

観測データ同化による長周期地震動リアルタイム予測へ向けた試み

中島 研吾^{†1} 古村 孝志^{†2} 鶴岡 弘^{†2} 松葉 浩也^{†1} 坂口 吉生^{†3} 住元 真司^{†3}
笠井 良浩^{†3} 池田 輝彦^{†3} 八代 尚^{†4} 荒川 隆^{†5} 埴 敏博^{†1}

大地震の際に高層ビル等と共振し、大きく長く揺すり被害をもたらす長周期地震動の予測と災害軽減に向けて、全国に展開された地震観測網の観測データと地震波動伝播のシミュレーションを融合した長周期地震動の予測手法が提案されており、古村らは三次元長周期強震動シミュレーションと観測データによる同化を実施するコード Seism3D/OpenSWPC-DAF (Data-Assimilation-Based Forecast) を開発している。本研究は、全国地震観測データ流通ネットワーク「JDXnet」による観測データを受信、処理し、Seism3D/OpenSWPC-DAF による「データ同化・シミュレーション」融合をリアルタイムに実施するフレームワークをスーパーコンピュータ上に構築し、長周期地震動に対する安全な防災行動、減災を実現することを最終的な目標とする。本発表では、Wisteria/BDEC-01 (東京大学情報基盤センター) 及び「計算・データ・学習」融合を実現する革新的ソフトウェア基盤「h3-Open-BDEC」を使用し、新潟県中越沖地震 (2007 年) について、「データ同化・シミュレーション」融合を実施した事例について、ソフトウェア、アプリケーションの概要、将来への展望も含めて紹介する。

1. 背景と本研究の目的

我が国は自然災害の多い国であるが、中でも地震による被害は甚大で、最近 20 数年の間にも、神戸、新潟、東北、熊本等の大地震が発生している。地震に関連したシミュレーションとしては、①断層周辺に蓄積した応力分布から地震発生を予測するもの、②地震発生時の断層の動的な破壊のシミュレーション、更に、③その後の地震波の伝播と増幅に関する強震動シミュレーション、という 3 つのフェーズに分かれている。いずれのフェーズにおいても、特に深部の地下構造がよくわかっていないため、災害軽減に資する正確な予測の実現には課題が多い。観測結果を取り入れることによってより精度の高いシミュレーションを実施することが期待されるが、地震シミュレーションの分野で、両者を融合した研究事例はほとんどない。古村らは、強震動シミュレーションにおいて、両者を融合する手法を提案し、三次元長周期強震動シミュレーションと観測データによる同化を実施するコード Seism3D/OpenSWPC-DAF (Data-Assimilation-Based Forecast) [1,2] を開発している。

著者等は、学際大規模情報基盤共同利用共同研究拠点 (JHPCN) [3] 2021 年度共同研究課題「三次元強震動シミュレーションとリアルタイムデータ同化の融合 (jh210022-MDH, 代表: 中島研吾)」において、全国地震観測データ流通ネットワーク「JDXnet (Japan Data eXchange network)」[4] によって得られる地震観測データを利用したリアルタイムデータ同化と Seism3D/OpenSWPC-DAF による高精度な三次元強震動シミュレーションの融合による計算手法、機械学習による三次元地下構造モデルの改良手法、及び関連するソフトウェア群の研究開発に取り組んでいる。東京大学情報基盤センター [5] の Oakforest-PACS (OFP) [6]、Oakbirdge-CX (OBCX) [7]、Wisteria/BDEC-01 (Wisteria)

[8,9,10] を使用した検証を実施する他、革新的ソフトウェア基盤「h3-Open-BDEC」[11,12] 等のソフトウェア群を「富岳」を含む HPCI 計算資源群等に展開し、「シミュレーション (Simulation)・データ (Data)・学習 (Learning) (S+D+L)」融合の促進に貢献することを目指している。本研究は、「データ同化・シミュレーション」融合をリアルタイムに実施するフレームワークをこれらスーパーコンピュータ上に構築し、長周期地震動の即時予測・警報の実現による安全な防災行動、減災を実現することを最終的な目標とする。

本稿では、Wisteria/BDEC-01 (東京大学情報基盤センター) 及び「計算・データ・学習」融合を実現する革新的ソフトウェア基盤「h3-Open-BDEC」[11, 12] を使用し、新潟県中越沖地震 (2007 年) について、「データ同化・シミュレーション」融合を実施した事例について、ソフトウェア、アプリケーションの概要、将来への展望も含めて紹介する。

2. Seism3D/OpenSWPC-DAF

大地震の際に、震源から数百 km 以上も伝わり平野で強く増幅される周期 3~10 秒程度の長周期地震動は、超高層ビルや大型石油タンク等と共振して大きく長く揺ることによって被害を起こす恐れがある。2003 年十勝沖地震 (M8.0) では、震源から 230 km 離れた苫小牧の石油タンクが長周期地震動に共振して破損、出火した。2011 年東北地方太平洋沖地震では、震源から 750 km 離れた大阪 (咲洲) の 55 階建て超高層ビルが長周期地震動により最大で 1.7 m 揺れ、施設に被害が発生した。将来発生が危惧される南海トラフ沿いの巨大地震に備え、長周期地震動の即時予測は災害軽減に向けた重要な課題である。

長周期地震動は、周期 3~10 秒程度の表面波により構成され、その伝播・増幅は三次元的に不均質な地下構造、特に表面波の波長に相当する地下数 km 程度の堆積層の幾何

^{†1} 東京大学情報基盤センター
Information Technology Center, The University of Tokyo
^{†2} 東京大学地震研究所
Earthquake Research Institute, The University of Tokyo
^{†3} 富士通株式会社
Fujitsu, Ltd.

^{†4} 国立環境研究所
National Institute for Environmental Studies
^{†5} 高度情報科学技術研究機構
Research Organization for Information Science and Technology (RIST)

形状と地質の影響を強く受ける。こうした特性は、緊急地震速報など一般的な強震動予測に用いられる地震規模、距離、及び地盤特性を入力パラメータとした地震動予測式 (GMPE, Ground Motion Prediction Equation) に基づく評価は難しい。さらに、長周期地震動は、大きな揺れ振幅だけでなく、特定の周期成分が卓越した揺れが数分以上続くことが建物被害の拡大要因となる。このため、現実的な被害予測には、揺れの強度 (震度階、長周期地震動震度階) だけでなく、卓越周期や継続時間を含めた、揺れの波形自体を予測する地震波伝播シミュレーションが必要になる。

古村らは、防災科学技術研究所が運用する強震観測網 (K-NET, KiK-net) [13] から得られる強震観測データを入力として、三次元差分法により長周期地震動の波動場データ同化と、各地点の波形の予測を行うフレームワークとして、Seism3D/OpenSWPC-DAF (Data-Assimilation-Based Forecast) [1,2] を開発した。本手法では、地震波伝播を空間4次・時間2次のスタガード格子差分法計算コード (Seism3D) を用いて陽的に計算する。本コードは、大規模並列計算に向け MPI/Open-MP による並列化と性能チューニングが図られている (OpenSWPC)。データ同化手法としては、計算負荷が小さくリアルタイム処理に有利な最適内挿法 (Optimal Interpolation Technique) を用いている。最適内挿法は、粗い観測点で記録された地震波形を、観測誤差と計算誤差の大きさを考慮して適切に空間補完し、シミュレーション結果と同化させる技法である。これを、差分法計算のタイムステップ毎に行う。観測記録は地表に限られるが、長周期地震動を形成する表面波 (基本モード) のエネルギーは地表付近に集中するため、データ同化の効果は大きい。地表での同化結果は、表面波の伝播とともに地下深部にも反映される。

