Development of the Upgraded Tangent Linear and Adjoint of the Weather Research and Forecasting (WRF) Model

XIN ZHANG *

National Center for Atmospheric Research, Boulder, Colorado 80307, USA

XIANG-YU HUANG

National Center for Atmospheric Research, Boulder, Colorado 80307, USA

NING PAN

Fujian Meteorological Bureau, Fuzhou, Fujian, China

* *Corresponding author address:* Dr. Xin Zhang, NCAR,MMM, P.O. Box 3000, Boulder, CO 80307.

E-mail: xinzhang@ucar.edu

ABSTRACT

Tangent linear and adjoint models have been re-developed for the Weather Research and Forecasting (WRF) Model with its Advanced Research WRF dynamic core (WRFPLUS). The automatic differentiation engine (TAPENADE) has been used in the development. The WRFPLUS has the following improvements:

- A complete check interface ensures developers to write the tangent linear and adjoint codes in accuracy.

- To parallelize the WRFPLUS model, we adopted an innovative technique based on the nature duality that existed between MPI communication routines. The Registry in WRF is extended to automatically generate the tangent linear and adjoint of the required communication operations. This approach dramatically speeds up the software development cycle of the parallel tangent linear and adjoint codes and leads to impressive parallel efficiency.

- Module interfaces have been constructed for coupling tangent linear and adjoint codes of the WRF with other applications, such as four-dimensional variational data assimilation, forecast sensitivity to observation etc.

1. Introduction

During the past two decades, the use of the adjoint technique in meteorology and oceanography has been rapidly increasing. The adjoint model is a powerful tool in many applications, such as data assimilation, parameter estimation, sensitivity analysis etc. (Errico and Vukicevic (1992);Errico (1997);Rabier et al. (1996);Langland et al. (1999);Li et al. (1999);Xiao et al. (2002);Xiao et al. (2008);Kleist and Morgan (2005a);Kleist and Morgan (2005b))

The Weather Research and Forecasting (WRF) modeling system (Skamarock et al. (2005)) is a multi-agency effort intended to provide a next-generation mesoscale forecast model and data assimilation system to advance both the understanding and prediction of mesoscale weather and accelerate the transfer of research advances into operations. The WRF model is designed to be an efficient massively parallel computing code to take advantage of advanced high-performance computing systems. The code can be configured for both research and operations and offers numerous physics options. WRF is maintained and supported as a community model to facilitate wide use internationally, for research, operations, and teaching. It is suitable for a broad span of applications across scales ranging from large-eddy to global simulations. Such applications include real-time NWP, data assimilation development and studies, parameterized-physics research, regional climate simulations, air quality modeling, atmosphere-ocean coupling, and idealized simulations. As of this writing, the number of registered WRF users exceeds 19905, and WRF is in operational and research use around the world.

The first version of adiabatic WRF tangent linear (TLM) and adjoint model (ADM) system (WAMS) was developed by NCAR around 2007 (Xiao et al. (2008)). It has been used in the adjoint sensitivity analysis (Xiao et al. (2008)) and four-dimensional variational data assimilation (4D-Var) (Huang et al. (2009)). In the past few years, due to the hand-coded parallellisation, WAMS has failed to follow the rapid development of WRF model and data assimilation system (WRFDA)(Barker et al. (2012)). The growing gap between WAMS and WRF/WRFDA makes WAMS inconvenient to be used with other systems.

Furthermore, because WAMS uses the disk input/output (I/O) for storing basic states and exchanging data, parallel efficiency is unsatisfactory on modern high performance computers with distributed memory parallelization, especially, for 4D-Var applications.

Encouraged by the rapid developments of 4D-Var, cloud analysis, forecast sensitivity to observations and chemistry data assimilation, we re-developed the WRF tangent linear model and adjoint model (called WRFPLUS) based on the latest repository WRF. Compared with the WAMS developed by Xiao et al. (2008) and Huang et al. (2009), the new system is an all-in-one system which includes the WRF full-physics forward model (FWM), tangent linear model (TLM) and adjoint model (ADM); it also includes the tangent linear check and adjoint check procedure. A set of module interfaces is developed for easily coupling WRFPLUS with other systems such as data assimilation and adjoint sensitivity applications. An innovative approach has been applied to develop the parallel code which dramatically reduces the software development cycle of parallel TLM and ADM, and the derived parallel TLM and ADM have better parallel efficiency compared to the FWM.

The purpose of this paper is to describe the technical aspects of the newly developed WRFPLUS model. A brief introduction of the development of WRFPLUS is presented in section 2, followed by demonstration of the linearity and adjoint tests in section 3. A detailed description of parallelization strategy for the tangent linear and adjoint codes, as well as the parallel performance are in section 4. Section 5 introduces the module interfaces which are constructed in WRFPLUS for coupling purposes, such as in WRF 4D-Var. The concluding remarks are given in section 6.

2. Description of the WRF tangent linear and adjoint model

After the release of WRF model version 3.2, we started to use TAPENADE (Hascoët and Pascual (2004)) to re-develop the tangent linear and adjoint models of the WRF ARW core based on the latest WRF model. The development of WRFPLUS system follows the same three phases proposed by Xiao et al. (2008). Firstly, numerical experiments were conducted to make sure the adiabatic version of WRF with simplified physics paramterisation routines can produce the major features that the full-physics model does. Secondly, the tangent linear model and its adjoint were generated by TAPENADE and modified manually whenever necessary. The third step is to verify the correctness of the tangent linear and adjoint models.

