**様式 C-19**

# 科学研究費補助金研究成果報告書

平成 21 年 6 月 26 日現在

**研究種目：若手研究(B)**
**研究期間：2006～2008**
**課題番号：18760067**
**研究課題名（和文）３次元形状計測装置に基づくメッシュレス構造解析**

**研究課題名(英文)　Meshless structural analysis using three-dimensional scanning system**

**研究代表者**
**仲田　晋（NAKATA SUSUMU）**
**立命館大学・情報理工学部・准教授**
**研究者番号：00351320**

研究成果の概要：本研究課題は，実在する構造物の形状を 3 次元形状計測装置により取得し，その力学的特性をシミュレーションにより解明するための数値計算の一手法を確立することを目的としている．ここで提案する手法はメッシュレス解析技術と曲面モデリング技術を組み合わせた方法であり,特に計算の効率化や精度向上を目的とした新しい計算アルゴリズムの開発,並列化による高速化，および 3 次元複雑形状の応力解析の定式化とシミュレーションの実現が研究成果として挙げられる．

交付額

（金額単位：円）

| | 直接経費 | 間接経費 | 合　計 |
|---|---|---|---|
| 2006 年度 | 2,000,000 | 0 | 2,000,000 |
| 2007 年度 | 500,000 | 0 | 500,000 |
| 2008 年度 | 500,000 | 150,000 | 650,000 |
| 総　計 | 3,000,000 | 150,000 | 3,150,000 |

研究分野：工学
科研費の分科・細目：応用物理学・工学基礎・工学基礎
キーワード：数値解析，シミュレーション工学，コンピュータグラフィックス，形状モデリング，構造解析

## １．研究開始当初の背景

3 次元物体の力学的構造を解析するための手法としては有限要素法が一般的であり，建築物の耐性の解析などに広く利用されている．この場合，解析対象の 3 次元形状は CAD などのソフトウェアを利用して単純な形状の組み合わせとしてモデリングし，これをメッシュ構造に変換することで有限要素解析を実現している．一方，メッシュ構造の代わりに解析対象内部に分布する点群（節点）を利用して解析するメッシュレス法も存在する．この手法は流体解析の分野では比較的古くから利用され，SPH 法（Smoothed particle hydrodynamics）に代表されるように，流体を粒子として表現することで解析を実現している．また，1994 年に発表された EFG 法(Element Free Galerkin method, Belytschko, et al.）以降，構造解析分野においてもメッシュレス法が多く見られるようになり，有限要素法に代わる手法として発展してきた．特に 2002 年に発表された RPIM（Radial Point Interpolation Method, Wang, et al.）は境

界における変位の条件(Dirichlet 境界条件)を厳密に満たすという性質を持ち，3 次元応力解析においてその有効性が示されている．

我々はこのメッシュが不要であること，さらに3次元応力解析においても有効に活用できるという性質に着目し，彫刻のような複雑な形状の応力解析を実現するための計算手法の開発に着手した．解析対象の形状は3次元形状計測装置により取得されるデータを利用してモデリングする．ここでは計測点群からの形状モデリング手法として陰関数曲面モデルを採用する．陰関数曲面モデルによる点群補間法は2002年のTurkらによる手法や2003年のOhtakeらによる手法など，数多くの手法が提案され，計測点群を補間あるいは近似する曲面の自動生成が可能となっている．また，陰関数曲面モデルは曲面の内外判定が用意であるという特徴を持ち，メッシュレス法における節点の生成，および曲面内部における定積分の計算に活用できる．我々の研究グループでは，この陰関数曲面モデルと RPIM を組み合わせたメッシュレス解析手法を提案し，3次元の応力解析へ適用した．

## ２．研究の目的

本研究課題は，複雑な形状を持つ構造物に対し，実測値に基づく構造解析を行うための一手法を確立することを目的としている．ここでは前述のように，実在する構造物の形状を3次元形状計測装置により取得することを想定し，応力解析や動解析のための数値計算アルゴリズムの開発に関連する以下3点の項目についての研究を主に進めてきた．