一般的な強震動 (長周期地震動) シミュレーション (図1の (A) Traditional Method : Pure Simulation) では、震源断層運動モデルを設定して、波動伝播の初期条件とするが、地震発生直後に震源モデルを精度良く推定することは難しい。これに対して、新手法 (図1の (B) New Method : Assimilation (同化) + Simulation) では、地震波動場の初期条件を観測データから直接的に設定するため、高速かつ高い精度でシミュレーションを開始することができる。

図2は、2007年新潟県中越沖地震 (Mw6.6) の強震観測データ同化と、未来の波動場の予測結果である。480 km × 480 km × 55 km の領域を一辺 0.24 km の立方体で分割し (合計約9億メッシュ)、長周期 (>2.7秒) 地震動の同化・予測シミュレーションを実施した。現在のところ、強震観測 (K-NET, KiK-net) データはリアルタイムに配信されていないが、本研究では、将来のリアルタイム強震データ流通を想定した数値実験を行った。シミュレーション結果は、計算領域内の全482の強震観測点の加速度3成分 (南北, 東西, 上下動) 波形と同化させた。そして、地震発生から

30, 50, 70, 及び90秒後の同化結果 (図2 (a)) に基づき、地震発生から160秒後の未来の波動場を予測した (図2 (b))。新宿地点での長周期地震動の予測波形と実際の観測波形、およびそれらの速度応答スペクトルを図2 (c,d) で比較する。データ同化が進むにつれ、予測と観測波形が揺れ振幅と継続時間を含めて良く一致する様子がわかる。100秒先の波動場の予測に要する時間は、Oakforest-PACS [6] の2,048ノードを使用して12.4秒であった [1,2]。これは、実際の長周期地震動の伝播より数倍高速であり、警報と防災行動のための十分な猶予時間が見込める。なお、予測精度と猶予時間にはトレードオフがあるため、データ同化の進展に合わせたシミュレーションの繰り返し実施による、予測の更新が必要である。

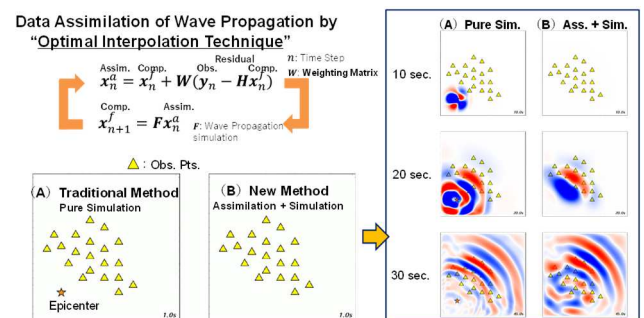


図1 データ同化・シミュレーション融合による長周期地震動予測の新手法 [1,2]

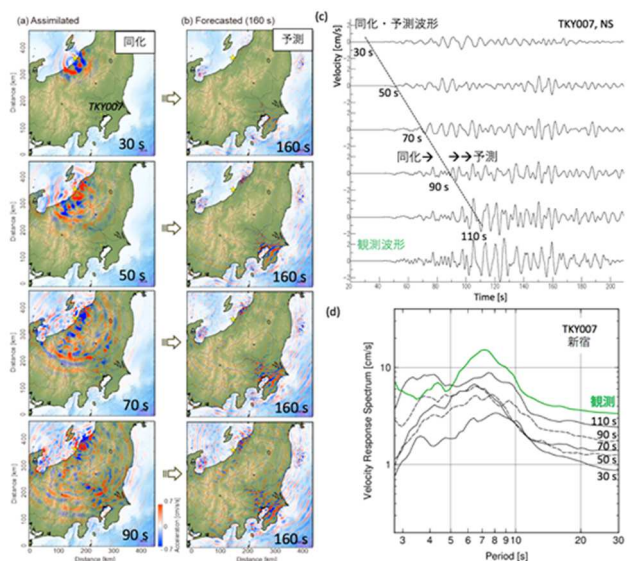


図2 Seism3D/OpenSWPC-DAFによる「データ同化+シミュレーション」融合事例, 2007年新潟県中越沖地震 [1,2]

3. JDXnet

地震観測網には大きく分けて、高感度地震観測網、広帯域地震観測網、強震観測網 (2. で使用した K-NET, KiK-net 等) がある。JDXnet (全国地震データ流通システム, Japan Data exchange network) [4] が対象とするのは、観測点からデータがリアルタイムで24時間365日常時送られてくる高感度地震観測網 (Hi-net (High Sensitivity

Seismograph Network Japan) 等) [14] と広帯域地震観測網 (F-net (Full Range Seismograph Network of Japan) 等) [15] である。日本列島には約 1,300 の高感度地震観測点 (Hi-net 等) と約 120 の広帯域地震観測点 (F-net 等) が設置され、日本列島周辺で発生した微小地震から巨大地震までの地震をいち早く検知して、その発生場所や地震の規模を求めるのに使用されている。気象庁が緊急地震速報を出すための観測点もこれらの観測点に含まれている。広帯域地震観測網は、日本列島周辺だけでなく世界の大地震の観測も可能で、周辺海域で発生する地震の津波警報の発表等にも利用されている。

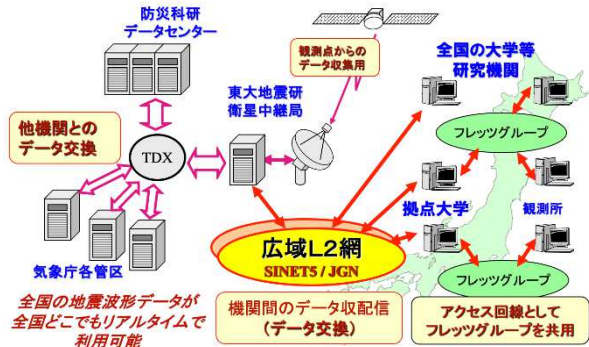


図3 全国地震データ流通基盤システム JDxnet の概要

これらの高感度地震観測網と広帯域地震観測網は、全国の9国立大学、気象庁、防災科学技術研究所、海洋研究開発機構、産業技術総合研究所などが観測点を設置して実施している。全国地震データ流通システム JDxnet [4] はこれらの各機関のすべての地震観測データを SINET/JGN 経由でリアルタイムにそれぞれの機関のみならず全国の大学等の研究機関に流通させている (図3)。

4. Wisteria/BDEC-01

東京大学情報基盤センターで2021年5月14日に運用を開始した「Wisteria/BDEC-01」[8,9,10] は、「シミュレーション (Simulation) ・データ (Data) ・学習 (Learning) (S+D+L)」融合を目指す BDEC システム構想 (Big Data & Extreme Computing) に基づくシステムの第1号機であり、シミュレーションノード群 (Odyssey (オデッセイ)) とデータ・学習ノード群 (Aquarius (アクエリアス)) の2つの計算ノード群を有したシステムである (図4)。総ピーク性能はそれぞれ 25.9 PFLOP (Odyssey), 7.2 PFLOPS (Aquarius), 合計 33.1 PFLOPS である。

シミュレーションノード群 (Odyssey) は「FUJITSU Supercomputer PRIMEHPC FX1000」20ラックから構成され、「A64FX」を7,680ノード (368,640コア) 搭載する。「A64FX」は、Armv8.2-A 命令セットアーキテクチャーをスーパーコンピュータ向けに拡張した「SVE (Scalable Vector Extension)」を、世界で初めて実装したプロセッサである。最先端の7nmプロセスで製造され、48個の演算

コアと2個または4個のアシスタントコアを有し、倍精度浮動小数点演算で3.3792 TFLOPSの理論ピーク性能を実現する。合計ピーク性能は25.9 PFLOPSであり、各ノードは32 GiBのHBM2メモリを搭載し、シミュレーションノード群 (Odyssey) の総メモリ容量は240 TiB、総メモリバンド幅は7.8 PB/秒である。各ノードはバイセクションバンド幅が13.0 TB/秒のノード間相互結合ネットワーク (Tofu インターコネク ト D) で結合されている。

