

に対応して、ターボ機械システム全体を俯瞰的に理解することに意義を見出し、詳細構造解析の適用を検討することになった。その結果、具体的なシミュレーションの利用を模索するため協力し、具体的に解析を進めることになった。荏原製作所殿は、現在、社会インフラで活躍する一般プラントに不可欠なターボ機械の改善に着手しており、その検討過程において丸ごとシミュレーション等の適用計画を立案中である。本課題では、次世代耐震シミュレーション技術を設計問題の解決や効率化に資する。

平成 24 から 26 年度までの期間で、立軸ポンプの 3 次元幾何モデルと、その有限要素分割モデルを作成し、立軸ポンプの部品解析を実施しながら、入力データの精緻化と詳細化を実施した。立軸ポンプの 3 次元幾何モデルは、Fig.5-13(a)に示すように約 9 万点の解析部品からなるものとして完成させた。立軸ポンプは、大きくは Fig.5-13(b)と(c)のケーシング部と内部部品に二つからなる。立軸ポンプの幾何モデルの作成時間は、約 490 時間、解析形状モデル(サブストラクチャモデル)の作成時間は、約 2,550 時間、有限要素解析モデルの作成時間は、約 700 時間、有限要素解析条件設定の作業時間は、約 80 時間であった。解析のため集合させた解析部品数は、92,075 である。これらの定量的なデータは、純粋に CAD/CAE システム等の実操作時間のみを抽出したものであるため、実作業時間として加えるべきデータのレビュー時間や手戻り等に含まれる修正検討時間等は、加えていない。また、データ完成後、データの改修は必要に応じて行っているため、作業時間および部品数は、初期モデルの作成段階のものである。

平成 25 から 27 年度の期間で、部品ごとの解析や立軸ポンプ全系の解析を実施した。Fig.5-14(a)、(b)と(c)にケーシング部、内部部品、立軸ポンプ全体のそれぞれの解析結果例の一部を示す。平成 26 年度ではケーシング部のみを取り上げ、本課題で開発したコードの妥当性を立証するために、従来解析手段との比較検証を実施した。その結果を Fig.5-15(a) と(b)に示すように、全体的な挙動はよく一致することを確認するとともに、詳細部における解析結果が得られることを確認した。これにより、詳細かつ一体的な耐震シミュレーション技術(丸ごとシミュレーション技術)の特長を確認した。また、ケーシングを題材とした部品接続部における解析を行い、そのデータ収集に努め、シミュレーションコードの改良やモデルリング技術について考察を継続した。ケーシングの据付時や定常運転時における振動シミュレーションに資するため、異なる地震波における時刻歴応答解析を実施し、回転機として機能維持を確認すべき部位における物理量の取得や構造の健全性等を調査した。これらの解析結果例として、部品接続部における解析結果例を Fig.5-16 に示す。

平成 27 年度は、本取り組みの集大成として、立軸ポンプ全体の解析を中心に実施した。Fig.5-17 (a)は、エル・セントロ波を入力波とする解析結果である。エル・セントロ波とは、1940 年 5 月 18 日に米国カリフォルニア州最南部(インペリアル郡)発生した M7.1 の地震の波形であり、その地域の名称であるエル・セントロに由来する。この波形の特長は、(1)南北方向の加速度が最大 342 ガルと大きい、(2)ランダム性(卓越周期の散乱)がある、(3)直下型地震における震源近傍の地盤の固いところでの地震波、である。Fig.5-17(b)は、タフト波を入力波とする解析結果である。タフト波とは、1952 年 7 月 21 日に米国カリフォルニア州南部(カーン群)発生した M7.3 のカーン・カントリー地震の波形であり、エル・セントロ波同様に、その地域の名称であるタフトに由来する。この波形の特長は、地盤の固いところでの地震波である。Fig.5-17(c)は、八戸波を入力波とする解析結果例である。八戸波は正しくは、1968 年十勝沖地震の八戸港湾での強震記録というべきであろうが、本稿では八戸波と呼ぶ。八戸波は 1968 年発生した十勝沖地震を八戸港湾で観測したもので、過去の強震記録に比べて長い周期の成分が卓越することに特長がある。長周期成分等を含む地震波形の代表例の一つであり、エル・セントロ波とタフト波と並ぶ動的解析用地震動の標準 3 波である。これらの解析結果の分析には、情報可視化技術等を用いて詳細な顕現化を図り、その結果の討論を実施した。特定箇所の変位グラフや、空間的なデータの分布等を可視化した結果の一例を、Fig.5-18 と

