

第3章 数値解析予報実験システム (NAPEX)

3.1 数値解析予報実験システム (NAPEX) の必要性和その歴史¹

数値予報システムは非常に多くのプログラム (ジョブ) で構成され、多くの種類の入力データを必要とする。そのため、プログラムの相互依存性を考慮した実行 (ジョブスケジューリング)、入出力データのハンドリングが複雑になっており、開発に必要な実験システムの構築は容易ではない。また、複数のモデル開発者によってさまざまな開発実験が行われるようになっていく中で、基準となる実験設定 (用いる観測データ、モデル、実験期間など) を明確にして開発者間で統一することは、実験間の比較や評価をしやすくするために必要かつ重要なことである。

現在の数値予報モデルの開発手順では、予報モデルや同化スキームの変更をまずは単体で評価し、期待する結果が得られたことを確認して、データ同化サイクル実験を行うのが一般的である。単体での評価では良好な結果が得られても、データ同化サイクルを通じて予期しない悪影響が顕在化することがあるため、その変更の現実化の可否の判断に当たってはデータ同化サイクル実験が不可欠である。しかし、大野木・入口 (2003) によれば、1990年代には開発段階でデータ同化サイクルを簡便に実行できるシステムが存在しておらず、データ同化サイクル実験を行うには、それぞれのモデル開発者が数値予報ルーチンに組み込まれた環境を自分の開発環境にコピーし、それを手動で修正することによって実験を行っていた。その修正箇所は、基準実験のための環境構築だけでも、入力データの入力元や出力データの出力先を変更したり、数値予報ルーチン専用の記述を取り除いたりするなど、多岐にわたった。そのため、実験環境を誤りなく構築し、正しい実験を行うことは非常に手間のかかる作業であった。また、モデル開発者によって基準となる実験設定が異なっている場合が多く、モデル開発者間でお互いの実験結果を比較することにも困難を伴った。

そのような中で、1998年から2000年に欧州中期予報センター (ECMWF) に派遣されて再解析システムの開発に携わった数値予報課の職員が ECMWF におけるデータ同化実験システムや予報実験システムの充実ぶりを見て、気象庁のモデル開発における同様のシステムの必要性を強く認識し (大野木 2000)、帰国後の2001年に第7世代スーパーコンピュータシステム (2001年3月~2006年2月運用) が導入され OS が UNIX に切り替えられたのを機に構築したのが、数値解析予報実験システム (NAPEX: Numerical Analysis and Prediction EXperiment system) である (大野木・

入口 2003)。NAPEX は数値予報ルーチンで運用されている解析サイクル、モデル予測をモデル開発者の開発環境でも簡便に実行できるようにしたシステムであり、このシステムの登場によって、モデル開発者が実験システムの構築に煩わされることなく、そのシステムを用いた実験の評価に集中できるようになった。次節で説明するように、NAPEX はその後、さまざまな改良を経て²、今日でも開発基盤として最も重要なシステムの一つとして活用されている。

2001年に構築された初版の NAPEX (大野木・入口 2003) では、標準の実験環境が管理者によって準備されており、ユーザは指定の設定ファイルを編集して、環境構築のためのシェルスクリプトを実行すれば、その設定に従って標準の実験環境のユーザの開発環境へのコピー、および必要な修正が自動的に行われるようになっていた。また、バージョン管理を厳格に行っており、各モデル開発者の実験がどのバージョンの標準環境を用いたかを明確にして、実験間の比較がしやすいようにしている。これらの点は、15年経過した現在の NAPEX でも大きく変わることはない。一方、その当時の NAPEX では、標準の実験環境を管理者が構築する際に数値予報ルーチン特有の処理を取り除いたり、使用する計算機資源を節約するために、数値予報ルーチンで使用されているスクリプトに多数の修正を加えて利用していた。NAPEX のバージョン更新のたびに同様の作業を繰り返す必要があり、その管理コストは大きかった。

2006年3月に運用が開始された第8世代スーパーコンピュータシステム (2012年6月運用終了) への数値予報ルーチンの移行の際に、数値予報ルーチンの実行スクリプトの生成に JCL、Makefile の生成に PBF と呼ばれるファイルの記述法および関連ツールが導入された (JCL や PBF については第3.2節を参照)。それをきっかけに、NAPEX も数値予報ルーチンと同様に JCL と PBF を用いたシステムに移行し、かつ、数値予報ルーチンとして登録された JCL や PBF ファイルには極力、手を加えないこととする大きな変更がモデル開発者によって行われた。これは、前世代のスーパーコンピュータシステムと比べて開発に利用できる計算機資源が相対的に増加し、使用する計算機資源を節約するために高い管理コストを投じて数値予報ルーチンで使われているファイルを修正する意義が小さくなったことが背景にある。この変更により、数値予報ルーチンと NAPEX の間の差異が小さくなり、管理コストも小さくなった。また、検証システムなど、数値予報ルーチンにはない NAPEX 独自のジョブを簡単にブラ

¹ 原 旅人

² 2006年、2012年のスーパーコンピュータシステム更新のタイミングで NAPEX の大きな改良が行われている。

ゲインできるようになった。

その後、2008年から2010年まで英国気象局に派遣された著者が、英国気象局においては実験設定がデータベースで管理され、それを他のユーザが参照したり、コピーして同じ実験を行うシステムになっていること、また、そのようなシステムを構築するなどの開発支援を専門に行う部署があることを報告し(原・高谷 2013)、帰国後、気象庁のモデル開発においても同様のことが必要であると主張した。従来は、標準の実験環境は管理されているものの、それに対するユーザによる変更履歴の管理機能がNAPEX 本体には実装されていなかったため、どのような変更が加えられたのかはすぐには判然とせず、後日、同じ実験を行おうとしても再現できない場合すらあった。また、第 1.1 節で述べた通り、それまでの NAPEX の構築はシステム面での専門家ではないモデル開発者が行ってきたことから、先に述べたような、実験設定のデータベース管理とそれを簡単に利用できる機能、ユーザによる実験設定の変更履歴の管理機能、数値予報ルーチンシステムからの差異を小さくすることによるさらなる構築・管理コストの縮小など、システムの専門家が構築すればできたと考えられることが不十分になっている面もあった。

そのような状況を受けて、第 9 世代スーパーコンピュータシステム向けの NAPEX の開発には数値予報ルーチンシステムの専門家である数値予報課プログラム班(P 班)が参画し、数値予報ルーチンシステムで用いている技術を活用しながら、数値予報ルーチンの環境からの差異がより小さく、また、実験の履歴の記録や再現性の確保を兼ね備えた現在の NAPEX が構築された。数値予報ルーチン向けに開発されたツールを開発で利用するための拡張などが開発元の P 班で積極的に行われるようになり、モデル開発者とシステムの専門家がお互いの得意分野を活かしながら、数値予報モデルの効率的な開発に寄与している。数値予報課外でも、NAPEX は気候情報課が実施した長期再解析 JRA-25、JRA-55 のデータ同化実験・実行システムのベースとして導入され、その業務に大きく貢献している。

本章では、これまでに述べた NAPEX の歴史を踏まえ、現在用いられている NAPEX の設計思想および実装、管理および利用の実際について報告する。

参考文献

- 原旅人, 高谷祐平, 2013: 海外数値予報センターの開発管理の例. 数値予報課報告・別冊第 59 号, 気象庁予報部, 195-199.
- 大野木和敏, 2000: 開発環境の整備. 数値予報課報告・別冊第 47 号, 平成 12 年度数値予報研修テキスト合併号, 気象庁予報部, 134-137.
- 大野木和敏, 入口武史, 2003: 数値解析予報実験システム NAPEX. 気象庁測候時報, 70, 171-187.