(1) RPIM法の高速化

前述のとおり，RPIM法は3次元構造解析においても効果的であることが示されているが，一方でその計算量が問題となっている．解析精度の向上のためには節点数の増加や積分点数の増加が必要となるが，RPIMの計算量はそのいずれにも大きく影響されるため，高速化を実現する新たなアルゴリズムの開発が不可欠である．

(2) アダプティブ節点への適用

3 次元の構造解析では計算の効率化のため，解の性質に応じて節点を適応的に配置された節点を利用することがある．この適応的に配置された節点に対し，RPIMを効果的に適用し，精度の向上や計算量の削減を実現するアルゴリズムの開発を目的とする．

(3) 並列計算アルゴリズム

RPIM を用いたメッシュレス法の並列計算アルゴリズムを開発し，計算の高速化を図る．ここでは特にグラフィックスハードウェア（以下，GPU: Graphics Processing Unit）を利用した並列計算を想定している．GPU 上での並列計算は SIMD (Single Instruction Multiple Data) の構成を持っており，これに適したアルゴリズムの開発が必要となる．

## ３．研究の方法

(1) RPIMの基本的な考え方は，偏微分方程式の解を構成する基底関数を，クロネッカーのデルタ関数の性質を満たすように構成することである．すなわち，解析領域$\Omega$と境界$\Gamma$上に$N$個の節点$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$が与えられたとき，方程式

$$\begin{cases} -\Delta u = f \quad \text{in} \quad \Omega \subset \mathrm{R}^2, \\ u = \bar{u} \quad \text{on} \quad \Gamma_{\mathrm{D}}, \quad \dfrac{\partial u}{\partial n} = \bar{q} \quad \text{on} \quad \Gamma_{\mathrm{N}}, \end{cases} \tag{1}$$

の解$u(\boldsymbol{x})$ $(\boldsymbol{x} = [x, y]^{\mathrm{T}} \in \Omega)$を，各節点に対応して定義される基底関数$\phi_1(\boldsymbol{x}), \ldots, \phi_N(\boldsymbol{x})$の線形結合$u(\boldsymbol{x}) = u_1\phi_1(\boldsymbol{x}) + \cdots + u_N\phi_N(\boldsymbol{x})$で表現し，このとき，基底関数が

$$\phi_i(\boldsymbol{x}_j) = \begin{cases} 1 & (i = j) \\ 0 & (i \neq j) \end{cases}$$

を満たす．なお，ここでは簡単のため2次元Poisson方程式を例として用いるが，一般性を失うことは無く，3 次元の応力解析にも拡張可能である．ここでの基底関数の定義を以下に示す．解析領域内の点$\boldsymbol{x} = [x, y]^{\mathrm{T}} \in \Omega$に対し，これを中心とする円または矩形領域（以下，サポート）を考え，サポート内に含まれる節点を$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$とする．このとき，サポート内の節点に対応する基底関数の値は以下で定義される．

$$\begin{aligned} &[\phi_1(\boldsymbol{x}), \ldots, \phi_{n+3}(\boldsymbol{x})]^{\mathrm{T}} \\ &\quad = G(\boldsymbol{x})^{-1}[R_1(\boldsymbol{x}), \ldots, R_n(\boldsymbol{x}), 1, x, y]^{\mathrm{T}}. \end{aligned} \tag{2}$$

ただし，$R_i(\boldsymbol{x})$は対応する節点を中心とするRBFであり，例えば

$$R(\boldsymbol{x}) = (\|\boldsymbol{x} - \boldsymbol{x}_i\|^2 + (4d_c)^2)^q, \ q = 1.03$$

