Fredholm equation:

$$(I-K)u(x) = u(x) - \int_0^1 k(x,y)u(y)\,dy = f(x),$$

$f \in X = C[0,1]$, $k \in C([0,1]\times[0,1])$
Approximating space: $V_h =$ span $\{\phi_i\}$
$P_h$ is a projection onto $V_h$, and we seek $u^h \in V_h$.

$$u^h(x) - K_h u^h(x) = u^h(x) - \int_0^1 k_h(x,y)u^h(y)\,dy = P_h f(x)$$

where, $k_h(x,y) = \sum_{i,j=1}^{N_h} k(x_i,x_j)\phi_i(x)\phi_j(y)$

# Properties of Discretization

- $K_h$ operates on the function space

# Properties of Discretization

- $K_h$ operates on the function space
- $K_h \longrightarrow K$ in the operator norm

# Properties of Discretization

- $K_h$ operates on the function space
- $K_h \rightarrow K$ in the operator norm
- Lots of flexibility in $P_h$
  Strong convergence to $I$ is all you need.

# Properties of Discretization

- $K_h$ operates on the function space
- $K_h \to K$ in the operator norm
- Lots of flexibility in $P_h$
  Strong convergence to $I$ is all you need.
- If $I - K$ is nonsingular, then

$$u^h = (I - K_h)^{-1} P_h f \to (I - K)^{-1} f$$

Solve finite dimensional system for nodal values.

# Properties of Discretization

- $K_h$ operates on the function space
- $K_h \to K$ in the operator norm
- Lots of flexibility in $P_h$
  Strong convergence to $I$ is all you need.
- If $I - K$ is nonsingular, then

$$u^h = (I - K_h)^{-1} P_h f \to (I - K)^{-1} f$$

  Solve finite dimensional system for nodal values.
- Other choices of $K_h$ are possible
  Standard quadrature rule + fine-to-coarse by averaging

# Nystrom interpolation

- Solve $\tilde{u}^h - K_h \tilde{u}^h = f$ rather than $u^h - K_h u^h = P_h f$.
- Multiply by $P_h$ and use $K_h = K_h P_h = P_h K_h$ to get

$$(P_h \tilde{u}) - P_h K_h (P_h \tilde{u}) = P_h f.$$

  Finite dimensional system.
  Solve for $u^h = P_h \tilde{u}^h$.
- $\tilde{u}^h = f + K_h u^h$

# Performance of GMRES

Avoid the $O(N_h^3)$ cost of a direct solver, and compute

$$u^h = (I - K_h)^{-1} P_h f = \sum_{i=1}^{N_h} u_i^h \phi_i \in V_h.$$

with GMRES.

- Continuous problem: superlinear convergence
- Discrete problem: mesh independent performance
- Cost: One $K_h v$ evaluation/linear iteration
  Think $N_h \log N_h$ work if done slickly.

Nested iteration (aka grid sequencing) is a good idea.

# Multilevel Method; K 95

Since $K_h \to K$ in the operator norm,

- $(I - K_H)$ ($h << H$) might be a good preconditioner for GMRES

# Multilevel Method; K 95

Since $K_h \to K$ in the operator norm,

- $(I - K_H)$ ($h << H$) might be a good preconditioner for GMRES
- Richardson iteration is a better idea thanks to LOW STORAGE.

$$u \leftarrow u - (I - K_H)^{-1}((I - K_h)u - P_h f)$$

# Multilevel Method; K 95

Since $K_h \to K$ in the operator norm,

- $(I - K_H)$ ($h << H$) might be a good preconditioner for GMRES
- Richardson iteration is a better idea thanks to LOW STORAGE.

$$u \leftarrow u - (I - K_H)^{-1}((I - K_h)u - P_h f)$$

- $H$ suff small implies

# Multilevel Method; K 95

Since $K_h \to K$ in the operator norm,

- $(I - K_H)$ ($h << H$) might be a good preconditioner for GMRES
- Richardson iteration is a better idea thanks to LOW STORAGE.

$$u \leftarrow u - (I - K_H)^{-1}((I - K_h)u - P_h f)$$

- $H$ suff small implies
  - Krylovs independent of $H$.