3.2 現在の NAPEX の設計思想と実装¹

3.2.1 はじめに

NAPEX は 2001 年の運用開始 (大野木・入口 2003) から 15 年が経過し、その間に複数の管理者及びユーザの利用を通じて頻繁に改良が続けられ、現在では開発に不可欠な共通の実験基盤として広範囲に利用されるようになっている。

近年の NAPEX 開発史で最も大きな転機は、第 9 世代スーパーコンピュータシステム (西尾 2011) への移植作業時に訪れた。この移植時には第 8 世代スーパーコンピュータシステム (中山 2005) での NAPEX の利用を通じて、以下の項目が課題として挙げられていた。

管理コストが大きい

当時の NAPEX は実際に数値計算を行うためのプログラム群である数値予報システム本体のほかに、制御系とよばれる数値予報システムの実行制御を行う管理ツールが存在した。制御系は数値予報システムの大幅な変更が発生すると改修が必要となっており、特に初期値・境界値・観測データといった、数値予報システムの入力として利用される引継ぎデータの設定や、データ保存に関する部分では、毎度慎重な設定作業が要求され管理者負担が大きかった。これに対して数値予報ルーチン管理の現場では、こうした引継ぎデータ・保存データ処理の自動化が開発・実装されており、数値予報ルーチンの手法を用いることで制御系のメンテナンスを大幅に簡素化できる可能性が指摘されていた。

また、全球数値予報システム、週間・台風アンサンブル予報システム (週間・台風 EPS)、メソ数値予報システムで、NAPEX 環境が異なっており、個別管理を行う必要があったことから、統一的な NAPEX 環境によって管理コストを低減することが求められていた。

実験が再現できない

当時の NAPEX では制御系のバージョン管理はされていたものの、数値予報システム本体はバージョン管理されておらず、各ユーザが実行した実験の設定や実行プログラム、定数ファイルなどの管理はユーザに一任されていた。こうした背景からユーザが過去に行った実験を再現する必要性に迫られた際、管理が不十分で当時の実験設定を紛失してしまった場合、実験が再現できないという状況だった²。

こうした指摘を受け、現在の第 9 世代スーパーコンピュータシステム向けの NAPEX 構築では「数値予報ルーチンとの互換性確保」と「実験再利用性の確保」

¹ 雁津 克彦

² なお、週間・台風 EPS の NAPEX は全球数値予報システムやメソ数値予報システムの NAPEX より後に実装され、ユーザ実験にバージョン管理を導入したため、こうした状況が生じにくかった。

の二つの設計思想を中心に据え、NAPEX 環境の実装を行った。本節では現在の NAPEX の基礎となっているこれらの設計思想について紹介した後、実装方法について解説する。解説には数値予報ルーチンに関する知識が不可欠であることから、必要に応じて数値予報ルーチンに関する事項を説明する。

NAPEX は第 8 世代スーパーコンピュータシステム向け、第 9 世代スーパーコンピュータシステム向けなど、スーパーコンピュータシステムの更新により数値予報ルーチンの運用や開発環境が変更される際に仕様変更されている。これらを明示的に区別する正式な名称は存在しないが、説明の便宜上 NAPEX8, NAPEX9 のように、末尾に対象となるスーパーコンピュータシステムの世代数を付加することで以降の説明を簡略化したい。

3.2.2 基本思想

第 3.2.1 項で示した NAPEX8 での課題を踏まえ、現在運用している NAPEX9 では以下の二つを基本的な設計思想と位置付けて実装を行った。

数値予報ルーチンとの互換性確保

NAPEX9 では数値予報ルーチンで標準的に利用されている JCL (Job Control Language), PBF (Program Build File-format), JDF (Job Definition File) といった各種数値予報ルーチン用のツールを活用して環境を構築した。これによって NAPEX で実行する数値予報システムを、ほぼそのままルーチンで利用する、あるいはその逆を容易に実行できる。また、現在の数値予報ルーチンで利用されている JCL, PBF, JDF は、必要となるファイルの依存関係を解決することができ³、例えば JCL であればジョブを実行する際に必要な引継ぎデータの一覧を、PBF であれば実行プログラムのコンパイルに必要なソースファイルの一覧を、自動的に取得できるようになっている。NAPEX8 でも JCL や PBF の利用自体はされていたが、NAPEX9 ではこうした自動的に環境を構築できる機能も導入し、実験構築時の設定作業や実験環境の維持管理に掛かるコストを大幅に軽減できるようになった。

実験再利用性の確保

NAPEX9 では個人が設定した実験設定の全ての情報をデータベースとバージョン管理システムに記録し、任意の過去の実験を再利用できることを第二の基本思想とした。これにより NAPEX8 で問題となっていた過去実験の再現が保障されるとともに、自分が実施した実験に更なる修正を加えて新たな実験とすることも簡単にできるようになり、実験の再利用性が飛躍的に向上した。さらに、自分以外の実験も再利用可能とする

³ 数値予報ルーチンの場合はファイルから直接依存関係を解決するのではなく、ファイルの内容を一度 RENS (第 2.10.3 項) に格納してから解決している。

ことで、他の開発者の開発成果を引用して自分の開発成果とマージして新たな実験を行うといった、グループとしての開発に不可欠なプロセスを簡潔に行うことが可能となった。また、実験の設定だけでなく実験結果のデータを引き継いで自分の実験を行うことも簡単にできるようになった。例えば、他の開発者が実験した全球数値予報システムの結果を境界値として引き継ぎ、メソ数値予報システムの実験を実行することができる。このように、実験の再利用性の確保によって開発者間で開発成果を相互利用することが可能になり、開発効率の向上が図られた⁴。

これらの基本思想は、単に設計に関する思想というだけでなく、運用面においても重視されることとなった。例えば、NAPEXで構築済みの実験をユーザが後から自前で修正したり、実行期間を延長して実行したりすることができるが、NAPEXを利用しない修正では設定が記録されないため、基本思想の「実験再利用性の確保」を大きく損なってしまふ。結果として後に実験の再現が必要になった際に、他のユーザや開発者自身の時間を浪費することに繋がりがかねない。このため、NAPEXで設定した実験を独自に修正することは運用ルールで禁止している。

このように、単にシステム設計面にとどまらず、実験時の運用ルールでも上記基本思想が重視されており、各ユーザはルールを遵守して実験を構築することが求められる。

3.2.3 NAPEX 環境の実装方法

NAPEX9では数値予報ルーチンとの互換性に重きをおいて実装することになったことから、それまでNAPEX環境を整備していた数値予報課数値予報班に代わって、実際に数値予報ルーチン環境の整備を行っている数値予報課プログラム班が主体となって実装が進められた。新たに加えられた主な機能、すなわちNAPEX8環境までとの差異は主に以下の四つの項目に集約することができる。

- 全ての実験の設定をデータベースで管理する
- ジョブの管理や実行プログラムの管理に数値予報ルーチン管理の言語を用いる
- ファイル管理にバージョン管理システムを利用する
- 実験の実行に独自開発のジョブスケジューラを用いる

以下、順を追って見ていく。

⁴ 過去実験の再現には実験の設定だけでなく利用した引継ぎデータの存在が不可欠である。このため再利用性を完全に確保するには、引継ぎデータも保持しておくことが必要であるが、NAPEX9では容量の都合などで入力した引継ぎデータの保持は行っておらず、古いデータの場合は消去されてしまうことも多くあることから、今後の課題である。

(1) 実験設定をデータベースで管理する

NAPEX9の最大の特徴は実験設定をRDBMS (Relational DataBase Management System) を用いて管理している点であり、これにより過去の実験を容易に再現できるようになったのは非常に革新的であった。具体的にはPostgreSQLを用いて実験に必要な情報をデータベース上に保持することで、実験の再現を可能にしている。実験に必要な情報とは、実験を実行する期間、データの保存先、実行するサーバの指定、利用する引継ぎデータ、保存するデータの種類など、数値予報システム本体以外の各種設定を指す。

