Evaluation of Linear Solvers for an Astrophysics Problem

O.A. Marques

P.B. Vasconcelos

U.PORTO

CMUP

Centro de Matemática
Universidade do Porto

PARA’06
Description of the astrophysics problem

• Solve radiative transfer equation in stellar atmospheres

\[
T \varphi = z \varphi + f
\]

Fredholm integral equation 2nd kind

• \( T \) integral operator defined on \( X = L^1(I), I = [0, \tau^*] \)

\[
(Tx)(\tau) = \int_{\tau^*}^\tau g(|\tau - \tau'|) x(\tau') \, d\tau'
\]

– \( \tau \) is the optical depth of a stellar atmosphere
– \( \tau^* \) optical thickness of the atmosphere
– \( z \) is on the resolvent set of \( T \)
– \( f \in L^1(I) \) is the source term
Description of the astrophysics problem

- $g$ is the kernel defined by $g(\tau) := \frac{\bar{\omega}}{2} E_1(\tau), \ 0 < \tau \leq \tau^*$
  - $\bar{\omega} \in ]0,1[\$ is the albedo and
  - $E_1$ is the first exponential-integral function and it belongs to the family
    $$E_v(\tau) := \int_0^\infty \frac{\exp(-\tau \mu)}{\mu^v} \, d\mu, \tau > 0, v \geq 1$$
    $$E_{v+1}'(\tau) = -E_v(\tau); \ E_v(0) = \frac{1}{v - 1}, v > 1$$
  - $g$ is weakly singular in the sense that
    $$\lim_{\tau \to 0^+} g(\tau) = +\infty; \ g \in C^0([0, \tau^*]) \cap X; \ \sup_{\tau \in [0, \tau^*]} \int_0^{\tau^*} g(|\tau - \tau'|) \, d\tau' < \infty$$
    $$g(\tau) > 0 \text{ for all } \tau \in [0, \tau^*]; \ g \text{ decreasing function on } [0, \tau^*]$$
Projection method: Kantorovich

- Approximate $T\varphi = z\varphi + f$ by $T_n\varphi_n = z\varphi_n + f$
  - consider a grid $0 = \tau_{n,0} < \tau_{n,1} < \cdots < \tau_{n,n-1} < \tau_{n,n} = \tau^*$
  - define $X_n = \text{span} \left\{ e_{n,j}, \ j = 1, \ldots, n \right\}$, $e_{n,j} \in X$

- Let $\pi_n$ be the projection op. $\pi_n x = \sum_{j=1}^{n} \langle x, e_{n,j}^* \rangle e_{n,j}$

$$T_n x = \pi_n T x = \sum_{j=1}^{n} \langle x, \ell_{n,j} \rangle e_{n,j}, \quad \ell_{n,j} = T^* e_{n,j}^*$$
  - where $e_{n,j}^*$ is the adjoint basis of $e_{n,j}$ in $X^*$
The solution of the approximate problem
\[ T_n \varphi_n = z \varphi_n + f \]
leads to the solution of a linear system with \( n \) eq’s and \( n \) unknowns
\[
(A_n - zI_n) x_n = b_n
\]
- \( A_n \) is the restriction of \( T_n \) to \( X_n \): \( A_n = \left( \langle e_{n,j}, \ell_{n,i} \rangle \right)_{i,j=1}^n \)
\[
b_n = \left( \langle f, \ell_{n,i} \rangle \right)_{i=1}^n \quad x_n = \left( \langle \varphi_n, \ell_{n,i} \rangle \right)_{i=1}^n
\]
- we recover \( \varphi_n \) from \( x_n \) by
\[
\varphi_n = \frac{1}{z} \left( \sum_{j=1}^n x_n(j) e_{n,j} - f \right)
\]
Matrix coefficients: \( A_n \)

grid \( (\tau_{n,j})_{j=0}^n \) defined on \([0, \tau^*]\), for \( i, j \in [1, n] \)

\[
A_n (i, j) = \frac{\bar{\omega}}{2h_{n,i}} \int_{\tau_{n,i}}^{\tau_{n,i+1}} \int_0^{\tau^*} E_1(|\tau - \tau'|) e_{n,j}(\tau') d\tau' d\tau
\]