というように定義される．ここで，$d_c$は平均節点間距離を表す．行列$G(\boldsymbol{x})$は以下で定義される．

$$G = \begin{bmatrix} B_0 & P_0 \\ P_0^{\mathrm{T}} & 0 \end{bmatrix}, \quad P_0 = \begin{bmatrix} 1 & x_1 & y_1 \\ \vdots & \vdots & \vdots \\ 1 & x_n & y_n \end{bmatrix},$$
$$B_0 = \begin{bmatrix} R_1(\boldsymbol{x}_1) & \cdots & R_n(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ R_1(\boldsymbol{x}_n) & \cdots & R_n(\boldsymbol{x}_n) \end{bmatrix}. \tag{3}$$

ここで定義された基底関数を用いて以下の連立１次方程式を構成し，解くことで，偏微分方程式の近似解を得る．

$$\begin{aligned} &A[u_1, \ldots, u_N]^{\mathrm{T}} = [y_1, \ldots, y_N]^{\mathrm{T}}, \\ &a_{ij} = \int_\Omega \nabla\phi_i(\boldsymbol{x}) \cdot \nabla\phi_j(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \\ &y_i = \int_\Omega f(\boldsymbol{x})\phi_i(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}. \end{aligned} \tag{4}$$

行列 $A$ と右辺ベクトルの各要素は解析領域上の定積分として定義され，数値積分則を適用して計算する．以上が RPIM の基本的な計算手順であり，メッシュ構造を用いずに近似解を得ることが可能である．ここで，RPIM の計算量について考察すると，基底関数を評価するためには連立 1 次方程式(2)を構築して解く必要があり，これが計算量の増加をもたらしている．特に連立 1 次方程式(4)の構築のための数値積分の計算では，各積分点で基底関数とその偏微分を評価する必要があり，このため,連立１次方程式(4)を解くよりも，その構築のために計算時間が費やされることも多い．この問題を解決するため，本研究課題では RPIM に高速化のための修正を加え，基底関数を少ない計算量で評価するためのアルゴリズムを開発した．このアイデアを以下に示す．

1) 解析領域 $\Omega$ を複数の矩形領域 $\Omega_1, \ldots, \Omega_L$ に分割する．
2) $l = 1, \ldots, L$ について，以下を実行する．
   a．$\Omega_l$ を拡大した矩形領域 $\hat{\Omega}_l$ をサポートとし，$\hat{\Omega}_l$ に含まれる節点を $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ とする．
   b．節点 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ を利用し，(3)と同様に行列 $G_l$ と右辺ベクトルを構成する．
   c．LU 分解 $G_l = L_l U_l$ を求める．
3) 点 $\boldsymbol{x} \in \Omega$ における形状関数を，以下の手順で評価する．
   a．$\boldsymbol{x}$ を含む分割領域を探索し，$\Omega_l$ とする．
   b．先に計算した $G_l$ の LU 分解 $L_l, U_l$ を用い，方程式(2)を解くことで，基底関数の値を得る．

以上が修正版のアルゴリズムである．基底関数のサポートが従来の RPIM と異なるため，このアルゴリズムで定義される形状関数は従来の RPIM とは異なる．ただし，本質的な意味での性質，すなわち，クロネッカーのデルタ関数の性質を満たし，かつサポートがコンパクトであるという点については同様である．また，後述するように，計算精度の面でも従来の RPIM と差が少ないことが実験的に確認されている．修正版アルゴリズムの特徴は，前述のアルゴリズムの 2) において，方程式(2)の係数行列の LU 分解をあらかじめ前処理として計算するという点である．この前処理をより，基底関数の評価時に行列(3)を構築する必要がなく，さらに方程式(2)を解くための LU 分解がすでに与えられているため，高速に解を得ることが可能となる．