Fig.5-19 に示す。

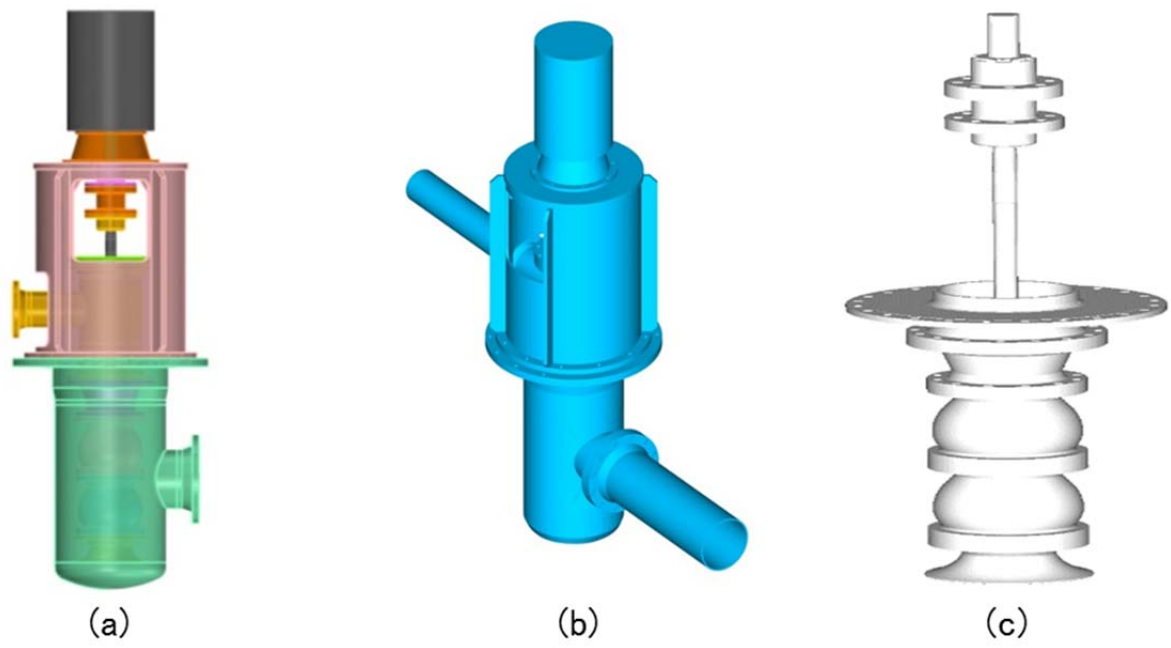


Fig.5-13 Pump geometric modeling

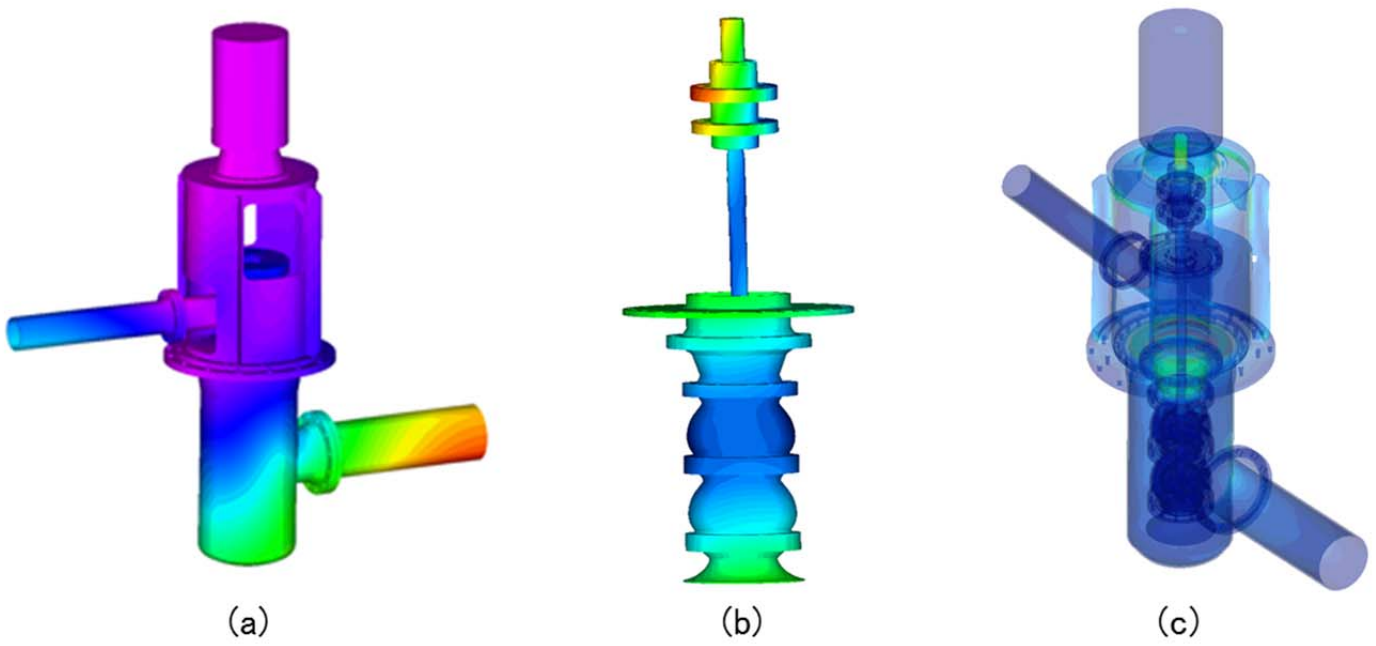


Fig.5-14 Pump Analyses

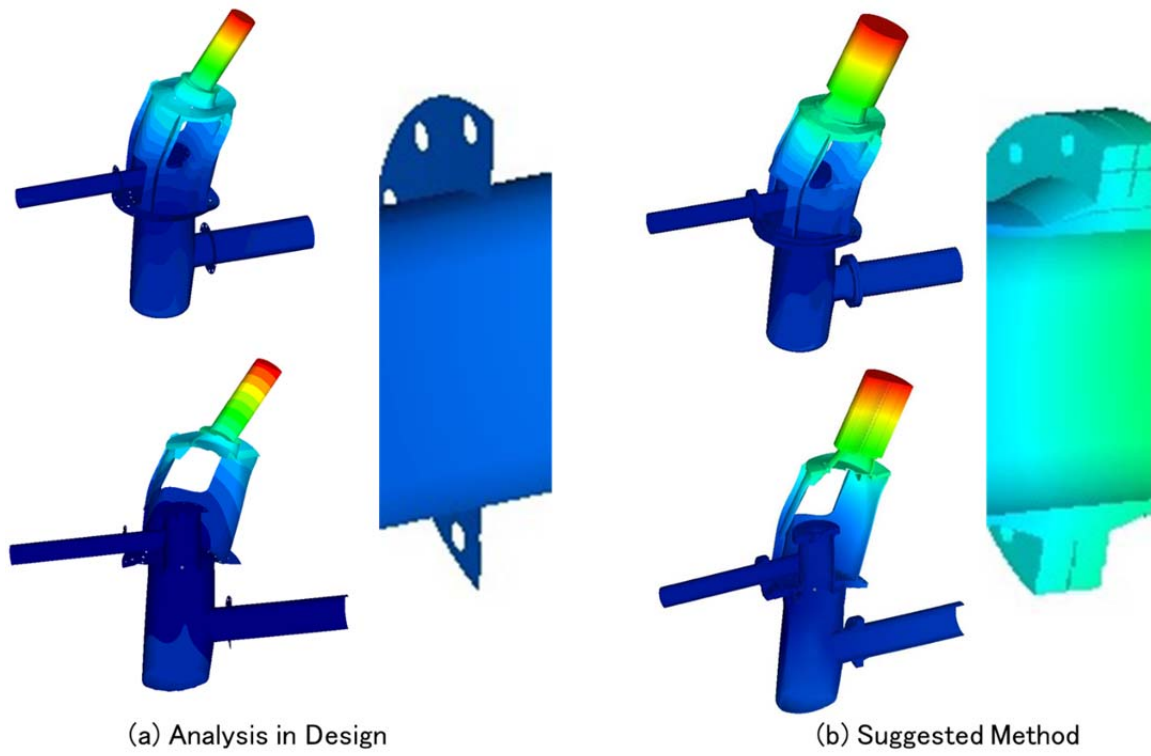


Fig.5-15 Comparison with suggested method

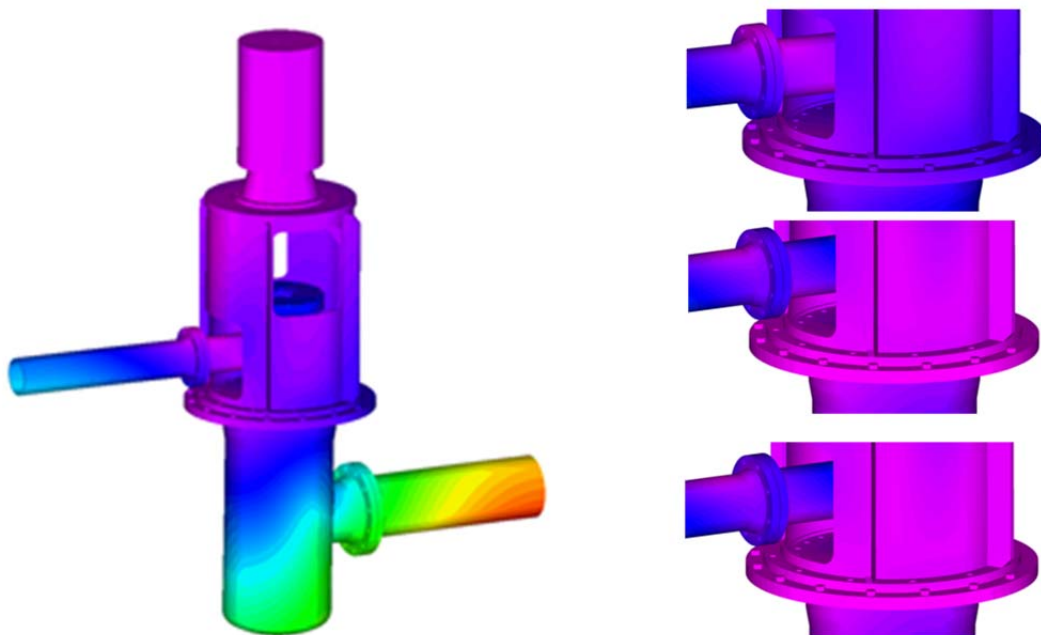
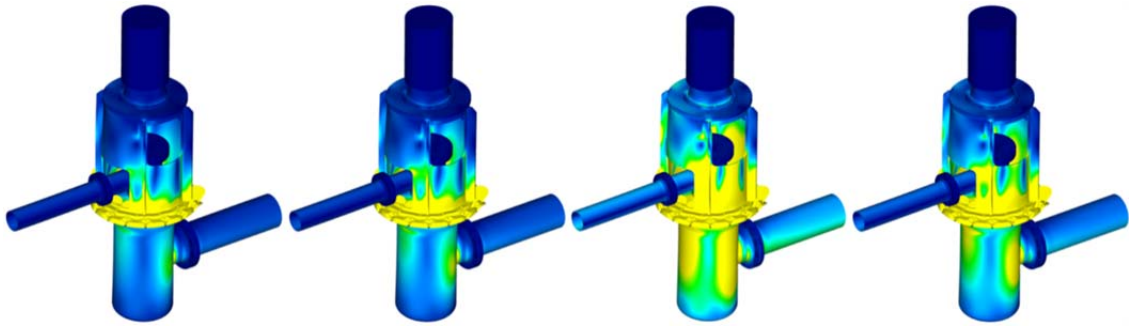
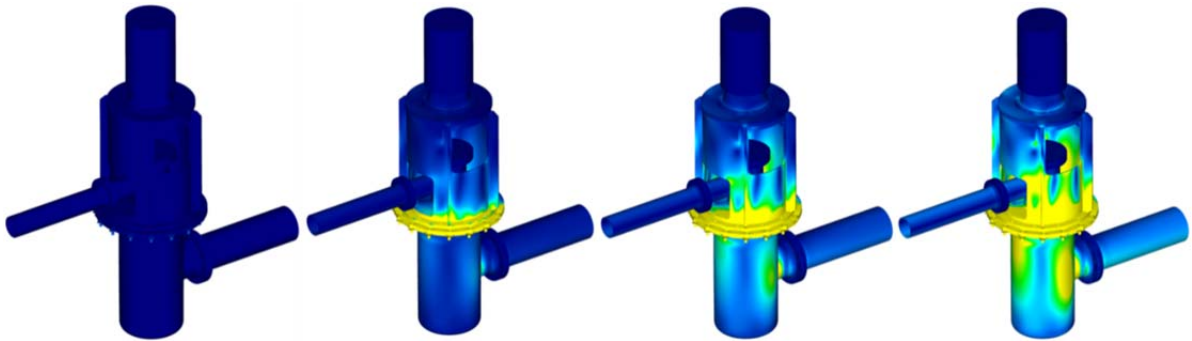


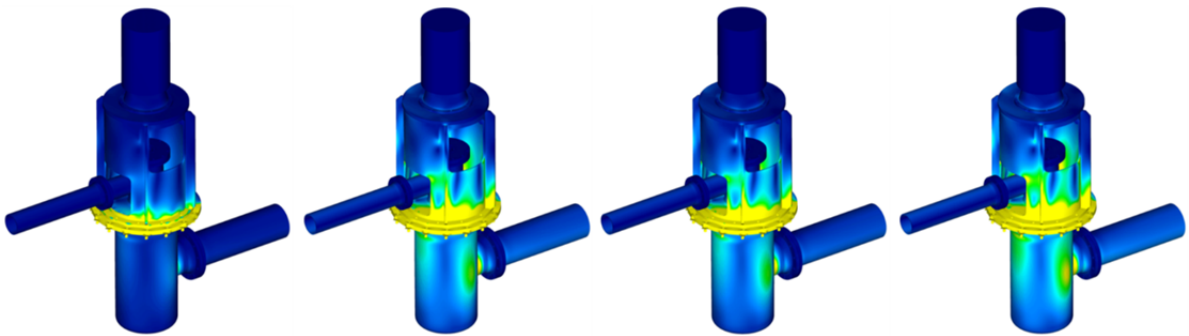
Fig.5-16 Analyses for interaction among components



(a) Time-history response analysis by El-Centro wave

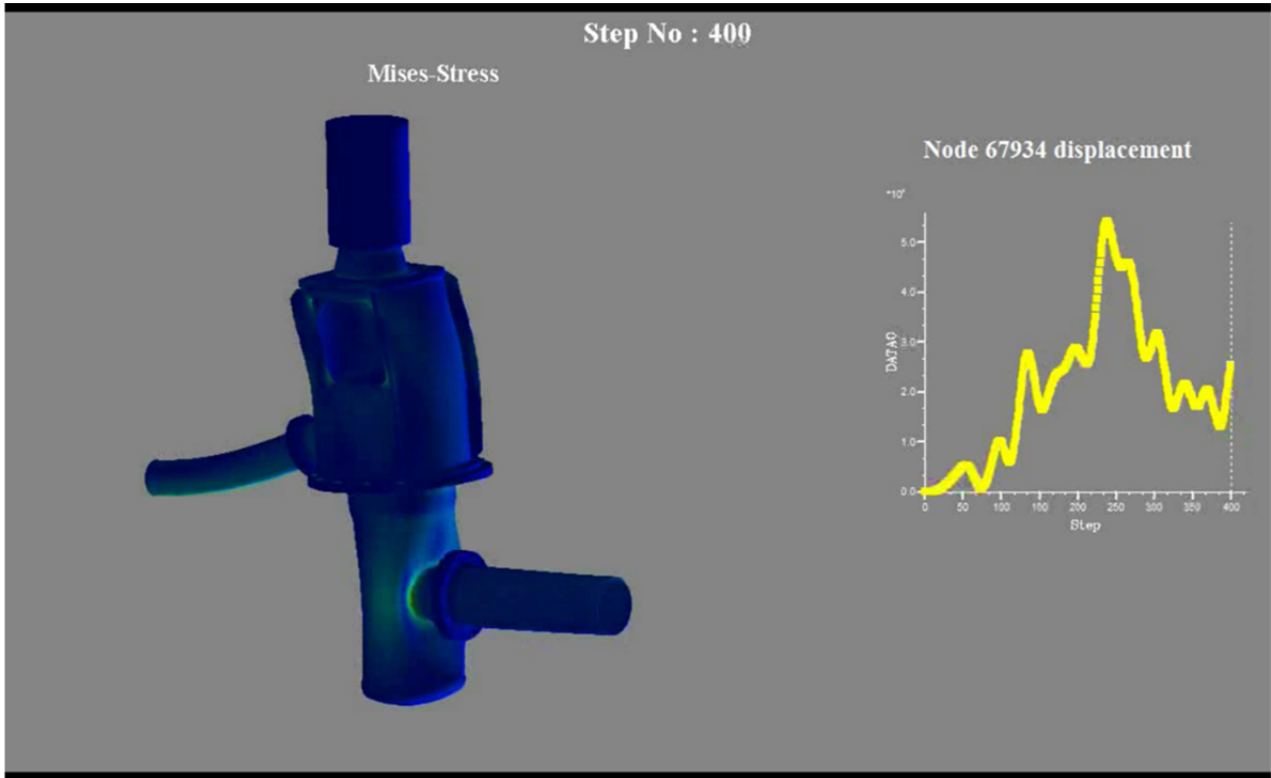


(b) Time-history response analysis by Taft wave



(c) Time-history response analysis by Hachinohe wave

Fig.5-17 Results for time-history response analysis



Note: the deformation in the picture is exaggerated to visualize its behavior.

Fig.5-18 Visual analysis for deformation

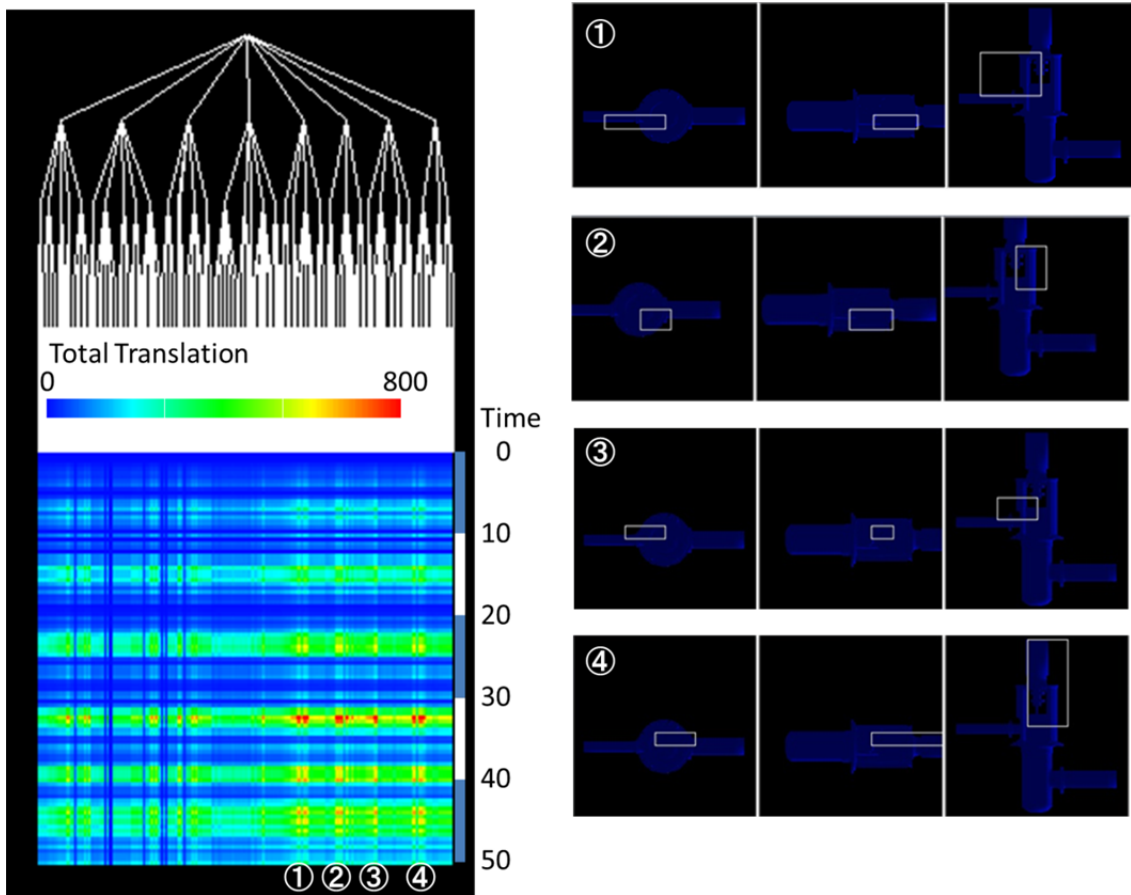


Fig.5-19 Visual analysis by spatio-temporal methodology

(ii) 千代田化工建設 株式会社

千代田化工建設(株)は、エネルギー、化学、医薬品、バイオ、FA 等のプラント・施設等にかかわる幅広い分野での大型重要施設事業を牽引している。大型重要施設の計画や設計等では、耐震設計の観点等から構造解析が様々な形で利用されている。千代田化工建設(株)におけるプラントは、大きく分けてストラクチャ、機器、配管から構成されている。現状では、これら3つのカテゴリーはその用途、設計・製作会社が異なることから、その荷重情報を互いにやり取りしながら、別々に構造設計や耐震解析が行われるのが普通である。しかし、個別の耐震解析ではその相互作用を考慮できないことから、より現実に近い状態を解析するためには、これら3つを合わせた一体シミュレーションをする手段を必要としている。ここで、ストラクチャとは鋼またはRC 構造物であり、配管、機器の重量支持がその主機能となる。機器には、化学プラントの場合、一般に、交換器(regenerator)、反応器(reactor)、精留器(fractionator)などがある。配管は、口径、長さのさまざまに異なる多数のパイプが、複雑に張り巡らされている。Fig.5-20 に3つの構造物群のイメージ図を示す。