TAPENADE is a source-to-source automatic differentiation (AD) tool for programs written in Fortran 77-95, i.e. TAPENADE generates a TLM or ADM from the source code of a given model. Like other AD tools, TAPENADE has difficulty to handle some of the complicate codes in WRF, such as third-order Runge-Kutta large time steps and small acoustic time steps (Xiao et al. (2008)), and manual intervention is necessary to check and improve the TAPENADE generated code. To represent first-order physical effects on the model evolution while also minimizing the code length to make developing its adjoint simple, and to allow the code to run quickly in the iterations required to minimize the cost function, we adopted three simplified physics packages which have maximum impact on the forecast compared to a no-physics model. They are surface friction, cumulus parameterization and large-scale condensation developed by Jimy Dudhia (personal communication), which are similar to the those physics packages in WAMS (Xiao et al. (2008)).

3. Linearity test and adjoint test

It is important and necessary to test the TLM consistency with the FWM and the ADM consistency with the TLM before they are used in any real application (Vukicevic (1991); Errico and Vukicevic (1992); Gilmour et al. (2001)). We developed the tangent linear and adjoint check procedure following Navon et al. (1992).

Let $\mathbf{f}(\mathbf{x})$, $g\_\mathbf{f}(\mathbf{x}, g\_\mathbf{x})$ and $a\_\mathbf{f}(\mathbf{x}, a\_\mathbf{x})$ denote a forward, tangent linear and adjoint model, respectively, where $\mathbf{x}$, $g\_\mathbf{x}$ and $a\_\mathbf{x}$ is column vector of model state variables, perturbations of state variables and adjoint of state variables, respectively. The correctness of tangent linear model can be tested as:

$$\Phi(\lambda)=\frac{\|\mathbf{f}(\mathbf{x}+\lambda g\_\mathbf{x})-\mathbf{f}(\mathrm{x})\|}{\|g\_\mathbf{f}(\mathbf{x},\lambda g\_\mathbf{x})\|},\lim_{\lambda\to 0}\Phi(\lambda)=1 \tag{1}$$

where $\|\|$ denotes the norm of the vector. The adjoint relation is tested by :

$$\langle g\_\mathbf{f}(\mathbf{x},g\_\mathbf{x}),g\_\mathbf{f}(\mathbf{x},g\_\mathbf{x})\rangle=\langle a\_\mathbf{f}(\mathbf{x},g\_\mathbf{f}(\mathbf{x},g\_\mathbf{x}),g\_\mathbf{x}\rangle \tag{2}$$

If the tangent linear and adjoint codes are correct, the above two relations should hold up to the machine accuracy. Because different model variables have different magnitudes, we also design the capability to perform checks on individual variables separately. We performed the tangent linear and adjoint check with the test case being integrated up to 24h. We sequentially reduced the initial perturbations by a factor (λ) of 10 and repeatedly calculate $\Phi(\lambda)$ in Eqs. (1) on NCAR IBM machine with 64 bits precision. Table 1 shows the value of $\Phi(\lambda)$ from the tangent linear forecasts and the differences of two nonlinear forecasts over the whole domain. It indicated that the tangent linear forecast approximates the difference of two nonlinear forecasts as the initial perturbations decrease and approach zero.

In the adjoint relation, the left-hand side (lhs) involves only the tangent linear code, while the right-hand side (rhs) involves also the adjoint code. If lhs and rhs have the same value with machine accuracy, the adjoint code is correct compared with the tangent linear code. Using the same test case with 24h integration, lhs and rhs for the test case are 0.14182720729878E+14 and 0.14182720729883E+14, respectively. This indicates that the adjoint code is correct.

4. Parallelization of the WRFPLUS model

TAPENADE has few problems to generate tangent linear and adjoint codes from the sequential forward codes. However, it still has some ways to go forward to derive tangent linear and adjoint of parallel communication routines inside a parallel forward model. In most of the atmospheric and oceanic models, the communication routines to parallelize finite difference algorithms are linear operators, hence as matrices, the adjoint is simply the transpose, which is the dual operator and the tangent linear is the original linear operator acting on the perturbations (see Cheng (2006) and Utke et al. (2009)). With the proper coding structure and available parallel communication routines in forward codes, it is straightforward to write communication routines for tangent linear model. We only need to use the same parallel communication templates in forward codes and add the corresponding perturbation variables. In the adjoint code, the data flow of the original program needs to be reversed and any communication needs to be reversed as well. Due to the duality between MPI_SEND and MPI_RECV calls, in ADM, send message to where we receive in FWM and receive message from where we send previously. Please note that it is the adjoint variable which is subjected to be exchanged in ADM. Figure 1(a) and 1(b) could be helpful to understand the data flow of communication in FWM and ADM, respectively. In the FWM model, the variable U in the ghost region of processor P1 will be overwritten by the value of U received from processor P0. In ADM, the communication needs to be reversed. The adjoint variable a_U in the ghost region of P1 will be send to P0 and add to the value of a_U of P0, then the a_U in the ghost region of P1 will be set to zero.

In the WRF model, hundreds of thousands of lines of the code are automatically generated from a user-edited table, called the Registry (Michalakes and Schaffer (2004)). The Registry provides a high-level single-point-of-control over the fundamental structure of the model data. It contains lists describing state data fields and their attributes: dimensionality, binding to particular solvers, association with WRF I/O streams, communication operations, and run time configuration options (namelist elements and their bindings to model control structures). Adding or modifying a state variable to WRF involves modifying a single line of a single file; this single change is then automatically propagated to scores of locations in the source code the next time the code is compiled.