# Multilevel Method; K 95

Since $K_h \to K$ in the operator norm,

- $(I - K_H)$ ($h << H$) might be a good preconditioner for GMRES
- Richardson iteration is a better idea thanks to LOW STORAGE.

$$u \leftarrow u - (I - K_H)^{-1}((I - K_h)u - P_h f)$$

- $H$ suff small implies
  - Krylovs independent of $H$.
  - One iteration/level suffices.

# Nonlinear Problems

Generalization to the nonlinear case is easy,

$$u \leftarrow u - (I - \mathscr{K}_H'(u^H))^{-1} F_h(u)$$

if you're careful about the fine-to-coarse transfer.
If coarse mesh suff fine,

- Krylovs/Newton independent of $H$
- one Newton/level suffices.

# Nested Iteration: Bottom up; K 95

$h = H$, $i = 0$
Solve $F_H(u^H) = 0$ to high accuracy.
$u \leftarrow u^H$
**for** $i = 1, \ldots m$ **do**
  $h \leftarrow h/2$
  $u \leftarrow u - (I - \mathscr{K}_H'(u^H))^{-1} F_h(u)$
**end for**

- All the linear solver work is on the coarse mesh.
- Only two grids $H$ and $h$ active at any time.
- Cost of solve to truncation error:
  $< 3$ fine mesh evals, depending on cost of $\mathscr{K}_h$

# Works great for OZ! K., Pettit 2004

Iteration statistics for three nested iterations

- Multilevel, Newton-GMRES, Picard
- Formulation in $c$:
  $c \rightarrow h(c)$ via integral equation
  $c = \mathscr{K}(c)$ via constraint
- Tabulate:
  $i_G^f$ = fine mesh GMRES/Newton (average)
  $i_G^c$ = coarse GMRES/Newton (average)
  incoming nonlinear residual $R_h$ ($R_{2h} \approx 4R_h$)

# Iteration Statistics: h = 1/(N-1)

| | Picard | | Newton-GMRES | | Multilevel | |
|---|---|---|---|---|---|---|
| $N$ | $R_\delta$ | $i_G^f$ | $R_\delta$ | $i_G^f$ | $R_\delta$ | $i_G^c$ |
| 65 | 3.5900e+00 | 650 | 3.5900e+00 | 85 | 3.5900e+00 | 85 |
| 129 | 1.3696e-01 | 11 | 1.3696e-01 | 4 | 1.3696e-01 | 8 |
| 257 | 2.0031e-02 | 3 | 2.9413e-02 | 5 | 4.1900e-02 | 7 |
| 513 | 4.8144e-03 | 9 | 6.9937e-03 | 5 | 9.4120e-03 | 7 |
| 1025 | 2.3568e-03 | 14 | 1.5400e-03 | 5 | 2.0205e-03 | 7 |
| 2049 | 3.6543e-04 | 15 | 3.5596e-04 | 7 | 4.6015e-04 | 8 |
| 4097 | 8.2396e-05 | 22 | 8.4570e-05 | 5 | 1.0831e-04 | 8 |
| 8193 | 2.2253e-05 | 38 | 2.0784e-05 | 7 | 2.6411e-05 | 8 |
| 16385 | 4.0075e-06 | 48 | 5.2729e-06 | 8 | 6.5042e-06 | 8 |
| 32769 | 9.7738e-07 | 32 | 1.2263e-06 | 5 | 1.6132e-06 | 8 |
| 65537 | 2.3869e-07 | 44 | 3.0647e-07 | 7 | 4.0169e-07 | 8 |

# Mission Accomplished?

- We found two solutions;

# Mission Accomplished?

- We found two solutions; one was wrong.

# Mission Accomplished?

- We found two solutions; one was wrong.
- Monte knew which one was correct.

# Mission Accomplished?

- We found two solutions; one was wrong.
- Monte knew which one was correct. Tim did not.