モデル詳細度

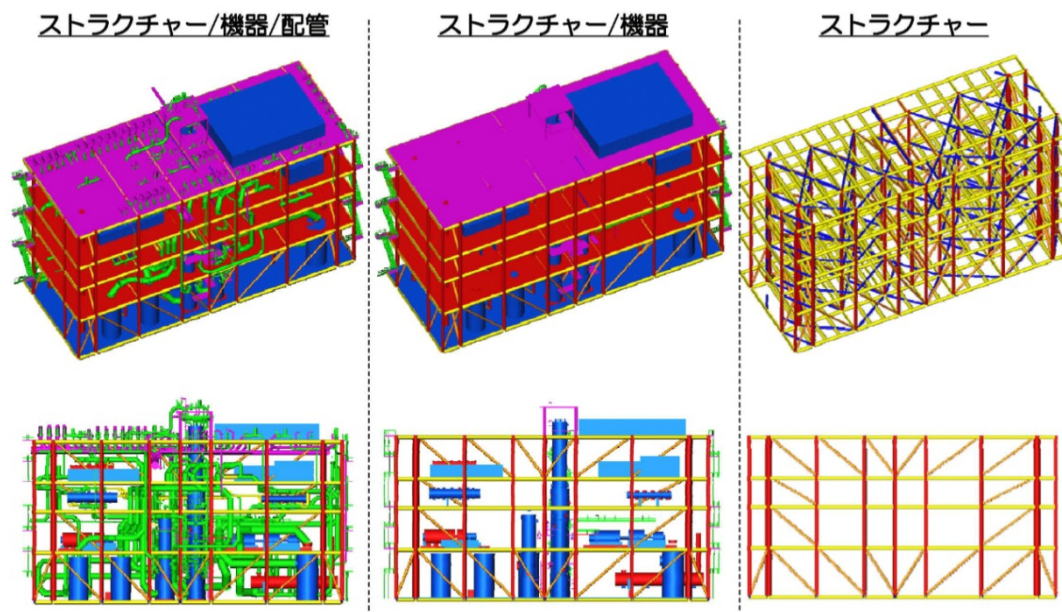


Fig.5-20 Three modules which constitute a plant

平成24年6月から、これら3つの構造物群を合わせてシミュレーションする手段として、本課題で進める「次世代耐震シミュレーション」技術の適用を検討している。平成25年度は、日本原子力研究開発機構によってストラクチャの3次元幾何モデルの作成が行われ、その詳細な有限要素解析モデルを完成させ、このモデルを用いた固有振動数解析とオンサイトで発生した地震波を用いた時刻歴応答解析を実施した。Fig.5-21に、そのモデル図を示す。Fig.5-22には、芸予地震波を用いた時刻歴応答解析結果例を示す。これらの成果は、千代田化工建設(株)ホームページで広報されるとともに、2015年2月12日の日刊工業新聞150212朝刊13面と業界新聞である化学工業新聞の一面に2015年2月10日に掲載された。

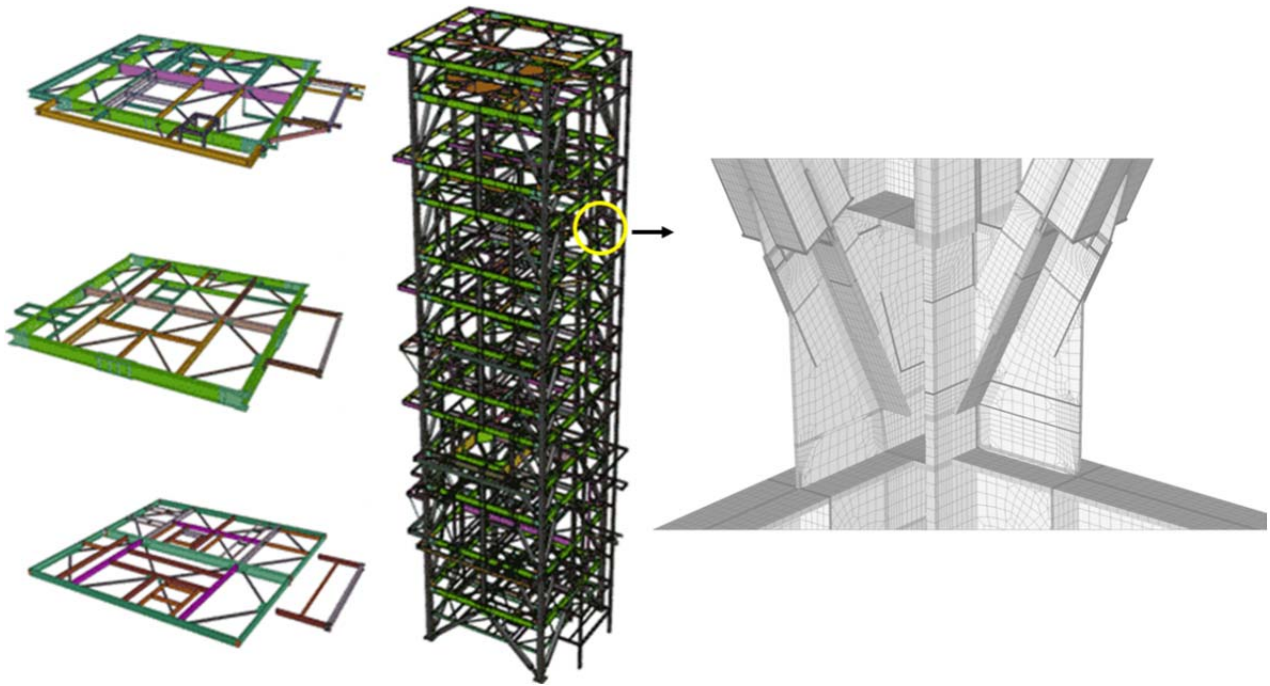


Fig.5-21 Finite element models for“Structure” of a petroleum plant

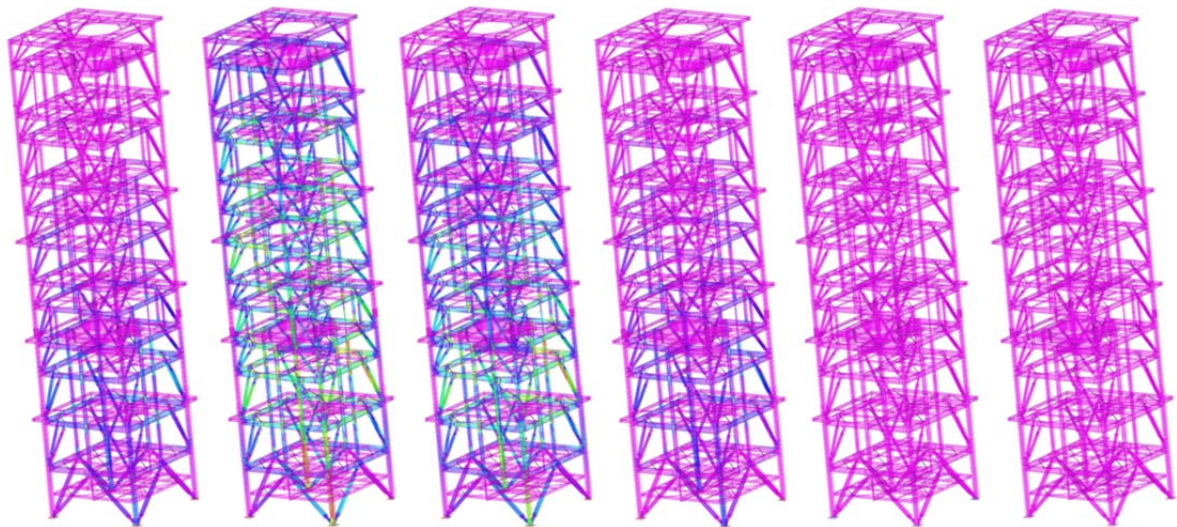


Fig.5-22 A time domain response analysis with Geiyo

平成 26 年度から東京大学が 千代田化工建設(株)と NDA を締結した上でストラクチャ、機器、配管のモデル化、統合シミュレーションを実施した。平成 26 年度は入力地震波として代表的な 3 波 El Centro(1940)、Taft (1952)、八戸 (1968)を用いて統合シミュレーションを行い、応答応力の検定等を実施してきた。

平成 27 年度はレベル 2Taft 波を入力地震動とし、ストラクチャ、機器、配管の統合シミュレーションによる弾塑性地震応答解析を実施した。3 次元大規模統合シミュレーションにより、これまでのストラクチャ、機器、配管それぞれ個別の耐震評価では得られなかった、振動モード等が得られるなど、より高度な耐震評価が可能となった。

(6) 計算科学技術推進体制構築

本分野における計算科学技術推進体制構築の主要施策としては、そのユーザーを、HPC ものづくりインフラの整備・普及と HPC 人的資源の強化・育成に分け、前者については①計算機資源の効率的なマネジメント、②研究成果の普及を、後者については③人材育成、④人的ネットワークの形成、⑤分野を超えた取り組みの推進を具体的な実施項目として組み入れた。なお、本分野における体制構築の進め方の特徴を踏まえ、4 つの事業に括って対応しており、その括りでの各実施項目の概要について、Fig. 6-1 に示す。



Fig. 6-1 Key measures of the Promotion of Computational Science and Technology Program

以下に、それぞれの項目の成果を報告する。

(6-1) 計算機資源の効率的なマネジメント

計算機資源の効率的なマネジメントとしては、「先導的研究課題に対する対応」と「多様な産業ニーズへの対応」を実施している。先導的研究課題への対応では、スーパーコンピュータ「京」の活用を前提とする課題推進のため、本格活用に至るまでの最適化を含めた課題別の具体的な利用計画を策定し実行していく。多様な産業ニーズへの対応では、多様な産業ニーズに応えるために戦略機関に専用窓口を設け、ニーズに応じた適正な HPCI の利用マネジメントを実施した。

(6-1-1) 計算機資源の有効活用に関する全体的な構想

次世代ものづくり分野においては、計算科学技術が牽引することによって達成できる世界最先端の技術開発によるトップランナーの輩出を強化することに加えて、ものづくりを広範にわたって支え

ている数多くの中堅・中小企業層の HPCI への参入を積極的に促す仕組みづくりを行うことにより、我が国ものづくり分野全体の抜本的な国際優位性強化への貢献を目指している。その実現のため、スーパーコンピュータ「京」を中核とする HPCI をシームレスに連携させ、目的や規模に応じた適切なハードウェアやソフトウェアが効率的に選択・利用できる環境の整備を目指す。民間企業のスパコンの利用促進を図るためには、準備段階・試用段階に適した HPCI の活用から最先端インフラであるスーパーコンピュータ「京」に至るまでの段階的・効率的利用マネジメントと利用支援を行うことが必須であり、そのための具体的な施策立案・実施体制を戦略機関に構築した。

(6-1-2) 研究支援協力

(1) 多様な産業ニーズへの HPCI 利活用の促進と支援活動の状況

本事業においては、当初より計画的・戦略的に実施する先導的研究開発課題以外にも公募によって広範の産業ニーズを満たす課題を採択し、それらの課題に取り組むための計算機資源の利用枠を一定量確保している。これにより、本分野における多様な目的を有する研究者・技術者にも広く HPCI を開放する。この公募事業を含んだ HPCI 普及推進施策の概要を Fig.6-2 に示す。

産業界ニーズを広く取り込むための取り組みについて、実証研究課題公募事業で公募を行い平成 23 年度は 3 件の申し込みがあり 2 件の課題を採択した。平成 24 年度は 8 件の申し込みがあり 6 件(うち 2 件は継続課題)の課題を採択した。課題の採択に際しては、本事業に参加している有識者数名に応募された課題について、技術的な観点より研究実施計画と課題内容のスパコン産業利用への貢献期待度などについて事前審査を依頼し、申込書からは判断できない内容については面談によるヒアリングを実施した。

