Current Status

- Updates toward the GSI-based WRF 4D-Var have been done.
- All Changes have been synchronized and tested with the revision 1061 of Boulder repository (2\textsuperscript{nd} April, 2013).

- The GSI-based WRF 4D-Var is able to run parallel.
- Multiple outer-loops setup works (with script controlled)
- JcDFI as weak constraint for WRF MASS CORE is added and works. (more investigation needed for regional WRF model.)
- All test cases pass the adjoint test.
- All test cases pass the gradient test.

- Most of the changes related to the GSI-based WRF 4D-Var were isolated using #ifdef.
Milestones:

• Re-development of the WRF tangent linear and adjoint model.

• Experiences gained from upgrading the WRF 4D-Var from MPMD to SPMD.

• Existed 4D-Var developments in current GSI framework.
  • Thank Dr. R. Todling and others.
Outline

• Linking GSI with WRF TLM and ADM

• GSI-based WRF 4DVAR

• What to do ...
Outline

• Linking GSI with WRF TLM and ADM.

• GSI/WRF 4DVAR

• What to do ......
What do we need to link GSI with WRF+?

**GSI Space**
- U,V on A-Grid
- Virtual Temp.
- Specific Humidity
- Surf. Press.
- model_tl
- model_ad

**WRF+ Space**
- U, V on C-grid
- Potential Temp.
- Mixing Ratio
- Dry Surf. Press.
- wrf_run_tl
- wrf_run_ad
Linking GSI with WRF+

**GSI Space**
- control2state
- model_tl
- intall
- model_ad
- state2control

**Coupler**
- Grids Conversion
  - gsi2wrf_tl
  - gsi2wrf_ad
  - wrf2gsi_tl
  - wrf2gsi_ad

**WRF+ Space**
- wrf_run
- wrf_run_tl
- wrf_run_ad

**Model interfaces**
- model_tl_wrf
- model_ad_wrf
Data flow – 3DVar

controlvars.

sf, vp, ps, tv, q

control2state

u, v, ps, p3d, tv, tsen, q

Tangent Linear

intall

Adjoint

controlvars.

sf, vp, ps, tv, q

state2control

u, v, ps, p3d, tv, tsen, q
Outline

• Linking GSI with WRF TLM and ADM

• GSI-based WRF 4DVAR

• What to do ......
Single pseudo observation Experiments. Scenario I

iwnbgn
idade4
regional_time
ntguessig

\text{min\_offset}=0
\text{time\_offset}=0

ntguessig\_ref = 1

\text{nhr\_assimilation}=6

\text{prepbufr\_01}

\text{prepbufr\_02}

4DVAR
Single Obs. (6h obs)

**SINGLE OBS GSI/WRF4DVAR**

init: 2000-01-26 00:00:00
valid: 2000-01-26 06:00:00

delta T at 500 hPa
Height (m) at 500 hPa
Wind (ks) at 500 hPa

Height Contours: 5320 to 5880 by 80

delta T

- .28 .24 .2 .16 .12 .08 .04 .0 .04 .08 .12 .16 .2 .24 .28
Single pseudo observation Experiments. Scenario II

iwnbgn
idate4
regional_time
ntguessig

min_offset=0
time_offset=0

ntguessig_ref = 4

ntguessig = 4

nhr_assimilation=6

4DVAR

500hPa T
prepbufr.01

21 00 03 06 09 12
Single Obs. (3h obs.)

3DVAR

4DVAR
GSI Tutorial case – downgraded to lower resolution

- **Analysis time:**
  - 2011·03·22·12
- **Domain size:**
  - 175×83×L51
- **Domain horizontal resolution:**
  - 90km
- **Assimilation window**
  - 6 hours (2209 to 2215) for both 3dvar and 4dvar
- **Minimization algorithm**
  - lsqrtb = .true.
  - lcongrad = .true.
  - ltint = .true.
  - 1 outer loop, 50 inner loops
Question: We expect that the 4DVar mode should use more data (especially surface obs.) than 3DVar mode does. What’s wrong?
## Adjoint Test