(2) 前述の高速化のアルゴリズムは，解析領域の分割に基づく手法であり，この分割領域をどのように作成するかが問題となる．もし節点の密度が領域全体で一様であれば，解析領域を等間隔の矩形領域に分割すればよい．しかし，アダプティブに生成された節点の場合，等間隔の矩形領域を用いることは可能であるが，サポート内の節点数が節点密度に応じて変わる．これは節点が高密度に存在する分割領域において行列 $G_l$ のサイズが大きくなることを意味し，前述のアルゴリズムの 2) の前処理，および 3) における基底関数の評価の計算量の増加を招く．この状況を解消するために，領域分割も節点密度に合わせて適応的に設定する手法を提案する．具体的には，領域分割内の最大節点数をユーザ入力の条件として与え，個の条件を満たすまで再帰的に分割を繰り返すことで，適応的な領域分割を実現する．これにより，各サポートに含まれる節点数をほぼ一定に保つことができ，計算の効率化が図れる．図 1,2,3 にこの例を示す．図 1 は領域内に適応的に配置した節点，図 2 は一様な領域分割の結果，図 3 は適応的に配置した分割領域を表している．

(3) 前述の修正版 RPIM は並列計算にも適している．ここでは CUDA (NVIDIA 社) と呼ばれる GPU 上の並列プログラミング環境を想定し，アーキテクチャの特性を活かした並列計算アルゴリズムを構築する．前述の高速化アルゴリズムのうち，特に高速化が必要となるのは 3)-b の連立 1 次方程式の解を求める処理であり，これを並列に処理することで高速化を図る．並列化のための重要な性質として，各部分領域では 3)-b を独立に処理できるということが挙げられる．この性質を利用し，(4)の係数行列の生成処理を以下のように変形する．

$$
\begin{aligned}
a_{ij} &= \int_{\Omega} \nabla\phi_i(\boldsymbol{x}) \cdot \nabla\phi_j(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\
&= \sum_{l=1}^{L} \left( \int_{\Omega_l} \nabla\phi_i(\boldsymbol{x}) \cdot \nabla\phi_j(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \right) \\
&\approx \sum_{l=1}^{L} \left( \sum_{k=1}^{M} w_k \nabla\phi_i(\boldsymbol{x}_k^{(l)}) \cdot \nabla\phi_j(\boldsymbol{x}_k^{(l)}) \right).
\end{aligned}
\tag{5}
$$

これを行列の形で表せば，

$$
\begin{aligned}
&A = \sum_{k=1}^{M} A^{(k)}, \quad A^{(k)} = \{a_{ij}^{(k)}\}, \\
&a_{ij}^{(k)} \equiv \sum_{l=1}^{L} w_l \nabla\phi_i(\boldsymbol{x}_l^{(k)}) \cdot \nabla\phi_j(\boldsymbol{x}_l^{(k)})
\end{aligned}
\tag{6}
$$

となる．ここで，$M$ は分割領域内の積分点の数，$\boldsymbol{x}_1^{(l)}, \ldots, \boldsymbol{x}_M^{(l)}$ は分割領域 $\Omega^{(l)}$ 内の積分点，$w_1, \ldots, w_M$ は対応する重みを表している．この式より，以下の方針で階層的な並列化を施すこととする．

1) $L$ 個の行列 $A^{(1)}, \ldots, A^{(L)}$ を並列に求める．
2) 各行列 $A^{(l)}$ の計算には，$M$ 個の積分点での基底関数の勾配が必要であるこの $M$ 個の積分点における勾配を並列に計算する．

以上がここで適用する並列化手法である．な

お，この階層的な並列化は CUDA の環境にも適合している．図 4 に示すように，CUDA では全 プ ロ セ ッ サ が SM (Streaming Multiprocessor) と呼ばれるプロセッサの集合から構成され，さらに，各 SM は SP (Streaming Processor) と呼ばれるプロセッサの最小単位の集合から構成される．この設計に合わせて，以下の方針で並列化を行うことで，並列計算の効率的を図る．

1) $L$ 個の行列 $A^{(1)}, \mathrm{K}, A^{(L)}$ の計算処理を SM 単位で分配し，並列に求める．
2) 各 SM は $M$ 個の積分点での基底関数の勾配の計算処理を SP に分配し,計算する．