平成 24 年度の採択課題は以下の通りであり、最初の 2 件が平成 23 年度からの継続課題である。

- ・ 一般財団法人日本造船技術センター「CFD による船舶の推進性能推定精度向上に関する研究」
- ・ JX 日鉱日石エネルギー株式会社「水素ステーション用 CFRP 蓄圧器の開発」
- ・ 株式会社インテック「音響業界における FrontFlow/Blue Acoustic モジュールの活用範囲の拡大に関する実証研究」
- ・ 株式会社 IHI「メゾスケールモデルを用いた FRP の損傷シミュレーション」
- ・ ソニー株式会社「全電子カノニカル分子軌道計算による金属タンパク質の電子移動反応解析」
- ・ 株式会社日立製作所「ガスタービン燃焼装置の燃焼解析のハイパーフォーマンスコンピューティング適用と実効性把握」

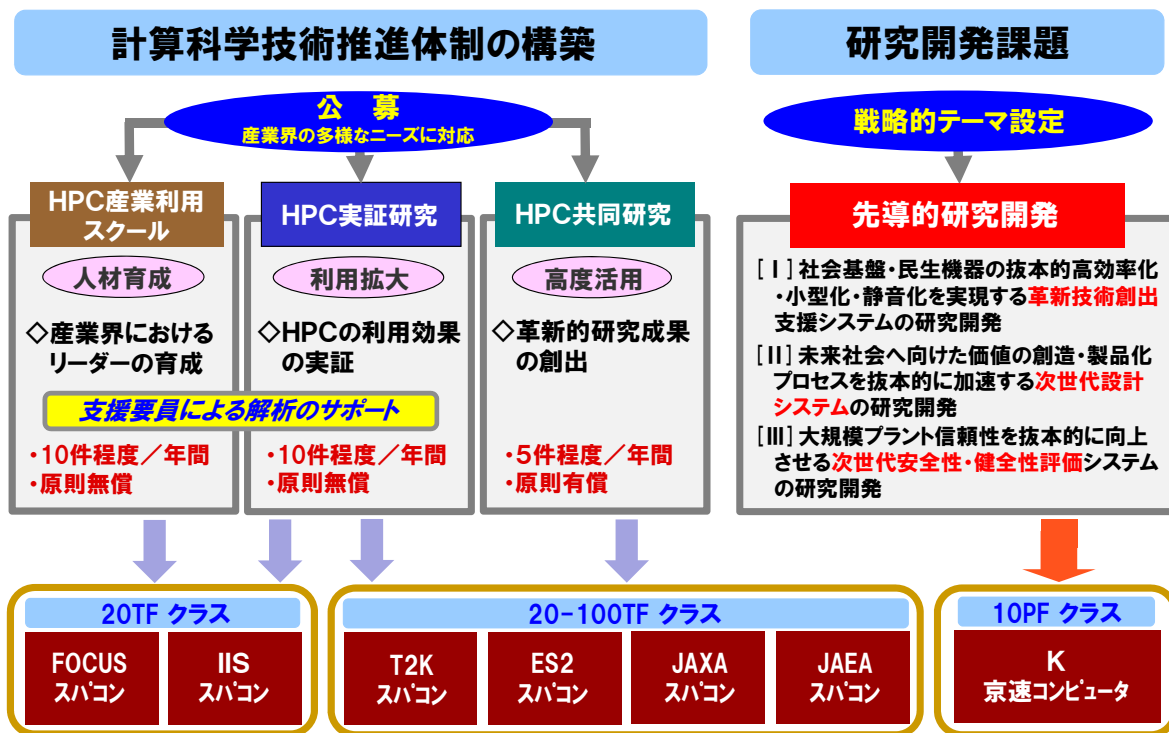


Fig.6-2 Programs for utilizing HPCI and their features

(6-1-3) 計算機資源の利用状況

計算機資源の具体的利用計画については、先導的研究課題を推進するためのスーパーコンピュータ「京」を中心とした利用計画と、計算科学技術推進体制構築に係わる諸施策(多様な産業ニーズ等への対応)を推進するための、目的に応じたレベルのスパコン利用計画を設定した。

戦略的・計画的に推進する研究開発課題を円滑に実施するため、平成23年3月に導入した「次世代ものづくり分野研究支援環境プロトタイプシステム(36ノード・5.06TFLOPS)」をはじめとし、計算科学振興財団より借り受けた「FOCUSスパコン(48ノード・5.2TFLOPS)」、東京大学情報基盤センターより借り受けた「FX10スーパーコンピュータシステム(600ノード・141.9TFLOPS 契約)」等の計算機資源を用意し、先導的課題の研究開発を実施した。

平成24年3月には研究開発課題をより効率良く実施するために、演算システムとして「次世代ものづくり分野研究支援環境システム(88ノード・24.78TFLOPS)」「次世代ものづくり分野大規模データの可視化処理支援システム(48ノード・13.52TFLOPS+GPU8ノード・CPU:1.08TFLOPS+GPU単精度:8.24TFLOPS)」を導入したほか、大規模なプレ・ポスト処理に活用できるための「次世代ものづくり分野前処理用大容量メモリ装置(64コア・2TBメモリ搭載)」「大容量メモリーサーバ(64コア・512GBメモリ搭載)」を導入した。

当分野で有している計算機環境(Table6-1)と「京」コンピュータの利用実績(Table6-2)を以下に示す。

Table 6-1 List of computer resources

名称	計算機名称	ノード単体性能	利用可能ノード数 (ログインノード等含まず)	システム全体性能	ノード間ネットワーク性能
次世代ものづくり分野 研究支援環境 プロトタイプシステム	FUJITSU PRIMERGY CX1000クラス	Intel Xeon X5670(Westmere-EP 2.93GHz, 6コア) x 2CPU 11.72GFLOPS x 6core x 2CPU = 140.64GFLOPS	36ノード	5.06TFLOPS	InfiniBand QDR x 1(40Gbps)
次世代ものづくり分野 研究支援環境システム	DELL PowerEdge R620クラス	Intel Xeon E5-2660(SandyBridge-EP 2.20GHz, 8コ ア) x 2CPU 17.6GFLOPS x 8core x 2CPU = 281.6GFLOPS	88ノード	24.78TFLOPS	InfiniBand QDR x 2(80Gbps)
次世代ものづくり分野 大規模データの可視化処理 支援システム	DELL PowerEdge R620クラス	Intel Xeon E5-2660(SandyBridge-EP 2.20GHz, 8コ ア) x 2CPU 17.6GFLOPS x 8core x 2CPU = 281.6GFLOPS	48ノード	13.52TFLOPS	InfiniBand QDR x 2(80Gbps)
	ビジュアルテクノロジー VT64 Server 9500 X-1SG IIクラス	Intel Xeon X5660(Westmere-EP 2.8GHz, 6コア) x 2CPU 11.2GFLOPS x 6core x 2CPU = 134.4GFLOPS	8ノード	CPU 1.08TFLOPS GPU(単精度) 8.24TFLOPS	InfiniBand QDR x 1(40Gbps)
情報基盤センター 大規模超並列システム FX10スーパーコンピュータ システム	FUJITSU PRIMEPC FX10	SPARC64 IXfx(1.848GHz, 16コア) x 1CPU 14.78GFLOPS x 16core x 1CPU = 236.48GFLOPS	624ノード 最大1440ノード	147.6TFLOPS 最大340TFLOPS	6次元メッシュ/トラス 5GB/s x 10 x 2(100GB/s)
次世代ものづくり分野前処理用大 容量メモリ装置	VT64 Server Extreme 8800 X-5	Intel Xeon E7-8837(Westmere-EX 2.66GHz, 8コ ア) x 8CPU 32GB x 64 = 2048GB Memory	1ノード	680.96GFLOPS	10Gbps Ethernet x 2
大容量メモリサーバ	VT64 Server Extreme 8800 X-5	Intel Xeon E7-8837(Westmere-EX 2.66GHz, 8コ ア) x 8CPU 32GB x 16 = 512GB Memory	1ノード	680.96GFLOPS	10Gbps Ethernet x 2

Table 6-2 Resource of “K” computer

		H24下期			
		割当資源量(NH)	利用実績(NH)	差異	利用率
分野4	研究開発課題(1)	5,130,605	5,117,324	13,281	99.7%
	研究開発課題(2)	4,828,685	4,442,046	386,639	92.0%
	研究開発課題(3)	15,575,502	11,183,072	4,392,430	71.8%
	研究開発課題(4)	3,530,605	2,660,103	870,502	75.3%
	研究開発課題(5)	1,730,605	1,712,767	17,838	99.0%
	計算科学推進体制構築	530,601	471,199	59,402	88.8%
分野4合計		31,326,603	25,586,511	5,740,092	81.7%

		H25			
		割当資源量(NH)	利用実績(NH)	差異	利用率
分野4	研究開発課題(1)	25,298,751	24,281,808	1,016,943	96.0%
	研究開発課題(2)	10,300,000	8,946,983	1,353,017	86.9%
	研究開発課題(3)	30,239,775	26,297,840	3,941,935	87.0%
	研究開発課題(4)	9,520,000	7,922,760	1,597,240	83.2%
	研究開発課題(5)	4,810,000	1,925,242	2,884,758	40.0%
	計算科学推進体制構築	1,886,000	1,355,518	530,482	71.9%
分野4合計		82,054,526	70,730,151	11,324,375	86.2%

		H26			
		割当資源量(NH)	利用実績(NH)	差異	利用率
分野4	研究開発課題(1)	12,050,000	11,874,501	175,499	98.5%
	研究開発課題(2)	8,200,000	7,936,713	263,287	96.8%
	研究開発課題(3)	27,664,277	27,418,446	245,831	99.1%
	研究開発課題(4)	6,078,000	4,541,400	1,536,600	74.7%
	研究開発課題(5)	5,770,000	5,336,900	433,100	92.5%
	計算科学推進体制構築	498,502	468,403	30,099	94.0%
分野4合計		60,260,779	57,576,363	2,684,416	95.5%

		H27(2015年12月末時点)			
		割当資源量(NH)	利用実績(NH)	差異	利用率
分野4	研究開発課題(1)	11,000,000	8,172,618	2,827,382	74.3%
	研究開発課題(2)	8,000,000	7,302,762	697,238	91.3%
	研究開発課題(3)	24,550,000	17,454,795	7,095,205	71.1%
	研究開発課題(4)	6,300,000	2,400,452	3,899,548	38.1%
	研究開発課題(5)	7,000,000	3,035,097	3,964,903	43.4%
	計算科学推進体制構築	855,734	196,247	659,487	22.9%
	分野4合計	57,705,734	38,561,971	19,143,763	66.8%

		H24下期～H27(2015年12月末時点)合計			
		割当資源量(NH)	利用実績(NH)	差異	利用率
分野4	研究開発課題(1)	53,479,356	49,446,251	4,033,105	92.5%
	研究開発課題(2)	31,328,685	28,628,504	2,700,181	91.4%
	研究開発課題(3)	98,029,554	82,354,153	15,675,401	84.0%
	研究開発課題(4)	25,428,605	17,524,715	7,903,890	68.9%
	研究開発課題(5)	19,310,605	12,010,006	7,300,599	62.2%
	計算科学推進体制構築	3,770,837	2,491,367	1,279,470	66.1%
	分野4合計	231,347,642	192,454,996	38,892,646	83.2%

(6-2) 研究成果の普及

(6-2-1) HPC/PF 開発整備

我が国のものづくりを担う産業界では、設計対象となる形状や物理が複雑になり、単純なシミュレーションだけでは設計支援の要求に応えることができなくなっている。「HPCI 戦略プログラム」分野4次世代ものづくりでは、産業界のエンジニアが抱える設計の諸問題の解決支援に資するために、将来を見据えた先端的機能を有するシミュレーションコードに加えて、プリポストなどの周辺技術も含めた統合的なシミュレーションシステムを、ユーザの効果的・効率的利用に資するためのソフトウェア利用ノウハウ等を含んだデータベースシステムとともに提供する。このシミュレータ群と関連するプリポスト処理を統合したシミュレーション環境をHPC/PFシステムとして整備した。また、「京」を中心とするHPCIで利用可能なアプリケーションラインナップの整備については、産業ニーズを考慮して順次増強してきた。

これまで、H22年度には企業ユーザに対するヒアリングを実施し、企業の設計利用のために必要とされる機能の洗い出しと利用シナリオを検討し、HPC/PFシステムの概念設計を行った。H23年度には、大規模並列計算の設計適用を促進するため、想定する利用シナリオから必要なシステム機能を絞り込み、システムの概念設計とプロトタイプ開発を実施した。H24年度は、プロトタイプを用いて、利用シナリオの実現性を検討し、運用版の事例データベースを開発した。H25年度以降は、開発したプロトタイプを基に、従来開発してきた国プロアプリのHPC/PF対応とシミュレータの事例を利用ノウハウとともに蓄積し、利用ガイドラインを構築してきた。また、戦略プログラムの各研究課題への展開を図った。