Halo entries in the Registry define communication operations in the model. Halo entries specify halo updates around a patch horizontally. A typical halo entry is :

```
halo     HALO_EM_C    dyn_em    4:u_2,v_2
```

The first field is the keyword ”halo”. The second entry –”HALO_EM_C”, which is the given name of the halo exchange template and will be used in the model to refer to the communication operation being defined. The third entry denoting the associated solver. The fourth entry is a list of information about the operation. This example specifies that 4 points (one cell each in North,South,East, and West direction respectively) of the stencil are used in updating the state arrays for fields u_2 and v_2. During compilation, the WRF Registry will automatically generate a code segment based on this halo entry:

```
...
CALL HALO_EM_C_sub ( grid, local_communicator, ...)
...
```

The code snippet will be included in the place where the communications are needed. At the same time, the registry also generates the subroutine ”HALO_EM_C_sub”:

```
...
! for Y direction
CALL RSL_LITE_PACK ( local_communicator, grid%u_2, ...,0,...
CALL RSL_LITE_PACK ( local_communicator, grid%v_2, ...,0,...
CALL RSL_LITE_EXCH_Y
CALL RSL_LITE_PACK ( local_communicator, grid%u_2, ...,1,...
CALL RSL_LITE_PACK ( local_communicator, grid%v_2, ...,1,...
...
! for X direction
CALL RSL_LITE_PACK ( local_communicator, grid%u_2, ...,0,...
CALL RSL_LITE_PACK ( local_communicator, grid%v_2, ...,0,...
CALL RSL_LITE_EXCH_X
CALL RSL_LITE_PACK ( local_communicator, grid%u_2, ...,1,...
CALL RSL_LITE_PACK ( local_communicator, grid%v_2, ...,1,...
...
```

In the above code segment, the outgoing slices of u_2 and v_2 for Y direction (South-North) exchanging are packed by ”RSL_LITE_PACK” into a local contiguous memory region. Therefore, one call of ”RSL_LITE_EXCH_Y” is able to finish the data exchanges on south-north direction. Once every processor received the incoming data, ”RSL_LITE_PACK” will be called again with different arguments to unpack the data to the ghost area position. The same operations are performed on the X direction (East-West) followed.

The forward communication codes show an efficient chain of relationships established. Therefore, perturbing and adjointing the code are simple. However, it is a error-prone and time consuming task to manually write all tangent linear and adjoint of communication subroutines. Since WRF Registry is able to generate halo exchange routines, there is the possibility to let Registry generate the tangent linear and adjoint of halo exchanges too. To enable WRF Registry to generate the corresponding tangent linear and adjoint communication codes automatically, we modified the Registry and added a new entry ”halo_nta” as:

```
halo_nta      HALO_EM_C     dyn_em     4:u_2,v_2
```

With this new entry, the Registry will not only generate ”HALO_EM_C_sub”, but also generate ”HALO_EM_C_TL_sub” for tangent linear codes and ”HALO_EM_C_AD_sub” for adjoint codes. The subroutine ”HALO_EM_C_TL_sub” looks like:

```
...
! for Y direction
CALL RSL_LITE_PACK ( local_communicator, grid%u_2, ...,0,...
CALL RSL_LITE_PACK ( local_communicator, grid%v_2, ...,0,...
CALL RSL_LITE_PACK ( local_communicator, grid%g_u_2, ...,0,...
CALL RSL_LITE_PACK ( local_communicator, grid%g_v_2, ...,0,...
CALL RSL_LITE_EXCH_Y
CALL RSL_LITE_PACK ( local_communicator, grid%u_2, ...,1,...
CALL RSL_LITE_PACK ( local_communicator, grid%v_2, ...,1,...
CALL RSL_LITE_PACK ( local_communicator, grid%g_u_2, ...,1,...
CALL RSL_LITE_PACK ( local_communicator, grid%g_v_2, ...,1,...
...
! for X direction
```

TLM has exactly the same exchange stencil and data flow as FWM, however, in addition to the basic state fields (u_2 and v_2), the perturbation fields (g_u_2 and g_v_2) exchanged as well, this is required by tangent linear model.

The adjoint codes ”HALO_EM_C_AD_sub” looks like:

```
...
! for Y direction
CALL RSL_LITE_PACK_AD ( local_communicator, grid%a_u_2, ...,0,...
CALL RSL_LITE_PACK_AD ( local_communicator, grid%a_v_2, ...,0,...
CALL RSL_LITE_EXCH_Y
CALL RSL_LITE_PACK_AD ( local_communicator, grid%a_u_2, ...,1,...
CALL RSL_LITE_PACK_AD ( local_communicator, grid%a_v_2, ...,1,...
...
! for X direction
```

ADM also has the same exchange stencil as FWM and TLM, but the data flow is reversed. On each processor, the entire tile of basic state variables (including halo) are stored in memory stack during the forward re-computation stage and will be restored from memory stack during the adjoint calculation. Therefore, in adjoint communication codes, only the adjoint variables (a_u_2 and a_v_2) need to be exchanged and the new subroutines ”RSL_LITE_PACK_AD” will pack the data slices from where we receive data in FWM and TLM for sending and unpack the received data to the slices where we send out data in FWM and TLM.