# Mission Accomplished?

- We found two solutions; one was wrong.
- Monte knew which one was correct. Tim did not.
- One can get one or the other by
    - varying the initial iterate,
    - varying the initial grid, or
    - varying the details of the algorithm,

# Mission Accomplished?

- We found two solutions; one was wrong.
- Monte knew which one was correct. Tim did not.
- One can get one or the other by
  - varying the initial iterate,
  - varying the initial grid, or
  - varying the details of the algorithm,
- which motivates a parametric $(\sigma, \varepsilon, \rho \ldots)$ study of the OZ equations.

# Path Following

$F : X \times [a,b]$, $F$ smooth, $X$ a Banach space.
Objective: Solve $F(u,\lambda) = 0$ for $\lambda \in [a,b]$
Obvious approach:

Set $\lambda = a$, solve $F(u,\lambda) = 0$ with
Newton-(MG, GMRES, ...) to obtain $u_0 = u(\lambda)$.
**while** $\lambda < b$ **do**
Set $\lambda = \lambda + d\lambda$.
Solve $F(u,\lambda) = 0$ with $u_0$ as the initial iterate.
$u_0 \leftarrow u(\lambda)$
**end while**

The implicit function theorem says: You will not find two solutions with identical parameter values this way.

# What's the problem?

- Multiple solutions, hysteresis
- No solutions

# What's the problem?

- Multiple solutions, hysteresis
- No solutions

A fix: Pseudo-arclength continuation.
Set $x = (u, \lambda)$ and solve $G(x, s) = 0$, where, for example

$$G(x,s) = \begin{pmatrix} F \\ N \end{pmatrix} = \begin{pmatrix} F(u(s), \lambda(s)) \\ \dot{u}^T(u - u_0) + \dot{\lambda}^T(\lambda - \lambda_0) - (s - s_0) \end{pmatrix}.$$

# What's the problem?

- Multiple solutions, hysteresis
- No solutions

A fix: Pseudo-arclength continuation.
Set $x = (u, \lambda)$ and solve $G(x, s) = 0$, where, for example

$$G(x,s) = \begin{pmatrix} F \\ N \end{pmatrix} = \begin{pmatrix} F(u(s), \lambda(s)) \\ \dot{u}^T(u - u_0) + \dot{\lambda}^T(\lambda - \lambda_0) - (s - s_0) \end{pmatrix}.$$

$s$ is an artificial "arclength" parameter.
$u_0$ and $\lambda_0$ are from the previous step.
$\dot{u} \approx du/ds$ and $\dot{\lambda} \approx d\lambda/ds$,

(say by differences using $s_0$ and $s_{-1}$).

# Simple Folds

We follow solution paths $\{x(s)\}$.
Assume that $F$ is smooth and

- $G_x$ is nonsingular (not always true)
  So implicit function theorem holds in $s$.

We are assuming that there is no true bifurcation and that the singularity in $\lambda$ is at worst simple fold.

$$\text{dim}(\text{Null}(F_u)) = 1, F_\lambda \neq \text{Ran}(F_u)$$

# Arclength Continuation Algorithm

Set $\lambda = a$, $s = 0$ solve $F(u, \lambda) = 0$ with Newton-(MG, GMRES, $\ldots$) to obtain $u_0$.

Estimate $ds$, $\dot{u}$, $\dot{\lambda}$.

**while** $s < s_{max}$ **do**

  $s \leftarrow s + ds$.

  Solve $G(x, s) = 0$ with $u_0$ as the initial iterate.

  $x_0 \leftarrow x$

  Update $ds$, $\dot{u}$, $\dot{\lambda}$.

**end while**

# How should compactness help?

- Newton-Krylov solvers: Ferng-K(00), K, Kevrekidis, Qiao (04)

# How should compactness help?

- Newton-Krylov solvers: Ferng-K(00), K, Kevrekidis, Qiao (04)
  - Mesh-independent performance for compact ranges of $s$,

# How should compactness help?

- Newton-Krylov solvers: Ferng-K(00), K, Kevrekidis, Qiao (04)
  - Mesh-independent performance for compact ranges of $s$,
  - Preconditioning easy or unnecessary(?).