Fig.6-3には、HPC/PFの実装と展開の概要を示す。以下、HPC/PFが提供するメニューと代表的なユーザシナリオ、機能要件/非機能要件、システム構成を述べた後、サブシステムのプロトタイプの開発進捗とHPC/PFの展開について報告する。

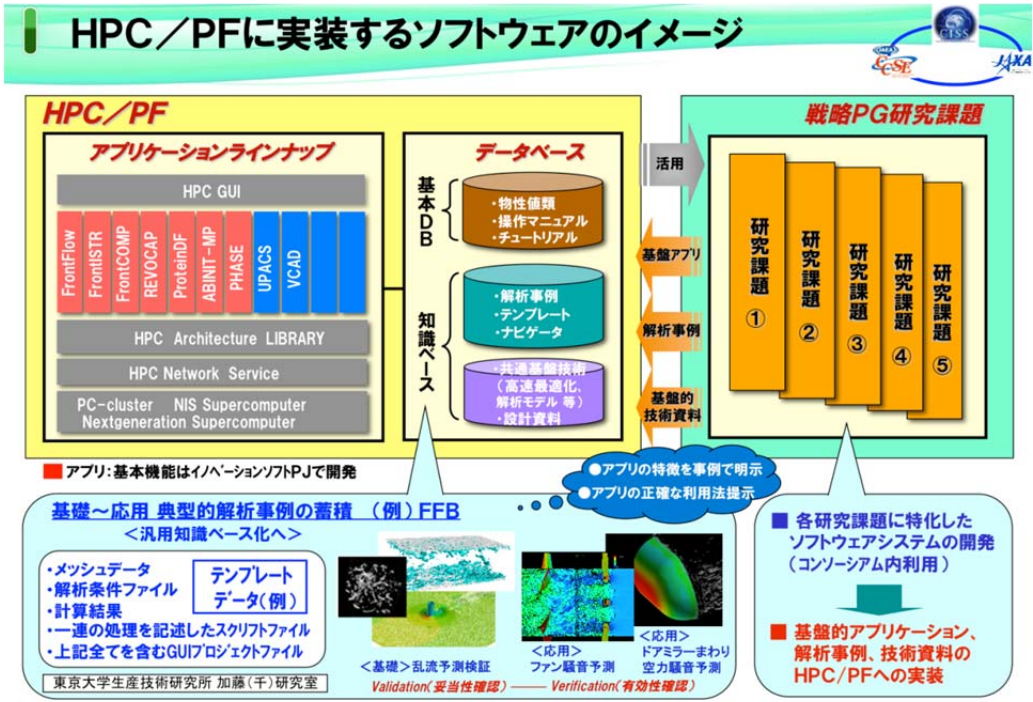


Fig. 6-3 Usage and Deployment Scenario of HPC/PF system.

(1) 中間評価までの成果

【HPC/PF の概念設計とプロトタイプ】

HPC/PF システムが提供する標準的なサービスメニューを Fig.6-4 に示す。ユーザの標準的な利用方法を提案したもので、HPC/PF が提供できるサービスの一例を示している。ここでは「アプリケーション利活用」、「事例閲覧・検索」、「情報交換」の 3 つのユーザに分類している。この標準サービスメニューは、後述するユーザの HPC/PF システムの使い方(シナリオ)の事例から導出した。

HPC/PF ポータルサブシステム 標準サービスメニュー

	アプリケーション利活用			事例閲覧・検索		情報交換	
	ユーザ事例解析	既解析事例追体験	テンプレート利用 応用解析	基礎～応用 事例概観 キーワード検索	アプリ機能 情報閲覧	ユーザ間の 活用情報・ 広報・閲覧	開発情報 共有
サブシステム連携フロー	PTL	PTL	PTL	PTL	PTL	PTL	PTL
	PJM/KDB	KDB	KDB-PJM	KDB	KDB	KDB	RSC
	VOX	PJM	VOX			LOG	DEV
	PDI	PDI	PDI			DEV	
	ENG	WFM	WFM				
	LSV/LDP	LSV/LDP	LSV/LDP				
	WFM/KDB	KDB	KDB				
メニュー	◆ プロジェクト・格子作成 ◆ ユーザ事例の計算 ◆ 可視化分析処理 ◆ 計算手順の編集・テンプレート作成・保存	◆ サンプルで提供される事例の追体験 ◆ 同一/別ハードでの結果確認(計算時間、計算精度等)	◆ 提供テンプレートを利用した解析 ◆ パラメータスタディ ◆ 同上に関する解析ノウハウ ◆ 結果のDB化	◆ 各アプリ別解析事例一覧 ◆ 各技術分野別事例一覧 ◆ 類似ソフトとの比較 ◆ キーワード検索事例一覧	◆ 各アプリ主要機能一覧 ◆ 各アプリ利用マニュアル ◆ 各アプリ関連論文類一覧	◆ 各アプリ間の活用情報・広報・閲覧	◆ 開発情報共有
	◆ 各アプリ別利用情報一覧 ◆ 各技術分野別利用情報一覧 ◆ Q & A 情報	◆ 開発状況 ◆ 最新ソース ◆ バグトラック ◆ 対応ハード情報 ◆ その他					

Fig.6-4 Service items of HPC/PF system.

まず、想定している基本的なユーザシナリオを以下に示す。

- ・ 事例データベースの閲覧： 利用者は登録された事例を見て、HPC/PF システムでどのような計算ができるのか、計算精度や計算時間、性能などのデータを理解する。
- ・ 追試： 事例データベースにリンクされたプロジェクトファイルをダウンロードし、ユーザのローカル環境に合わせてパラメータを調整、提供される入力ファイルを用いて事例の計算を実行し、動作と精度確認を行う。
- ・ テンプレートを利用した計算： 事例データベースに登録されているテンプレートファイルをダウンロードし、ユーザのもつ課題の実行を補助する。
- ・ 定形作業の支援： 作業プロセスが決まっている一連のタスクを自動実行する。このため、作業プロセスの記述を行うスクリプトを記述し、それを実行する。
- ・ パラメータスタディ： あるパラメータ空間での最適設計を行う利用方法を支援する。最適化を図るためにパラメータ空間を設計し、各パラメータに応じた入力ファイルを自動作成する。また、バッチジョブの投入と結果の回収を自動化するスクリプトをホストサーバーとクライアントの両側でシームレスに実行する。
- ・ 大規模格子の作成： 大規模並列計算に必要な格子を自動生成する。
- ・ 大規模データのポスト処理： 計算結果から設計に必要な情報を集約する。
- ・ 解析結果の活用： シミュレーションした結果とそこから得られた知見を再利用可能な形式でアーカイブ化し、設計に活用する。

以上のシナリオを実現するために、Fig.6-5 のようなシステム構成を設計した。特徴としては、HPC/PF システムは、シミュレーションを実施し、結果を得るために必要なツール類を備える統合的なシミュレーション環境である。つまり、シミュレータを中心としてプリポスト機能やシステムの有効利用を図るデータベースなどを備えたシステムの総称である。そのシステム開発の基本的なポリシーは、柔軟性と利便性の両立を図る点にある。特に、大規模並列システムでの稼働も視野に入れているため、大規模システム故に生じる雑多な設定や手続きを緩和する仕組みを提供する。HPC/PF には、サブシステムがその構成要素として抽象化されている。各サブシステムは、それぞれさらに細かい機能プログラムから構成される場合もある。稼働する計算機環境としては、京コンピュータ、地球シミュレータ、大学の情報基盤センタなどに設置されているクラスタシステム、企業のプライベートクラウド、ベンダーのパブリッククラウド、個人の PCなどを想定している。

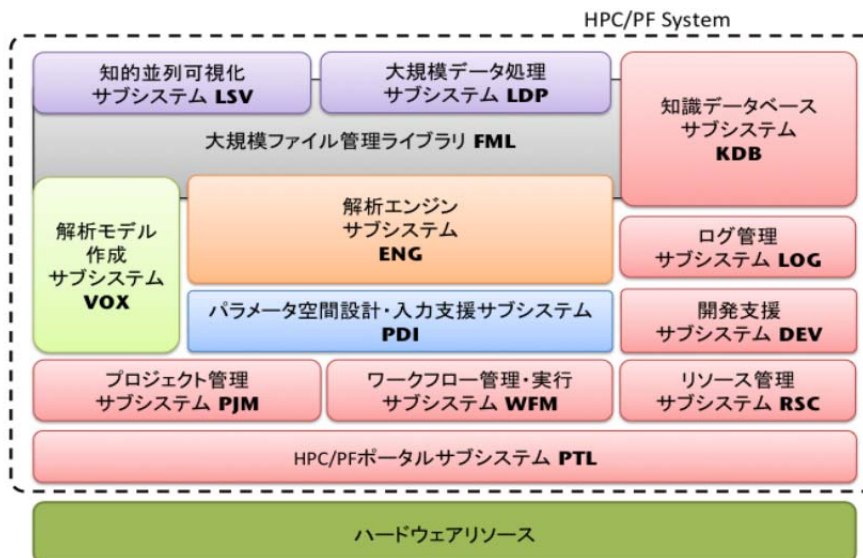


Fig. 6-5 Functional diagram of HPC/PF

以下、これまで開発してきた主要なコンポーネントについて説明する。

大規模格子生成技術

数百億から1兆規模の大規模格子を扱う場合、メモリ使用量の点からプリプロセスで格子生成を行うのは不可能であるため、ソルバー実行時に直接格子を生成する方法を開発、検討した (Fig. 6-6)。この方法を実施するために必要な基本情報 (ジオメトリデータ、計算空間分割情報、境界条件情報) は、専用のプリプロセスアプリケーション FXgen を用いる。FXgen は FFV のプリプロセッサとして動作するが、生成した階層型直交格子の情報を非構造格子へ変換し、今後他ソルバーでも利用可能な方法も提供する。

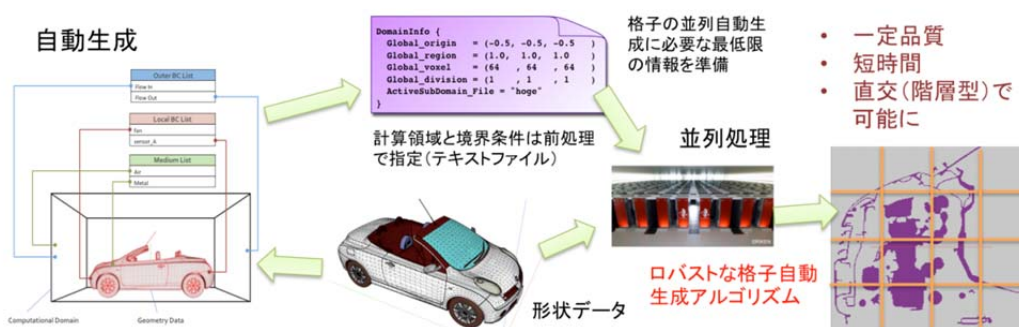


Fig.6-6 Approach of large-scale grid generation

事例データベース

HPC/PF 上で提供される各種シミュレータの活用支援を、以下の3段階のシナリオに分類し、各シナリオで必要となる情報支援機能を検討した。

- 第1段階: 事例を通したシミュレータ機能の理解
- 第2段階: 例題の追試
- 第3段階: 自課題の計算

事例データベースは、HPC/PF 上で提供される各種ソフトウェアに関する情報、およびそれを利用した解析事例情報を管理し、さまざまな視点から探索する機能、探索された解析事例を再現するために必要となる一連のデータ等を提供する機能、また、利用者自身の課題解決に各種ソフトウェアを自由に使いこなすための支援情報を集約し、提示する機能を持つ。また、解析事例コンテンツの自動登録機能については、解析実行ワークフローにより登録の自動化を進めることによりユーザの労力を軽減し、これによりシステムの実際的な運用を支援する(Fig. 6-7)。



Fig. 6-7 Case example database

アウトリーチサイトのプロトタイプ

分野4の研究開発成果を産業界へ普及展開するため、アウトリーチ活動推進 Web サイト運用に向け、Web サイトのプロトタイプ(Fig.6-8)を構築した。このサイトのコンテンツとして、HPC/PF の事例データベースとの連動、開発ソフトウェアのダウンロード、ユーザフォーラム、編集コンテンツなどがある。現在プロトタイプの評価を行っており、本 Web サイトを分野4全体のアウトリーチ活動のひとつとして展開している。



Fig. 6-8 A proto-type of outreach web site

(2) 中間評価以降の成果

本節では、中間評価以前に開発した HPC/PF のプロトタイプに対して、機能やアプリ対応のために追加した機能について述べる。

【HPC/PF ポータル】

HPC/PF ポータル GUI は、ユーザが本システムの提供する各種機能の利用の基点となるユーザーインターフェイスである。HPC/PF を構成する各種サブシステムの起動、特定の目的のために作成したワークフローの実行ナビゲーションなどのサービスをユーザに提供している。本 GUI については、平成25年度までに基本機能の開発は終了し、以降はユーザ利便性向上のための機能拡充を行ってきた。ここでは中間評価以降の開発成果について報告する。