### 3DVAR -before

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADTEST starting</td>
<td></td>
</tr>
<tr>
<td>ADTEST use random_cv(xhat)</td>
<td></td>
</tr>
<tr>
<td>ADTEST</td>
<td>0.123456789012345678</td>
</tr>
<tr>
<td>ADTEST $&lt;F*F,Y,X&gt;$</td>
<td>1.905341354607271496E+06</td>
</tr>
<tr>
<td>ADTEST $&lt;F,Y,F,Y&gt;$</td>
<td>1.905341354607257294E+06</td>
</tr>
<tr>
<td>ADTEST 14 digits are identical</td>
<td></td>
</tr>
<tr>
<td>ADTEST rel. err.</td>
<td>7.454133732279982765E-16</td>
</tr>
<tr>
<td>ADTEST mach.eps</td>
<td>2.220446049250313081E-16</td>
</tr>
</tbody>
</table>

### 4DVAR-before

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADTEST starting</td>
<td></td>
</tr>
<tr>
<td>ADTEST use random_cv(xhat)</td>
<td></td>
</tr>
<tr>
<td>ADTEST</td>
<td>0.123456789012345678</td>
</tr>
<tr>
<td>ADTEST $&lt;F*F,Y,X&gt;$</td>
<td>1.139774043173196539E+07</td>
</tr>
<tr>
<td>ADTEST $&lt;F,Y,F,Y&gt;$</td>
<td>1.139774043173203059E+07</td>
</tr>
<tr>
<td>ADTEST 14 digits are identical</td>
<td></td>
</tr>
<tr>
<td>ADTEST rel. err.</td>
<td>5.719781092889553150E-15</td>
</tr>
<tr>
<td>ADTEST mach.eps</td>
<td>2.220446049250313081E-16</td>
</tr>
</tbody>
</table>

### 3DVAR-after

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADTEST starting</td>
<td></td>
</tr>
<tr>
<td>ADTEST use random_cv(xhat)</td>
<td></td>
</tr>
<tr>
<td>ADTEST</td>
<td>0.123456789012345678</td>
</tr>
<tr>
<td>ADTEST $&lt;F*F,Y,X&gt;$</td>
<td>1.145017507932603126E+06</td>
</tr>
<tr>
<td>ADTEST $&lt;F,Y,F,Y&gt;$</td>
<td>1.145017507932609413E+06</td>
</tr>
<tr>
<td>ADTEST 14 digits are identical</td>
<td></td>
</tr>
<tr>
<td>ADTEST rel. err.</td>
<td>5.49024563825641478E-15</td>
</tr>
<tr>
<td>ADTEST mach.eps = 2.220446049250313081E-16</td>
<td></td>
</tr>
</tbody>
</table>

### 4DVAR-after

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADTEST starting</td>
<td></td>
</tr>
<tr>
<td>ADTEST use random_cv(xhat)</td>
<td></td>
</tr>
<tr>
<td>ADTEST</td>
<td>0.123456789012345678</td>
</tr>
<tr>
<td>ADTEST $&lt;F*F,Y,X&gt;$</td>
<td>1.034088584772372805E+07</td>
</tr>
<tr>
<td>ADTEST $&lt;F,Y,F,Y&gt;$</td>
<td>1.034088584772372805E+07</td>
</tr>
<tr>
<td>ADTEST 14 digits are identical</td>
<td></td>
</tr>
<tr>
<td>ADTEST rel. err.</td>
<td>4.142859621812996916E-15</td>
</tr>
<tr>
<td>ADTEST mach.eps = 2.220446049250313081E-16</td>
<td></td>
</tr>
<tr>
<td>4DVAR</td>
<td></td>
</tr>
<tr>
<td>-----------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>ADTEST_OBS</strong> ®&lt;F*F.Y,X&gt; = 6.758543423470276175E+05</td>
<td></td>
</tr>
<tr>
<td><strong>ADTEST_OBS</strong> ®&lt;F.Y,F.Y&gt; = 6.758543423470275011E+05</td>
<td></td>
</tr>
<tr>
<td><strong>ADTEST_OBS</strong> 15 digits are identical</td>
<td></td>
</tr>
</tbody>
</table>
### Gradient Test

