

文部科学省次世代IT基盤構築のための研究開発
「革新的シミュレーションソフトウェアの研究開発」

RSS21 フリーソフトウェア

HEC ミドルウェア (HEC-MW)

PC クラスタ用ライブラリ型 HEC-MW

(hecmw-PC-cluster) バージョン 2.01

API リファレンス

本ソフトウェアは文部科学省次世代IT基盤構築のための研究開発「革新的シミュレーションソフトウェアの研究開発」プロジェクトによる成果物です。本ソフトウェアを無償でご使用になる場合「RSS21フリーソフトウェア使用許諾条件」をご了承頂くことが前提となります。営利目的の場合には別途契約の締結が必要です。これらの契約で明示されていない事項に関して、或いは、これらの契約が存在しない状況においては、本ソフトウェアは著作権法など、関係法令により、保護されています。

お問い合わせ先

(公開／契約窓口) (財)生産技術研究奨励会

〒153-8505 東京都目黒区駒場4-6-1

(ソフトウェア管理元) 東京大学生産技術研究所 計算科学技術連携研究センター

〒153-8505 東京都目黒区駒場4-6-1

Fax : 03-5452-6662

E-mail : software@rss21.iis.u-tokyo.ac.jp

目 次

1. I/O, 可視化	1
hecmw_init	2
hecmw_finalize	3
hecmw_get_mesh	4
hecmw_dist_free	5
hecmw_put_mesh	6
hecmw_result_init	7
hecmw_result_add	8
hecmw_result_write	9
hecmw_result_write_by_name	10
hecmw_result_write_st	11
hecmw_result_write_st_by_name	12
hecmw_result_read	14
hecmw_result_read_by_name	15
hecmw_restart_add_int	16
hecmw_restart_add_real	17
hecmw_restart_write	18
hecmw_restart_write_by_name	19
hecmw_restart_open	20
hecmw_restart_open_by_name	21
hecmw_restart_read_int	22
hecmw_restart_read_real	23
hecmw_restart_close	24
hecmw_visualize_init	25
hecmw_visualize(mesh, result, tstep, max_step, is_force)	26
hecmw_visualize_finalize	27
2. 線形ソルバ	29
hecmw_solve_11	30
hecmw_solve_22	31
hecmw_solve_33	32
hecmw_solve_direct	33
hecmw_solve_direct_parallel	34
hecmw_barrier	35

hecmw_allREDUCE_R	36
hecmw_allREDUCE_I	37
hecmw_bcast_R	38
hecmw_bcast_I.....	39
hecmw_bcast_C	40
hecmw_update_1_R	41
hecmw_update_2_R	42
hecmw_update_3_R	43
hecmw_update_m_R	44
hecmw_matvec_11	45
hecmw_matvec_22	46
hecmw_matvec_33	47
hecmw_innerProduct_I.....	48
hecmw_innerProduct_R	49
3. 有限要素演算機能	50
hecmw_mat_con	51
hecmw_mat_ass_elem	52
hecmw_mat_ass_equation	53
hecmw_mat_ass_bc	54
hecmw_Jacob_231	55
hecmw_Jacob_241	56
hecmw_Jacob_341	57
hecmw_Jacob_361	58
4. 適応格子機能	59
hecmw_adapt_init.....	60
hecmw_adapt_proc.....	61
hecmw_adapt_new_mesh.....	62
hecmw_adapt_edge_info.....	63
5. 動的負荷分散機能	64
hecmw_transfer_data_f2c.....	65
hecmw_adapt_dynamic_load_balancing.....	66
hecmw_transfer_data_c2f.....	67
6. 連成カップリング機能	68
hecmw_couple_get_mesh	69
hecmw_couple_init	70
hecmw_couple_finalize.....	72

hecmw_couple_startup	73
hecmw_couple_cleanup	74
hecmw_couple	75
hecmw_couple_is_member	76
hecmw_couple_is_unit_member	77
hecmw_couple_is_unit_member_u	78
hecmw_couple_is_root	79
hecmw_couple_is_unit_root	80
hecmw_couple_is_unit_root_u	81
hecmw_intercomm_get_size	82
hecmw_intracomm_get_size	83
hecmw_intracomm_get_size_u	84
hecmw_intercomm_get_rank	85
hecmw_intracomm_get_rank	86
hecmw_intracomm_get_rank_u	87
hecmw_intercomm_get_comm	88
hecmw_intracomm_get_comm	89
hecmw_intracomm_get_comm_u	90
hecmw_intercomm_get_group	91
hecmw_intracomm_get_group	92
hecmw_intracomm_get_group_u	93

hecmw_get_mesh

メッシュデータをファイルから読み込みます。

```
subroutine hecmw_get_mesh(name_ID, mesh)

character(len=HECMW_NAME_LEN) :: name_ID
type(hecmwST_local_mesh) :: mesh
```

