Tutorial 2: Multiple Outer-Loops
Primal 4D-Var Algorithm (I4D-Var)

Choose $\delta z$

Run TLROMS

Run ADROMS

Conjugate Gradient Algorithm

$$\delta z$$

$$X_b(t), \ d$$

$$G\delta z, \ (G\delta z - d), \ J$$

$$R^{-1} (G\delta z - d)$$

$$G^T R^{-1} (G\delta z - d)$$

$$\frac{\partial J}{\partial \delta z} = D^{-1} \delta z + G^T R^{-1} (G\delta z - d)$$

$$X_a(t)$$
Choose $\delta z$

Run TLROMS

$R^{-1}$

Run ADROMS

Conjugate Gradient Algorithm

NLROMS, $z_a$

Primal 4D-Var Algorithm (I4D-Var)

Outer-loop

Inner-loop
Multiple Outer-Loops

• It is generally advantageous to perform more than one outer-loop.
• During each outer-loop the NL ROMS solution is updated using the increments $\delta z$ from the last inner-loop of the previous outer-loop.
• The updated NL ROMS solution is that about which TL ROMS and AD ROMS are linearized during the current outer-loop.
• Equivalent to solving a sequence of linear least-squares minimization problems.
• Can potentially help in identifying the global minimum of $J_{NL}$ although not proven.

$$J_{NL}(x) = \frac{1}{2} (z - z_b)^T D^{-1} (z - z_b) + \frac{1}{2} (y - H(x))^T R^{-1} (y - H(x))$$
Choose $v$

Run TLROMS

Run ADROMS

Conjugate Gradient Algorithm

Outer-loop, $n$

Inner-loop, $m$

$1 \leq m \leq M$

$Z_n = Z_b + \delta Z_{n-1}$

$Z_0 = Z_b; \quad \delta Z_0 = 0$

$v = 0; \quad v = D^{-1/2} \delta z$

Linearize about $x_n(t)$

Linearize about $x_n(t)$

NLROMS, $z_a$
Implementation in ROMS 4D-Var

• Number of inner-loops, $m$, and outer-loops, $n$, are controlled by $N_{\text{inner}}$ and $N_{\text{outer}}$ in the ocean.in file.

• Second level preconditioning option available using info gleaned about the shape of $J (J_{NL}?)$ from previous outer-loops:
  - spectral & Ritz preconditioning (Tshimanga et al, 2008)
  - s4dvar.in
    - Lprecond (T (on) or F (off))
    - Lritz (T (ritz) or F (spectral))
    - NritzEV (number of vectors to use)
Preconditioning

$J$

1st level preconditioning

2nd level preconditioning

$\mu$

$A_s = \mu s$
2\textsuperscript{nd} Level Preconditioning

Recall 1\textsuperscript{st} level preconditioning via: \( v = D^{-1/2} \delta z \)

2\textsuperscript{nd} level preconditioning proceeds via: \( u = U_n^{-1} v \) \hspace{1cm} (\( n = \text{outer-loop #} \))

Recall: \( \tilde{K}_m = K_b \Sigma C^{1/2} V_m T_m^{-1} V_m^T C^{T/2} \Sigma^T K_b^T G T R^{-1} \) \hspace{1cm} (\( m = \# \text{of inner-loops} \))

Following Tshimanga et al (2008):

Spectral (Lritz=F) \hspace{1cm} \[
U_n^{-1} = \prod_{i=m}^{1} \left( I - \left( 1 - \theta_i^{1/2} \right) \hat{y}_i \hat{y}_i^T \right)
\]

Ritz (Lritz=T) \hspace{1cm} \[
U_n^{-1} = \prod_{i=m}^{1} \left( I - \left( 1 - \theta_i^{1/2} \right) \hat{y}_i \hat{y}_i^T - \theta_i^{-1} \left( e_m^T y_i \right) \gamma_m \hat{y}_i q_{m+1}^T \right)
\]

where \( (\theta_i, y_i) \) are eigenpairs of \( T_m \) and \( \hat{y}_i = V_m y_i \)
2nd Level Preconditioning

Finally: \[ v_n = \prod_{j=1}^{n-1} U_j u_j \]

and: \[ \delta z_n = D_{n}^{1/2} v_n = (K_b)_n \Sigma C^{1/2} v_n \]

ROMS CCS example:

<table>
<thead>
<tr>
<th>Ritz #</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>J</td>
<td>9173</td>
<td>3131</td>
<td>1877</td>
<td>1855</td>
<td>1833</td>
<td>1972</td>
</tr>
</tbody>
</table>

(Moore et al, 2010b)
References