<table>
<thead>
<tr>
<th>3DVAR -before</th>
<th>4DVAR-before</th>
</tr>
</thead>
<tbody>
<tr>
<td>grtest: gradient T1= 1.000799663437064</td>
<td>grtest: gradient T1= 0.9990476095484848</td>
</tr>
<tr>
<td>grtest: gradient T1= 1.000801257924277</td>
<td>grtest: gradient T1= 0.9990423154564589</td>
</tr>
<tr>
<td>grtest: gradient T1= 1.000795318307842</td>
<td>grtest: gradient T1= 0.9990367849678288</td>
</tr>
<tr>
<td>grtest: gradient T1= 1.000735631073795</td>
<td>grtest: gradient T1= 0.9989816676095885</td>
</tr>
<tr>
<td>grtest: gradient T1= 1.000138699719116</td>
<td>grtest: gradient T1= 0.998430588591904</td>
</tr>
<tr>
<td>grtest: gradient T1= 0.9941693692653023</td>
<td>grtest: gradient T1= 0.9929206249020169</td>
</tr>
<tr>
<td>grtest: gradient T1= 0.9344760659344273</td>
<td>grtest: gradient T1= 0.9378202843136800</td>
</tr>
<tr>
<td>grtest: gradient T1= 0.3375430327161567</td>
<td>grtest: gradient T1= 0.3868168788307939</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3DVAR -after</th>
<th>4DVAR -after</th>
</tr>
</thead>
<tbody>
<tr>
<td>grtest: gradient T1= 1.000015804044360</td>
<td>grtest: gradient T1= 0.9999606139497992</td>
</tr>
<tr>
<td>grtest: gradient T1= 1.000010961587316</td>
<td>grtest: gradient T1= 1.000010809460202</td>
</tr>
<tr>
<td>grtest: gradient T1= 1.000000368712534</td>
<td>grtest: gradient T1= 1.000001562918812</td>
</tr>
<tr>
<td>grtest: gradient T1= 0.999990975675600</td>
<td>grtest: gradient T1= 0.9999995022610166</td>
</tr>
<tr>
<td>grtest: gradient T1= 0.999994246943746</td>
<td>grtest: gradient T1= 0.9999934531702166</td>
</tr>
<tr>
<td>grtest: gradient T1= 0.999914007240680</td>
<td>grtest: gradient T1= 0.9993453113466101</td>
</tr>
<tr>
<td>grtest: gradient T1= 0.9991400604212262</td>
<td>grtest: gradient T1= 0.9934531128869484</td>
</tr>
</tbody>
</table>

- Parallel problem in gradient test? (32 PE used)
- One-processor run produces good gradient test results.
- No impact on final analysis.
- Conventional data only works fine.
Minimization (pcgsoi)

- 4DVAR has smaller initial cost function.
- 4DVAR has smaller final cost function.
- Note: 4DVAR uses more data than 3DVar.
Increments (Temperature at 200hPa)
Increments (Temperature at 500hPa)
Increments (Temperature at 850hPa)
Increments (U component at 200hPa)
Increments (U component at 500hPa)
Increments (U component at 850hPa)

GSI WRF 3DVAR

Init: 2011-03-22_12:00:00  
Valid: 2011-03-22_12:00:00

delta U at 850 hPa
Height (m) at 850 hPa
Wind (kts) at 850 hPa

GSI WRF 4DVAR

Init: 2011-03-22_12:00:00  
Valid: 2011-03-22_12:00:00

delta U at 850 hPa
Height (m) at 850 hPa
Wind (kts) at 850 hPa

Height Contours: 1240 to 2000 by 40
Increments (RH at 200hPa)
Increments (RH at 500hPa)

GSI WRF 3DVAR

Init: 2011-03-22_12:00:00
Valid: 2011-03-22_12:00:00

GSI WRF 4DVAR

Init: 2011-03-22_12:00:00
Valid: 2011-03-22_12:00:00

delta RH at 500 hPa
Height (m) at 500 hPa
Wind (kts) at 500 hPa

Height Contours: 5320 to 5880 by 80

delta RH
-40 -32 -24 -16 -8 0 8 16 24 32 40
Increments (RH at 850hPa)