With the upgraded WRF Registry, all tangent linear and adjoint communication routines are generated automatically during compilation. It is very straightforward to manually insert the tangent linear and adjoint communication interfaces to TLM and ADM. In the TLM, the tangent linear communication routines are inserted to the same locations where the forward communication routines reside in FWM. In the ADM, the calling sequence of the adjoint communication routines is the reverse of which in FWM and TLM. From the code snippets presented above, we found that, for the TLM, because both basic state variables and perturbation variables are packed together, although the amount of data to communicate is doubled, the communication latency or overhead are kept the same as in FWM. This is highly desirable for modern distributed memory parallel computer system with high bandwidth. In the ADM, in general, there are two stages, the first stage is the forward re-computation part, which propagate the basic states within one time-step. It has the same latency and amount of data to communicate with FWM. The second stage is the adjoint backward part, which has the same communication latency and amount as in the first stage. Therefore, the total communication latency and amount of data to communicate are doubled in ADM. It should not impact the parallel scalability of ADM as adjoint code may has 3-4 times amount of computation than forward code. It is worth mentioning that, with this approach, we completed the parallelization of serial WRFPLUS model within a week by one person.

5. Parallel performance

To demonstrate the parallel efficiency of the WRFPLUS codes, we prepared the initial condition and boundary conditions for a 15km-resolution domain (not shown). There are 665×363 grid points in the horizontal and 45 levels in the vertical direction and the time step is $90s$. We tested this case on NCAR's two supercomputers: Lynx and Bluefire. Lynx is a single cabinet Cray XT5m supercomputer, comprised of 76 compute nodes, each with 12 processors on two hex-core AMD 2.2 GHz Opteron chips, with a total of 912 processors. Each Lynx compute node has 16 gigabytes of memory, for 1.33 gigabytes per processor, and totaling 1.216 terabytes of memory in the system. Bluefire is an IBM clustered Symmetric MultiProcessing (SMP) system, comprised of 4,096 Power 6 processors. The 4,096 processors are deployed in 128 nodes, each containing 32 processors. Nodes are interconnected with an InfiniBand switch for parallel processing using MPI. We use the default compilation options and the default processors topology calculated by WRF model. We did not do any further optimization to get the best performance. Therefore, the following results are not the best performance on the specific supercomputer.

We ran WRFPLUS on 16, 32, 64, 128, 256, 512, 1024 and 2048 processors of Bluefire and measured the parallel performance. Figure 2 shows the results for the average wall clock time for one-time-step integration (Fig.2(a)) and speedup for FWM, TLM and ADM respectively (Fig.2(b)). Please note that the timing results for FWM is different from the standard WRF run. The standard WRF is compiled with 4 bytes long real size. In WRFPLUS, the FWM is compiled with 8 bytes long real size. In general, due to the higher precision requirement in WRFPLUS, the actual computational performance of FWM is slower than standard WRF. Speedup for N processors was calculated as the wall clock time using 16 processors divided by the wall clock time using N processors. The computing times for all models are considerably reduced with increased number of processors up to 2048. In general, the TLM has a better speedup than FWM and the ADM has a slightly better or comparable speedup than FWM. The results confirm the successful implementation of the new parallel approach.

We also ran WRFPLUS on 16, 32, 64, 128, 256 and 512 processors of Lynx and measured the parallel performance. Figure 3 shows the results for the average wall clock time for one-time-step integration (Fig.3(a)) and speedup (Fig.3(b)). We could draw similar conclusion and confirm the high efficiency of the parallelization strategy of WRFPLUS again.

6. Implementation of an inline coupling interface in the WRFPLUS model

One of the motivations to upgrade the WRF tangent linear and adjoint model is to improve the computational performance of WRF 4D-Var. The old WRF 4D-Var system contains a two-way coupling between WRFDA and WAMS (Huang et al. (2009)). It exchanges information between WRFDA and WAMS via disk files and is a multiple program multiple data (MPMD) system. During the coupling, the exchanged data are written to disk and a signal file is prepared to inform the other component that data is ready to be read. Exchange of a field between WRFDA and WAMS consists of gathering and scattering operations across the processors, which are very inefficient on modern distributed memory supercomputers. This limits the number of processors that can be used for high resolution modeling.

Since WRFDA and WRFPLUS share the same software infrastructure including parallelization, field definition, I/O, Registry etc., it is straightforward to couple WRFDA and WRFPLUS to a single executable 4D-Var system. In the single executable coupled system, all information (grid and fluxes) from the WRFPLUS is passed as arguments to the coupling interface and WRFDA fetches the data from the coupler instead of disk files.

For this purpose, three major developments were needed:

i. Enable the WRFPLUS model to be callable from WRFDA with a simple application programming interface consisting of:

    (a) Initialization of the WRFPLUS model.

    (b) Advance one of the WRFPLUS model components (eg. forward model, tangent linear model and adjoint model) .

    (c) Finalize the WRFPLUS model.

ii. Development of a set of regridding routines that can interpolate data on the WRFPLUS grid to the WRFDA grid (and vice versa), which can be called by the WRFDA in full MPI parallel mode.

iii. Modify the WRFDA to allow it to call WRFPLUS with forcing data from the WRFDA and retrieve field data from WRFPLUS (e.g. gradients).

In this paper we only discuss the first development, the other two developments will be introduced in a separate paper.

The WRF model already has a well defined routine that advances the integration, which makes it fairly straightforward to be able to call it from an external model. New initialization and finalization routines had to be coded mostly to deal with tangent linear model and adjoint model. A namelist option *dyn_opt* was borrowed to allow WRFPLUS to decide which model will be advanced. *dyn_opt* = 2 will activate FWM, *dyn_opt* = 202 will activate TLM and *dyn_opt* = 302 will activate ADM.