\[
= \begin{cases} 
\frac{\bar{\omega}}{2h_{n,i}} \left[ E_3(d_{n,i-1,j}) - E_3(d_{n,i-1,j-1}) + E_3(d_{n,i,j}) - E_3(d_{n,i,j-1}) \right], & i \neq j \\
\frac{\bar{\omega}}{2h_{n,i}} \left[ 1 + \frac{1}{h_{n,j}} \left( E_3(h_{n,j} - \frac{1}{2}) \right) \right], & i = j 
\end{cases}
\]

\[
d_{n,i,j} = |\tau_{n,i} - \tau_{n,j}|, \quad i, j \in [0, n] \quad h_{n,j} = \tau_{n,j} - \tau_{n,j-1}, \quad j \in [1, n]
\]
for $i \in [1, n]$

$$b_n(i) = \frac{\mathcal{O}}{2h_{n,i}} \int_{\tau_{n,i-1}}^{\tau_{n,i}} \int_{0}^{\tau} E_1(|\tau - \tau'|) f(\tau') d\tau' d\tau ,$$

$$f(\tau) = \begin{cases} 
-1 & \text{if } 0 \leq \tau \leq \frac{\tau^*}{2} \\
0 & \text{if } \frac{\tau^*}{2} < \tau \leq \tau^* 
\end{cases}$$

$$= \begin{cases} 
\frac{\mathcal{O}}{2h_{n,i}} \left[ E_3 \left( \frac{\tau^*}{2} - \tau_{n,i} \right) - E_3 \left( \frac{\tau^*}{2} - \tau_{n,i-1} \right) \cdots + E_3 \left( \tau_{n,i} \right) - E_3 \left( \tau_{n,i-1} \right) - 2h_{n,i} \right], & \tau_{n,i} \leq \frac{\tau^*}{2} \\
\frac{\mathcal{O}}{2h_{n,i}} \left[ E_3 \left( \tau_{n,i} - \frac{\tau^*}{2} \right) - E_3 \left( \tau_{n,i-1} - \frac{\tau^*}{2} \right) \cdots - E_3 \left( \tau_{n,i} \right) + E_3 \left( \tau_{n,i-1} \right) \right], & \tau_{n,i} > \frac{\tau^*}{2} 
\end{cases}$$
$A_n - zI_n, \ z = 1$

band and sparse matrix

strong decay in magnitude from the diagonal

Typical coefficient matrix
Approximate solution

How to solve $T_n \varphi_n = z \varphi_n + f$ when the associated coefficient matrix $A_n - zI_n$ has large dimension?

one can use:
  – direct methods,
  – preconditioned nonstationary iterative methods, or
  – iterative refinement methods (Newton-type method):

\[
\begin{cases}
given \quad x^{(0)} \\
x^{(k+1)} = x^{(k)} - (T - zI)^{-1} \left( Tx^{(k)} - zx^{(k)} - f \right)
\end{cases}
\]
Iterative refinement methods

- Jacobian \((T - zI)^{-1}\) can be approximated by
  
  - scheme A (Atkinson’s algorithm): \((T_n - zI)^{-1}\)
  
  - scheme B (Brakhage’s algorithm): \(\left( T \left( T_n - zI \right)^{-1} - I \right)/z\)
  
  - scheme C (Ahues algorithm): \(\left( \left( T_n - zI \right)^{-1} T - I \right)/z\)
Iterative refinement methods

- In practice $T$ is not used. The problem is restricted to $X_m$, $m \gg n$, considering a finer projection discretization of $T$, $T_m$

- $T_m$ restricted to $X_m$: $A_m = \left( \langle e_{m,j}, \ell_{m,i} \rangle \right)_{i,j=1}^{m}$

- $T_m$ restricted to $X_n$: $C = \left( \langle e_{m,j}, \ell_{n,i} \rangle \right)_{i,j=1}^{n,m}$