実装上の基盤となるのは、各個人の実行する実験に対して一意の「実験番号」を与えて区別する点である。ユーザが実験を行う際は新規に実験番号を取得して実験を構築していくわけだが、実験構築時の作業では設定内容の修正が頻繁に行われることが想定される。そこで個々の修正内容を記録するために実験番号とは別に「枝番」を用意し、修正が行われるたびに枝番が更新される実装としている。ユーザが設定した内容は実験番号と枝番に紐付けられ、データベース上に格納する仕組みとなっている。再利用時はこの仕組みを逆に利用して、過去に実験を行った際の実験番号と枝番をキーに、データベースに対してSQLコマンドを発行することで当時の実験設定を検索する。

SQLによるデータベースの検索や格納は、NAPEX環境を利用するためのツール (Ruby スクリプト) を通じて行う。ユーザが直接SQLコマンドを実行する必要はない。例えば「新規の実験番号を取得する」際は `napex_init.rb` というツールを、「実験の修正内容の記録」では `napex_setup.rb` というツールを利用することで、必要な情報がデータベースに登録されるようになっている。別の実験を再現する場合も、再現したい実験番号と枝番を手元に用意し、NAPEX環境の標準的なツールを利用することで、対象の実験設定を自分の環境で再現することができる。

過去に設定された全ての実験は、CGIを利用してイントラサーバ上に一覧が公開されている。ユーザはウェブブラウザによる閲覧で利用したい実験の実験番号と枝番を調べることができる。また、多くの開発者が共通して利用することを前提に構築された実験は「雛形実験」(第3.3.3項も参照)として、開発管理サーバ上のWikiのリンクなどを通じて実験番号と枝番を広く周知する運用を行っている。

NAPEXで必要となる実験の設定内容は多岐に渡ることから、データベースには設定保持のためのテーブルを複数用意している。利用されているテーブルの一例を表3.2.1に挙げる。テーブルアクセスの応答速度は、テーブルの設計方法、とりわけインデックスの作成方法で大きく変化する。NAPEXのデータベース設計では頻りに利用される実験番号と枝番を中心に、各テーブルの主要なカラムに対しインデックスを作成す

表 3.2.1 NAPEX9 で実験情報を保存するために利用しているテーブルの例

テーブル名	格納されている主要な情報 (抜粋)
exp_control	基本設定 (実験名、ユーザ名、実行ディレクトリ等)
exp	実験設定 (実行期間、保存先ディレクトリ等)
prec_exp	引継ぎ設定
set_exp	詳細設定 (実行ホスト、スケジュール等)
jcl	利用する JCL ファイルの情報
pbf	利用する PBF ファイルの情報
jdf	利用する JDF ファイルの情報
const	利用する定数ファイルの情報
var	保存するデータの設定

ることで高速化を実現している。

(2) ジョブの管理や実行プログラムの管理に数値予報ルーチン管理の言語を用いる

NAPEX9 では数値予報ルーチンのツールを利用した管理が徹底されるようになった。数値予報ルーチンではジョブの管理や実行プログラムの管理に、以下に挙げる独自開発の言語を利用している。

JCL (Job Control Language)

JCL は気象庁が開発したジョブを記述するための言語である⁵。

Unix 系マシンでのジョブの記述は通常シェルスクリプトが利用される。気象庁でも 2001 年の第 7 世代数値解析予報システムへの更新時に OS が Unix 系 OS (HI-UX/MPP) に変更となり、ジョブ制御にシェルスクリプトが採用されるようになった (平 2001)。しかし、シェルスクリプトは高機能で記述の自由度が高いためループや条件分岐を手軽に利用でき、複雑な動作をするジョブを簡単に書いてしまう。これにより機械的なジョブ内容のチェックが困難で、数値予報ルーチンの管理上弊害があった。また、データハンドリングのような数値予報ルーチンで定型的に利用する処理に複数行の記述が必要であり、ミスを誘発しやすい問題もあった。そこで第 7 世代数値解析予報システム運用中に、数値予報ルーチン用のジョブを記述するため独自開発したのが JCL である。

⁵ 本来 JCL とはメインフレームに代表される企業向け汎用計算機で、バッチ処理を記述するスクリプト言語を指す。気象庁でも第 6 世代数値解析予報システムまでは、ジョブ制御言語として本来の JCL が採用されていた。第 7 世代数値解析予報システムで一時シェルスクリプトが利用されたものの、本文中のような問題が発生したため、本来の JCL を参考に気象庁で独自開発したものが JPP/JCL と呼ばれる言語である。JPP/JCL はその後改名され、現在は JCL が正式名称となっており、本文中の JCL はこちらを指す。

JCL はシェルスクリプトと比較して簡素な記述が特徴であり、数値予報ルーチン用途の定型的な処理を簡潔に記述することができる。また、構文構造がシンプルであり、構文解析器を相対的に低コストで実装することができる。現在では jcl2sh.rb という Ruby スクリプトで JCL からシェルスクリプトへの変換を行い、変換後のシェルスクリプトでジョブ制御を行うという運用を行っている。

さらに、JCL は構文解析の時点でジョブが必要とする実行プログラムと、実行時に必要な引継ぎデータ、定数ファイルの一覧を得ることができる。このことは、ジョブに必要な各種ファイルの用意を自動化できることを意味し、実行時に必要な引継ぎデータを一括して準備するシェルスクリプト (全てのジョブに先駆けて実行されることから START と呼ぶ) を自動で生成できるようにになっている⁶。

PBF (Program Build File-format)

PBF は実行プログラムをソースコードから生成するルールを記述するため、気象庁が開発した言語である。

実行プログラムを一定の規則に基づいてソースコードから生成する手法としては、Makefile を用いる方法が一般的である。実際に第 7 世代数値解析予報システムでは、数値予報ルーチンの実行プログラムの生成に Makefile を利用していた (平 2001)。一方で、数値予報ルーチンでは実行プログラムのディレクトリ構成や利用するコンパイラなどに運用上一定のルールを定めている。このことからシェルスクリプトの事例と同様、ユーザが複雑な生成規則を利用できることは管理上望ましくない。そこで複雑な生成規則を抑制するとともに、プログラム生成規則に関する各要素を簡単に取り出せるようにすることを目的に、第 8 世代スーパーコンピュータシステムの運用に向けて開発されたのが PBF である。

JCL からシェルスクリプトを生成できるように、PBF からは Makefile を生成することができ、生成された Makefile を用いて数値予報ルーチンの実行プログラムがコンパイルされる。また、PBF は Makefile と同様にコンパイル時に必要なソースコード間の依存関係を解決することが可能であり、PBF があれば実行プログラムを生成するために必要なソースコード一覧を自動的に作成することができる。

この他、数値予報ルーチンでは実行プログラムをスクリプト言語で書くこともできる。スクリプト言語で実行プログラムを作成する場合はコンパイル不要であるが、数値予報ルーチンで利用する実行プログラムを一律に管理するという観点から、利用するスクリプトファイルを PBF に記載することになっている。このた

⁶ 数値予報ルーチンの場合、正確には直接 JCL から START を生成するのではなく、一度 JCL の内容を RENS に登録した後、db2auto.rb で START 用の JCL を出力し、jcl2sh.rb で START を作成するという手順を踏む。

めスクリプト言語で記載された実行プログラムについても、PBFを通じて必要なファイルが把握できるようになっている。

JDF (Job Definition File)

JCL がジョブの動作を記述する言語であったのに対し、JDF は複数のジョブが集まったジョブグループ (JG: Job Group) の動作やジョブの環境を記述するために気象庁が開発した言語である。

ジョブと JG の関係の一例として全球サイクル解析の例を挙げる。全球サイクル解析では以下のような処理を行っている。

1. 解析に必要な入力データを集める
2. 前回の解析結果から予報を実行して第一推定値を作成する
3. 観測データの品質管理を行う
4. 品質管理済みデータと第一推定値を用いて 4 次元変分法によるデータ同化を行う
5. 結果を保存する