・プロジェクト再開機能

実行中断、あるいは実行中にプロジェクトエディタウィンドウを閉じたプロジェクトを確認することができる、resume project ボタンをメインメニュー画面に追加した。(Fig.6-9) これにより実行中、あるいは実行中断したプロジェクトの一覧が確認でき、一覧からプロジェクトを選択し再開できるようになった。(Fig.6-10) なお、表示されるプロジェクトの状態は「Ready」「Running」「Finished」「Failed」のいずれかである。

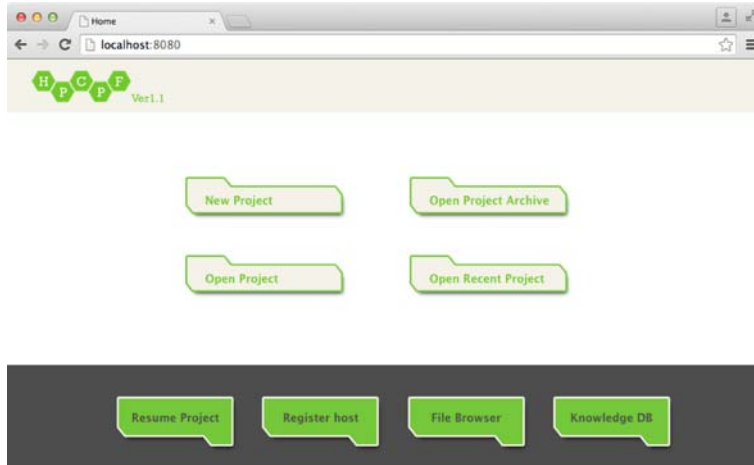


Fig. 6-9 Main menu of HPC/PF Portal GUI

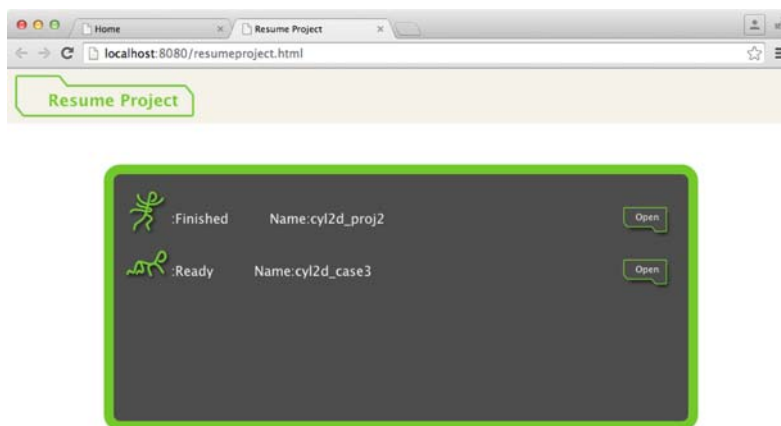


Fig. 6-10 A screen shot of Resume Project window

・ファイルブラウザ画面におけるドラッグ・アンド・ドロップ機能

ローカルマシンとリモートマシン、あるいは異なるリモートマシン間でのファイルのコピーや移動といった操作をドラッグ・アンド・ドロップで実現できるようになった。また、ファイルの圧縮や展開なども直感的な操作で実行できる。(Fig. 6-11)

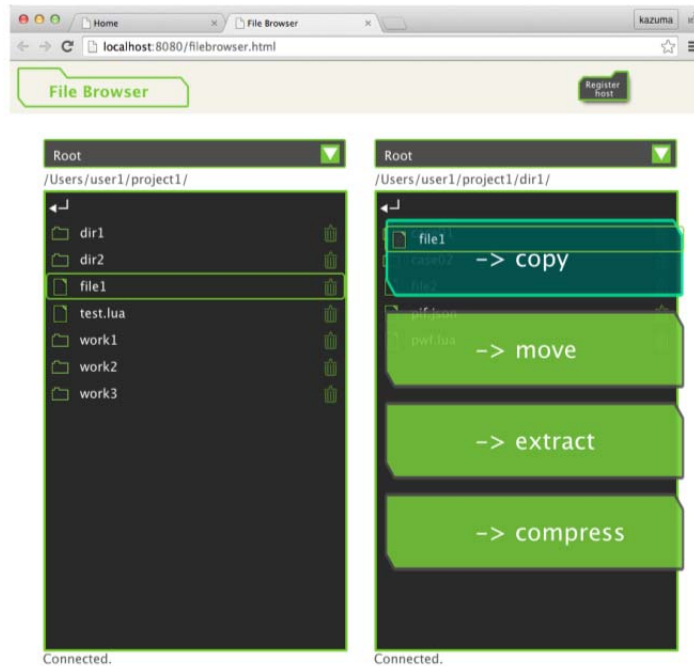


Fig. 6-11 A screen shot of File Browser window

・グラフィカルなワークフローエディタの実装

プロジェクトのワークフローはプリ処理、解析処理、ポスト処理といった単位(ケースと呼ぶ)で構成されており、各ケースの内部処理はスクリプト言語 Lua とデータ定義言語 JSON で記述されている。ワークフローエディタではプロジェクトに用意されたケースをユーザが必要な処理順に接続する機能を提供する。各ケースには固有の入出力定義があり、それがケース間の接続可不可のルールとなっているため、ワークフローエディタは各ケースの入出力定義を参照することで、ケース間の正しい接続をナビゲートする。(Fig. 6-12)

ワークフローエディタ上でケースを選択すると、画面右側にケースのプロパティ編集タブが表示される。ユーザは各ケース情報の閲覧や、ケース実行に必要な各種パラメータ(実行する計算機資源や並列実行数など)をここで設定することができる。(Fig. 6-13)

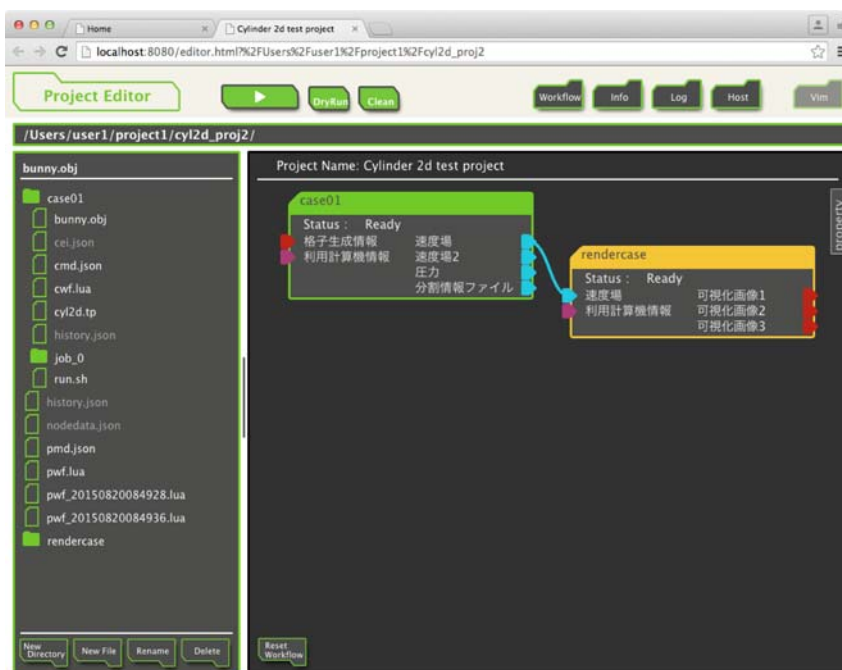


Fig. 6-12 A Screen shot of Workflow Editor window

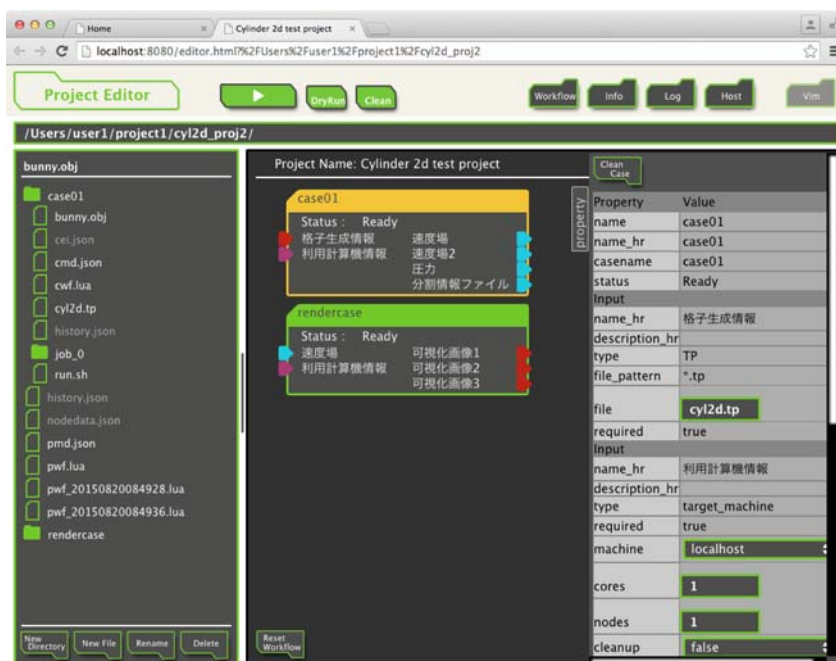


Fig. 6-13 A Screen shot of Case Property Tab of Workflow Editor window

・プロジェクト管理機能の拡充

HPC/PFにおけるプロジェクトとは、HPC/PF環境下で実行される解析処理ワークフローと入力・出力データを纏めたファイルセットのことである。知識データベースに掲載されている各種解析事例には、当該解析内容のプロジェクト圧縮ファイル(プロジェクト・アーカイブ)を添付しているものがあり、ユーザが自由にダウンロードして手元のHPC/PFポータルGUI上で閲覧し、解析の再現実行が出来るようになっている。ユーザ独自のプロジェクトはポータルGUI上で新規作成するか、既存プロジェクトを

編集して作成することが可能である。HPC/PF ポータルが実装するプロジェクト管理機能について平成26年度以降は以下の開発を実施した。

1. プロジェクト、ケース単位での変更履歴の管理

プロジェクト、あるいはケースが変更された場合、変更前のプロジェクトメタデータファイル、ケースメタデータファイルをバックアップ保存する仕組みを実装した。これによりユーザの試行錯誤の履歴がトレース可能となる。

2. プロジェクトの中断・再開機能を実装

前述のポータル GUI の項でも説明したとおり、プロジェクトの実行ステータスを記録することで、中断されたプロジェクトの再開実行が可能となった。

【大規模データ管理】

「京」コンピュータを始めとする大規模並列計算環境におけるシミュレーションは、必然的に並列計算となり、出力されるデータは大規模かつ並列ファイルとなる。解析モデル作成、計算準備、計算、後処理というシミュレーションのプロセスにおいて、シミュレーションから後処理にかけては、大規模なデータの管理とハンドリングがシミュレーション実行上の問題点となる。この問題点を解決するために、大規模並列ファイル管理のしくみを構築した。対象とするデータは、構造格子系データ(Cartesian)、非構造格子系データ(Unstructured)、粒子データ(Particle)の3つのカテゴリである。これらを総称して xDMLib(x Data Management library)と称する。

並列ファイル管理の設計には、ファイル入出力性能、および管理の容易さの両方の観点が必要となる。並列ファイル入出力性能は、ファイルシステムの構成と密接に関連する。Lustre ファイルシステム上での MPI-IO の積極的な利用も考えられるが、本プロジェクトでは産業展開時の汎用的なファイルシステムへの展開を考慮し、1 プロセスにつき 1 ファイルを入出力する形式を基本として管理方式を検討した。

管理のしくみとして、多数出力されるファイルに命名規約を設け、ファイル群を管理するメタファイルにより、ファイル群全体を管理する。Fig.6-14 にデータファイル管理ライブラリ概念を示す。xDMLib はアプリケーションのファイル入出力機能を担当し、リスタート時に任意のプロセス数に対応する入出力、ソルバーとポストプロセスアプリ間のデータ授受、データ圧縮、ファイル変換などの機能を提供する。

Fig.6-15 に示すメタデータは、テキストパーサー形式のテキストにより管理情報を記述する。テキストパーサーは json に準じる階層記述が可能な簡易なデータ記述フォーマットである。メタデータとして、proc.dfi と index.dfi の 2 種類を用いる。Proc.dfi は並列プロセスの情報を記録し、index.dfi は出力された変数ファイル毎に作成され、ファイル命名規則、endianness、ファイルフォーマット、データの値域や出力タイムスタンプなどの情報を保持する。メタデータを解釈することにより、ファイルシステムに存在するデータの場所や種類などを知ることができる。また、ライブラリを利用することで、データの読み書きを省力化することもできる。特に、Fig.6-16～Fig.6-19 に示すように、異なるプロセス数でリスタートする場合の自動データロードや、細分化してリスタートする場合の補間処理などにも対応可能で、アプリケーションの柔軟なリスタート処理、あるいはアプリ間のデータ授受の処理を統一して扱うことができる。