引数

name_ID

!MESH または!MESH GROUP を特定する識別子

mesh

読み込まれたメッシュデータの格納先

説明

ファイルからメッシュデータを読み込みます。

この API は、全体制御ファイルから入力ファイルの情報を取得します。

読み込み可能なメッシュファイルの種類は以下のとおりです。

- HEC-MW 分散メッシュデータ
- HEC-MW 単一領域メッシュデータ
- GeoFEM メッシュデータ
- ABAQUS メッシュデータ

メッシュファイルの種類は全体制御ファイルで指定します。

読み込むメッシュファイルは、全体制御ファイルの!MESH で定義されており、かつ NAME が name_ID のものです。!MESH GROUP によってメッシュファイルがグループ化されている場合は、グループ内の全てのメッシュファイルを読み込みます。

読み込まれるメッシュタイプが分散メッシュデータの場合、実際に読み込むファイルのファイル名は、全体制御ファイルから取得したファイル名の末尾に「.<ランク番号>」を付加したものとなります。

hecmw_result_add

結果ファイルに出力するデータを指定します。

```
subroutine hecmw_result_add(node_or_elem, n_dof, label, data)

integer(kind=kint) :: node_or_elem
integer(kind=kint) :: n_dof
character(len=HECMW_NAME_LEN) :: label
real(kind=kreal) :: data
```

引数

node_or_elem

指定する値が節点値なのか要素値なのかを示す

1:節点 2:要素

n_dof

自由度数

label

ラベル

data

結果データ

説明

結果ファイルに出力するデータを指定します。

これは、複数回呼び出すことが可能です。この呼び出しによって指定されたデータの情報は、一旦 HEC-MW の内部に蓄えられます。蓄えられたデータは、hecmw_result_write または hecmw_result_write_by_name によって出力されます。

この呼び出し以前に、hecmw_result_init によって初期化が行われていなければなりません。

hecmw_result_write

結果データをファイルへ出力します。

```
subroutine hecmw_result_write()
```

引数

なし

説明

結果データをファイルへ出力します。

出力されるデータは、`hecmw_result_add` で指定されたデータです。

出力されるファイルは、全体制御ファイルの `!RESULT` で定義されており、`IO=OUT` かつ全体制御ファイル内で最初に定義されているファイルです。

出力されるファイルのファイル名は、全体制御ファイルから取得したファイル名の末尾に「.<ランク番号>.<tstep>」を付加したものとなります。

hecmw_visualize(mesh, result, timestep, max_step, is_force)

可視化を行います。

```
subroutine hecmw_visualize(mesh, result, timestep, max_step, is_force)

type(hecmwST_local_mesh) :: mesh
type(hecmwST_result_data) :: result
integer(kind=kint) :: timestep
integer(kind=kint) :: max_step
integer(kind=kint) :: is_force
```

引数

mesh

メッシュデータ

result

可視化用結果データ

tstep

タイムステップ

max_step

最大タイムステップ

is_force

最後のステップも強制的に描くかどうかを示す。

0:描かない 1:描く

説明

可視化を行います。

hecmw_visualize_init によって事前に初期化されている必要があります。

可視化に必要な結果データには result、メッシュには mesh が使用されます。

hecmw_solve_33

線形ソルバ（1 節点あたり 3 自由度）を呼び出します。

```
subroutine hecmw_solve_33(mesh, matrix)

type(hecmwST_local_mesh) :: mesh
type(hecmwST_matrix      ) :: matrix
```