個々の処理に相当するのがジョブ、これらの処理全体が JG に相当する⁷。ジョブが必要とする環境はそれぞれ異なり、4 次元変分法の実行や予報の実行では複数のノードを利用した大規模なプログラムを十数分間実行する必要があるが、観測データの品質管理ではノード数の小さな数分程度のプログラムを並列に多数実行することになる。こうしたノード数や実行時間などのジョブに関する資源割り当てや、各ジョブの実行順序などの依存関係を記述するのが JDF である。

さらに、JG やジョブの依存関係を制御してジョブを実行することができるジョブスケジューラ ROSE (第 2.10.2 項) を数値予報課プログラム班が主体となって開発を進めており、JDF はジョブスケジューリングそのものを記述できる言語として、利用目的が拡大している。

これらのツールはいずれも自主開発の言語で構文解析器が整備されており、設定ファイルから依存するファイルを自動で解決できる特徴がある。こうした機能を NAPEX に導入することで、多くの処理が自動化され管理コストの削減に繋がった。

(3) ファイル管理にバージョン管理システムを利用する

実験の各種設定はデータベース上に保存されるが、JCL, PBF, JDF といった各種ファイルの内容をデータベース上に直接保持すると、レコード数が膨大となり問合せ時のレスポンス低下によって実験構築のボトルネックとなることが懸念される。そこで NAPEX ではデータベース上に直接ファイルの内容を登録することは行わず、バージョン管理システム Subversion のリ

⁷ 説明のために簡略化しているが、実際はこれより複雑なジョブ構成となっている。例えば観測データの品質管理は、一つのジョブではなく複数のジョブに分割して実行している。

ポジトリにコミットする実装としている⁸。まず、実験番号とユーザ ID の情報を元に Subversion のリポジトリに新規のブランチを作成し、このブランチに必要なファイルをコミットする。次に、ブランチのパスとコミット時に与えられたリビジョン番号を実験番号・枝番と紐付ける形でデータベースに登録する。これによってデータベースとリポジトリが関連付けられることになる。コミットするファイルは、前述の JCL, PBF, JDF に加え、PBF が利用するソースコードとスクリプトファイル、プログラム実行時に必要となる定数ファイルも対象となる。

定数ファイルの多くはバイナリ形式であり、ファイルサイズが大きいという特徴がある。Subversion はテキストファイルについては差分情報が保存されるが、バイナリファイルは差分ではなくファイル全体が直接保存されるため、サイズが大きなファイルを複数コミットすると、リポジトリの実体サイズが急増する危険性がある。幸い、数値予報システムで利用する定数ファイルの多くは、他の実験で利用した定数ファイルをそのまま利用することが多い。そこで NAPEX では、他の実験を引用して実験を行う場合、明示的に変更するファイル以外はリポジトリにコミットせず、過去のデータベースのレコードをそのまま流用する仕様としている。これにより、直接リポジトリにコミットする機会は僅かとなり、実体サイズが急増する危険性は低くなっている。

(4) 実験の実行に独自開発のジョブスケジューラを用いる

JDF の項目で述べたが、数値予報課では独自開発のジョブスケジューラ「ROSE」の整備が進められており (第 2.10.2 項)、NAPEX9 では実験用のジョブスケジューラとして採用している。

NAPEX8 までのジョブスケジューリングは、数値予報システムによって手法が異なり、全球数値予報システムや週間 EPS では欧州中期予報センターが開発した SMS (Supervisor Monitor Scheduler) をジョブスケジューラとして採用し、メソ数値予報システムではシェルスクリプトでジョブスケジューリングを行っていた。しかし、数値予報ルーチンと NAPEX で JG を別途管理する必要があったこと、SMS の開発が既に停止していること、各システムで統一的なスケジューリングが利用できる方が複数システムを利用するユーザにとって望ましいことなどを踏まえ、NAPEX9 からは自主開発の ROSE を統一的に採用することとなった。ROSE への移行によって、数値予報ルーチンで利用されている設定をある程度流用して JG の設定を行うことができるようになり、この点でも NAPEX 管理にかかるコストを抑えることができるようになっている。

現在、ROSE はバッチジョブのキューイングで

⁸ 数値予報ルーチンでも同様である。

LoadLeveler を始めとする複数のキューイングシステムに対応しており、移植性がかなり高くなっている。これによって、多様な機器で数値予報システムを実行できると期待されている。現在の ROSE は BCP サーバ（第 2.10.2 項）及び NAPEX で利用されているが、2018 年度に運用開始が予定されている第 10 世代スーパーコンピュータシステムでは数値予報ルーチンの運用にも利用される予定となっており、今後移植される NAPEX10 でも引き続き ROSE が利用される予定である。

3.2.4 まとめ

現在の NAPEX9 では中心となる基本思想を最初に設定した上で、必要となる各種技術を組み合わせて実装を行った。これによって NAPEX8 で課題とされていた点の多くが改善され、より利便性の高い実験システムとなっている。加えて数値予報ルーチンとの互換性が向上することによって、NAPEX 自体の管理にかかるコストを低減するとともに、数値予報ルーチンとの相互利用も以前より容易になっている。

参考文献

- 中山博義, 2005: 新数値解析予報システムについて. 平成 17 年度数値予報研修テキスト, 気象庁予報部, 10-11.
- 西尾利一, 2011: 計算機 (スーパーコンピュータシステム). 平成 23 年度数値予報研修テキスト, 気象庁予報部, 68-70.
- 大野木和敏, 入口武史, 2003: 数値解析予報実験システム NAPEX. 気象庁測候時報, 70, 171-187.
- 平隆介, 2001: NAPS の数値予報運用スケジュールと実行システム (3.4). 気象庁測候時報, 68, 53-56.

3.3 NAPEXの管理¹

3.3.1 はじめに

NAPEXは数値予報システムの開発で広汎に利用されていることもあり、適切な管理が重要となっている。数値予報課ではNAPEXの管理を目的として2002年度からNAPEX管理グループを組織し、NAPEXの管理に係る作業の調整を行っている(大野木・入口 2003)。現在の参加メンバーは、NAPEXのデータベースや実験設定ツールといったNAPEX環境の開発を担当しているプログラム班の担当者、NAPEX環境で動作させる数値予報システム(NAPEXモデルと呼ぶ)の管理を担当している数値予報班基盤整備グループの担当者、数値予報班の各グループ(全球・台風グループ、メソモデルグループ、観測データ処理グループ)の担当者及びアプリケーション班の担当者となっている。

NAPEXの管理はNAPEX環境の管理とNAPEXモデルの管理に大別される。NAPEX環境はスーパーコンピュータシステムの更新に連動して開発するため、新しいスーパーコンピュータシステムへの移植が終わると基本的に不具合修正を除き大規模な開発は行われな。それに対し、NAPEXモデルは数値予報システムの更新が計画的に実施されているため、定期的にメンテナンスを行うことが必要となっており、スーパーコンピュータシステム更新時を除き、NAPEX管理作業の大半はNAPEXモデルの管理が占めることになる。このため本節ではNAPEXモデルの管理を中心に解説する。

3.3.2 NAPEXモデルの管理

NAPEX8まではNAPEXモデルの管理に大きなコストがかかっており、現業的に実行される数値予報システムである数値予報ルーチンの変更申請とは別に、NAPEXモデルの変更にも申請が必要であった。NAPEX9ではNAPEXモデルと数値予報ルーチンとの互換性が向上したこともあり(第3.2節)、NAPEXモデルの変更申請を廃止し、数値予報ルーチンの変更申請の状況を鑑みながら数値予報班内での協議を踏まえ、NAPEX管理グループの担当者が計画的に更新作業を行っている。2017年1月時点でのNAPEXモデル一覧を表3.3.1に示す。

数値予報システムに変更が生じた際は、変更内容を対象となるNAPEXモデルに反映させる作業が必要となる。作業は大まかに、引継ぎデータの確認と、NAPEXモデル本体の更新の二つに分かれる。