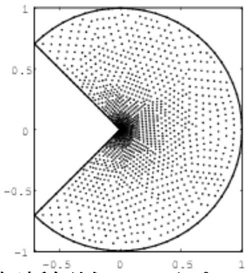

図 1．解析領域とアダプティブ節点

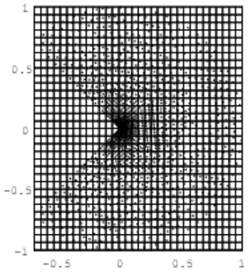

図 2．一様な領域分割

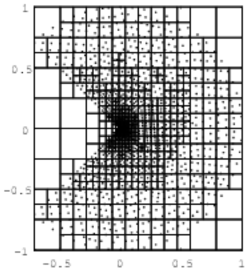

図 3．アダプティブ領域分割

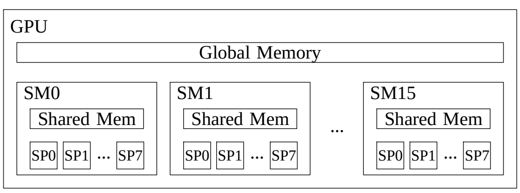


図 4．CUDA のプログラミングモデル例

## 4．研究成果

(1) 前述の RPIM の高速化アルゴリズムの効果を数値実験により確認する．ここでは図 5 に示す正方形領域上における(1)の偏微分方程式に対し，同じく図 5 に示す節点を用いて近似解を求め，RPIM と修正 RPIM の精度と計算量を比較する．図 6,7 はそれぞれ相対誤差と計算時間の比較を表す．なお，図中 MRPIM は修正 MRPIM を指し，横軸はサポートのサイズを決定するための係数である．一般にサポートサイズが大きいほど解の精度は高いが，その分多くの計算量を要する．この例ではパラメータによって 1/3 程度の高速化が実現された．3 次元の問題では効果が顕著であり，問題やパラメータによっては 1/7 程度の計算時間で解が得られた．詳細は発表論文を参照されたい．

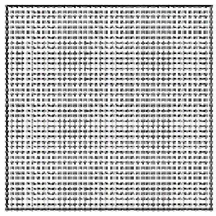

図 5．解析領域と節点分布

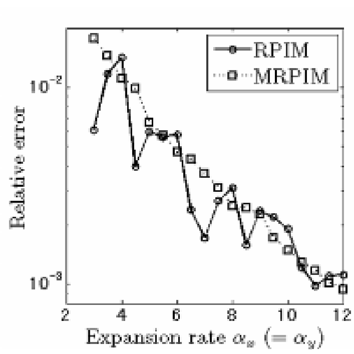


図 6．サポートの大きさと相対誤差

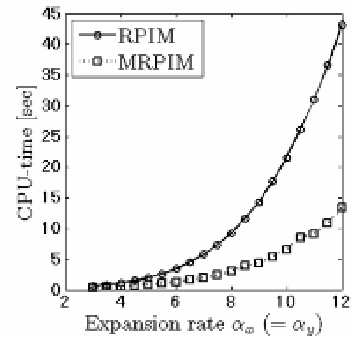


図 7．サポートの大きさと計算時間

図 8．解析対象の計測点群と陰関数曲面

図 9．曲面上と曲面内部の節点

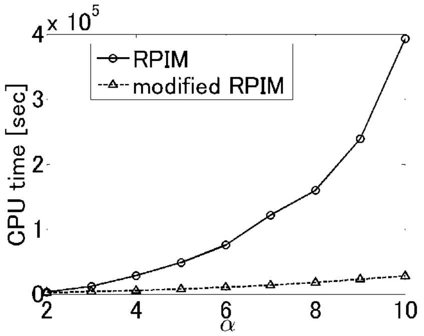


図 10．サポートの大きさと計算時間

(2) 陰関数曲面モデルとして表現した解析対象に対して，アダプティブな領域分割を利用した修正 RPIM を適用し，3 次元応力解析を行った例を示す．図 8 は解析対象の計測点群（約 6 万点）と，生成された陰関数曲面である．これに対し，図 9 に示すように節点を生成し（5823 節点），境界条件を設定して 3 次元ソリッドとしての応力解析を行った．このときの計算時間の比較を図 10 に示す．この例ではおよそ 5 倍から 10 倍の高速化が実現され，提案手法の有効性が確認された．

