

Rules for Interchange of Physical Parameterizations

E. Kalnay,* M. Kanamitsu,*
J. Pfaendtner,[†] J. Sela,*
M. Suarez,[†] J. Stackpole,*
J. Tuccillo,* L. Umscheid,[‡]
and D. Williamson[§]

1. Preface

Progress in operational numerical weather prediction and atmospheric modeling has been slowed by the difficulties of testing new schemes developed at different centers, and even within a center, using codes which are not modular and have incompatible structures. To address this problem, we formed a working group to draft a set of rules to facilitate the interchange and testing of FORTRAN subroutines developed in different centers to parameterize atmospheric subgrid-scale physical processes. The goal of these rules is to make the "physics" routines easily transferable between models with only a few hours of work, and at the same time minimize the degradation in model efficiency.

Drafts of the rules were widely circulated among national and international modeling centers for comments. We received a very positive response, indicating that such an effort towards standardization is perceived as necessary and timely (Pielke and Arritt 1984). Several modelers sent detailed comments on the specific rules. Based on these comments we have expanded, modified, and clarified the rules, including where necessary a justification for their choice.

Several operational and research centers have indicated that they plan to follow these rules (or to violate them deliberately when necessary) in their future model development. These centers include (as of December 1988) the U.S. National Meteorological Center (NMC), the National Aeronautics and Space Administration's (NASA) Goddard Laboratory for Atmospheres (GLA), the Air Force Geophysical Laboratory (AFGL), the United Kingdom's Meteorological Office (UKMO), the Canadian Recherche en Prevision Numerique (RPN), the French Centre de Recherche en Meteorologie Dynamique (CRMD), the Brazilian Center de Previsao de Tempo e Estudos Climaticos (CPTEC), the New Zealand Meteorological Service and the Community Climate Model Development Group at the National Center for Atmo-

spheric Research (NCAR). The National Oceanographic and Atmospheric Administration's (NOAA) Geophysical Fluid Dynamics Laboratory (GFDL) plans to implement these rules when transferring physical packages. Many individual modelers have also indicated that they will follow the rules in their modeling research. The NMC has expressed interest in testing new physical parameterizations cast in this modular fashion.

The "plug compatibility" rules are meant to apply to physics packages that do not have a significant number of prognostic variables of their own. More complex submodels that do have a large number of state variables not used by the atmospheric model (e.g., ocean models, complex biosphere models) should be connected using a more complex system. We have assumed that the physical parameterizations take place in single vertical columns, i.e., they do not involve neighboring columns, and that vectorization in a horizontal direction is important and desirable if at all possible. It is likely that experience with the use of these rules in the field will suggest additional future improvements.

The following issues raised in the comments deserve to be addressed.

(1) *Enforcement of the rules, and the problem of efficiency versus compatibility.* These rules are designed to facilitate the interchange of physical packages and do not rule out violations designed to increase code efficiency during in-house operational use. However, we believe that conscious violations of the rules will still result in better and more exchangeable code than designs chosen independently.

(2) *Programming standards and code documentation.* As indicated by several comments, rules for plug compatibility should be complemented by rules for programming standards, especially with respect to structured programming and program documentation, which are particularly important for program maintenance. Particular sets of programming standards have been adopted at NMC, the European Centre for Medium-range Weather Forecasts (ECMWF), and other operational centers. It would be desirable to consolidate them in the future.

(3) *Multitasking.* We have assumed in these rules that multitasking is performed by the driver code at a higher level than the physics packages.

(4) *Cataloging.* David Williamson (NCAR) has

* National Meteorological Center, Washington, DC 20233
[†] NASA/Goddard Laboratory for Atmospheres, Greenbelt, MD 20771
[‡] NOAA/Geophysical Fluid Dynamic Laboratory, Princeton, NJ 08540
[§] National Center for Atmospheric Research, Boulder, CO 80307

volunteered to produce a catalog documenting plug compatible subroutines submitted by their authors, including a brief description of their use, references to more complete documentation when available, and a mailing address from where the subroutines can be obtained in machine readable form. The catalog will be updated and distributed periodically to interested centers or scientists. Please contact David Williamson at NCAR (P.O. Box 3000, Boulder, CO 80307) to be placed on the mailing list to receive the catalog and/or obtain information on how to contribute to it.