(1) 引継ぎデータの確認

NAPEX9ではJCL(第3.2.3項(2)参照)の機能を用いて実験開始時に必要な引継ぎデータを自動で取得することができるため、多くの場合は特に設定を行う必

表 3.3.1 2017年1月時点のNAPEXモデル一覧。サポートが終了したものは除く。

NAPEXモデル	内容
Dc	観測データデコード処理
Da	全球サイクル解析
Df	全球サイクル解析からの予報
Ea	全球速報解析
Ef	全球速報解析からの予報
Vn	全球検証スコア作成
Ma	メソ解析
Mf	メソ予報
La	局地解析
Lf	局地予報

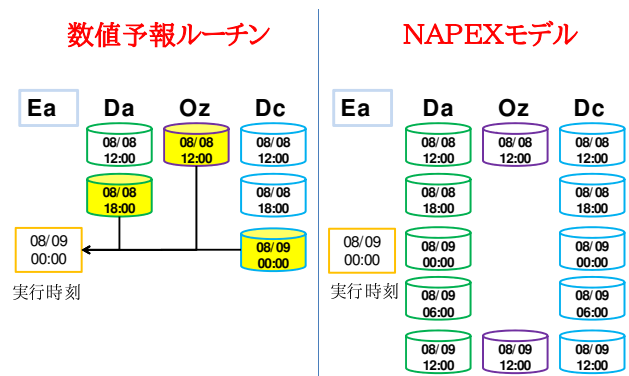


図 3.3.1 数値予報ルーチンとNAPEXモデルにおける最新データの引継ぎ概念図。Eaの実行にDa, Oz(オゾン解析), Dcの最新データが必要であると仮定する。数値予報ルーチンはリアルタイムで実行するため、Da, Oz, Dcの最新データが黄色で示したデータであると容易にわかる。しかし、NAPEXモデルは過去の実験を行うため、当時の最新データがどの時刻のデータなのかを判別しなければならない。

要がない。ただし、数値予報システムで新規に利用開始されるデータについては、自動取得の対象となるように登録作業を行う必要があり、こうした場合はNAPEX管理グループの担当者が登録作業を実施している。

NAPEX特有の事情に由来する作業もある。図3.3.1は数値予報ルーチンとNAPEXモデルでの最新データの引継ぎを示した概念図である。数値予報ルーチンの場合、実行時にどのデータが最新のデータであるかは、ほぼ自明である。しかし、NAPEXモデルの場合は過去の実験を行う必要があるため、どの時刻のデータが当時の最新データだったかを判別しなければならない。この判別にはNAPEXデータベースに登録された実行時刻の情報が利用されるため、必要に応じてNAPEX管理グループの担当者が時刻の前後判別に関する設定を調整する必要がある。

¹ 雁津 克彦


```

% if NAPEX
  dd out nusdas=17 data=jpp:/jgdir/Fcst/fcst_phy3m_1lpp.nus &
    nusdef=jpp:/const/Gnaps/<%=prefix1%>nusdef/Comm/<%=prefix2%>phy3m_1lpp.def
  dd out nusdas=18 data=jpp:/jgdir/Fcst/fcst_phy2m_1l.A.nus &
    nusdef=jpp:/const/Gnaps/<%=prefix1%>nusdef/Comm/<%=prefix2%>phy2m_1l.def
  dd out nusdas=19 data=jpp:/jgdir/Fcst/fcst_physub_1l.A.nus &
    nusdef=jpp:/const/Gnaps/<%=prefix1%>nusdef/Comm/<%=prefix2%>physub_1l.def
% end

```

図 3.3.2 Ef での JCL による NAPEX 専用処理の記載例。if NAPEX から end までの処理は、NAPEX で実行する場合だけ利用される。この例では出力するファイルを追加している。

(2) NAPEX モデル本体の更新

NAPEX モデルは基本的に数値予報ルーチンと同じものが登録される。基本的に、と述べたのは実際には異なる部分が存在するためである。

数値予報ルーチンを実行する場合は、障害などで観測データや初期値のデータが存在しない場合でも、システムが停止することなく動作することが求められる。こうした場合、データの出力自体は行われるが本来とは異なる予測特性を持つデータが出力される。しかし、実験では新たに開発した数値予報システムが現在の数値予報システムと比較して統計的に改善されているか、あるいは予測特性にどういった変化が生じているかを評価することが重要で、予測特性が異なるデータの混入は望ましくない。また、実行に異常が生じた場合は、異常時の状態が上書きされないように実験を停止して、数値予報システムの安定化のために、異常の原因について調査することも重要である。したがって、数値予報ルーチンとは逆に実験が継続しないことが望ましい。

実行時に異常が検知された場合、数値予報ルーチンでは関係者に警告メールを送り、後続処理を行うなど障害対応のための構成が不可欠であるが、こうした処理も実験では不要である。業務用のデータ配信処理が含まれる場合も、誤って実行されないよう停止しておく必要がある。この他、開発ではモデルを改善するための調査として数値予報ルーチンでは出力されない情報を追加で出力したり、本来とは異なる調査用のモジュールを実行する場合がある。NAPEX モデルでは、こうした実験用の設定も追加しておく必要がある。

このように、数値予報ルーチンと NAPEX モデルで異なる処理を行うために、NAPEX9 では数値予報ルーチン用の設定と NAPEX モデル用の設定を自動的に切り替えられるよう配慮している。参考として Ef での JCL の記述例を図 3.3.2 に挙げる。この例では NAPEX モデルの場合だけ特定のファイルを出力する処理を記述しているが、JCL に if NAPEX という文を挿入することで一連の処理が NAPEX モデルの場合だけ利用されるようになっている。こうした機能を活用することで、数値予報ルーチンと NAPEX モデルとで異なるシェルスクリプトが生成され、適切に実験専用の処理を実行できるようになっている。また、数値予報ルーチンには存在しない実験専用の定数ファイルや実行プログラ

ムを追加する場合には、登録したファイルが NAPEX 専用のファイルであることをデータベース上のフラグの有無で判別できるようになっている。

数値予報システムの更新では新規に機能が追加されることがあるが、数値予報ルーチンの処理だけ記述され、NAPEX モデル用の処理が含まれない場合がある。こうした際は、NAPEX モデル用の処理と自動判別のための設定を NAPEX 管理グループの担当者が手作業で導入するなどして、ユーザが誤って数値予報ルーチン用の処理を実行しないよう、NAPEX モデルの管理を行っている。

3.3.3 コントロールデータと雛形実験設定

前項でも述べたが、実験では新たに開発した数値予報システムが現在の数値予報システムと比較して統計的に改善されているか、あるいは予測特性にどういった変化が生じているかを評価することが重要である。評価時の比較対象としては一見すると現業的に実行されている数値予報ルーチンの結果が利用できると考えられる。しかし多くの場合、数値予報ルーチンの結果は比較対象として適切ではない。

統計的に改善しているかを確認するためには、比較対象の予測特性が均質である必要がある。ところが、数値予報ルーチンは開発成果の反映や不具合修正などで随時変更が行われるため、程度に差はあれ変更の前後で予測特性に変化が生じることになる²。こうした理由により数値予報ルーチンの予測特性は一定とならず、比較対象として適切ではない。

そこで NAPEX では同じ数値予報システムを用いて過去の一定期間を計算し、予測特性が均質なデータを作成している。このデータのことをコントロールデータと呼ぶ。開発者が NAPEX で実験した結果を評価する際はコントロールデータとの比較を行う。このため開発者は、コントロールデータが作成された期間と同じ期間で実験を行うことが求められるが、過去の実験を行う場合、特に解析実験を行う場合は観測データの利用で煩雑な設定が必要となる場合がある。利用開始から間もない観測データの場合には過去のデータが存

² この他、障害で初期値や観測値などのデータが存在しない場合も、業務継続のための代替処理が行われ、通常とは異なる予測特性を持つ結果が作成されるが、頻度としては稀。