File management function for Cartesian data structure on distributed parallel environment

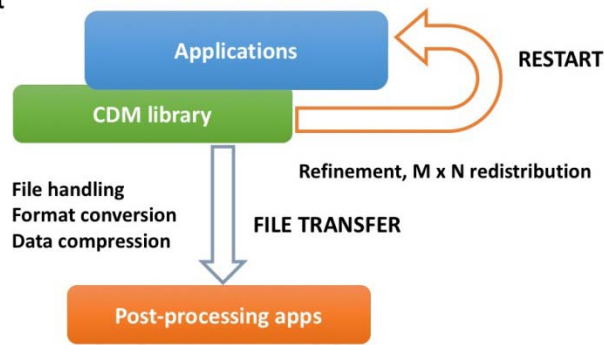


Fig.6-14 Concept of xDMLib

```

FileInfo {
  DFItpe = "Cartesian" // "Cartesian" , "Non_Uniform_Cartesian"
  DirectoryPath = "./ "
  TimeSliceDirectory = "off" // Option to generate dir. for each time
  Prefix = "field " // Base file name ※1
  FileFormat = "bov " // "bov" , "fun" ※1
  FieldFilenameFormat = "step_rank"
  GuideCell = 0 // Halo
  DataType = "Float32 " // Data length ※2
  Endian = "little " // ※3
  Variable[@] { name = "u " } // 登録変数名、記述順に登録
  Variable[@] { name = "v " }
  Variable[@] { name = "w " }
  . . .
}

FilePath {
  Process = "proc.dfi " // specify the name of 'proc.dfi' file
}

TimeSlice { // for time varying data
  Slice[@] { // repeat time slices for each fileout timing
    Step = 0 // step
    Time = 0.0 // time
    AverageTime = 20 // as may be necessary
    AverageStep = 30
  }
}
:
(to be continued)
  
```

Fig.6-15 Example of metadata file

Same # of processes, same resolution => Std. restart

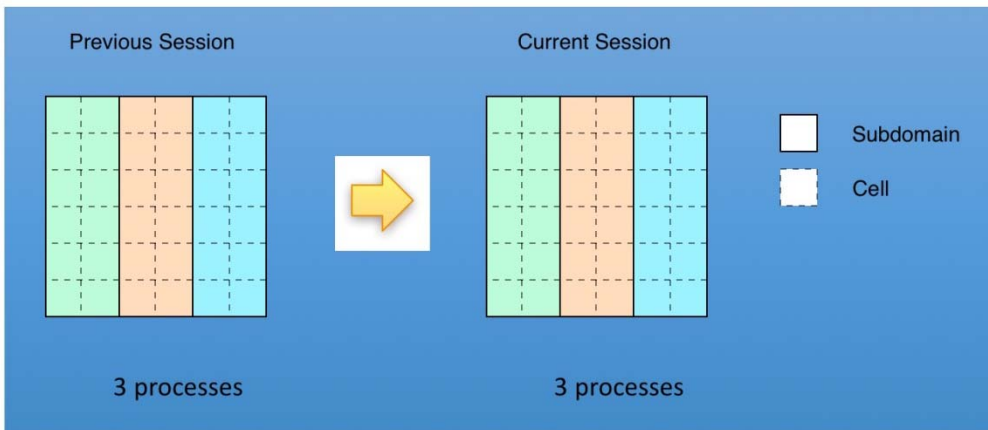


Fig.6-16 Restart pattern : restart by same number of process.

Same # of processes, different resolution => Refinement restart

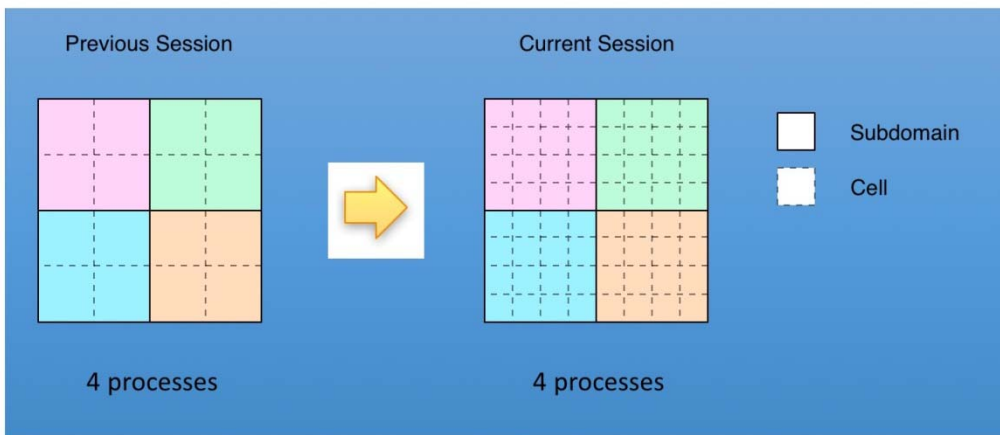


Fig.6-17 Restart pattern : refinement restart.

Different # of processes, same resolution => M X N restart

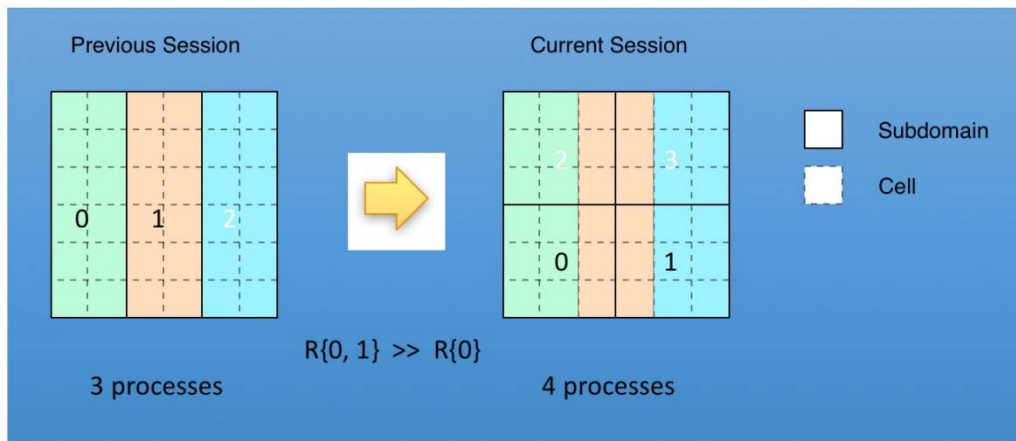


Fig. 6-18 Restart pattern : MxN restart.

Different # of processes, different resolution => M x N /w refinement

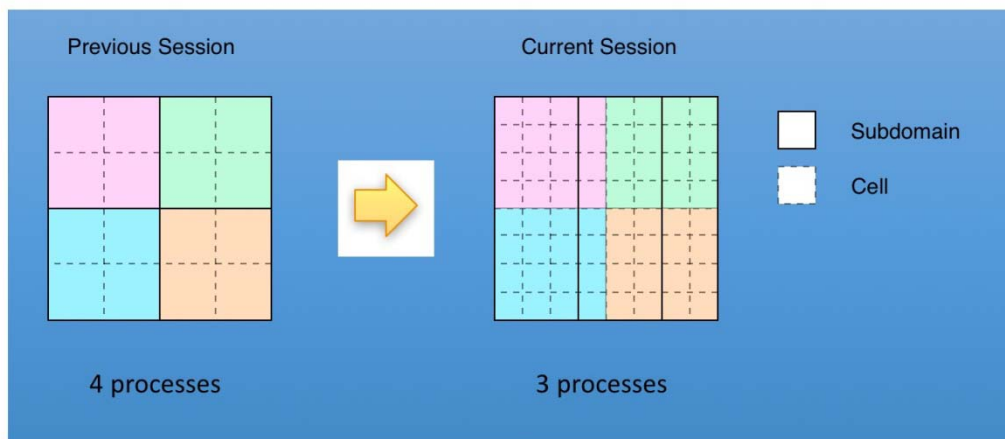


Fig. 6-19 Restart pattern : MxN with refinement.

【ワークフロー】

前述の HPC/PF ポータル GUI においてワークフローをグラフィカルに組み立て可能なワークフローエディタを実装したが、これに先立ちワークフローを記述するスクリプト言語を、Xcrypt と Lua の混成から Lua 言語のみに変更した。これは元々 Xcrypt が担当していたリモート計算資源へのジョブ投入制御機能を Lua 言語による各種ライブラリの組み合わせでプロトタイピングを実施し、置き換え可能であると判断できたことによる。

ワークフロースクリプトは、プロジェクト全体のワークフローを定義する ProjectWorkflowFile (PWF) と、個別のケース内部の処理を定義する CaseWorkflowFile (CWF) に分かれている。PWF は CWF の実行順を定義するが、実行順序はポータル GUI のワークフローエディタ上でユーザがグラフィカルに指定可能であり、ワークフローエディタでのケース実行順の指定に従い PWF をシステムが自動生成する (Fig. 6-20)。

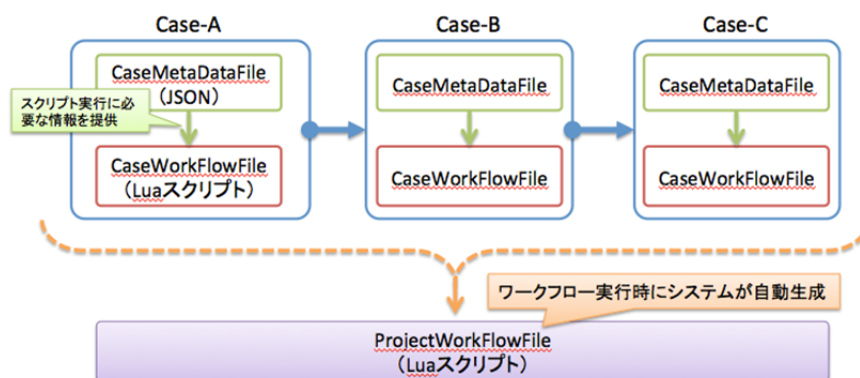


Fig. 6-20 Construction of Workflow Script

CWF はプロジェクト作成者がスクリプトコードを作成する。リモートマシンへの接続や、ファイル転送、ジョブ投入、実行監視といった HPC を利用したバッチジョブ実行において必要な処理は HPC/PF システムによる API が提供されており、CWF コード作成者は簡略化されたスクリプト記述のみでリモートマシンでのバッチジョブ実行を記述できる。以下に示す CWF サンプルコードはリモートマシンでのバッチジョブ実行の典型的な例だが、大部分のコードは後述する PDI サブシステムが自動生成するスケルトンコードであり、コード作成者は必要があれば一部を修正追加するだけでよい。以下の例の場合、利用する解析アプリの事情に合わせるためにリモートマシンに転送したファイルを配置するディレクトリ作成処理を追加している (Fig. 6-21)。

```
Require('hpcpf')

local delimiter = '/'
if (getPlatform() == 'Windows') then
    delimiter = '¥¥¥'
end

local dxjob = require('dxjob')
local caseDir = getCurrentDir() .. getBasePath()

print('Case dir = ' .. caseDir)

-- Get remote host info.
```

```

Local targetConf = require('targetconf')

-- Create job manager
local dj = dxjob.new(targetConf)

-- Define job
require("paramsweep_wf")
for I, v in ipairs(jobs) do
    v.name = 'FFB_HEX_PIPE'
    v.core = 20
    v.node = 1
end

local CasePath = string.gsub(getBasePath(), "/", "") .. delimiter
for I, v in ipairs(jobs) do
    local JobPath = CasePath .. v.path .. delimiter
    -- Create subdirectories
    makeDir(JobPath .. "PRE")
    makeDir(JobPath .. "PRE" .. delimiter .. "data0")
    makeDir(JobPath .. "PRE" .. delimiter .. "data1")
    makeDir(JobPath .. "PRE" .. delimiter .. "data1" .. delimiter .. "DDD")
    makeDir(JobPath .. "POST")
    makeDir(JobPath .. "POST" .. delimiter .. "FV")
    makeDir(JobPath .. "POST" .. delimiter .. "HISTORY")
    -- Copy files
    copyFile(CasePath .. "PIPE.A.grd", JobPath .. "PRE" .. delimiter .. "data0" ..
delimiter .. "PIPE.A.grd")
    copyFile(CasePath .. v.job, JobPath .. v.job)
end

-- Add job
for I, v in ipairs(jobs) do
    dj:AddJob(v)
end

-- Generate boot script
dj:GenerateBootSh()

-- Send caseDir
dj:SendDir(caseDir)

-- Get casedirectory name
local uppath, casename = getDirAndName(caseDir)

-- Submit job
for I, v in ipairs(jobs) do
    print('Job ` .. I .. `submitted')
    dj:SubmitAndWait(casename)
    print('Job ` .. I .. ` finished')
end

-- Retrieve results
dj:GetDir(casename, uppath)

print('Case finished')

```

Fig.6-21 A example of Case Workflow Script File.