The fully implemented interfaces as seen from WRFDA point of view looks like as below:

- **Components routines :** The interfaces to activate forward, tangent linear and adjoint model
  - **wrf_run :** Interface to run forward (nonlinear) model
  - **wrf_run_tl :** Interface to run tangent linear model
  - **wrf_run_ad :** Interface to run adjoint model
- **Data exchange routines :** The interfaces to exchange data between WRFDA and WRFPLUS
  - **read_xtraj :** Read the trajectories from FWM integration
  - **save_xtraj :** Save the trajectories from FWM integration
  - **read_tl_pert :** Read initial perturbation for TLM integration
  - **save_tl_pert :** Save trajectories of perturbation from TLM integration.
  - **read_ad_forcing :** Read adjoint forcing for ADM integration
  - **save_ad_forcing :** Save initial adjoint forcing from ADM integration

These interfaces are written not only for the coupling with WRFDA and they are designed for general coupling purpose. In addition to the coupling of WRFDA and WRFPLUS to construct WRF 4D-Var, we successfully coupled WRFPLUS with the community GSI to construct a GSI-based WRF 4D-Var (Zhang and Huang (2012)). Figure 4 shows the parallel performance of WRF 4D-Var run with 15KM resolution CONUS domain (not shown), This domain has 450X450 horizontal grids and 51 vertical levels. The assimilation window is 6 hours and the integration time step is 90s. Only GTS conventional data is assimilated. WRF 4D-Var shows impressive scalability, i.e. with the addition of more processors, the total performance of WRF 4D-Var increased accordingly. The implementation and coupling work quite well and it already replaced the old WRF 4D-Var system since the V3.4 release of WRF 4D-Var. Beyond the performance benefit, there are other advantages for the approach compared to the old WRF 4D-Var system. The execution of the new 4D-Var is simpler than the old since we do not need to launch a multiple program multiple data (MPMD) collection of different executables.

7. Conclusion

The implementation technique of the new tangent linear and adjoint of WRF ARW core has been discussed. With free AD tool-TAPENADE, the WRFPLUS has been developed and carried forward into later versions. Compared to WAMS, the new WRFPLUS has following improvements: 1) The tangent linear and adjoint codes have been maintained to be consistent with the latest WRF changes; 2) Improved parallelization strategy and efficiency; 3) Complete tangent-linear and adjoint checks are included to ensure the accuracy of the existing codes and new developed codes; 4) Module interfaces for coupling are constructed in WRFPLUS, which have led to a single executable WRF 4D-Var system with nice parallel performance and scalability.

8. Figures and tables

a. Figures

b. Tables

Acknowledgments.

The National Center for Atmospheric Research is sponsored by the National Science Foundation. This work is supported by the Air Force Weather Agency. The WRFPLUS system has benefitted greatly from close collaboration with Dr. Qiang Chen and his ADG tools. We thank Dong-Kyou Lee and Gyu-Ho Lim of Seoul National University for their comments on the manuscript and generous support through KUWCC. We thank Fuqing Zhang and

Jon Poterjoy of Penn State University for their valuable suggestions and comments on the manuscript.

REFERENCES

Barker, D., et al., 2012: The weather research and forecasting (wrf) model's community variational/ensemble data assimilation system: Wrfda. *Bull. Amer. Meteor. Soc.*, **93**, 831–843.

Cheng, B., 2006: A duality between forward and adjoint MPI communication routines. *Computational Methods in Science and Technology*, Polish Academy of Sciences, 23–24.

Errico, R., 1997: What is an adjoint model? *BAMS*, **78**, 2577–2591.

Errico, R. M. and T. Vukicevic, 1992: Sensitivity analysis using of the psu-ncar mesoscale model. *Mon. Wea. Rev.*, **120**, 1644–1660.

Gilmour, I., L. Smith, and R. Buizza, 2001: Linear regime duration: Is 24 hours a long time in synoptic weather forecasting? *J. Atmos. Sci.*, **58**, 3525–3539.

Hascoët, L. and V. Pascual, 2004: Tapenade 2.1 user's guide. Technical Report 0300, INRIA. URL `http://www.inria.fr/rrrt/rt-0300.html`.

Huang, X.-Y., et al., 2009: Four-dimensional variational data assimilation for wrf: Formulation and preliminary results. *Mon. Wea. Rev.*, **137**, 299–314.

Kleist, D. T. and M. C. Morgan, 2005a: Interpretation of the stucture and evolution of adjoint-derived forecast sensitivity gradients. *Mon. Wea. Rev.*, **133**, 466–484.

Kleist, D. T. and M. C. Morgan, 2005b: Application of adjoint-derived forecast sensitivities to the 2425 january 2000 u.s. east coast snowstorm. *Mon. Wea. Rev.*, **133**, 31483175.

Langland, R. H., R. Gelaro, G. D. Rohaly, and M. A. Shapiro, 1999: Target observations in fastex: Adjoint based targeting procedure and data impact experiments in iop17 and iop18. *Quart. J. Roy. Meteor. Soc.*, **125**, 3241–3270.

Li, Z., A. Barcilon, and I. M. Navon, 1999: Study of block onset using sensitivity perturbations in climatological flows. *Mon. Wea. Rev.*, **127**, 879–900.

Michalakes, J. and D. Schaffer, 2004: Wrf tiger team documentation: The registry. Technical report, INRIA. URL `http://www.mmm.ucar.edu/wrf/WG2/software_2.0/registry_schaffer.pdf`.

Navon, I. M., X. Zou, J. Derber, and J. Sela, 1992: Variational data assimilation with an adiabatic version of the nmc spectral model. *Mon. Wea. Rev.*, **120**, 1433–1446.