# How should compactness help?

- Newton-Krylov solvers: Ferng-K(00), K, Kevrekidis, Qiao (04)
  - Mesh-independent performance for compact ranges of $s$,
  - Preconditioning easy or unnecessary(?).
- Multilevel solvers

# How should compactness help?

- Newton-Krylov solvers: Ferng-K(00), K, Kevrekidis, Qiao (04)
  - Mesh-independent performance for compact ranges of $s$,
  - Preconditioning easy or unnecessary(?).
- Multilevel solvers
  - Easy to build. Compactness smooths for you.

# How should compactness help?

- Newton-Krylov solvers: Ferng-K(00), K, Kevrekidis, Qiao (04)
  - Mesh-independent performance for compact ranges of $s$,
  - Preconditioning easy or unnecessary(?).
- Multilevel solvers
  - Easy to build. Compactness smooths for you.
  - Appropriate coarse grid data depend on $s$.

# Multilevel Approach

Pathfollowing on coarse mesh + nested iteration fails.

- $F(u,\lambda) = u - \mathscr{K}(u,\lambda)$
- $\lambda(s)$ is sensitive to the mesh.
- Track path on fine mesh.
- Use coarse mesh problem to approximate $\mathscr{K}^u$
  Apply GMRES to new problem.

# Coarse mesh problem construction

For continuation in $\lambda$

- $x^h = x^h + dx$, Euler predictor on fine mesh.
- $u^H = I_h^H(u^h)$, $\lambda = \lambda^H = \lambda^h$.
- Build $K_H = I_H^h \mathscr{K}_u^H(u^H, \lambda) I_h^H$
- Norm convergent (K, 1995) if $I_h^H$ is done right degenerate kernel approximation
- Approximate Newton step by solving $s - K_H s = -F_h(u^H, \lambda)$.
  Fine mesh residual and coarse mesh solve.

# Continuation in $s$

Approximate $G_x$ by

$$G^{H,h}_{u,\lambda}(u,\lambda) \equiv \begin{pmatrix} I - \partial \mathscr{K}^H(I_h^H u,\lambda)/\partial u & -\partial \mathscr{K}^H(I_h^H u,\lambda)/\partial \lambda \\ \\ (I_h^H \dot{u})^T & \dot{\lambda} \end{pmatrix}.$$

and apply GMRES.

# Continuation in $s$

Approximate $G_x$ by

$$G^{H,h}_{u,\lambda}(u,\lambda) \equiv \begin{pmatrix} I - \partial \mathscr{K}^H(I_h^H u, \lambda)/\partial u & -\partial \mathscr{K}^H(I_h^H u, \lambda)/\partial \lambda \\ \\ (I_h^H \dot{u})^T & \dot{\lambda} \end{pmatrix}.$$

and apply GMRES.

- Operator-function product is now on coarse mesh.
- Works for "black-box" functions. Flexible choice of $\mathscr{K}^H$.
- Theory follows from older work,
  if you coarsen only in $\mathscr{K}$, not in $G$.

# Details

- $F$ and $N$ may require scaling to make Newton terminate properly

# Details

- $F$ and $N$ may require scaling to make Newton terminate properly
- $ds$ must be controlled by watching for

# Details

- $F$ and $N$ may require scaling to make Newton terminate properly
- $ds$ must be controlled by watching for
  - deviation of Newton's/(step in s) from target

# Details

- $F$ and $N$ may require scaling to make Newton terminate properly
- $ds$ must be controlled by watching for
  - deviation of Newton's/(step in s) from target
  - curvature estimation

# Details

- $F$ and $N$ may require scaling to make Newton terminate properly
- $ds$ must be controlled by watching for
  - deviation of Newton's/(step in s) from target
  - curvature estimation
  - true bifurcation

# Details

- $F$ and $N$ may require scaling to make Newton terminate properly
- $ds$ must be controlled by watching for
  - deviation of Newton's/(step in s) from target
  - curvature estimation
  - true bifurcation
- occasional testing for bifurcation