在しない、又は存在しても通常と異なる場所に保存されていることがある。こうした場合、過去のデータを適切に利用するための設定を実験に組み込んでおく必要があるが、設定すべき項目が多くミスも生じやすい。そこで、過去の観測データを適切に利用できる実験をNAPEX 管理グループの担当者が予め用意し、開発者が設定を行わなくて済むよう簡素化を図っている。このように、過去の観測データに関する設定を施し、広く開発者に利用されることを想定した実験のことを雛形実験と呼ぶ。前述のコントロールデータは、この雛形実験をベースに実行することで作成されている³。

3.3.4 まとめ

NAPEX モデルの管理においては、最新の数値予報システムを用いて実験できるよう NAPEX モデルを整備するとともに、開発者が実験を行う上で標準的に利用するコントロールデータと雛形実験の提供を適宜行っている。

NAPEX モデルの管理作業とユーザの開発の流れを時系列でまとめると次のようになる。

1. 開発者が作成・変更した数値予報システムを、数値予報ルーチンの担当者が数値予報ルーチンに反映する
2. NAPEX 管理グループの担当者が、反映後の数値予報ルーチンを NAPEX 環境用に修正して NAPEX モデルとして登録する⁴
3. NAPEX 管理グループの担当者が、NAPEX モデルを過去の実験期間でも実行できるように修正した雛形実験を作成する
4. NAPEX 管理グループの担当者が、雛形実験を実行してコントロールデータを作成する
5. 開発者は雛形実験を利用して自分の実験を構築・実行し、実験結果とコントロールデータの比較を行う
6. 数値予報課内で新たな数値予報システムを数値予報ルーチンに導入することが合意されれば、開発者は 1. を行うための申請を行う

このように、現在では数値予報モデル開発において、NAPEX モデルの管理業務が開発サイクルの一部に組み込まれており、効率的な開発を行うためにも適切な NAPEX モデルの管理が重要となっている。

³ Da, Df, Ea, Ef などでは、雛形実験で保存するデータ種別をコントロールデータより少なく設定している。これは大量のデータを出力する設定で全ユーザが実験を行うと、ストレージを逼迫するおそれがあるためである。こうした事情から、コントロールデータを作成する実験と雛形実験の設定には異なる箇所が存在する。

⁴ 開発スケジュールによっては、数値予報ルーチンに反映する前の段階で NAPEX モデルを構築する場合もある。

参考文献

大野木和敏, 入口武史, 2003: 数値解析予報実験システム NAPEX. 気象庁測候時報, 70, 171-187.

3.4 NAPEXによる実験の実行手順¹

3.4.1 はじめに

本章でこれまで述べられてきたように、開発者はNAPEXを利用することで、単発実験やサイクル実験を容易に実行、再現することが可能となっている。

本節では、実際にNAPEXを利用するに当たっての方法などについて述べる。まず第3.4.2項で実験をそのまま再現して実行するための手順について述べ、第3.4.3項でさらにそこから変更を加えた実験を設定する方法について示す。第3.4.4項では、設定した実験を実行するためのジョブスケジューラROSE（詳細は第2.10.2項参照）について、ユーザの視点から説明する。第3.4.5項に本節のまとめを記述する。

3.4.2 コントロール実験の再現

数値予報課で数値実験を行う場合、通常は基準となる実験（コントロール実験）と何らかの変更を加えた実験（テスト実験）の差を比較、評価する。テスト実験を行う場合、そもそも基準となるコントロール実験を再現することができなければ、コントロール実験との正しい比較を行うことはできないだろう。

第3.2.3項で述べられているように、NAPEXでは各実験に対して実験番号と枝番が一意に設定されるが、それはコントロール実験についても同様である。コントロール実験の実験番号と枝番についてはイントラネット内のウェブページにまとめられており、開発者は基準としたいコントロール実験の実験番号と枝番を容易に取得することができる。

あるコントロール実験をそのまま再現して実行する手順は以下の通りである²。

手順1. 参照するコントロール実験の実験番号と枝番のほか、自分のユーザID、実験登録時の作業ディレクトリ、実行プログラムや定数ファイル（地形データ、パラメータファイルなど、実験実行時刻で内容の変わらない入力ファイル）の格納ディレクトリなどを設定ファイルに記述し、基本初期設定用のコマンドを実行することで、実験を初期化する。

手順2. 実験の期間、実行ディレクトリ、出力ファイルの保存ディレクトリなどを設定ファイルに記述し、実験設定用コマンドを実行することで、内容をリポジトリ³に登録する。

手順3. 実行プログラムや定数ファイル、ジョブを実行するシェルスクリプトを、それぞれコンパイル、コピーまたは作成するための設定用コマンドを実

行することで、実験をスーパーコンピュータ上にセットアップする。

手順4. 実験をジョブスケジューラROSEに登録するためのコマンドを実行する。

手順5. ROSEに登録した実験を開始するためのコマンドを実行する。

指定した実験番号や各種ディレクトリの設定等が適切であれば、実行結果は実行時間等のログを除いて元のコントロール実験と一致するはずである。

3.4.3 テスト実験の実行

初めてNAPEXに触れて、動作を試してみるような場合を除いて、前項の様にコントロール実験をそのまま実行するようなことはほとんどないだろう。実際に開発者が変更を加えたテスト実験を行うためには、基本的に前項の手順2.で、変更についての記述を設定ファイルに追加すればよい。

以下では、実際に変更を加える方法について概説する。

(1) 実行プログラムや定数ファイルの差し替え

入出力ファイルの名称が変わらないような実行プログラムの変更、あるいは定数ファイルの差し替えの場合には、元のファイルの配置のパスと、PBF⁴や定数ファイルの格納ディレクトリをそれぞれ設定ファイルで指定すればよい。

手順2.の実験設定用コマンドを実行すると、実験の枝番の数字が1つ増える。また、登録した実行プログラムのソースコードとPBF、定数ファイルがリポジトリに登録される。手順3.で配置されるファイルはリポジトリに登録されたものから取得されるため、登録時に格納していたファイルが後に削除されても問題なく設定を再現することができる。

変更を繰り返す場合には、再度手順2.を実行する。これにより、実験の枝番の数字がさらに1つ増える。手順3.から手順5.までは、いずれも適切な実験番号と枝番を指定する必要がある。

(2) 実行プログラムや定数ファイルの追加など

入出力ファイルの変更を伴うような実行プログラムの修正を行う場合には、バッチ処理を記述するJCL⁴ファイルの変更を行う必要がある。JCLファイルで、変更した実行プログラム名や入出力ファイルの記述を修正する。手順2.の設定ファイルには追加するPBFと定数ファイルの記述を加え、さらに修正したJCLファイルの格納ディレクトリのほか、実行されるジョブの種類や時刻について指定する。さらに保存データの追加を行う場合には、データのパスのほか、長期保存するか（データバンクストレージに圧縮して保存する）、短期保存するか（NFS (Network File System) に基本的に非圧縮で保存する）の設定などをそれぞれ記述する。

⁴ 詳細は第3.2.3項参照。

¹ 江河 拓夢

² このほか、NAPEXを初めて実行する場合のみ、リモートサーバへのアクセス設定や、スケジューラの環境設定が必要となる。

³ Subversionが用いられているが、ユーザはSubversionのコマンドを直接実行することはない。

(3) ジョブの追加など

ジョブの追加のほか、スーパーコンピュータの使用ノード数の変更、実行時のジョブフローの依存関係の変更などを行う場合には、それらを記述するための JDF⁴ ファイルの変更を行う。手順 2. の設定ファイルには、変更した JDF ファイルの格納ディレクトリのほか、ジョブグループ⁵ の実行間隔などを設定する。

(4) 新規の引継ぎデータの登録

新規の観測データなど、実験で使用する外部データ（引継ぎデータ）を変更する場合には、手順 2. に先駆けて、データの格納ディレクトリなどをデータベースに登録する必要がある。登録用の設定ファイルには、データを格納しているディレクトリのほか、ファイル名、データの保存期間、圧縮の有無などを記述する。引継ぎデータ登録用のコマンドを実行すると、ここで登録したファイルに一意的な番号が割り当てられる。この番号を、手順 2. のリポジトリ登録時に使用する。