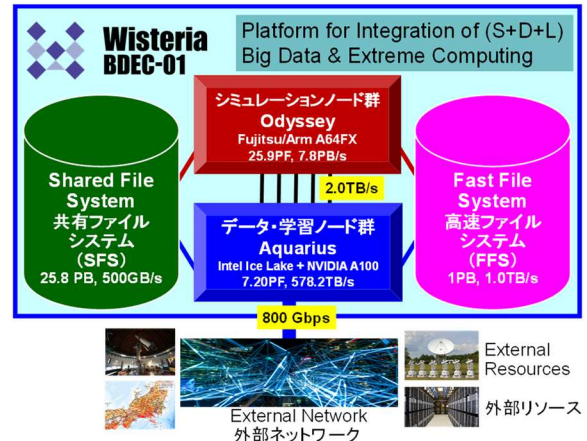


図4 Wisteria/BDEC-01 の概要 [8,9,10]

データ・学習ノード群 (Aquarius) 各ノードは汎用 CPU 2基 (Intel Xeon Platinum 8360Y (Ice Lake), 36core, 2.4GHz), 演算加速装置 (GPU) 8基 (NVIDIA A100 Tensor コア (SXM4, 40GB)) から構成されており、ノード間インターコネク トには NVIDIA Mellanox HDR InfiniBand ネットワークが採用されている。データ・学習ノード群 (Aquarius) の合計ピーク性能は7.2 PFLOPS、総メモリ容量は36.5 TiB、総メモリバンド幅は578.2 TB/秒である。各ノードは、データ転送速度が200 Gbpsの帯域を有する InfiniBand HDR を4リンク用いて、フルバイセクションバンド幅を持つノード間相互結合ネットワークで結合されている。さらに、外部接続のために25 Gbps Ethernet インタフェースも有している。

FEFS (Fujitsu Exabyte File System) による、共有ファイルシステム (容量: 25.8 PB, データ転送速度: 0.504 TB/秒) および SSD を搭載した高速ファイルシステム (容量: 1.0 PB, データ転送速度: 1.00 TB/秒) を有し、それぞれシミュレーションノード群 (Odyssey), データ・学習ノード群 (Aquarius) からアクセスし、大規模なデータを高速に処理することが可能である。

シミュレーションノード群 (Odyssey) とデータ・学習ノード群 (Aquarius) は、合計160本の InfiniBand EDR (100Gbps) を用いて2.0 TB/秒のネットワークバンド幅で結合されている。また、データ・学習ノード群 (Aquarius) は合計800 Gbpsのネットワーク転送速度で外部との通信が可能である。Aquariusの一部のノードは SINET 等の外部ネットワークを介して、サーバー、スト

レージ、センサーネットワークを含む様々な外部リソースに直接アクセス可能であり、観測データをリアルタイムに取り込んで解析、シミュレーションに利用することも可能である。現在は、Odyssey と Aquarius は別々に運用されている。

ソフトウェアとしては、Fortran, C/C++コンパイラ, Python インタープリタ, MPI 通信ライブラリ等を使用できる。計算科学, データ科学, 機械学習, 人工知能等幅広い分野のライブラリ, ツール, アプリケーションを提供する。オープンソースアプリケーションとしては, OpenFOAM (数値流体力学), MateriApps アプリケーション群 (物質科学), 東京大学生産技術研究所で開発された革新的シミュレーションソフトウェア群などを利用できる。

Wisteria/BDEC-01 は最先端の研究だけでなく, 計算科学・データ科学・機械学習や HPC 分野の人材育成にも共同利用されている。Wisteria/BDEC-01 の導入および運用により, 特に「計算・データ・学習」融合が推進され, サイバー空間 (仮想) とフィジカル空間 (現実) を高度に融合させた Society 5.0 の実現に大きく貢献することが期待される。

5. h3-Open-BDEC

5.1 「計算・データ・学習」融合実現

「計算・データ・学習 (S+D+L)」融合のためには, Wisteria/BDEC-01 のようなこれまでにない革新的なハードウェアが必要であるが, 様々なアプリケーション, ワークロードを Wisteria/BDEC-01 上で開発, 実行していくためのソフトウェア群も重要である。

当センターで開発した「ppOpen-HPC (自動チューニング機構を有するアプリケーション開発・実行環境)」

[16], 「h3-Open-BDEC (「計算・データ・学習」融合のための革新的ソフトウェア基盤)」[11,12] を利用し, 高性能なアプリケーションを容易に開発することが可能である。

5.2 h3-Open-BDEC 概要

当センターでは, センター内外の計算科学, 計算機科学, 数値アルゴリズム, データ科学, 機械学習の専門家と協力して, エクサスケール時代のスパコンの能力を最大限活用し, 科学的発見を持続的に促進するために, (計算・データ・学習) 融合による革新的シミュレーション手法を提案し, 最小限の計算量・消費電力で融合シミュレーションを実現する研究開発, ソフトウェア基盤実装を実施している [11,12]。本研究では, Wisteria/BDEC-01 を (計算・データ・学習) 融合のためのプラットフォームと位置付け, ①変動精度演算・精度保証・自動チューニング (Automatic Tuning, AT) による新計算原理に基づく革新的高性能・高信頼性・省電力数値解法, ②機械学習に基づく革新的手法である階

層型データ駆動アプローチ (hDDA) の 2 項目を中心に研究開発を実施し, 革新的ソフトウェア基盤「h3-Open-BDEC」を開発する (図 5) [11,12]。

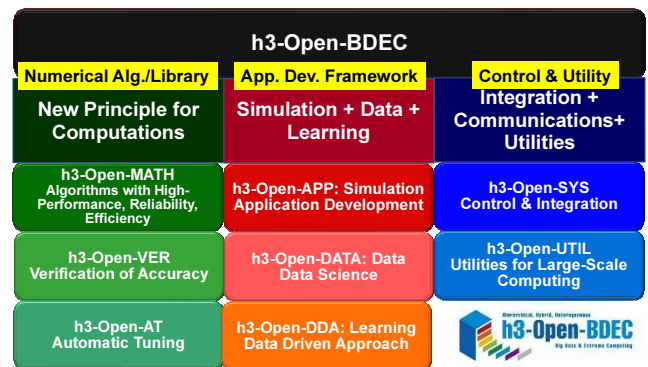


図 5 h3-Open-BDEC の概要 [11,12]

革新的ソフトウェア基盤「h3-Open-BDEC」(図 5) を構成する h3-Open-MATH (数値アルゴリズム), h3-Open-VER (精度保証), h3-Open-AT (自動チューニング), h3-Open-APP (アプリ開発), h3-Open-DATA (データ科学), h3-Open-DDA (データ駆動アプローチ), h3-Open-UTIL (並列ユーティリティ), h3-Open-SYS (統合・制御) は複数の構成要素を含み, 緊密に関連し, 以下の 3 層を構成している:

- ① 「変動精度演算に基づく新計算原理」層 (h3-Open-MATH, h3-Open-VER, h3-Open-AT)
- ② 「(計算・データ・学習) 融合」層 (h3-Open-APP, h3-Open-DATA, h3-Open-DDA)
- ③ 「統合・通信・ユーティリティ」層 (h3-Open-SYS, h3-Open-UTIL)

エクサスケールシステムにおける高性能数値アルゴリズム実現には, メモリ・ネットワークの階層の深化に対応した通信最適化 (Serial, Parallel), 省電力・省エネルギーに向けた検討が必要である。数値計算による近似解 (数値解) は様々な計算誤差を含み, 計算結果の信頼性の観点から, 数値解の正しさを数学的に保証する必要がある。低精度・混合/変動精度使用時, 悪条件問題には重要であるが, 実問題で現れる大規模疎行列・H 行列への応用例はほとんどない。本研究では, 「変動精度演算に基づく新計算原理」確立を目指して, 高性能アルゴリズム, 精度保証, 最適精度選択のための自動チューニング手法の研究開発を実施する。