Rabier, F., E. Klinker, P. Courtier, and A. Hollingsworth, 1996: Sensitivity of forecast errors to initial conditions. *Quart. J. Roy. Meteor. Soc.*, **122**, 121–150.

Skamarock, W. C., J. B. Klemp, J. Dudhia, D. O. Gill, D. M. Barker, W. Wang, and J. G. Powers, 2005: A description of the advanced research wrf version 2. Technical report, NCAR Tech. Note NCAR/TN-468+STR, 100 pp. [Available from UCAR communications, P. O. Box 3000, Boulder, Co 80307-3000.].

Utke, J., L. Hascoët, P. Heimbach, C. Hill, P. Hovland, and U. Naumann, 2009: Toward adjoinable mpi. *Proceedings of the 10th IEEE International Workshop on Parallel and Distributed Scientific and Engineering, PDSEC'09.*

Vukicevic, T., 1991: Nonlinear and linear evolution of initial forecast errors. *Mon. Wea. Rev.*, **119**, 1602–1611.

Xiao, Q., X. Zou, M. Pondeca, M. A. Shapiro, and C. S. Velden, 2002: Impact of gms-5 and goes-9 satellite-derived winds on the prediction of a norpex extratropical cyclone. *Mon. Wea. Rev.*, **130**, 507–528.

Xiao, Q., et al., 2008: Application of an adiabatic wrf adjoint to the investigation of the may 2004 mcmurdo, antarctica, severe wind event. *Mon. Wea. Rev.*, **136**, 3696–3713.

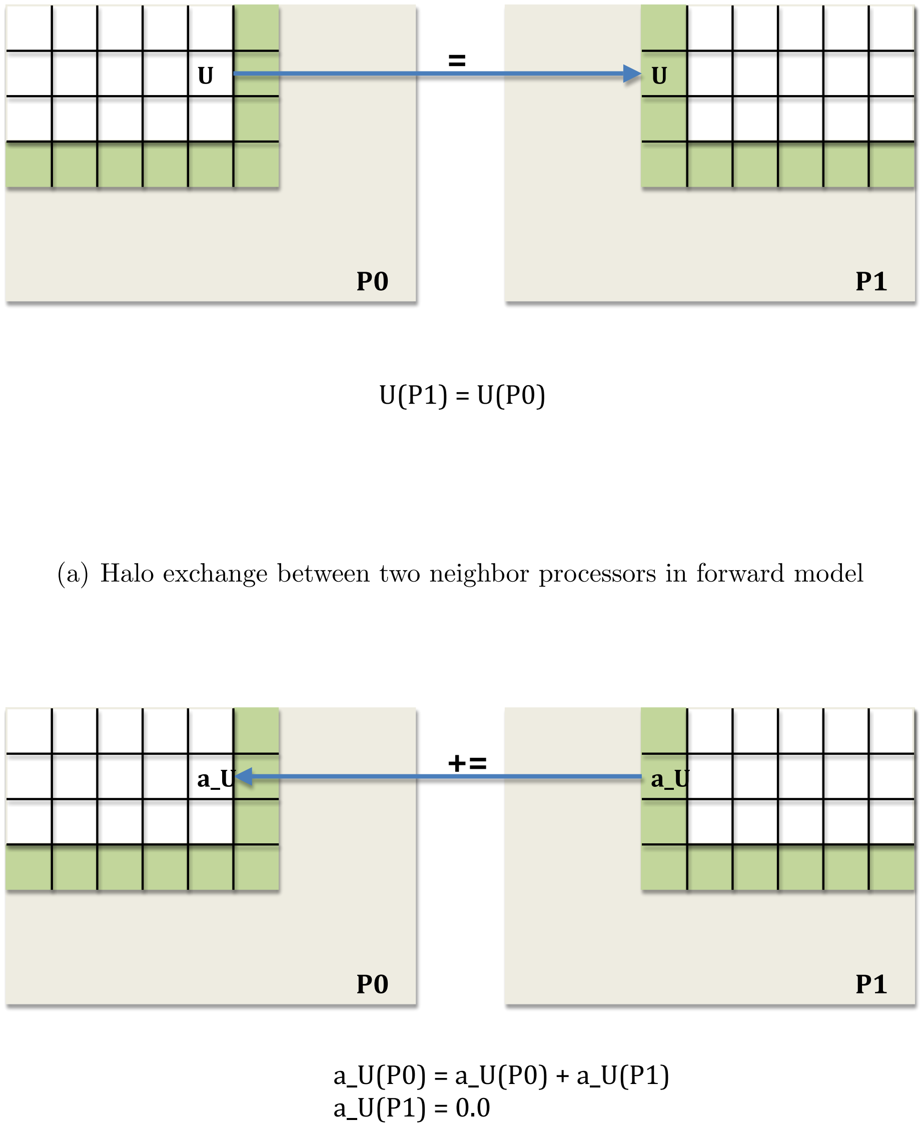
Zhang, X. and X.-Y. Huang, 2012: The development of regional wrf tangent linear and adjoint models and its applications in wrf 4d-var system and gsi based wrf 4d-var system. *92nd American Meteorological Society Annual Meeting*, New Orelean, USA, 000–000.

List of Tables

Table 1. Ratio of norms between the tangent linear forecasts and the differences of the two nonlinear model forecasts at 24 h. The norm is defined as the summation of the squares of all variables (perturbations of tangent linear model and difference of two nonlinear models) over the whole domain at 24 h. Here λ is the perturbation scaling factors of the initial perturbation.