引数

mesh

メッシュデータの格納先

matrix

マトリクスデータおよびソルバ制御情報の格納先

説明

- 線形ソルバ（1 節点あたり 3 自由度）を呼び出します。
- 係数マトリクスに関する情報，右辺ベクトル，ソルバ制御情報は全て「matrix」に格納されています。
- 「matrix」の内容については，「2.2」をご覧ください。

hecmw_solve_direct

直接法ソルバを呼び出します。

```
subroutine hecmw_solve_direct(mesh, matrix, ifmsg)

type(hecmwST_local_mesh) :: mesh
type(hecmwST_matrix      ) :: matrix
integer(kind=kint)      :: ifmsg
```

引数

mesh

メッシュデータの格納先

matrix

マトリクスデータおよびソルバ制御情報の格納先

ifmsg

エラー時のメッセージを出力するデバイス番号

説明

- 直接法を呼び出します。
- 係数マトリクスに関する情報、右辺ベクトル、ソルバ制御情報は全て「matrix」に格納されています。
- 「matrix」の内容については、「2.2」をご覧ください。

注

本バージョンでは並列実行時には機能いたしません。

hecmw_solve_direct_parallel

並列直接法ソルバを呼び出します。

```
subroutine hecmw_solve_direct_parallel(mesh, matrix, ifmsg)

type(hecmwST_local_mesh) :: mesh
type(hecmwST_matrix      ) :: matrix
integer(kind=kint)      :: ifmsg
```

引数

mesh

メッシュデータの格納先

matrix

マトリクスデータおよびソルバ制御情報の格納先

ifmsg

エラー時のメッセージを出力するデバイス番号

説明

- 並列直接法を呼び出します。
- 係数マトリクスに関する情報，右辺ベクトル，ソルバ制御情報は全て「matrix」に格納されています。
- 「matrix」の内容については，「2.2」をご覧ください。

hecmw_mat_con

係数行列の CRS テーブルを作ります。

```
subroutine hecmw_mat_con (mesh, matrix)

type(hecmwST_local_mesh) :: mesh
type(hecmwST_matrix)     :: matrix
```

引数

mesh

メッシュデータの格納先

matrix

マトリクスデータの格納先

説明

- 全体化された係数行列の圧縮格納用データテーブルを作ります。
- 生成されたテーブルデータは、`matrix` 構造体に格納されます。
- CRS フォーマットに関しては HEC ミドルウェアの線形ソルバを参照してください。
- 内部から「`hecmw_mat_con0`」および「`hecmw_mat_con1`」を呼び出しています。

hecmw_mat_ass_elem

要素行列を全体行列に足しこみます。

```
subroutine hecmw_mat_ass_elem (hecMAT, nn, nodLOCAL, matrix)

type(hecmwST_matrix)      :: hecMAT
integer(kind=kint) :: nn
integer(kind=kint) :: nodLOCAL(:)
real(kind=kreal) :: matrix(:, :)
```

引数

hecMAT

マトリクスデータ (CRS テーブル) の格納先

nn

要素の節点数

nodLOCAL

要素を構成する節点

matrix

要素行列

説明

- 要素行列を全体行列に足しこみます。
- 要素行列は、一辺が節点数×節点自由度数の正方行列 (2次元配列) として与えます。

hecmw_mat_ass_equation

全体行列に多点拘束の条件式をくみこみます。

```
subroutine hecmw_mat_ass_equation (hecMESH, hecMAT)

type(hecmwST_mesh)      :: hecMESH
type(hecmwST_matrix)    :: hecMAT
```

引数

hecMESH

メッシュデータの格納先

hecMAT

マトリクスデータ (CRS テーブル) の格納先

説明

- メッシュデータ内に定義されている多点拘束の条件式をくみこみます。
- 用いるペナルティ値は hecMAT%rarray(11) に予めセットしておきます。

hecmw_mat_ass_bc

全体行列に拘束条件をくみこみます。

```
subroutine hecmw_mat_ass_bc (hecMAT, inode, idof, RHS)

type(hecmwST_matrix)      :: hecMAT
integer(kind=kint) :: inode
integer(kind=kint) :: idof
real(kind=kreal) :: RHS
```

引数

hecMAT

マトリクスデータ (CRS テーブル) の格納先

inode

拘束される節点

idof

拘束される自由度

RHS

拘束する値

説明

- 指定した節点の指定した自由度を指定した値に拘束します。