シミュレーションに機械学習を適用して異なるパラメータでの解を予測するデータ駆動アプローチ (Data Driven Approach, DDA) では, 計算を繰り返して教師データを生成する必要がある。本研究で提案する階層型 DDA (hDDA) は, 特徴検知, MOR (Model Order Reduction), UQ (Uncertainty Quantification), スパースモデリング, 適応格子等の諸機能を駆使して, 計算量 (メッシュ数, 粒子数) を削減した簡

易モデルを、機械学習により自動生成し、教師データ生成用モデルとして利用する。「統合・通信・ユーティリティ」層は Wisteria/BDEC-01 のようなヘテロジニアスなシステム上で、「計算・データ・学習」融合を容易に実現するための環境を提供する。

h3-Open-BDEC はエクサスケール時代のスパコンで（計算・データ・学習）融合を実現する世界初の革新的ソフトウェア基盤であり、計算科学の専門家のみで（計算・データ・学習）融合を容易に実現できる。ソースコード、マニュアル類も含めて一般に公開し、様々な環境で利用できるよう、普及に努める。h3-Open-BDEC 利用による（計算・データ・学習）融合シミュレーションにより、従来手法と同等の正確さを保ちつつ、大幅な計算量・消費電力削減を目指す。

5.3 Wisteria/BDEC-01 と h3-Open-BDEC による「計算・データ・学習 (S+D+L)」融合

計算科学シミュレーションは多くの場合、非線形な問題を扱うため、多数のパラメータスタディが必要である。Wisteria/BDEC-01 では、機械学習による最適パラメータ推定を、外部から取り込んだ実験・観測データによる同化と組み合わせ、正確な解をより短時間で求めることを目指している。



図 6 Wisteria/BDEC-01 利用による「計算・データ・学習」融合のイメージ [11,12]

図 6 は、Wisteria/BDEC-01 における「計算・データ・学習 (S+D+L)」融合のイメージである。h3-Open-BDEC を使用することによって、シミュレーションノード群で計算科学シミュレーションコードを実行し、データ・学習ノード群では外部から取り込んだ観測データや、機械学習による推論等に基づきパラメータを最適化し、更に計算を実施するというサイクルを容易に実現することができ、またパラメータ最適化によって計算時間を全体として短縮できることが期待される。

h3-Open-SYS は Wisteria/BDEC-01 のようなヘテロジニアスなシステムにおいて、シミュレーションとデータ処理実行を統合するソフトウェア群である。h3-Open-SYS/WaitIO

[17] はその中核機能として、ファイルシステムを通じて複数の並列プログラムがデータの受け渡しを行うライブラリである。多くのスパコンで提供されている共有ファイルシステムをプログラム間のデータ連携手段として使用することで高い汎用性を確保し、ファイルを通信手段として用いる際に一般的に問題となる同期の問題を WaitIO ライブラリで解決する。WaitIO ライブラリは元々ファイルシステム経由で通信を前提としていたが、複数の並列プログラム間での通信を実施するために、Odyssey と Aquarius 間の通信ライブラリとしても開発が進められている [17]。h3-Open-SYS/WaitIO は、h3-Open-BDEC の多機能カプラー (Coupler) である h3-Open-UTIL/MP [18] と組み合わせることによって、Wisteria/BDEC-01 のようなヘテロジニアスなシステム上で「S+D+L」融合を実現できる。以下に h3-Open-SYS/WaitIO, h3-Open-UTIL/MP の概要を紹介する。Odyssey と Aquarius 間の通信は既に WaitIO ライブラリによって実現されており、h3-Open-UTIL/MP も Odyssey と Aquarius 両者を同時に利用可能となっている。詳細は著者等による先行研究 [17] を参照されたい。

5.4 h3-Open-SYS/WaitIO

h3-Open-SYS/WaitIO [17] は複数の並列プログラム間で通信を行うためのライブラリである。各並列プログラムは典型的には MPI プログラムであるが、MPI 以外の形の並列プログラムからも利用できるよう汎用的に設計されている。

h3-Open-SYS/WaitIO は多様な並列計算機環境を想定している。特に、通信を行う並列プログラムが同一の並列計算機で動作する場合はもちろん、図 4 に示す Odyssey と Aquarius のように各々の並列プログラムが異なる並列計算機で動作する場合にも対応する。そのために通信経路としてファイルと TCP/IP を想定する。

ファイル経由での通信を行う h3-Open-SYS/WaitIO を特に h3-Open-SYS/WaitIO -File (以下、WaitIO-File) と呼ぶ。WaitIO-File は同時実行される並列プログラム間の通信手段が共有ファイルシステムに限られる状況を想定し、共有ファイルを通信路として使用するものである。一般にスーパーコンピュータは、並列プロセスに対して割り当てたノード群から外への通信を許さない運用が多いため、共有ファイルを用いたデータの受け渡しは非効率であるものの現実的な手段である。WaitIO-File は通信を行うプロセスのペアそれぞれに対して通信方向別に 1 個ずつのファイルを準備する。送信側は送信データをファイル末尾に追記し続け、受信側はそのファイルを読み込む動作が基本であるが、特に受信側はファイルの末尾以降を読み込もうとした場合には新たなデータの追記を待つ点が通常のファイル I/O と異なる (WaitIO はこの特徴を表した名前である)。

TCP/IP を用いて通信を行う h3-Open-SYS/WaitIO を特に h3-Open-SYS/WaitIO-Socket (以下、WaitIO-Socket) と呼ぶ。WaitIO-Socket は並列プログラムを構成する各プロセスが、

別の並列プログラムの各プロセスに直接通信できる環境を想定している。前述のように現在のスーパーコンピュータの通常の運用ではこのような通信は許されていないことが多いが、並列計算機を構成するインターコネクトはTCP/IPでの通信をサポートしていることが多いため、セキュリティ上の問題を考慮の上、運用方針を明確にすれば技術的には容易に実現可能な環境と言える。WaitIO-Socketの基本アイデアは、並列プログラムを構成する各プロセスが通信相手の並列プログラムの全プロセスに対してTCP/IPのコネクションを張り通信を行うものである。この基本アイデアはシンプルであるが、各並列プロセスが1万を超える数のプロセスから構成されるような大規模環境を想定する際に、通信ソケットだけでメモリ資源を使い尽くすようなことがないように、設計、実装上の工夫が必須である。

表1 WaitIO-MPI Conversion ライブラリ API [17]

WaitIO-MPI API	概要
<code>int waitio_mpi_isend (const void *buf, int count, WAITIO_MPI_Datatype datatype, int dest, int tag, WAITIO_MPI_Comm comm, WAITIO_MPI_Request *request);</code>	WaitIO実装版 MPI_Isend
<code>int waitio_mpi_irecv (void *buf, int count, WAITIO_MPI_Datatype datatype, int source, int tag, WAITIO_MPI_Comm comm, WAITIO_MPI_Request *request);</code>	WaitIO実装版 MPI_Irecv
<code>int waitio_mpi_reduce (const void *sendbuf, void *recvbuf, int count, WAITIO_MPI_Datatype datatype, WAITIO_MPI_Op op, int root, WAITIO_MPI_Comm comm);</code>	WaitIO実装版 MPI_Reduce
<code>int waitio_mpi_bcast (void *buffer, int count, WAITIO_MPI_Datatype datatype, int root, WAITIO_MPI_Comm comm);</code>	WaitIO実装版 MPI_Bcast
<code>int waitio_mpi_allreduce (const void *sendbuf, void *recvbuf, int count, WAITIO_MPI_Datatype datatype, WAITIO_MPI_Op op, WAITIO_MPI_Comm comm);</code>	WaitIO実装版 MPI_Allreduce
<code>int waitio_mpi_waitall (int count, WAITIO_MPI_Request *array_of_requests, int *array_of_statuses);</code>	WaitIO実装版 MPI_Waitall
<code>int waitio_create_universe (WAITIO_MPI_Comm *commp);</code>	Waitio 初期化 サポート関数

h3-Open-SYS/WaitIO-Socketの実装、インタフェースの詳細については先行研究[17]を参照されたい。WaitIO-Socketは、WaitIO-MPI Conversion ライブラリ[17]を使用して、C/C++, Fortran からMPIと同様のインタフェースで呼び出すことが可能である。現状、実装されている関数を表1に示す。非同期送受信関数(isend, irecv)の他、集団通