- $T_n$ restricted to $X_m$: $D = \left( \langle e_{n,j}, \ell_{m,i} \rangle \right)_{i,j=1}^{m,n}$
Atkinson’s scheme

given $A_n, A_m, C', D, x_n^{(0)}, x_m^{(0)}, z$
repeat until convergence

$$y_n = A_n x_n^{(k)} - C' x_m^{(k)}$$

solve $(A_n - zI) w_n = y_n$

$$w_m = \frac{1}{z} \left( D(w_n - x_n^{(k)}) + A_m x_m^{(k)} \right)$$

$$x_n^{(k+1)} = x_n^{(0)} + w_n$$
$$x_m^{(k+1)} = x_m^{(0)} + w_m$$

$k = k + 1$

band block LU
or
sparse iterative methods

prolong. $w_n$

update $x_n$ and $x_m$
Solving the problem in the m-D space

- We can solve \( T_m \varphi_m = z \varphi_m + f \) for the finer grid approximated matricial problem \( A_m - zI_m = b_m \)
- Our goal is to experiment with robust and portable algorithm implementations (from the ACTS Collection)
- Direct methods:
  - SuperLU
- Preconditioned nonstationary iterative methods:
  - PETSc
  - Trilinos
Problem specification

- grid $\mathcal{G}^*$: nonuniform grid (4 zones)
- parameters: $z = 1$, $\sigma = 0.75$ and $\sigma = 0.9$; $tol: \varepsilon \leq 10^{-12}$
- machines: located at LBNL/NERSC
  - SGI Altix 350: 32 64-bit 1.4 GHz Intel Itanium-2 processors, with 192 GBytes of shared memory
  - AMD Opteron Cluster: 356 dual-processor nodes, 2.2 GHz/node, 6 GB/node, interconnected with a high-speed InfiniBand network
  - IBM SP: 380 compute nodes with 16 Power 3+ processors/node, 16 GB memory/node.
- software:
  - MPI, F77 & F95, PETSc, SuperLU
Normalized times for the generation phase and system solution with SuperLU, for various matrix sizes ($m$), on the SGI Altix

$$\bar{\omega} = 0.75$$

<table>
<thead>
<tr>
<th>$m$</th>
<th>generation</th>
<th>solution factor</th>
<th>solution solve</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>3.26E+03</td>
<td>6.95E+01</td>
<td>1.00E+00</td>
</tr>
<tr>
<td>2000</td>
<td>2.12E+04</td>
<td>1.65E+02</td>
<td>3.00E+00</td>
</tr>
<tr>
<td>4000</td>
<td>9.71E+04</td>
<td>3.59E+01</td>
<td>6.00E+00</td>
</tr>
<tr>
<td>8000</td>
<td>4.26E+05</td>
<td>7.51E+02</td>
<td>1.80E+01</td>
</tr>
<tr>
<td>16000</td>
<td>1.80E+06</td>
<td>1.54E+03</td>
<td>3.00E+01</td>
</tr>
<tr>
<td>32000</td>
<td>7.36E+06</td>
<td>3.12E+03</td>
<td>5.35E+01</td>
</tr>
</tbody>
</table>
**Normalized times and nb. it. for various matrix sizes (m) on up to 32 processors (p) on the Opteron cluster**

A constant memory use per node allows efficiency to be maintained.

<table>
<thead>
<tr>
<th>$\omega = 0.75$</th>
<th>generation</th>
<th>GMRES 22 iterations</th>
<th>BiCGStab 14 iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m$</td>
<td>$p$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10000</td>
<td>1</td>
<td>5.40E+03</td>
<td>7.54E+00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2.67E+03</td>
<td>4.02E+00</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1.39E+03</td>
<td>2.32E+00</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>6.90E+02</td>
<td>1.80E+00</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>3.51E+02</td>
<td>1.15E+00</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>1.79E+02</td>
<td>1.15E+00</td>
</tr>
<tr>
<td>25000</td>
<td>4</td>
<td>8.41E+03</td>
<td>5.42E+00</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>4.28E+03</td>
<td>3.02E+00</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>2.16E+03</td>
<td>2.05E+00</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>1.07E+03</td>
<td>1.00E+00</td>
</tr>
<tr>
<td>50000</td>
<td>16</td>
<td>8.57E+03</td>
<td>3.14E+00</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>4.24E+03</td>
<td>1.53E+00</td>
</tr>
</tbody>
</table>
## Normalized times and nb. it. for various matrix sizes (m) on up to 32 processors (p) on the Opteron cluster