GSI WRF 3DVAR

Init: 2011-03-22 12:00:00
Valid: 2011-03-22 12:00:00

GSI WRF 4DVAR

Init: 2011-03-22 12:00:00
Valid: 2011-03-22 12:00:00

delta RH at 850 hPa
Height (m) at 850 hPa
Wind (kts) at 850 hPa

Height Contours: 1240 to 2000 by 40

delta RH

-40 -32 -24 -16 -8  0  8  16  24  32  40
2 outer-loops Minimization
(pcgsoi, inner loops: 50/50)

Obs. Number Statistics

<table>
<thead>
<tr>
<th>Obs.</th>
<th>3DVAR</th>
<th>4DVAR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1st outer</td>
<td>2nd outer</td>
</tr>
<tr>
<td>Surf. Press.</td>
<td>13852</td>
<td>13859</td>
</tr>
<tr>
<td>Temp.</td>
<td>9233</td>
<td>9233</td>
</tr>
<tr>
<td>Wind</td>
<td>25582</td>
<td>25636</td>
</tr>
<tr>
<td>Moist.</td>
<td>4210</td>
<td>4210</td>
</tr>
<tr>
<td>GPS</td>
<td>8707</td>
<td>8837</td>
</tr>
<tr>
<td>Radiance</td>
<td>131221</td>
<td>154758</td>
</tr>
<tr>
<td>Total</td>
<td>192805</td>
<td>216533</td>
</tr>
</tbody>
</table>
Increments after 2 outer loops (Temperature at 500hPa)
Increments after 2 outer loops (RH at 850hPa)
Parallel Performance

- Domain size:
  - $349 \times 247 \times L_{51}$
- Domain horizontal resolution:
  - 30km
- Assimilation window
  - 6 hours (2209 to 2215)

Walltime for 5 iterations on Yellowstone

<table>
<thead>
<tr>
<th>Processing Cores</th>
<th>Minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>60</td>
</tr>
<tr>
<td>64</td>
<td>40</td>
</tr>
<tr>
<td>128</td>
<td>20</td>
</tr>
<tr>
<td>256</td>
<td>10</td>
</tr>
<tr>
<td>512</td>
<td></td>
</tr>
</tbody>
</table>

GSI does not allow us to use more processing tasks.
One-month regression test

- **Analysis time:**
  - 2012\textbullet{}02 12Z

- **Domain size:**
  - 175×124×L51

- **Domain horizontal resolution:**
  - 60km

- **Conventional data only**
Experiments design:

3DVAR

4DVAR

72h
Aggregated RMSE Profiles—oH

RMSE Profiles 01 - 29 Feb. 2012

- U (m/s)
- V (m/s)
- T (Degree)
- Q (g/Kg)

4D-Var WRFDA
3D-Var WRFDA
Aggregated RMSE Profiles—24H

RMSE Profiles 01 - 29 Feb. 2012

Pressure (hPa) vs U (m/s)

Pressure (hPa) vs V (m/s)

Pressure (hPa) vs T (Degree)

Pressure (hPa) vs Q (g/Kg)

4D-Var WRFDA
3D-Var WRFDA
Aggregated RMSE Profiles—48H

RMSE Profiles 01 - 29 Feb. 2012

- U (m/s)
- V (m/s)
- T (Degree)
- Q (g/Kg)

Graphs showing RMSE profiles for U, V, T, and Q over pressure levels from 1000 hPa to 10 hPa, with 4D-Var and 3D-Var WRFDA models compared.
Aggregated RMSE Profiles—72H

RMSE Profiles 01 - 29 Feb. 2012

U (m/s) vs. Pressure (hPa)

V (m/s) vs. Pressure (hPa)

T (Degree) vs. Pressure (hPa)

Q (g/Kg) vs. Pressure (hPa)