信関数(bcast, reduce, allreduce)であり、扱うDatatypeは数種の基本Datatypeに限定している。アプリケーションの必要に応じて拡張する予定である。現状、関数名とdatatype, opなどのMPIオブジェクト表記はWAITIO_のprefixを機械的に追加することでWaitIO向けに書き換えるようにしている。

5.5 h3-Open-UTIL/MP

h3-Open-SYS/WaitIOは並列プログラム間通信ライブラリである。このライブラリを基盤として複数の並列プログラムを連成するための上位ソフトウェア(カブラ)としてh3-Open-UTIL/MPが開発されている[18]。h3-Open-UTIL/MPは異なる格子系を持つ複数のモデルコンポーネントに対して設定された時間間隔でデータを交換し格子変換を行う。更に、これらの基本機能に加えて、結合されたモデル群を並列に実行し統計処理を行う結合アンサンブル機能やPythonアプリケーションを結合するためのPythonインタフェースを装備している[18]。モデルと結合されるPythonアプリケーションにはI/Oや作画ライブラリなどが想定されるが、有力なアプリケーションの一つが機械学習である。シミュレーションモデルを機械学習と結合することで、計算負荷の高いプロセスを機械学習に代替させる、パラメータを用いない高解像度計算の結果をパラメータ化が必要な低解像度計算に反映させるといった、多くの利点が得られる。このような背景に基づき、大気モデルNICAMと機械学習ライブラリPyTorchを結合するプロジェクトが進行中であり、既に予備的な結果が得られている。しかしながら、PyTorchによる学習は計算負荷が高く実行のボトルネックとなっている。この状況を改善するにはシミュレーションモデルをシミュレーションノード群(Odyssey)で、機械学習ライブラリをデータ・学習ノード群(Aquarius)で実行し結合する方法が考えられる。この機能を実現するためのプログラム構成を図7に示す。

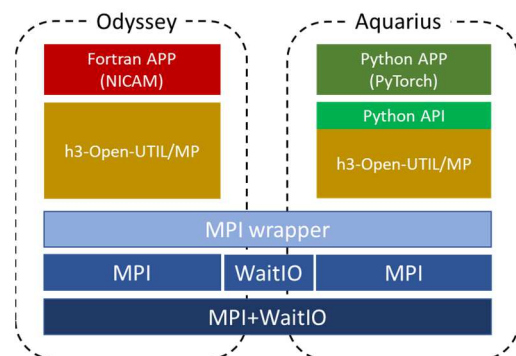


図7 WaitIO-SocketとMPIを併用した異機種間結合プログラムの構成[17]

最下層は機種をまたぐ大域通信を行うレイヤでMPIとWaitIO-Socketを併用し大域通信を実現する。第2層目はア

プリ内通信とアプリ間通信を行うレイヤで、アプリ内は従来通りの MPI を、アプリ間は WaitIO-Socket を用いて通信を行う。第3層目は MPI の wrapper で、これにより従来の MPI ルーチンコールとシームレスに異機種間通信ができるようにする。

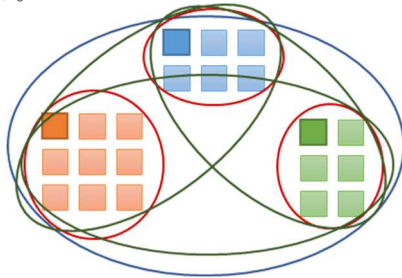


図 8 h3-Open-UTIL/MP が用いるコミュニケータ [17]

h3-Open-UTIL/MP が用いているコミュニケータの種類を図 8 に示す。図で色分けされた小さな四角の集合が一つのモデルコンポーネントを表す。用いるコミュニケータは全体通信に用いる Global, モデル間通信に用いる Model, 各モデルの 0 番プロセスの集合である Leader, モデル内通信に用いる Local の 4 種類である。これらのコミュニケータに対して使われる MPI ルーチンの種類を表 1, 表 2 に示す。表でハッチを施されたセルが WaitIO-Socket を用いる異機種間通信である。

表 2 各コミュニケータで用いられる局所通信ルーチン [17]

局所通信	Global	Model	Leader	Local
Isend		●	●	●
Irecv		●	●	●
Wait		●	●	●
WaitAll		●		

表 3 各コミュニケータで用いられる大域通信ルーチン [17]

大域通信	Global	Model	Leader	Local
Bcast	●		●	●
Gather			●	●
Scatter				●
GatherV				●
ScatterV				●
AllReduce	●			●
Reduce	●			●
Barrier			●	

表 2 の局所通信 (一対一通信) では MPI_Isend, MPI_Irecv, MPI_Wait, MPI_Waitall のみが用いられ、これらは WaitIO-Socket でも同様のルーチンが用意されていることからそのまま WaitIO-Socket に移行可能である。表 3 に示される大域通信 (多対多・一対多通信) で図 7 第 1 層にあたる

MPI+WaitIO-Socket の併用が必要なルーチンは特殊なルーチンである Barrier を除くと Bcast, Gather, AllReduce, Reduce の 4 種類のみであり、これらを適切に実装すれば h3-Open-UTIL/MP を異機種間結合に対応させることができる。

大域通信の実装に際しては WaitIO-Socket が担当するマシン間の通信は双方の 0 番プロセス(King)のみが行うものとしている。これは大域通信に関わる全プロセスが一斉に WaitIO-Socket による通信を行うと性能上の問題が生じる懸念があるためである。

MPI と WaitIO-Socket を併用した大域通信の例を図 9 に示す。図 9 は MPI_Bcast の事例で、黄色い四角が Bcast の起点となる root プロセスを表す。まず root から root が所属する King にデータを送受信する。この通信は MPI で行われる。次いで King 同士の通信を WaitIO-Socket で行い、最後に各マシンの King が MPI_Bcast でデータを分配する。

作成した通信ルーチン群を h3-Open-UTIL/MP に適用し NICAM と PyTorch を模したテストプログラムでの結合試験を実施した。想定される計算規模においても Odyssey-Aquarius 間のデータ交換は問題なく実行されており WaitIO-Socket の有用性が確認された。

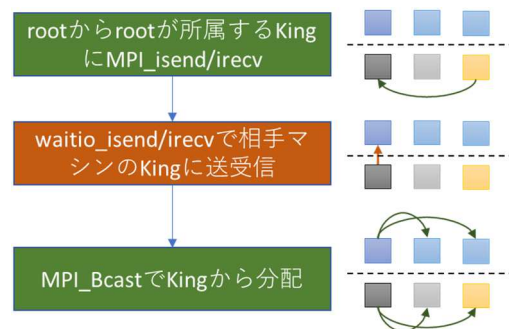


図 9 MPI と WaitIO-Socket を併用した大域分配の概要 [17]