<table>
<thead>
<tr>
<th>m</th>
<th>p</th>
<th>BiCGStab 37 iterations</th>
<th>GMRES 24 iterations</th>
<th>generation 22</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>1</td>
<td>2.83E+03</td>
<td>6.14E+00</td>
<td>6.99E+00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1.36E+03</td>
<td>3.62E+00</td>
<td>4.12E+00</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>7.20E+02</td>
<td>2.20E+00</td>
<td>2.31E+00</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>3.59E+02</td>
<td>9.99E+00</td>
<td>1.76E+00</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>1.80E+02</td>
<td>1.11E+00</td>
<td>1.30E+00</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>9.19E+02</td>
<td>1.02E+00</td>
<td>1.00E+00</td>
</tr>
<tr>
<td>25000</td>
<td>4</td>
<td>2.83E+03</td>
<td>4.96E+00</td>
<td>5.37E+00</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>2.22E+03</td>
<td>2.96E+00</td>
<td>3.39E+00</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>1.11E+03</td>
<td>1.78E+00</td>
<td>2.11E+00</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>5.55E+02</td>
<td>1.33E+00</td>
<td>1.46E+00</td>
</tr>
<tr>
<td>50000</td>
<td>16</td>
<td>4.36E+03</td>
<td>2.82E+00</td>
<td>3.14E+00</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>2.21E+03</td>
<td>2.15E+00</td>
<td>2.05E+00</td>
</tr>
</tbody>
</table>
Speedup up to 32 processors on the Opteron cluster

\[ t_p = \text{elapsed time using } p \text{ processors} \]

\[ S_p = \frac{t_1}{t_p} \]

\[ m = 10000 \]
Normalized times for Jacobi and block Jacobi preconditioners on the Opteron cluster

$m = 50000$
Conclusions

- We discussed the numerical solution of a radiative transfer equation for modelling the emission of photons in stellar atmospheres.
- The parallelization of the generation phase greatly reduces the time to solution and enables the solution of large systems.
- The selection of appropriate linear solvers is important for delivering performance and portability.
- Compared to iterative refinement techniques, the present approach
  - leads to 40% savings in time in the generation phase (for \( m=50000 \) and \( np=5 \))
  - reduces the number of communications required for mapping the coarse problem into the fine one (up to 5x for Atkinson and 4x for Brakhage and Ahues’ schemes for \( m=50000 \) and \( np=5 \))
Main references

- M. Ahues, F. D. d’Almeida, A. Largillier, O. Titaud and P. Vasconcelos
  An $L^1$ refined projection approximate solution of the radiation transfer
- B. Rutily, Multiple scattering theoretical and integral equations, *Integral
- L.A. Drummond and O. Marques, An Overview of the Advanced
- P.B. Vasconcelos and F. D. d’Almeida, Performance evaluation of a
Motivation

• In this work we consider the numerical solution of a radiative transfer equation for modeling the emission of photons in stellar atmospheres.
• Mathematically, the problem is formulated in terms of a weakly singular Fredholm integral equation defined on a Banach space.
• Computational approaches to solve the problem are discussed, using direct and iterative strategies that are implemented in open source packages.
Atkinson’s parallel scheme

given $A_m$, $x_m^{(0)}$, $x_m^{(0)}$, $z$
repeat until convergence
receive $C[i] * x_{r_m}[i]$ from all $P_i$
$y_n = A_n x_n^{(k)} - \sum_{i=1}^{p-1} C[i] * x_{r_m}[i]$
solve $(A_n - zI) w_n = y_n$
compute $w_n - x_n^{(k)}$
receive $A_m[i] * x_m^{(k)}$
y_m = y_m - \sum_{i=1}^{p-1} A_m[i] * x_m^{(k)}$
receive $D[i] * \left( w_n - x_n^{(k)} \right)$ from all $P_i$
in location $w_m[(i-1) * n + 1 : i * n]$
w_m = \frac{1}{z} (w_m - y_m)$
x_n^{(k+1)} = x_n^{(0)} + w_n; x_m^{(k+1)} = x_m^{(0)} + w_m$
k = k + 1

given $A_m[i]$, $C[i]$, $D[i]$, $z$, $x_n^{(0)}$, $x_m^{(0)}$
repeat until convergence
compute $C[i] * x_m^{(k)}$
compute $A_m[i] * x_m^{(k)}$
receive $\left( w_n - x_n^{(k)} \right)$ from $P_0$
compute $D[i] * \left( w_n - x_n^{(k)} \right)$
receive $x_n^{(k+1)}$ and $x_m^{(k+1)}$ from $P_0$
k = k + 1