(5) 任意の実験の引用

ここまでコントロール実験から変更を行うための方法について述べてきたが、あるテスト実験から変更を行うことも同様に可能である。設定された各種のファイルは全てリポジトリに登録されているので、テスト実験のために変更・追加したファイルを全て取得することが可能となっている。過去に実行された実験を再現することが容易であり、過去の NAPEX と比べて開発者の利便性が格段に向上している（第 3.2.3 項参照）。

3.4.4 ROSE の利用について

実験が開始された後は、ROSE によりジョブの投入、制御が行われる。ROSE の GUI (graphical user interface) には当初は Adobe AIR が用いられていた (平原 2012) が、保守性の観点から現在はウェブベースに移行している。モデル開発者は、GUI だけで様々な操作を行うことができる。

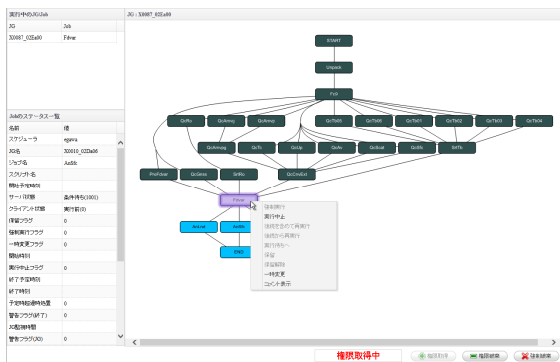


図 3.4.1 ROSE のウェブベースでの GUI の表示例（枠内）。枠内左側にジョブのステータスの一覧表、右側にジョブグループの進行状況のフローチャートなどが表示されている。枠外右側に進行状況の凡例を示す。

⁵ 特定の時刻に行われるジョブの集合。

図 3.4.1 に GUI でのジョブグループの表示例を示す。右側のフローチャートの部品はジョブとその依存関係を表しており、ジョブの状態が色別で表示される。ユーザは視覚的にジョブの進行状況を確認することができる。図の中央付近に表示されているポップアップは、ジョブをクリックすることで表示される。ジョブの実行状況によっても異なるが、ここでジョブの実行の中止、再実行、保留などの制御を行ったり、実験設定の一時変更を行うことができる。図の左側には、ジョブのステータス一覧が表示されている。これにより、ジョブ投入時の環境変数などの設定や、ジョブの終了ステータスを確認することができる。図の下部には、赤い文字で権限取得中と表示されているが、この権限はスケジューラに対して排他処理を行うために設定されている。様々なジョブに対する制御や設定の変更を行う場合には権限を取得してから作業を開始する。

なお、GUI と同様な操作を CUI (character user interface) で行うことも可能である。通常は GUI ベースで実行できるので、使用するかは任意となっている。例えば第 3.4.2 項の手順 5. は GUI と CUI のどちらでも行うことができるが、このケースでは CUI で実行する方が容易である。

3.4.5 まとめ

本節では、まず実験をそのまま再現するための手順について述べ、さらにそこから変更を加えた実験を設定する方法について示した。これらの大まかな流れは過去の NAPEX での設定と大きく変わらないが、使用するファイルをリポジトリに登録するようになったのは、実験の再現性の確保の点から大きく進歩したといえる。過去の NAPEX では、実行プログラムをただ特定の場所に配置するだけであったため、ソースコードが不明であったり、そもそも実行プログラムが削除される場合もあった。現 NAPEX では他の開発者が実験で使用した実行プログラムのソースコードをリポジトリから取得し、さらに変更を加えることも容易である。

ジョブスケジューラ ROSE の利用についてはユーザの視点から説明した。自主開発の ROSE を用いることで、数値予報ルーチンと実験の親和性が上がっているほか、スケジューラの機能に関する要望も挙げやすくなり、開発者の利便性を向上させやすくなったといえる。

NAPEX は数値予報モデルの開発において必要不可欠なツールになっている。今後もより良いものにするために改良を続けていくことが必要である。

参考文献

平原洋一, 2012: 気象庁における数値解析予報実験システムとバージョン管理. 地球流体データ解析・数値計算ワークショップ, https://www.gfd-dennou.org/library/davis-workshop/2012-12-12/hirahara_20121212.pdf.

3.5 NAPEX の利用の広がり¹

3.5.1 はじめに

NAPEX は数値予報課が主体となって開発したが、近年では数値予報課の枠を超えて他課室でも利用されている。本節では数値予報課以外での利用実績について簡単に紹介するとともに、第 10 世代スーパーコンピュータシステムに向けた取組についても概説する。

3.5.2 気候情報課での利用

全球数値予報システム及び全球アンサンブル予報システム（全球 EPS）の開発は、数値予報課と気候情報課が開発項目を分担・連携する体制で行われている（経田 2016）。このような背景から、数値予報課だけでなく気候情報課においても実験システムとして NAPEX が採用されている。両者が共通の開発基盤を利用することで、数値予報課と気候情報課の間で開発成果を速やかに共有できるようになっている。気候情報課での NAPEX の利用は数値予報課で実施しているような全球 EPS のコントロールメンバー（摂動なしメンバー）予報実験やアンサンブル予報実験の実施のみならず、再予報、1 年積分共通評価ツール、さらには長期再解析の実験システムとしても活用されている。

再予報とは、過去の多数の事例について調査対象となる同一のモデルで予測計算を実行することである。一般にモデルによる予測では、モデルの不完全性などにより積分が進むとともに誤差が時間発展し、現実的な場から離れていく。このため、予測期間が長い場合には、予測精度を事前に把握するとともにモデルの誤差を補正して予測結果を解釈する必要が生じる。再予報により過去の多数の事例の予測計算を行い、検証を通じて予測誤差特性に関する資料を得る（高谷 2012）。気候情報課では 1 か月及び季節アンサンブル予報システムの更新の際、それぞれ必要な再予報を実施し、予測値の特性の把握やバイアス補正、ガイダンスの作成などに役立っている（たとえば金浜ほか 2017; 高谷・石川 2015）。実行にあたっては、特定の期間の特定の計算開始対象日（たとえば金浜ほか 2017 では、期間として 1981~2012 年の 32 年間、対象日として毎月 10 日、20 日、最終日の 3 日×12 か月=年間 36 日）を初期値とした予報を実行する必要がある。全球 EPS では、一年のうちの特定の初期日（たとえば 1 月 10 日）の計算を 1 年刻みで全期間（32 年間）実行する設定を NAPEX で作成し、これを全ての初期日（1 月 10 日、1 月 20 日、...）で実行することにより実現している。NAPEX により、共通の実験設定を容易に複数の実験に対応させられることから、実験設定の労力軽減につながっている。

1 年積分共通評価ツールは、数値予報モデルの内容（力学過程や物理過程）を変更した際に影響を調査するためのツールで、特定期間で積分を実施し、予測特

性に問題がないかを調べるために用いられる。これまでは個々の開発者が自身で環境を構築する必要があったが、NAPEX で統一的に実施できるよう開発を行った。これにより、開発者が同様の実験を行う際に一から環境構築をする必要がなくなると共に、意図的に変更を加えた部分以外は共通の設定の実験を容易に行えるようになった。設定の違う部分が明確になることから、別の開発者の実験結果との比較も容易に行えるようになった。

長期再解析とは、数十年以上という長期のスパンで、最新のデータ同化システムと、当時の観測データを用いて均質な精度を持つ解析値を作成することである。現在準備が進められている次期長期再解析では実際に再解析プロダクトを作成する「本計算」を NAPEX を用いて実施することが予定されている。その際、利用可能な観測データ（特に衛星観測データ）は時代と共に変遷していく（たとえば古林ほか 2015）ことから、過去の期間の解析を行うためには、期間毎に異なる観測データを同一のデータ同化システムで利用していくことが要求される。NAPEX は本来府県天気予報・週間天気予報用途での開発を主目的に開発されてきたことがあり、このような複雑な状況での利用は想定されていなかったが、対応できるよう気候情報課で検討が進められているところである。