6. 「データ同化・シミュレーション」融合

6.1 概要

本研究では、図 10 に示すように、Seism3D/OpenSWPC-DAF による「三次元強震動シミュレーション+データ同化」と JDXnet によるリアルタイムデータ同化を融合させた新しいシミュレーション手法の開発を、5. で述べた h3-Open-SYS/WaitIO, h3-Open-UTIL/MP を使用して実施している [19,20]。図 11 は、Oakbridge-CX (東京大学情報基盤センター, OBCX) [7] 上に構築した、融合システムのプロトタイプである。まず、JDXnet から観測データをリアルタイムで取得、処理した結果を、Seism3D/OpenSWPC-DAF で読み込んでいる。通常時は、Seism3D/OpenSWPC-DAF は観測データの内挿のみを実施するが、地震時には「データ同化+シミュレーション (Assimilation+Simulation, A+S)」, 次いで「シミュレーション (Pure Simulation, Pure S)」に移行する (図 12)。

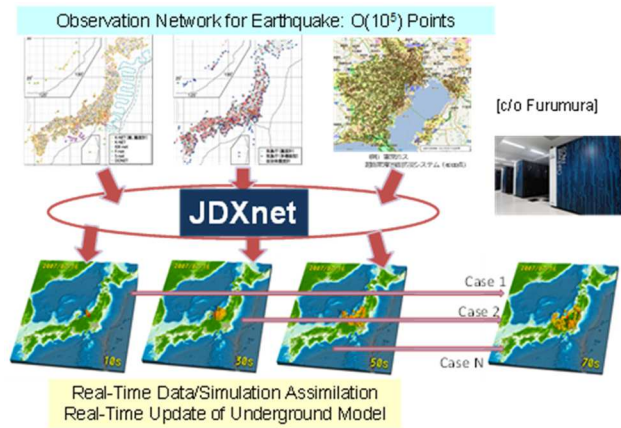


図 10 リアルタイムデータ同化と三次元強震動シミュレーションの融合 [19,20]

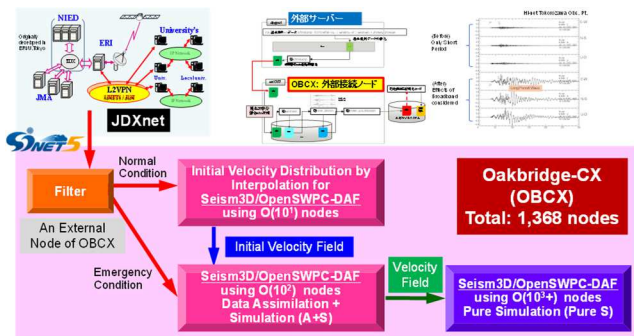


図 11 Oakbridge-CX (OBCX) 上に構築した「リアルタイム観測データ同化+強震動シミュレーション」融合システムのプロトタイプ

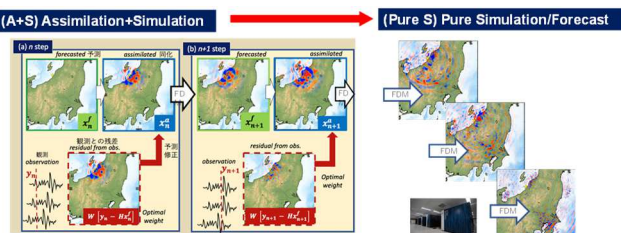


図 12 「リアルタイム観測データ同化+強震動シミュレーション」融合の手順：(A+S) ⇒ (Pure S)

6.2 観測データの処理

2.で紹介したように、[1, 2]では、防災科学技術研究所強震観測網 (K-NET, KiK-net) の観測結果が使用されているが、JDXnetによるリアルタイム配信はサポートされていない。従って、本システムでは、高感度地震観測網 (Hi-net) [14]、広帯域地震観測網 (F-net) [15] による観測データを使用している。これらのリアルタイム観測データを Seism3D/OpenSWPC-DAF に入力する前に、データパケットの時系列順の並べ替え、Hi-net 高感度地震計の広帯域化のための特性補正フィルタ処理、ファイルフォーマット変換等を実施する必要がある。現在は、Oakbridge-CX の「外部接続ノード」[7] 上で、JDXnet からの観測データ取得及び諸般の処理をリアルタイムで実施できる (図 11, 図 13)。処理結果は、WaitIO-Socket/File 経由で Seism3D/OpenSWPC-DAF に受け渡す。

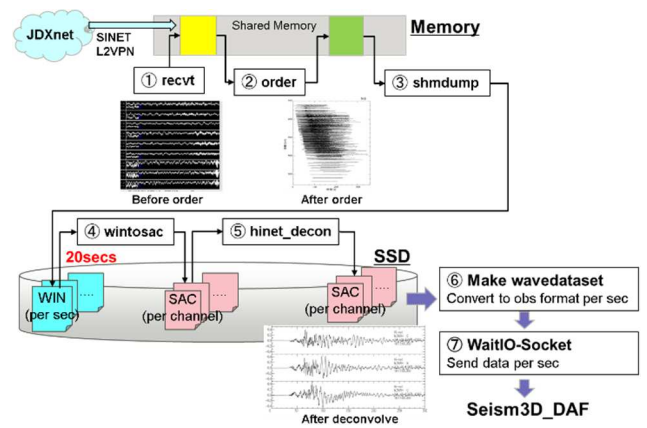


図 13 リアルタイムデータ処理

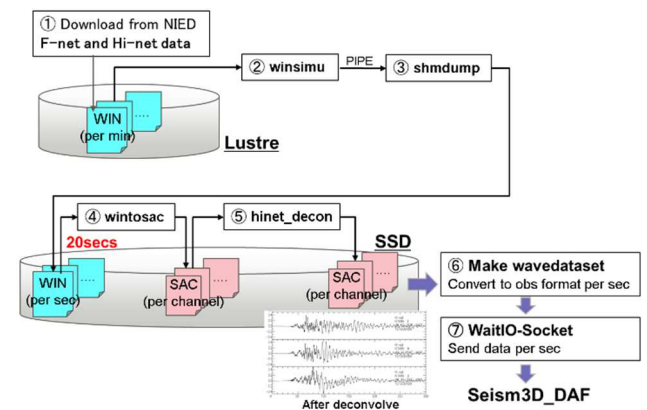


図 14 Playback 処理

図 13 に示すリアルタイムデータ処理機能の他、防災科学技術研究所 高感度地震観測網 (Hi-net) [14]、広帯域地震観測網 (F-net) [15] による過去の観測データをダウンロードして利用することも可能であり、これを Playback 処理と呼んでいる (図 14)。本稿では、過去の地震を対象としているため、Playback 処理を使用している。

6.3 Wisteria/BDEC-01 における実装

図 11 に示す一連の処理を Wisteria/BDEC-01 上で実装した。フィルタリング処理を Aquarius、データ同化・シミュレーションを Odyssey で実施する。

