

cosgene モジュール仕様書

2018 年 2 月 5 日

Copyright (C) 2006-2018 Next Generation Natural Product Chemistry (N²PC)

 目次

1. Adjust_Velocity.....	16
1.1. Control_Velocity : 系全体の運動制御.....	16
2. Analyze_Method.....	17
2.1. Analyze_Force : 原子間の力の解析.....	17
2.2. Analyze_ElectroPotential : 静電ポテンシャルエネルギーの解析.....	17
2.3. Analyze_Violation : van der Waals 接触の解析.....	17
2.4. Analyze_MinimizeEnergy : エネルギー最小化フェーズの結果解析.....	18
2.5. Analyze_MDEnergy : MD のエネルギートラジェクトリー解析.....	18
2.6. Analyze_DistanceTrajectory : 距離のトラジェクトリー解析.....	18
2.7. Analyze_DihedralTrajectory : 二面角のトラジェクトリー解析.....	19
2.8. Analyze_Fluctuation : 原子位置の揺らぎ解析.....	19
2.9. Analyze_Combine : Combine 法エネルギー解析.....	19
3. Calc_Angle.....	20
3.1. Calc_AngleEnergy : angle のエネルギー計算.....	20
3.2. Calc_ClusterAngleEnergy : angle のクラスタ間エネルギー計算.....	20
4. Calc_Bond.....	21
4.1. Calc_BondEnergy : bond のエネルギー計算.....	21
4.2. Calc_BondEnergy : bond のクラスタ間エネルギー計算.....	21
5. Boundary_Function.....	22
5.1. Adjust_EllipsoidBoundary : 楕円体セルの補正.....	22
6. Boundary_Method.....	23
6.1. Adjust_Boundary : 距離ベクトルの補正.....	23
6.2. Replica_Index : レプリカセルのインデックス取得.....	23
6.3. Adjust_BoundaryWithReplica : 1-5 相互作用テーブルでの距離ベクトル補正.....	23
6.4. Compress_ReplicaIndex : レプリカセルインデックスの一次元化.....	23
6.5. Create_ReplicaCellCoordinate : レプリカセルの座標設定.....	24
6.6. Create_BoundaryStatus : セル行列、セル行列の逆行列および行列式の計算.....	24
6.7. Calc_LengthAndAngle : セルの各辺の長さおよび角度の計算.....	24
7. Calc_Cap.....	25
7.1. Calc_CapEnergy : CAP エネルギー計算.....	25
7.2. Calc_ClusterCapEnergy : CAP のクラスタ間エネルギー計算.....	25
8. Check_Class.....	26
8.1. Check_AtomDistance : 原子間の距離チェック.....	26
8.2. Check_LinearTorsion : 二面角構成原子の直線チェック.....	26
8.3. Check_ChainInEllipsoid : 楕円体セルの境界チェック.....	26
8.4. Check_Boundary : 周期境界条件のチェック.....	27
8.5. Check_ReplicaPosition : 相互作用テーブルでの周期境界条件チェック.....	27
8.6. Check_AtomBelongCluster : 所属クラスタのチェック.....	27
8.7. Check_OptionCombination : オプションの組み合わせのチェック.....	27
9. Analyze_Control.....	28
9.1. Analyze_Main : ANALYZE 主制御.....	28
10. MD_Control.....	29
10.1. MD_Main : MD 主制御.....	29
10.2. Init_MD : MD 初期処理.....	29
10.3. Exec_MD : MD の実行.....	29
10.4. Final_MD : MD 終了処理.....	30
11. Minimize_Control.....	31
11.1. Minimize_Main : MINIMIZE 主制御.....	31
11.2. Init_Minimize : MINIMIZE 初期処理.....	31
11.3. Exec_Minimize : MINIMIZE の実行.....	31
11.4. Final_Minimize : MINIMIZE 終了処理.....	32