# Numerical Results: Three Solution Paths

For each solution we continue in $\rho$, and plot three scalars:

- Excess number

$$\int r^2 h(r)\,dr$$

- Pressure

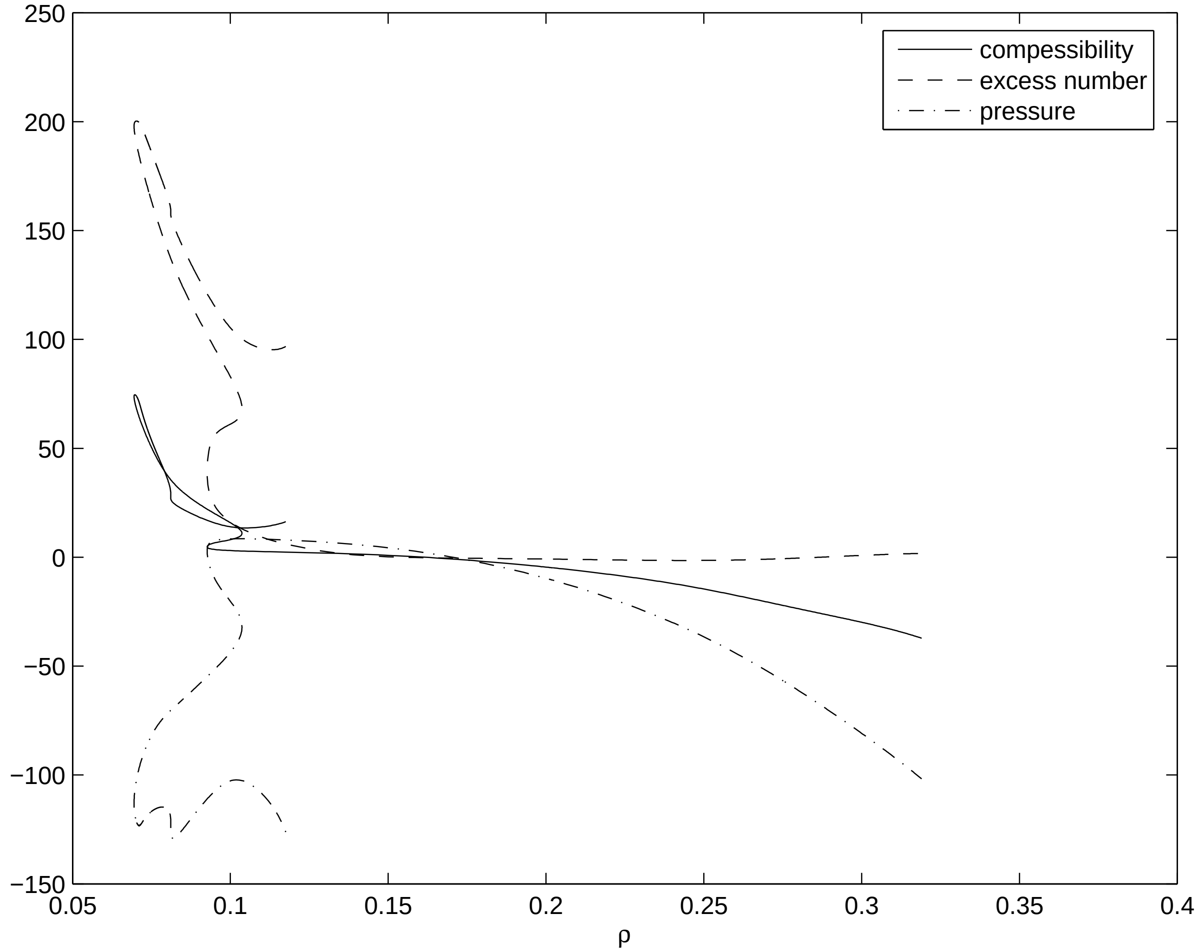
$$\int r^3 U'(r)(h(r)+1)\,dr$$

- Compressibility

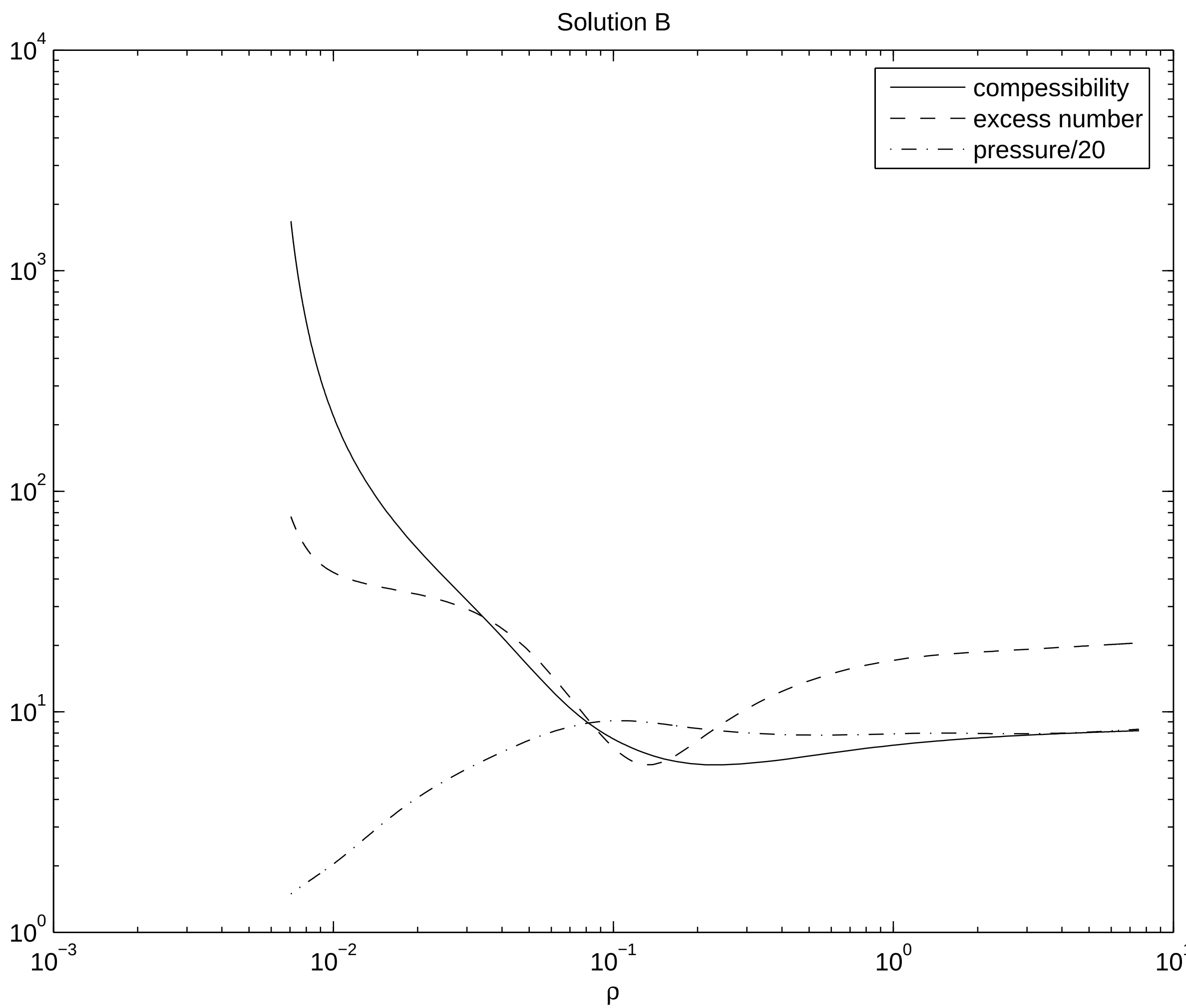
$$\int r^2 c(r)\,dr$$

as functions of $\rho$.