(3) ここでは図 1 に示す 2 次元の解析領域とアダプティブ節点を対象とし，本研究課題で提案する GPU 上での並列計算による高速化アルゴリズムを Poisson 方程式に適用した結果を示す．なお，ここで用いたメッシュレス法は全て修正 RPIM である．図 11 は本手法で得られた解の精度を，アダプティブ領域分割におけるサポート内の最大節点数に対してプロットしたものである．また，CPU と GPU における計算時間の比較を図 12 に示す．この例では GPU 上での並列計算を利用することで計算時間が 1/10 前後となり，提案する並列アルゴリズムの導入による高速化の効果が確認された．

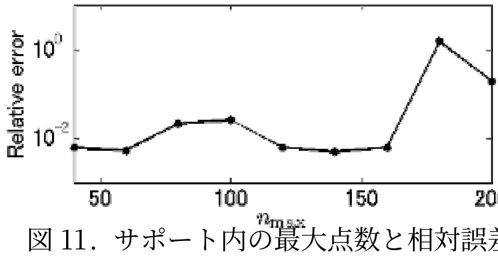


図 11．サポート内の最大点数と相対誤差

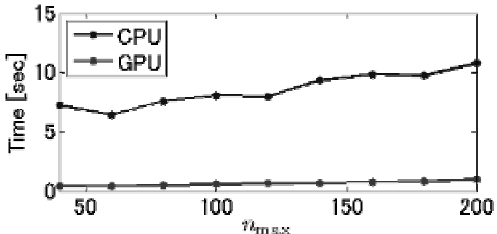


図 12．サポート内の最大点数と計算時間

５．主な発表論文等

〔雑誌論文〕（計 12 件）

[1] 木村彰徳，八田拓也，市村智和，仲田晋，田中覚，複雑な陰関数曲面モデルの確率過程的並列サンプリング，電子情報通信学会論文誌，J92-D(3)，439-442，2009．（査読有）

[2] S. Nakata, Acceleration of Meshfree Radial Point Interpolation Method on Graphics Hardware, AIP conference proceedings 1048, pp.396-399, 2008.（査読有）

[3] K. Hasegawa, S. Nakata, S. Tanaka, A Fast Solver for 3D Meshless Analysis Based on RPIM, Theoretical and Applied Mechanics Japan, 56, 439-444, 2008.（査読有）

[4] T. Ide, H. Isozaki, S. Nakata, S. Siltanen, G. Uhlmann, Probing for electrical inclusions with complex spherical waves, Communications on Pure and Applied Mathematics, 60(10), 1415-1442, 2007.（査読有）

[5] S. Nakata, Meshless Analysis with Non-Uniformly Distributed Nodes Using Hierarchical Cell Structure, AIP conference proceedings 936, 378-381, 2007.（査読有）