4D-Var WRFDA
3D-Var WRFDA
Outline

• Linking GSI with WRF TLM and ADM

• GSI-based WRF 4DVAR

• What to do ......
Proposed changes

- M src/main/makefile_DTC
- A + src/main/read_wrf_mass_files.F90
- M src/main/wrf_binary_interface.F90
- M src/main/gridmod.F90
- D src/main/read_wrf_mass_files.f90
- M src/main/stpcalc.f90
- M src/main/model_tl.F90
- M src/main/model_ad.F90
- M src/main/stub_pertmod.F90
- M src/main/fill_mass_grid2.f90
- A + src/main/gesinfo.F90
- M src/main/adjtest.f90
- M src/main/read_prepbufr.f90
- M src/main/gsi_4dvar.f90
- A src/main/wrf_pertmod.F90
- D src/main/gesinfo.f90
- M src/main/wrwrflmassa.F90
- M src/main/wrf_netcdf_interface.F90
- M src/main/gsimod.F90
- M src/main/unfill_mass_grid2.f90
- M src/main/compute-derived.f90
- M src/main/read_wrf_mass_guess.F90
- M src/main/Makefile.dependency
- M src/main/enorm_state.f90
- M src/main/evaljcdf.f90
- M src/main/obsmod.F90
- M arch/Config.pl
- A arch/postamble.4dvar
Major changes

- Add coupler interfaces of WRF+, use `#ifdef WRF_PERT` to isolate the changes for WRF 4DVar from NASA GMAO 4DVar.
  
  ```
  M src/main/model_tl.F90
  M src/main/model_ad.F90
  M src/main/stub_pertmod.F90
  A src/main/wrf_pertmod.F90
  ```

- Add configure option for GSI/WRF 4DVar compilation. Compiling WRFPLUS firstly; pointing the envir. var. `WRF_DIR` to the PATH of WRFPLUS; set the envir. var. `VAR4D=1`

  ```
  M arch/Config.pl
  A arch/postamble.4dvar
  ```
Minor changes

• Add new namelist variable in setup: \texttt{nhr\_inc\_update}, which determines which guess file will be updated after the analysis. For example, \texttt{nhr\_inc\_update=1} updates the first guess at the beginning of the time window (\texttt{wrf\_inou1}), \texttt{nhr\_inc\_update=4} updates the first guess at hour 3 (\texttt{wrf\_inou4}) for \texttt{nhr\_obsbin=1}

\begin{verbatim}
M src/main/gsi_4dvar.f90
M src/main/gsimod.F90
A src/main/read_wrf_mass_files.F90
D src/main/read_wrf_mass_files.f90
M src/main/wrf_netcdf_interface.F90
\end{verbatim}
Minor changes --- continued

• Add new namelist variable: `nhr_inc_update`, which determines which guess file will be updated after analysis. For example, `nhr_inc_update=1` updates the firstguess at the beginning of the time window (wrf_inou1), `nhr_inc_update=4` updates the firstguess at hour 3 (wrf_inou4) for `nhr_obsbin=1`. Isolated with `#ifdef`.

  M src/main/gsi_4dvar.f90
  M src/main/gsimod.F90
  M src/main/wrwrffmassa.F90
Minor changes --- continued

• Modified to accommodate adjoint test for GSI-based WRF 4DVar.

  Isolated with #ifdef
  
  A + src/main/adjtest.F90
  D src/main/adjtest.f90

• Add corresponding adjoint codes for fill and unfill.

  M src/main/fill_mass_grid2.f90
  M src/main/unfill_mass_grid2.f90

• Turn off trajectory init. when observer is true. Isolated with #ifdef

  M src/main/compute-derived.f90

• Remove some WRF interfaces, which will be available in WRF+ lib..

  Isolated with #ifdef.
  
  M src/main/wrf_binary_interface.F90
Minor changes --- continued

• For GSI-based WRF 4DVar, we always use the guess file at the beginning of the assimilation window (wrf_inou1) to determine the `regional_time`. Isolated with `#ifdef`.

  ```f90
  M src/main/gridmod.F90
  ```

• For GSI-based WRF 4DVar, `min_offset` is forced to be zero and the `nhr_assimilation` is used to calculate the end of the window.