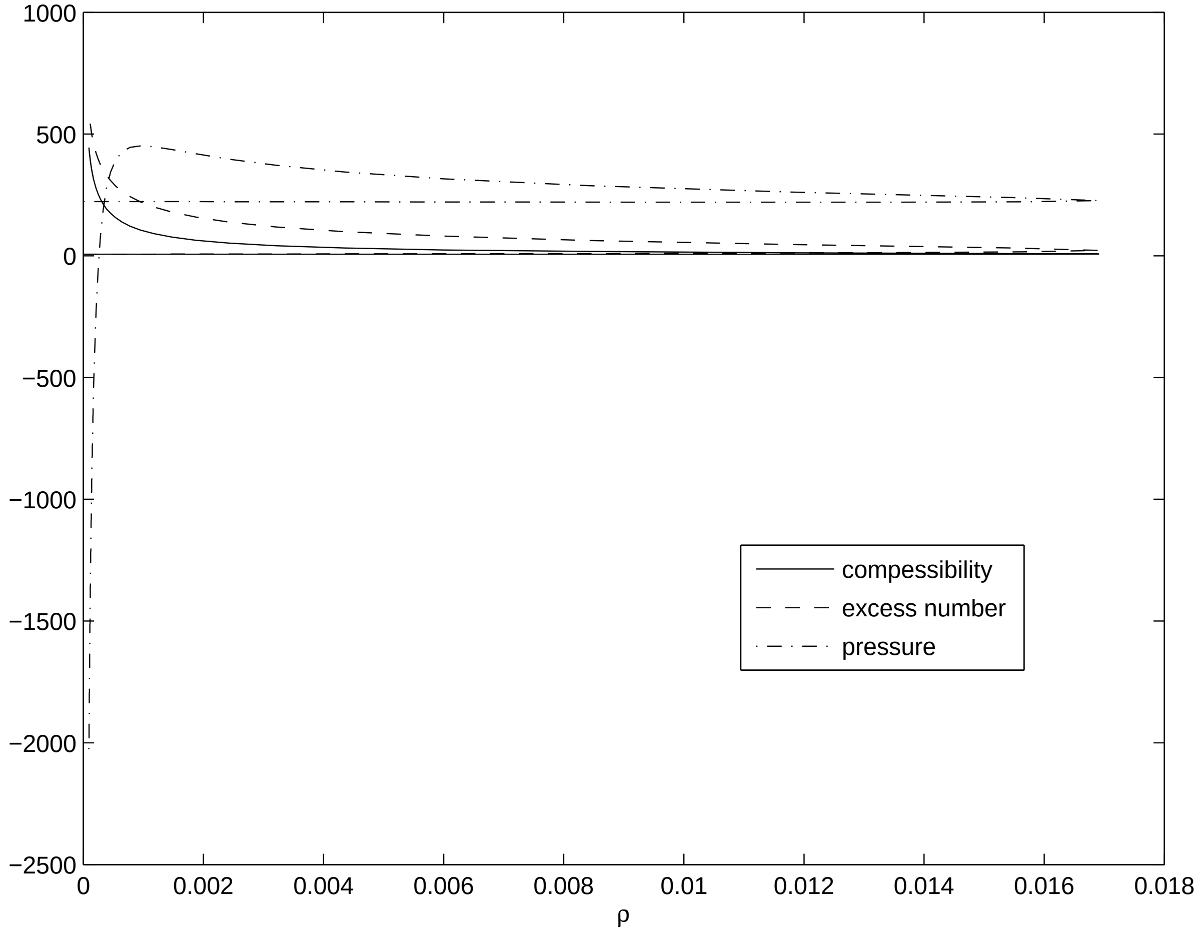
# Path through physical solution



# Path through non-physical solution



# Path through new solution



# Conclusions

- OZ integro-algebraic equations
  Elimination leads to compact fixed point problem
- Multilevel method for integral equations
- Solves OZ, but finds too many solutions
- Bottom-up nesting goes the wrong way for continuation
- Top down works; currently 30% faster than GMRES