[6] 小嶋一行，岡将史，柴田章博，仲田晋，田中覚，陰関数曲面上における粒子拡散法を

用いた高密度・大量点群のポリゴン化，可視化情報学会論文集，27(9)，77-83，2007．（査読有）
[7] Y. Jo, M. Oka, A. Kimura, K. Hasegawa, A. Saitoh, S. Nakata, A. Shibata, S. Tanaka, Stochastic Visualization of Intersection Curves of Implicit Surfaces, Computers & Graphics, 31(2), 230-242, 2007. （査読有）
[8] K. Hasegawa, S. Nakata, S. Tanaka, Meshless Structural Analyses of Complex Shape Models Using Implicit Surface Representations, Journal of Plasma Physics, 72(6), 1081-1086, 2006. （査読有）
[9] K. Hasegawa, S. Nakata, S. Tanaka, Meshless Method for Structural Analysis Based on Surface Reconstruction, Theoretical and Applied Mechanics Japan, 55, 247-252, 2006. （査読有）
[10] S. Nakata, An Efficient Scheme for Meshless Analysis Based on Radial Basis Functions, ICNAAM 2006 proceedings, WILEY-VCH, 263-266, 2006. （査読有）
[11] 長谷川恭子，仲田晋，田中覚，曲面モデルに基づくメッシュレス解析のための節点生成，計算数理工学論文集， 6(1)，61-64，2006．（査読有）
[12] 藤本大地，伊東拓，仲田晋，北川高嗣，岡将史，田中覚，MPU 法に基づく色情報付き陰関数曲面の生成，電子情報通信学会論文誌，J89-D(6)，1391-1402，2006．(査読有)

〔学会発表〕（計 14 件）

[1] 長谷川恭子，仲田晋，田中覚，修正 RPIM に基づくメッシュフリー振動解析，第 21 回計算力学講演会，沖縄，November 1-3, 2008.
[2] 長谷川恭子，仲田晋，田中覚，振動解析への修正 RPIM の適用，第 27 回日本シミュレーション学会大会，滋賀，June 19-20, 2008.
[3] 長谷川恭子，仲田晋，田中覚，修正 RPIM の 3 次元振動解析への適用，第 57 回理論応用力学講演会，東京，June 10-12，2008.
[4] 長谷川恭子，仲田晋，田中覚，修正 RPIM の振動解析への適用，計算工学講演会，仙台，May 19-21, 2008.
[5] 長谷川恭子，仲田晋，田中覚，形状モデリングとメッシュレス構造解析，JSIAM-JSST 連合発表会，東京，March 8-9, 2008.
[6] 長谷川恭子，仲田晋，田中覚，修正 RPIM によるメッシュレス解析への反復法の適用，第 20 回計算力学講演会，京都，November 26-28, 2007.
[7] A. Saitoh, S. Tanaka, S. Nakata, A. Kamitani, Accuracy Improvement of BNM, COMPUMAG 2007, Aachen, Germany, June 24-28, 2007.
[8] 長谷川恭子，仲田晋，田中覚，形状計測装置を利用した曲面生成と修正 RPIM による構造解析，歴史都市防災シンポジウム '07, June 23, 2007.
[9] M. Oka, S. Nakata, S. Tanaka, Preprocessing for Accelerating Convergence of Repulsive-particle Systems for Sampling Implicit Surfaces, Shape Modeling International 2007, Lyon, France, June 13-15, 2007.
[10] 長谷川恭子，仲田晋，田中覚，修正 RPIM に基づくメッシュレス構造解析，計算工学講演会，東京，May 22-24, 2007.
[11] 長谷川恭子，仲田晋，田中覚，RPIM に基づく 3 次元メッシュレス解析とその高速化，第 56 回理論応用力学講演会，東京，March 7-9, 2007.
[12] 長谷川恭子，仲田晋，田中覚，RBF に基づくメッシュレス法を用いた 3 次元複雑形状の解析，第 19 回計算力学講演会，名古屋，November 3-5, 2006.
[13] 齋藤歩，田中覚，仲田晋，神谷淳，境界節点法の高精度化，第 16 回日本応用数理学会，茨城，September 16-18, 2006.
[14] 長谷川恭子，仲田晋，田中覚，形状計測装置に基づく曲面モデルの生成と Radial Point Interpolation Method による構造解析，計算工学講演会，大阪，June 12-14, 2006.

６．研究組織

(1)研究代表者

仲田　晋（SUSUMU NAKATA）
立命館大学・情報理工学部・准教授
研究者番号：00351320