  Isolated with `#ifdef`.

  ```f90
  M src/main/gsimod.F90
  A + src/main/gesinfo.F90
  D src/main/gesinfo.f90
  ```
Minor changes --- continued

• For GSI-based WRF 4DVar, we always use the guess file at the beginning of the assimilation window (wrf_inou1) as the first guess. Isolated with `#ifdef`
  ```
  M src/main/wrf_netcdf_interface.F90
  ```

• Allow GSI-based WRF 4DVar to read more than one observation files for each obs. type. Isolated with `#ifdef`
  ```
  M src/main/obsmod.F90
  ```

• Make dependency and compilation updates
  ```
  M src/main/Makefile.dependency
  M src/main/makefile_DTC
  ```
Bug fixes

- **ier** was used before being initialized
  
  M src/main/read_wrf_mass_guess.F90

  ```fortran
  if (nguess>0) then
    ! Get pointer for each of the hydrometeors from guess at time index "it"
    it=ntguessig
    call GSI_BundleGetPointer (GSI_MetGuess_Bundle(it), 'ql', ges_qc, istatus);ier=ier+istatus
    call GSI_BundleGetPointer (GSI_MetGuess_Bundle(it), 'qi', ges_qi, istatus );ier=ier+istatus
    call GSI_BundleGetPointer (GSI_MetGuess_Bundle(it), 'qr', ges_qr, istatus );ier=ier+istatus
  end if
  ```

- **disterrmax** was used before being initialized
  
  M src/main/read_prepbufr.f90

  ```fortran
  + disterrmax=-9999.0
  ```
Bug fixes --- continued

- **psum** was calculated by processor zero only, but used by all processors immediately. Let it be calculated by all processors. Also, add the codes to calculate the energy norm for **WRF_MASS_CORE** in **enorm_state.f90**

```
Index: src/main/stpcalc.f90
===================================================================
--- src/main/stpcalc.f90 (revision 1061)
+++ src/main/stpcalc.f90 (working copy)
@@ -542,11 +542,11 @@
  stpinout = stp(istp_use)
 ! Estimate terms in penalty
+  do i = 1, ipen
+    psum(i) = pbc(1,i) + (stp(iis-1) - stp(iis)) * (2.0_r_quad * bsum(i) + &
+      (stp(iis-1) - stp(iis)) * csum(i))
+  end do
- do i = 1, ipen
-   psum(i) = pbc(1,i) + (stp(iis-1) - stp(iis)) * (2.0_r_quad * bsum(i) + &
-      (stp(iis-1) - stp(iis)) * csum(i))
- end do
write(iout_iter,101) (psum(i), i = 1, ipen)
end if
pjcostnew(1) = psum(1) ! Jb
```
Bug fixes --- continued

• **pjc** (cost function for Jcdfi) was multiplied by 0.5 more than one timer, the other is in grtest.f90. This error leads to gradient test fail with JcDFI.

```plaintext
src/main/evaljcdfi.f90

--- src/main/evaljcdfi.f90 (revision 1061)
+++ src/main/evaljcdfi.f90 (working copy)
@@ -67,7 +67,7 @@
     Jc = 1/2 * wgt * sfilter *sfilter
     afilter = wgt * sfilter
     call enorm_state(sfilter,pjc,afilter)
-     pjc=half_quad*pjc
+     pjc=half_quad*pjc
     if (mype=-0) write(6,*')'Jc DFI='',pjc

     ! Adjoint Jc multiplicative factor
```
Next

- JcDFI is only applied on the TLM trajectory of the observational sub-windows right now (only 7 snapshots for 6h window). It should be applied on the TLM trajectory of every time step during the assimilation window.

- Is it possible to relax the limit on the maximum allowable number of processing tasks?

- Data usage in 3DVar and 4DVar is suspicious?

- Multiple incremental 4DVAR.
Thank You

The NESL Mission is:
To advance understanding of weather, climate, atmospheric composition and processes; To provide facility support to the wider community; and,
To apply the results to benefit society.

NCAR is sponsored by the National Science Foundation