λ	*Ratio*
.1000E+0	0.10114281198481E+01
.1000E-01	0.10008545240448E+01
.1000E-02	0.10000832484054E+01
.1000E-03	0.10000095806229E+01
.1000E-04	0.10000007503957E+01
.1000E-05	0.10000001743469E+01
.1000E-06	0.10000000344215E+01
.1000E-07	0.99999998551913E+00
.1000E-08	0.10000001453468E+01
.1000E-09	0.10000007302081E+01
.1000E-10	0.10000775631370E+01

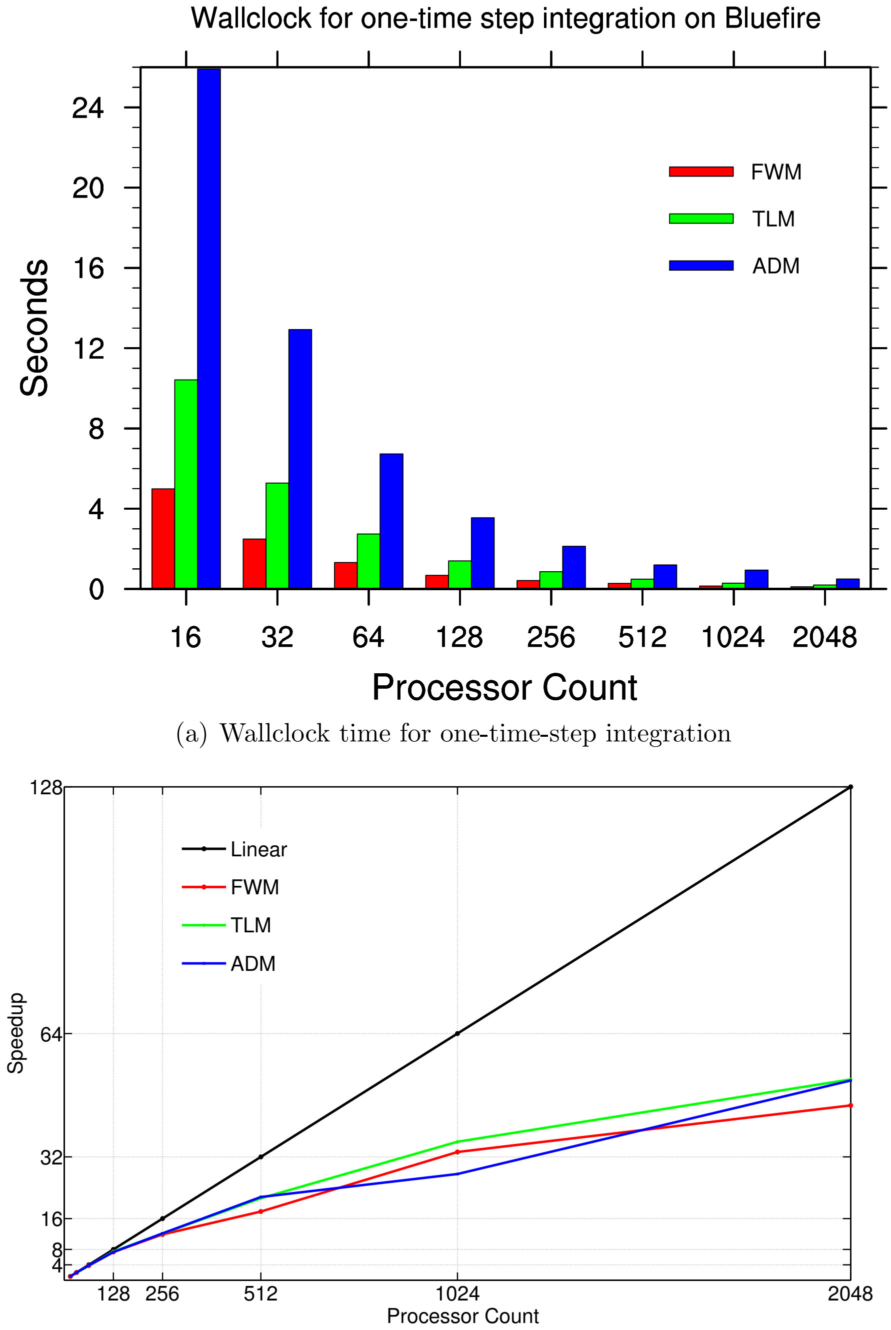
List of Figures



(a) Halo exchange between two neighbor processors in forward model

(b) Adjoint of halo exchange between two neighbor processors in adjoint model

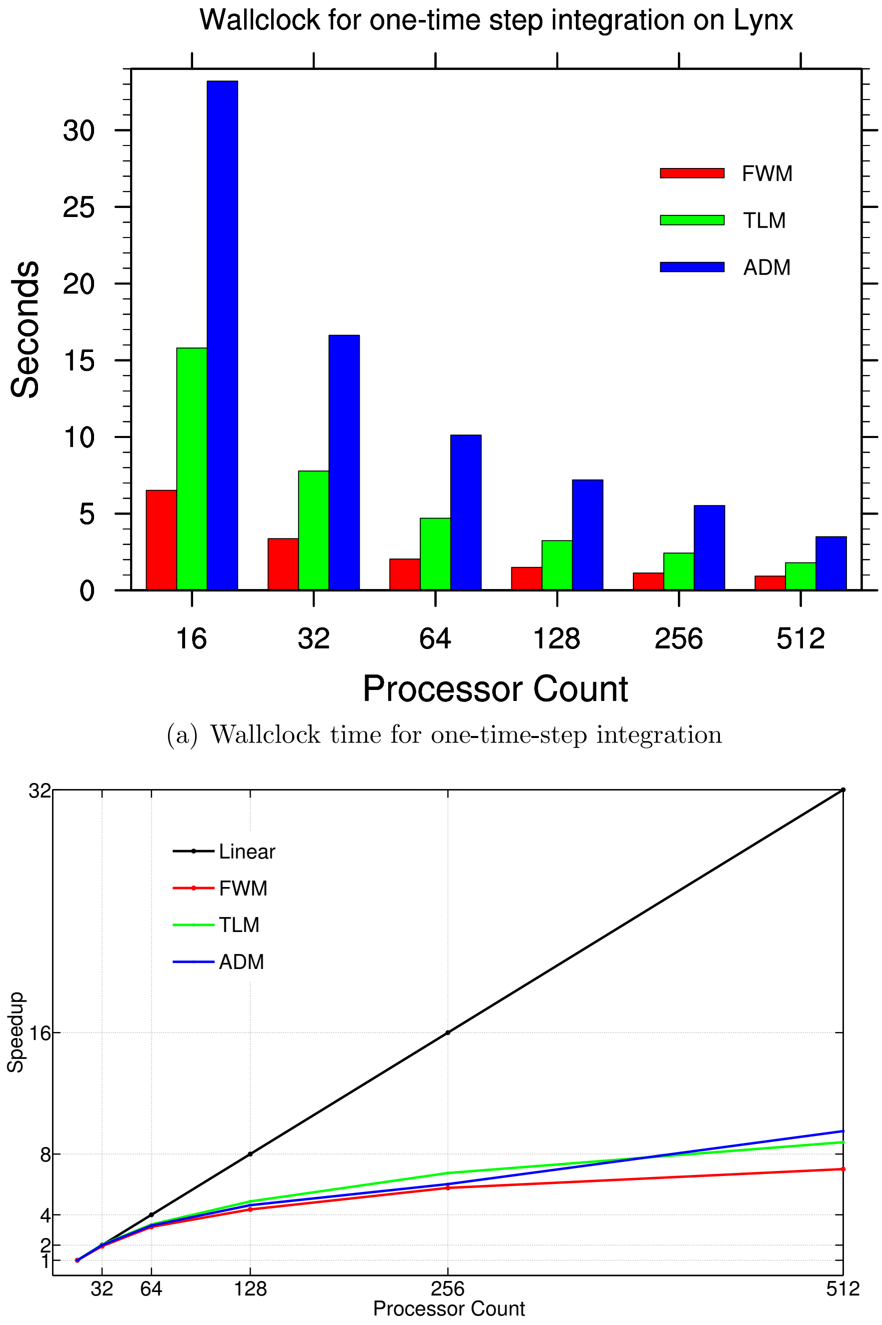