以上のように NAPEX は、府県天気予報・週間天気予報用途だけでなく、よりタイムスパンの長い数値予報システムの開発においても、活用が進められている。

3.5.3 気象研究所での利用

近年の数値予報システムの高度化により、その精度改善には最新の基礎研究の成果を速やかに現業システム設定での実験に適用できることが望ましく、気象研究所でも現業数値予報システムの改善・改良に資する研究開発が重要となっている。このような背景の下、重点研究課題「全球大気データ同化の高度化に関する研究（平成 23 年度～27 年度）」の開始を控えていた 2009 年度（平成 21 年度）後半から 2010 年度（平成 22 年度）にかけ、全球大気データ同化の研究を本格的に開始するためのシステムとして、第 8 世代スーパーコンピュータシステム向け NAPEX (NAPEX8) をベースに、第 6 世代気象研究所スーパーコンピュータシステム (MRI-SC6) 向けに NAPEX の移植作業が実施された² (MRI-NAPEX6)。

当時の NAPEX8 環境では既に JCL や PBF といった数値予報ルーチン管理のツールが利用されていたこともあり、これら関連ツールも同時に気象研究所へ移植が行われ、数値予報課と気象研究所との間で共通的な基盤で実験する環境が整備された（大野木ほか 2011）。そ

¹ 雁津 克彦、佐藤 芳昭（地球環境・海洋部 気候情報課）、石橋 俊之（気象研究所 台風研究部）

² 幸い MRI-SC6 と第 8 世代スーパーコンピュータシステムは同一系統の計算機であり、NAPEX8 と MRI-NAPEX6 は NAPEX モデルで同じ実行モジュールを利用することができた。ただし MRI-SC6 のスペックの問題で、現業水平解像度での実験ができず、解像度の変更が必要など運用に苦労を伴う点があった。

の後、気象庁本庁では2012年6月に第9世代スーパーコンピュータシステムの運用が開始され、第9世代スーパーコンピュータシステム向けNAPEX (NAPEX9) 環境ではROSEやデータベースの利用、引継ぎデータの自動解決やJDFによるノード数・ジョブ依存関係の管理等が実装された(第3.2節)。一方、気象研究所では2015年3月に第7世代気象研究所スーパーコンピュータシステム(MRI-SC7)の運用が開始された。気象研究所中期研究計画(平成26年度～平成30年度)では、重点研究(A3)「台風の進路予報・強度解析の精度向上に資する研究」において、引き続き全球数値予報システムを用いた研究開発が実施されており、気象研究所台風研究部の担当者によりMRI-NAPEX6の新システムへの移植・構築が行われた(MRI-NAPEX7)。

MRI-NAPEX7のうち、NAPEXモデルはNAPEX9で利用されているもの(第3.3.2項参照)を利用して、計算機環境依存への対応が行われた³。このためMRI-NAPEX7のNAPEXモデルはNAPEX9と同等である。一方、NAPEX環境はMRI-NAPEX6のNAPEX環境を新計算機に対応する形で利用し、NAPEX9環境で利用開始されたROSEやデータベースによる実験管理機能の移植は行わず、ジョブスケジューラはシェルスクリプトで構築された。これは、気象研究所には数値予報課プログラム班や基盤整備グループのような部署がなく、研究開発と並行してNAPEX環境の維持を行う必要があり、ROSEやデータベースを導入した場合、ROSEやデータベース自体の維持管理も必要となるため、少人数で利用するにあたって総合的にNAPEXのメンテナンスが容易となるようにこのような構成としている。

こうした構成は現行の気象研究所での研究目的を達成するためには十分な設計であり、不要な設定等を行う必要がないことから、最適化されたシステムとも言うことができる。試験的にはNAPEX9環境が気象研究所スーパーコンピュータシステム上で動作することは確認されており、移植自体は可能である。しかし、現行のMRI-NAPEX7がユーザニーズを十分満たしていることも注意する必要があり、不要な機能の移植はかえってユーザ利便性を損なうおそれもある。一方で、気象研究所で独自のNAPEX環境を運用するために、本庁のルーチン変更に追従するコストが高くなるといった弊害もある。他にも、第3.2節で紹介したNAPEX9環境で可能になった各種利点が反映されておらず、引継ぎデータの自動解決や、JDFによるノード数・ジョブ依存関係の管理といった、維持コストを大幅に軽減できるメリットも享受できない。このため、維持コストの低い利用形態を想定しつつ共通的なNAPEX環境の構築や維持管理体制を作っていくことが今後の課題と考えられる。

³ 現業水平解像度で実験ができないというMRI-SC6の問題は、MRI-SC7への移行でスペックが強化され解消したが、今度はMRI-SC7と第9世代スーパーコンピュータシステムが異なる系統の計算機となり、実行モジュール移植時に計算機に依存する箇所の修正が必要となっている。

なお、気象研究所では新たにNAPEXモデルとしてメソ数値予報システムの導入を検討しており、現在移植作業を進めているところである。

3.5.4 次期スーパーコンピュータシステムでの利用

気象庁では第10世代スーパーコンピュータシステムを2018年から運用する計画で、2017年1月現在各種移行作業に取りかかっている。NAPEXについても第10世代スーパーコンピュータシステム向けに仕様検討を進めているところである。次期NAPEXでも現行のNAPEXと同様に実験再利用性と数値予報ルーチンとの互換性を重視し、

- 全ての実験の設定をデータベースで管理する
- ジョブの管理や実行プログラムの管理に数値予報ルーチン管理の言語を用いる
- ファイル管理にバージョン管理システムを利用する
- 実験の実行に独自開発のジョブスケジューラ「ROSE」を用いる

といった現在の基本的な実装を踏襲する予定である。また、NAPEX9は利用開始から4年が経過し、いくつか課題も指摘されている。主な課題として

- 実験構築にかかる時間の短縮化⁴
- 実験環境構築時の設定ミス防止強化
- 引継ぎデータ利用の利便性強化

といったものが挙げられる。これらを踏まえ、2016年度中に数値予報課内のNAPEX管理グループ会合を通じ、次期NAPEXの仕様を決定する予定である。

参考文献

- 金浜貴史, 関口亮平, 足立恭将, 宮岡健吾, 久保勇太郎, 新保明彦, 西村明希生, 2017: 再予報と検証. 平成28年度季節予報研修テキスト, 気象庁地球環境・海洋部, 9-47.
- 古林慎哉, 太田行哉, 原田やよい, 海老田綾貴, 守谷昌己, 小野田浩克, 大野木和敏, 釜堀弘隆, 小林ちあき, 遠藤洋和, 宮岡健吾, 高橋清利, 2015: 気象庁55年長期再解析(JRA-55)の概要. 平成26年度季節予報研修テキスト, 気象庁地球環境・海洋部, 66-115.
- 経田正幸, 2016: 全球アンサンブル予報システムの開発. 数値予報課報告・別冊第62号, 気象庁予報部, 52-57.
- 大野木和敏, 江河拓夢, 小林ちあき, 石橋俊之, 田浦俊太郎, 石元裕史, 釜堀弘隆, 上清直隆, 2011: 全球大気データ同化実験システムNAPEXの気象研究所への移植～MRI-NAPEXの構築～. 気象庁測候時報, 78, 19-29.
- 高谷祐平, 2012: 新用語解説 再予報・ハインドキャスト. 天気, 59, 493-495.
- 高谷祐平, 石川一郎, 2015: 再予報による新システムの評価. 平成27年度季節予報研修テキスト, 気象庁地球環境・海洋部, 42-95.

⁴ ユーザの設定によっては、コンパイルに1時間近くかかる場合があり、NAPEX環境の改良によって冗長なコンパイルを省略することで時間の短縮が見込まれている。