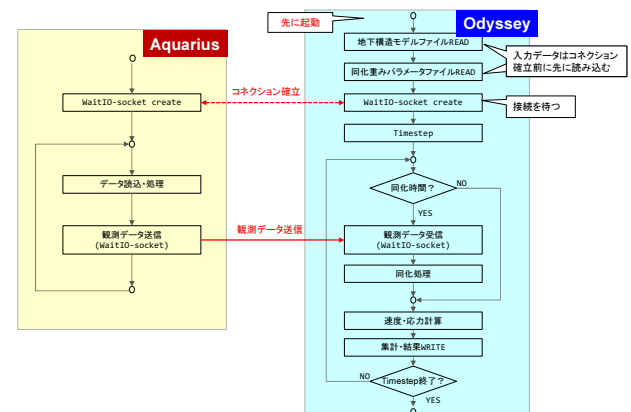


図 15 Aquarius・Odyssey におけるテストプログラム、WaitIO-Socket による通信


```

program dmy_filter
<変数: 変数宣言>
call mpi_init(ierr)
call mpi_comm_size(mpi_comm_world, nprocs, ierr)
call mpi_comm_rank(mpi_comm_world, myrank, ierr)
call waitio_create_universe(waitio_comm_universe, ierr)

if (myrank==0) then
open(100,file='./logfile_list.txt',form='formatted',status='old',iostat=ierr)
do 10,1000
<変数: obsデータ読み込み処理>
call waitio_mpi_isend(nmax1_o, 1, waitio_mpi_integer, 2,1, waitio_comm_universe,req(1,1), ierr)
call waitio_mpi_isend(dt_o, 1, waitio_mpi_float, 2,2, waitio_comm_universe,req(1,2), ierr)
call waitio_mpi_isend(nst_o, 1, waitio_mpi_integer, 2,3, waitio_comm_universe,req(1,3), ierr)
call waitio_mpi_isend(at_o, 1, waitio_mpi_float, 2,4, waitio_comm_universe,req(1,4), ierr)
call waitio_mpi_isend(tr_o, 1, waitio_mpi_float, 2,5, waitio_comm_universe,req(1,5), ierr)
call waitio_mpi_isend(iso_x_o, NSMAX, waitio_mpi_integer, 2,6, waitio_comm_universe,req(1,6), ierr)
call waitio_mpi_isend(iso_y_o, NSMAX, waitio_mpi_integer, 2,7, waitio_comm_universe,req(1,7), ierr)
call waitio_mpi_isend(iso_z_o, NSMAX, waitio_mpi_integer, 2,8, waitio_comm_universe,req(1,8), ierr)
call waitio_mpi_isend(isx_o, NST, waitio_mpi_integer, 2,9, waitio_comm_universe,req(1,9), ierr)
call waitio_mpi_isend(isy_o, NST, waitio_mpi_integer, 2,10, waitio_comm_universe,req(1,10), ierr)
call waitio_mpi_isend(isz_o, NST, waitio_mpi_integer, 2,11, waitio_comm_universe,req(1,11), ierr)
call waitio_mpi_isend(stc_o, 6*NST, waitio_mpi_char, 2,12, waitio_comm_universe,req(1,12), ierr)
call waitio_mpi_isend(vall_obs, NST*NBNS_LEN, waitio_mpi_float, 2,13, waitio_comm_universe,req(1,13), ierr)
call waitio_mpi_isend(vall_obs, NST*NBNS_LEN, waitio_mpi_float, 2,14, waitio_comm_universe,req(1,14), ierr)
call waitio_mpi_waitall(15,req, status, ierr)
call sleep(1)
enddo
close(100)
endif
call waitio_finalize(ierr)
call mpi_finalize(ierr)
end
    
```

図 16 WaitIO-Socket による送信 (Aquarius 側)

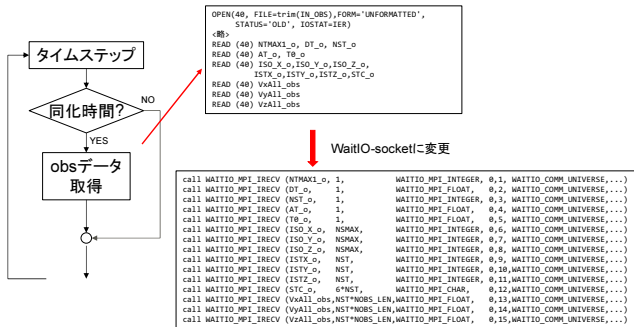


図 17 WaitIO-Socket による受信 (Odyssey 側)

現在は、WaitIO-Socket が正常に動作することを確認するために、フィルタリング済みデータを WaitIO-Socket より 1 秒ごとに Aquarius から Odyssey へ送り、Odyssey 側の Seism3D/OpenSWPC-DAF で受け取ってデータ同化・シミュレーションを実施するテストプログラムを作成した (図 15)。図 16、図 17 はそれぞれ、表 1 に示した WaitIO-MPI Conversion ライブラリによる Aquarius 側 (送信)、Odyssey 側 (受信) の記述例である。

Seism3D/OpenSWPC-DAF は Fortran で記述されているが、Fortran プログラムから MPI 関数を呼び出すのとはほぼ同じ方法で、WaitIO-Socket の機能を利用できることがわかる。

6.4 計算事例

6.3 で紹介したテストプログラムの動作検証、性能評価を Wisteria/BDEC-01 を使用して実施した。Wisteria/BDEC-01 では、図 15 に示す処理では、Aquarius 側では 1 ノード、Odyssey 側では 256~2,048 ノードを使用した。

2. で示したのと同じ、2007 年新潟県中越沖地震を対象として、動作確認を実施した。一辺 240m の立方体メッシュを使用し、メッシュ数は 1,920×1,920×240 (合計 8.85 億)、460.8 km×460.8 km×57.6 km の領域を扱っている、図 18 に示すように、対象領域内には、Hi-net : 349 点、F-net : 18 点の観測点があるが、本研究では、このうち 315 点の観測データを使用している。

図 19 は、図 2 と同様に、地震発生から 30、50、70、及び 90 秒後の同化結果に基づき、地震発生から 200 秒後ま

での未来の波動場を予測した結果と、東京都江東区における観測結果を比較したものである。図 2 の結果は強震観測網 (K-NET, KiK-net) における観測結果を使用しているのに対して、ここでは、Hi-net、F-net の観測結果を使用しているため、観測点も異なるが、両者の計算結果は良く一致しており、観測結果との一致も良好である。図 19 に示すように、(A+S) を 50 秒まで実施すれば、観測値に近い計算結果を得られることがわかる。

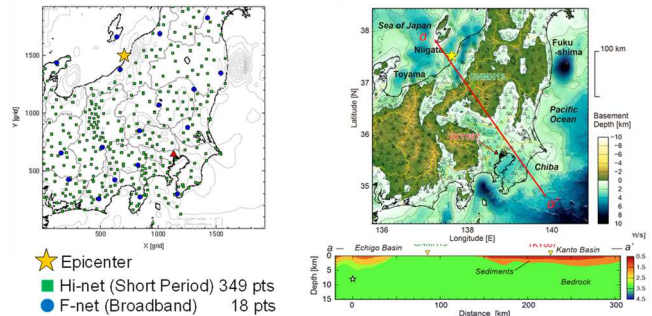


図 18 観測点分布・地下構造モデル [1]

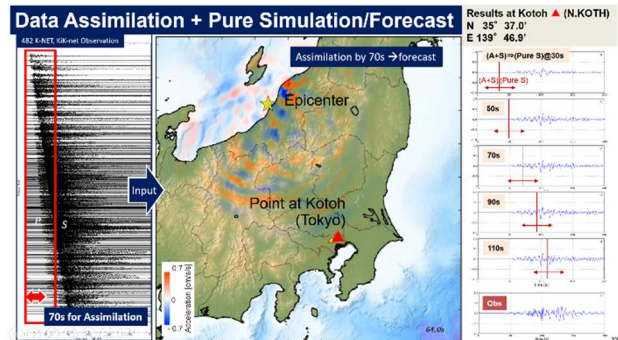


図 19 Seism3D/OpenSWPC-DAF による「データ同化+シミュレーション」融合事例、2007 年新潟県中越沖地震

表 4 Seism3D/OpenSWPC-DAF 計算時間

ノード数	(A+S) 60 秒分	(Pure S) 240 秒分
256	61.1	134.6
512	61.6	84.12
1,024	62.1	62.95
2,048	60.1	27.84

表 4 は、Seism3D/OpenSWPC-DAF の計算時間を比較したものである。60 秒分の「データ同化+シミュレーション (A+S)」, 続いて 240 秒分の「シミュレーション (Pure S)」を実施した場合について、可視化処理のための通信、I/O などを除いた時間を表示してある。(A+S) の部分は、フィルタリング済み結果を 1 秒毎に受信して、データ同化+シミュレーションを実施しているため、実時間 (60 秒) より短くなることはないが、(Pure S) の部分は、2,048 ノード利用時に実時間の約 10 分の 1 程度で計算が終了していることがわかる。2007 年新潟県中越沖地震の例では、図 19 に示すように、約 100 秒後に強い揺れが東京で発生している。地震発生 50 秒後に (A+S) から (Pure S) に切り替えた場