実行制御は、ポータル GUI のワークフローエディタで行う。ユーザーは目的の解析プロジェクトをポータル GUI 上で選択した上で、ワークフローファイルを指定し、RUN ボタンを押下すると、予め指

定されていた条件(利用する計算リソースや、ジョブの並列実行数など)に基づき、リモートマシンへの接続、ジョブ発行、ジョブの終了監視、結果の回収を行う。パラメータスタディなど、複数ジョブ発行が指定されているワークフローの場合、後述する PDI サブシステムのパラメータスイープ機能を利用することで複数ジョブ実行用スクリプトを自動生成し、全てのジョブ終了を待ち合わせた上でポスト処理を行い、最適解を選出するフロー制御が可能である。

【PDI】

PDI サブシステムはユーザのパラメータ空間設計およびパラメータ入力を支援する。中間評価以降の機能拡充について以下説明する。

・Lua 言語ワークフロースクリプト出力

ワークフロースクリプトを全て Lua 言語での記述に統一したことに伴い、バッチジョブ実行スクリプトモードも、Xcrypt で提供されていた API と同等の機能を HPC/PF システム側で用意したことでより簡素化され、HPC/PF システム利用者の学習コストも低減することが出来た。以下に示す CWF サンプルコードは PDI が出力するスケルトンコードであり、リモートマシンでのバッチジョブ実行の典型的な例である。リモートマシンでのファイルの配置変更など、ユーザ固有の事情による変更が必要ない限り、PDI が出力した CWF スケルトンコードはそのままワークフローで実行可能である(Fig. 6-22)。

```
require('hpcpf')
local ex = ...
local caseDir = ex.caseDir
print('Case dir = ' .. caseDir)
local dxjob = require('dxjob')
local dj = dxjob.new(ex)
require('pdi_generated')
jobs = GET_JOBS(ex)
for k, j in pairs(jobs) do
    dj:AddJob(j)
end
dj:GenerateBootSh()
dj:SendCaseDir()
dj:SubmitAndWait()
dj:GetCaseDir()
return ex.outputFiles
```

Fig. 6-22 An example of CaseWorkflow generated by PDI

・多目的最適化進化計算への対応

工学的な最適化問題は、最適化の対象である目的関数が複数(一般的には2個以上)存在する多目的最適化問題である場合が多く、しかもそれらは相反する性質を持つ場合も珍しくない。本プロジェクト研究課題4において JAXA(宇宙航空研究開発機構)が開発した Cheetah は、生物の進化のメカニズムを模倣した進化的アルゴリズム(EA)を多目的最適化問題に適用する MOEA(多目的進化計算)のソフトウェアフレームワークであり、PDI では、設定したパラメータ空間内での最適化計算を Cheetah を使用して実行するためのスクリプトの生成と、実行環境を設定する機能を実装した。PDI には元々グリッドサーチによるパラメータサーベイ機能が実装されており、Cheetah との連携においてはグリッドサーチ用の GUI を援用した。PDI のパラメータサーベイ画面において「MOEA param survey」選択時の状態を下図に示す(Fig.6-23)。なお、PDI の最適化進化計算対応機能は汎用的なインターフェース設計がされているため、Cheetah 意外の進化計算モジュール

にも対応可能である。

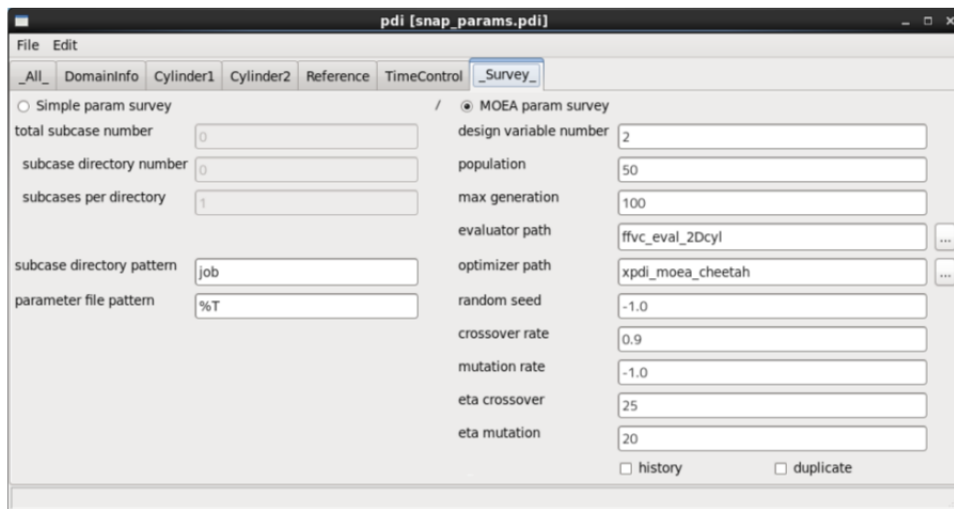


Fig.6-23 A Screenshot of Parameter Survey Window of PDI.

「MOEA param survey」選択時の各設定値は以下の通りである。

population

MOEA 計算における個体数。ここで設定した個数だけ作業ディレクトリが作成される。

max generation

MOEA 計算における世代数。ここで設定した回数だけ

{ ソルバー実行 ⇒ パラメータ値 (設計変数) の進化計算 }

がループ実行される。

evaluator path

MOEA 計算における評価モジュール (evaluator) のパス。評価モジュールは、ソルバーの実行結果から目的関数の値を計算する外部プログラムであり、最適化問題毎に使用者が用意する必要がある。

optimizer path

MOEA 計算における進化計算モジュール (optimizer) のパス。進化計算モジュールは、目的関数の値から、次の世代の設計変数を生成する外部プログラムであり、PDI では デフォルトで Cheetah を使用した進化計算モジュール (xpdi moea cheetah) を提供している。

random seed

MOEA 計算において使用する乱数の種。有効な値は 0.0~1.0 だが負値を設定した場合は乱数が設定される。

crossover rate

MOEA 計算における交叉率。

mutation rate

MOEA 計算における突然変異率。有効な値は 0.0~1.0 だが負値を設定した場合は自動的に (1.0/設計変数個数) が設定される。

eta crossover

MOEA 計算における交叉の distribution index。

eta mutation

MOEA 計算における突然変異の distribution index。

history

MOEA 計算において全評価個体の履歴ファイル (interface/history.txt) を作成するかどうか。

duplicate

MOEA 計算において重複する設計点を作成するかどうか。

MOEA 計算における設計変数空間の指定は、通常のグリッドサーチと同様の手順で行う。ただし PDI のパラメータ設定画面においてレンジ指定を表す「sweep range」の表記が「search range」に変わり、パラメータの刻み幅(delta)は有効精度の入力欄(precision)に変わる (Fig. 6-24)。

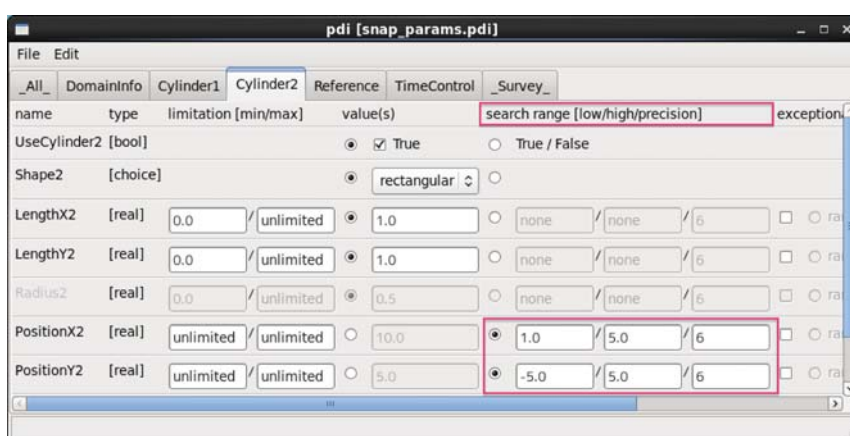


Fig. 6-24 A Screenshot of Parameter Setup Window of PDI.

MOEA 計算処理が設定されたケースワークフローの実行内容の概要は以下の通り。

1. optimizer(xpdi moea cheetah) を世代番号 0 で実行。これにより、探索空間内のランダムな設計点が個体数個生成され、初期化ファイルに出力される。
2. 内部コマンド xpdi genparam を実行。内部的に PDI がバッチモードで実行され、各サブケースディレクトリ配下にソルバ入力パラメータファイルが生成される。
3. 個体数個分のソルバー実行ジョブが発行される。
4. 全てのジョブの完了後、evaluator が実行される。これにより、各ジョブ毎の目的関数 (および制約条件) が計算され、ファイルに出力される。
5. 世代番号を+1し、最大世代数回、処理 1 から繰り返す。

【Cheetah 連携】

前述の PDI サブシステムによる多目的最適化問題用進化計算モジュールとの連携実装により、HPC/PF システムは Cheetah を利用した多目的関数に対応した進化計算による最適化設計が可能となった。その実践事例として、本プロジェクトのアウトリーチウェブサイトである「計算工学ナビ」の解析事例データベースに、下図に示す解析事例を掲載した(Fig. 6-25)。

2次元角柱周り流れの進化計算による最適化

●ソフトウェア
ものづくり>最適化設計>CHEETAH
ものづくり>流体解析>FFV

●V&V区分
Verification 検証
Validation 妥当性確認

●その他の分類
HPC/PF再現実行対応事例

概要
二次元の流れ場に2個の角柱を前後に配置した際の、後方角柱が受ける外力の大きさを最小化する位置を、CHEETAHによる遺伝的進化計算によって最適化する。

解析モデル

使用ソフトウェアモジュール

- FrontFlow/violet Cartesian(FFV-C) version 2.4.1
- HPC/PF PDI version 1.4.5
- HPC/PF GUI Portal (hpcpfGUI)
- CHEETAH version 1.0

計算格子
500 × 400 (x 5)

計算条件
レイノルズ数 Re = 1000, 計算時刻 t = 0.0 - 3.0

最適化パラメータ

- 設計変数: 角柱2の位置(角柱中心のX, Y座標)
- 目的関数: 第2角柱が受ける外力($t = 2.0 - 3.0$ での F_x, F_y の絶対値の平均)を最小化
- 制約条件: 流れ場が発散せずに解けること
- 個体数: 20
- 最大世代数: 10

上記の条件の下、以下の処理をワークフローで実行。

```

    graph TD
      A[初期状態の設計変数リストを乱数生成(個体数)] --> B[各設計変数においてソルバー(FFV-C)を実行]
      B --> C[ソルバー実行結果について評価関数値を計算  
(ffvo_eval_2Dcolを使用:この問題専用の評価モジュール)]
      C --> D{最大世代数に達した場合は終了  
(この時の設計変数がハレート候補となる)}
      D --> E[評価関数値を元に次世代の設計変数リストを計算  
(Cheetahを使用し、遺伝的アルゴリズムで計算)]
      E --> B
  
```

※、上記の計算条件は「最適化自動化」を検証する目的で設定されたものであり、実用的な最適化問題とはならない。
実用的な計算条件とするためには、計算時刻の範囲、第2角柱位置の探索範囲および個体数を選択に(十分大きく)設定する必要がある。

実行環境
FOCUSパソコン(1解析あたり1ノード×4コア)

解析時間

- 1解析: 約2分
- 1世代: 約10分
- 全体: 約100分

詳細説明
初期状態では探索範囲内にランダムに配置されていた第2角柱位置が、第10世代では、ほぼ2点の位置に集約されており、最適化が正しく実行されていると考えられる。ただし、計算時刻を短く設定しており、十分に発達していない流れ場で評価を行っているため、実際の流れ場における最適な角柱配置を示しているのではない点に注意。

・第1世代での角柱の配置

・第10世代での角柱の配置

・第10世代における設計変数の1つ(第2角柱位置: $x=3.227612, y=-0.180157$)での流れ場 (V-Isioによる可視化結果)

HPC/PF再現実行対応プロジェクトアーカイブ
ffv_cyls_moeda

問合せ先
keno@riken.jp
tkawanab@iis.u-tokyo.ac.jp
yoh@fujitsu.com

[更新日] 2015年12月4日

Fig.6-25 A Screenshot of sample case using MOEA on KDB.

本解析事例は、解析アプリケーションにFFV-Cを利用した「二次元角柱周り流れの進化計算による最適化」である。これは二次元の流れ場に2個の角柱を前後に配置した際の、後方角柱が受ける外力の大きさを最小化する位置を、Cheetahによる遺伝的進化計算によって最適化する問題である。最適化ワークフローを Fig. 6-26 に示す。