FIG. 1. Schematic diagram of the halo communication in FWM and ADM. Gray area denotes the entire model domain; Green zone denotes the ghost area; white zone is the computational patch for each processor; U denotes the basic state variable in FWM and a_U denotes the adjoint variable in ADM.



(a) Wallclock time for one-time-step integration

(b) Parallel speedup

Fig. 2. Parallel performance for one-time-step integration on Bluefire



(a) Wallclock time for one-time-step integration

(b) Parallel speedup

Fig. 3. Parallel performance for one-time-step integration on Lynx

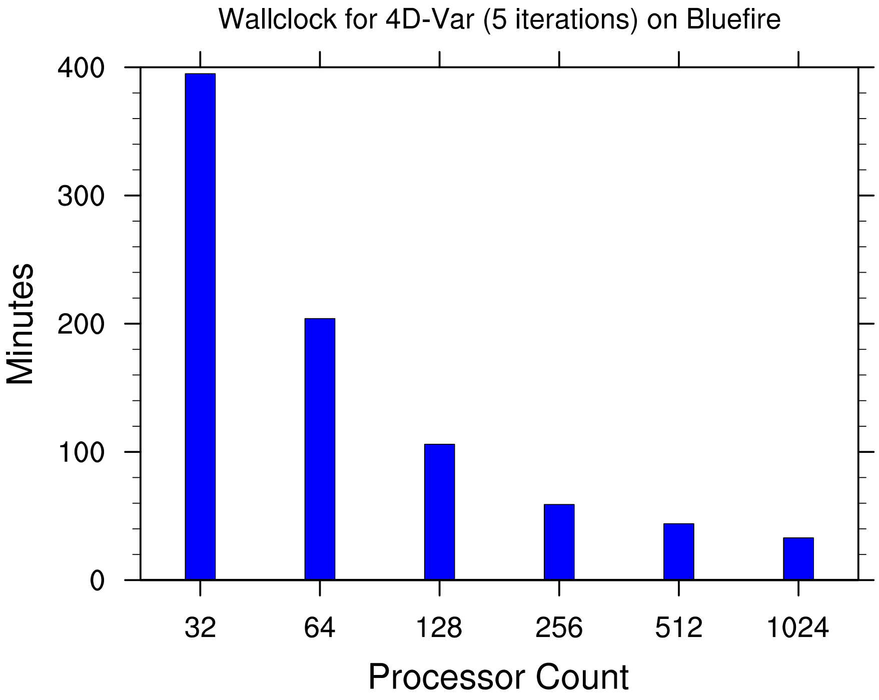


FIG. 4. Parallel performance for 4D-Var with 5 iterations on Bluefire