合、5秒程度で地震発生100秒後(切り替えから50秒後)の結果が予測可能である、ということになる。

7. まとめ

本稿では、著者等が学際大規模情報基盤共同利用共同研究拠点 2021 年度共同研究課題「三次元強震動シミュレーションとリアルタイムデータ同化の融合」において取り組んでいる、観測データ同化による長周期地震動リアルタイム予測へ向けた試みについて、シミュレーションコード、関連するソフトウェアの開発も含めて、現状を紹介した。Wisteria/BDEC-01 (Odyssey) 2,048 ノードによって、関東甲信越における約 8.85 億メッシュを対象とした Seism3D/OpenSWPC-DAF による三次元強震動シミュレーションを実時間の 10 分の 1 程度で実施可能であることが示された。

本稿では、h3-Open-SYS/WaitIO-Socket を利用することによって、Aquarius と Odyssey という異なるアーキテクチャによるノード群間の通信と連携を MPI とほぼ同じインタフェースで実現できることを示した。Wisteria/BDEC-01 と h3-Open-BDEC を使用した「計算・データ・学習」融合へ向けて有用な知見が得られた。現状では、フィルタリング済みの結果を Aquarius から Odyssey へ送っているが、今後は観測データ受信、フィルタリング処理も含めて実施するシステムを、本年度内を目標に構築する。5.5 で紹介した多機能カプラ h3-Open-UTIL/MP の活用も検討している。

現在、Odyssey 2,048 ノードを使用すると、約 8.85 億メッシュを対象とした Seism3D/OpenSWPC-DAF による三次元強震動シミュレーションを実時間の 10 分の 1 程度で実施できるが、これは Oakforest-PACS とほぼ同じ性能であり、A64FX の特性を活用することで、より高速化する必要がある。また、計算と通信のオーバーラップの導入により更なる最適化が可能である。

本研究の成果は、長周期地震動リアルタイム予測に適用可能であり、安全な防災行動、減災の実現に資するものと期待される。本研究によって、JDXnet の観測結果をリアルタイムにスパコンへ取り込んで「データ同化+シミュレーション」融合を実現することについては、ある程度見通しがついたが、計算結果のリアルタイム配信など、技術的には様々な課題が残されている。

地震波が伝播する媒体である地下地盤の構造は、波動伝播に影響を与えるが、広域にわたる地下構造の分析は十分に進んでおらず、より精密な三次元地下構造モデルの構築が強く望まれている。本研究では、通常の小規模な地震時に得られるデータを対象として、観測結果+シミュレーション結果に基づき、機械学習による地下構造法推定法の研究開発を併せて実施している。Seism3D/OpenSWPC-DAF における「データ同化・シミュレーション」融合とともに、より現実に近い三次元地下構造モデルを使用することによ

り、三次元強震動シミュレーションの更なる高精度化に貢献することが期待される。

謝辞

本研究の一部は、科学研究費補助金(19H05662, 20H02409)、学際大規模情報基盤共同利用共同研究拠点(JHPCN) 2021 年度共同研究課題(jh210022-MDH)、新エネルギー・産業技術開発機構(NEDO)「戦略的イノベーション創造プログラム(SIP) 第2期/ビッグデータ・AIを活用したサイバー空間基盤技術」、文部科学省「災害の軽減に貢献するための地震火山観測研究計画」の助成を受けたものである。

参考文献

- [1] Furumura, T., Maeda, T., Oba, A., Early Forecast of Long-Period Ground Motions via Data Assimilation of Observed Ground Motions and Wave Propagation Simulations, *Geophysical Research Letters*, <https://doi.org/10.1029/2018GL081163>, 2018
- [2] Oba, A., Furumura, T., Maeda, T., Data-Assimilation-Based Early Forecasting of Long-Period Ground Motions for Large Earthquakes along the Nankai Trough, *Journal of Geophysical Research*, <https://doi.org/10.1029/2019JB019047>, 2020
- [3] 学際大規模情報基盤共同利用共同研究拠点 (JHPCN) : <https://jhpcn-kyoten.ite.u-tokyo.ac.jp/ja/>
- [4] 卜部卓, 鷹野澄, 鶴岡弘, 中川 茂樹, 地震観測データ流通システム JDXnet の現状とクラウド化, *電子情報通信学会技術研究報告* 113 (256), 21-23, 2013
- [5] 東京大学情報基盤センター (スーパーコンピューティング研究部門・スーパーコンピューティングチーム) : <https://www.cc.u-tokyo.ac.jp/>
- [6] Oakforest-PACS (メニーコア型大規模スーパーコンピュータシステム) : <https://www.cc.utokyo.ac.jp/supercomputer/ofp>
- [7] Oakbridge-CX (大規模超並列スーパーコンピュータシステム) : <https://www.cc.utokyo.ac.jp/supercomputer/obcx>
- [8] Wisteria/BDEC-01 (「計算・データ・学習」融合スーパーコンピュータシステム) : <https://www.cc.utokyo.ac.jp/supercomputer/wisteria>
- [9] 中島研吾 他, 「計算・データ・学習」融合スーパーコンピュータシステム「Wisteria/BDEC-01」の概要, *情報処理学会研究報告* (2020-HPC-179-01), 2021
- [10] 塙敏博, 中島研吾 他, 「計算・データ・学習」融合スーパーコンピュータシステム「Wisteria/BDEC-01」の性能評価, *情報処理学会研究報告* (2020-HPC-180-22), 2021
- [11] h3-Open-BDEC : <https://h3-open-bdec.cc.u-tokyo.ac.jp/>
- [12] Iwashita, T., Nakajima, K., Shimokawabe, T., Nagao, H., Ogita, T., Katagiri, T., Yashiro, H., Matsuba, H., h3-Open-BDEC: Innovative Software Platform for Scientific Computing in the Exascale Era by Integrations of (Simulation + Data + Learning), *Project Poster, ISC-HPC 2020*
- [13] 防災科学技術研究所 強震観測網 (K-NET, KiK-net) : <https://www.kyoshin.bosai.go.jp/kyoshin/>
- [14] 防災科学技術研究所 Hi-net 高感度地震観測網 : <https://www.hinet.bosai.go.jp/>
- [15] 防災科学技術研究所 F-net 広帯域地震観測網 : <https://www.fnet.bosai.go.jp/>
- [16] ppOpen-HPC : <https://github.com/Post-Peta-Crest/ppOpenHPC>
- [17] 住元真司, 荒川隆, 坂口吉生, 松葉浩也, 八代尚, 塙敏博, 中島研吾, WaitIO-Socket : 異種システム上の複数 MPI プログラムを結合する通信ライブラリの試作, *情報処理学会研究報告*

告 (2020-HPC-181-07), 2021

- [18] Arakawa, T., Yashiro, H., Nakajima, K., Development of a coupler h3-Open-UTIL/MP, ACM Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region (HPC Asia 2022), 2022 (in press)
- [19] Nakajima, K., Matsuba, K., Hanawa, T., Furumura, T., Tsuruoka, H., Nagao, H., Integration of 3D Earthquake Simulation & Real-Time Data Assimilation on h3-Open-BDEC, MS290: Progress & Challenges in Extreme Scale Computing & Data SIAM Conference on Computational Science & Engineering (CSE21) (Online, March 4, 2021)
- [20] SIAM News (March 10, 2021), Supercomputer Simulations of Earthquakes in Real Time (by Jillian Kunze), <https://sinews.siam.org/Details-Page/supercomputer-simulations-of-earthquakes-in-real-time>
- [21] Stream Benchmark: <https://www.cs.virginia.edu/stream/>