2. Basic coding rules for physics packages, December 1988

Rule 1: A package shall refer only to its own subprograms and the ANSI FORTRAN intrinsic functions.

This ensures that everything needed will be provided with the package. It would be desirable to have a set of standard utilities (for tasks such as the computation of saturation vapor pressure) to include under this rule.

Rule 2: A package shall provide separate set-up and running procedures, each with a single entry point. All initialization of static data must be done in the set-up procedure, and the running procedure shall not modify the static data.

This limit on the number of entry points is intended to simplify implementing the package. Multiple entry points can be useful if some tasks need to be performed only occasionally, but they generally make the package more difficult to use and more prone to implementation errors.

The requirement that the initialization of static data be separate from the running procedure is intended to simplify multi-tasking implementations.

The use of named common blocks shared by the two procedures is allowed, but if it is not too cumbersome, communication through the argument list via the calling program is preferable. If the argument list is used, both procedures will satisfy rule 3 and will behave as though they were separate packages.

Rule 3: All communication with the package shall be through the argument list at the entry points.

This serves to require a disciplined and easily documented exchange of information. Since we are proposing these rules for large physics packages, the

overhead of using the argument list should be small. This rule, of course, forbids the use of COMMON blocks or MODULEs in FORTRAN 8X for exchanging data between the package and the rest of the model. Communication between routines within the package is not affected by this rule.

Rule 4: The package shall not use blank COMMON.

The use of named COMMONs or MODULEs in FORTRAN 8X that are local to the package is allowed, but not recommended.

Rule 5: Arguments shall be clearly documented. In particular, data items shall be defined in physical terms, and identified as being: (1) needed on input and not changed, (2) needed on input and modified, (3) simply output, or (4) work space; and EXTERNAL subprograms shall be described in detail. All data shall be in SI units.

It is left up to the author whether the results are returned in the form of tendencies or updated values.

The passing of EXTERNAL subprograms is allowed with some misgivings, since the practice could easily violate the spirit of rule 1.

For the time being this is our only documentation rule. The definition of serious documentation standards should be undertaken as soon as possible.

Rule 6: The horizontal index shall be the innermost of FORTRAN arrays. The range of this index processed on each call (we will call it the "run") shall be specifiable through the argument list.

The exchange of packages will be much easier if all models organize the data the same way, at least for the physics. The choice made here is the simplest for vectorization, which in physics is usually over the run.

If a maximum run is used for dimensioning arrays local to the package, it should be specified as a PARAMETER and be easily accessible so that it can be easily modified. This should not be necessary once automatic arrays are available in FORTRAN 8X.

Rule 7: The number of levels the package uses (we will call it the rise) shall be specifiable through the argument list.

It should be easy to change the number of levels. In some applications—such as z-coordinate models—one may need to vary the number of levels at

execution time. As in rule 6, if a maximum number of levels is used for dimensioning arrays local to the package, it should be specified as a PARAMETER and be easily accessible.

This rule, of course, does not apply to parameterizations that use only fixed levels. The number of levels is not required for, say, a surface hydrology parameterization.

Rule 8: All dimensions of dummy argument arrays shall appear in the argument list.

This is to aid vectorization and to allow array-bounds checking.

The rules do not require that horizontal and vertical dimensions be either identical or distinct from the rise and the run; but the code should be such that the user can easily go from one specification to the other. If the horizontal dimensions appearing in the argument list are distinct from those of the run, the package can be easily called on horizontal subdomains without copying input data to temporary arrays. This would be useful in multitasking. On the other hand, a "distinct" specification would disable outer-loop vectorization. The choice is left to the author, but it should not be too hard wired.

Rule 9: No array index shall exceed its declared dimension.

This is for clarity and safety. The package should be able to pass an array-bounds check.

Rule 10: The package shall not use the STOP statement.

A "polite" package will notify the calling program of catastrophic events by using an error flag in the argument list and providing appropriate output describing the event.

Rule 11: I/O from the package shall be limited to diagnostic output written to FORTRAN units specified in the argument list.

This rule allows the user to redirect the output. In addition, it would be desirable to be able to shut off this output and bypass expensive diagnostic calculations.

References

- Pielke, R. A., and R. W. Arritt. 1984. A proposal to standardize models. *Bull. Am. Met. Soc.* **65**: 1082. ●