11.5. Write_Histogram : 力のヒストグラム出力.....	32
12. Calc_Dihedral.....	33
12.1. Calc_DihedralEnergy : 二面角拘束のエネルギー計算.....	33
12.2. Calc_DihedralEnergy : 二面角拘束のクラスター間エネルギー計算.....	33
13. Calc_Distance.....	34
13.1. Calc_DistanceEnergy : 距離拘束のエネルギー計算.....	34
13.2. Calc_DistanceEnergy : 距離拘束のクラスター間エネルギー計算.....	35
14. Exception_Routine.....	36
14.1. Stop_Of : 処理の中断.....	36
14.2. Error_Of : エラー出力と処理の中断.....	36
14.3. Warning_Of : ワーニングの出力.....	36
15. File_Class.....	37
15.1. File_Open : ファイルオープン.....	37
15.2. Check_Keyword : 入出力パラメータのチェック.....	37
15.3. File_Close : ファイルクローズ.....	37
16. Input_Data.....	38
16.1. Input_TopologyFile : ASCII 形式のトポロジーファイルの読み込み.....	38
16.2. Update_MoleculeInformation : 分子情報の更新.....	38
16.3. Copy_MoleculeAsChains : 分子情報のコピー.....	38
16.4. Calc_PotentialParameter : ポテンシャル関数の設定.....	39
16.5. Set_ResidueInformation : 残基情報の設定.....	39
16.6. Init_Topology : トポロジーデータの初期化.....	39
16.7. Input_BinaryTopologyFile : バイナリ形式のトポロジーファイルの読み込み.....	40
16.8. Set_AdjustPhase : torsion 角の補正.....	40
16.9. Input_PDBFile : PDB ファイルの読み込み.....	41
16.10. Input_BinaryCoordinateFile : バイナリ形式の原子座標の読み込み.....	41
16.11. Input_VariableFile : 固定原子情報の読み込み.....	41
16.12. Input_ShakeFile : SHAKE ファイルの読み込み.....	42
16.13. Input_CapFile : CAP 情報の読み込み.....	42
16.14. Input_PositionFile : 位置拘束ファイルの読み込み.....	42
16.15. Input_BestfitFile : BESTFIT ファイルの読み込み.....	43
16.16. Input_DistanceFile : 距離拘束ファイルの読み込み.....	43
16.17. Input_DihedralFile : 二面角拘束ファイルの読み込み.....	44
16.18. Input_MonitorFile : モニター対象ファイルの読み込み.....	44
16.19. Input_MultiCanonicalFile : マルチカノニカルパラメータの読み込み.....	45
16.20. Search_Molecule : 対象分子 ID の検索.....	45
16.21. Input_RestartFile : リスタートファイルの読み込み.....	46
16.22. Input_EnergyFile : エネルギーファイルの読み込み.....	47
16.23. Input_EnergyAndAtomFile : エネルギー、原子位置データの読み込み.....	48
16.24. Get_HeadString : 先頭文字の取得.....	48
16.25. Input_HybridFile : Hybrid ファイル読み込み制御.....	49
16.26. Input_BinaryHybridFile : バイナリ形式の Hybrid モデルのファイル入力.....	49
16.27. Initialize_HybridModel : Hybrid モデル初期設定.....	49
16.28. Input_RigidFile : 剛体モデル入力.....	50
16.29. Init_RigidModel : 剛体モデル初期設定.....	50
16.30. Remove_InteractionInRigid : 剛体内の結合相互作用削除.....	51
16.31. Input_FlowForceFile : FlowForce ファイルの入力.....	51
16.32. Input_UmbrellaFile : Filled-Potential ファイルの入力.....	51
16.33. Remove_DummyBond : 計算しない BOND の削除.....	51
16.34. Remove_DummyAngle : 計算しない ANGLE の削除.....	52
16.35. Remove_DummyTorsion : 計算しない TORSION の削除.....	52
16.36. Remove_DummyImproper : 計算しない IMPROPER の削除.....	52
16.37. Input_ExtCapFile : 拡張 CAP 拘束データの読み込み.....	53

16.38.	Input_RangeFile : 範囲付き位置拘束データの読み込み	53
16.39.	Input_GeneralizedBornFile : GB/ASA データの読み込み	53
16.40.	Input_SatFile : SAT データの読み込み	53
16.41.	Input_RdcFile : RDC データの読み込み	53
16.42.	Input_Noefile : NOE データの読み込み	54
16.43.	Input_TplScalingFile : スケーリングデータの読み込み	54
16.44.	Init_FunkKind : 関数種別行列の初期化	54
16.45.	Set_AutoShake : SHAKE グループの自動解析	54
16.46.	Set_AutoRigid : RIGID の自動解析	55
16.47.	Input_MVOfFile : MVO データの読み込み	55
17.	Input_Common.....	56
17.1.	Input_CommonFile : cosgene 用共通ファイルの読み込み	56
17.2.	Calc_InteractNumber : 相互作用数の計算.....	56
18.	Input_Option.....	57
18.1.	Decide_Phase : 当該フェーズの決定	57
18.2.	Input_InputOption : INPUT フェーズのオプション入力	57
18.3.	Input_OutputOption : OUTPUT フェーズのオプション入力	57
18.4.	Input_MinimizeOption : MINIMIZE フェーズのオプション入力	58
18.5.	Input_MDOption : MD フェーズのオプション入力.....	58
18.6.	Input_AnalyzeOption : ANALYZE フェーズのオプション入力.....	59
18.7.	Read_LineAndSetValue : 行の読み込みとキーワード値設定.....	60
18.8.	Get_OptionKind : 行種別設定	60
18.9.	Get_LeftSideAssignment : 左辺文字列の取得	60
18.10.	Get_RightSideAssignment : 右辺値文字列の取得	61
18.11.	Is_LineEnd : 行終了チェック	61
18.12.	Error_Line : 読み込み行のエラー出力	61
18.13.	Get_Value_I4 : 整数型キーワード値の取得	61
18.14.	Get_Value_C : 文字型キーワード値の取得	62
18.15.	Get_Value_R8 : 実数型キーワード値の取得	62
18.16.	Get_Value_Select : 選択型キーワード値の取得	62
18.17.	Init_Value_I4 : 整数型キーワードの初期設定.....	62
18.18.	Init_Value_C : 文字型キーワード値の初期設定	63
18.19.	Init_Value_R8 : 実数型キーワード値の初期設定	63
18.20.	Init_Value_Select_Char : 選択型キーワード値の初期設定	63
18.21.	Init_Value_Select_Val : 選択型キーワード値の初期設定.....	63
18.22.	Init_Value_Select_Logical : 選択型キーワード値の初期設定	64
18.23.	Set_Value_I4 : 整数型キーワードの設定	64
18.24.	Set_Value_C : 文字型キーワードの設定	64
18.25.	Set_Value_R8 : 実数型キーワードの設定	65
18.26.	Set_Value_Select : 選択型キーワードの設定	65
18.27.	Set_Value_Select_Logical : 選択型キーワードの設定	65
18.28.	Chekc_Option : キーワード重複チェック	65
18.29.	Check_SelectOption : 選択型キーワードのチェック	66
18.30.	Check_DuplicateOption : 重複オプションのチェック	66
19.	Interact_Method.....	66
19.1.	Get_InteractType	66
20.	InteractTable_Method	67
20.1.	Get_InteractPairList : 1-5 相互作用リスト作成 (van der Waals + hydrogen bond)	67
20.2.	Get_VdwInteractList : 1-5 相互作用リスト作成 (van der Waals のみ)	67
20.3.	Get_HydInteractList : 1-5 相互作用リスト作成 (hydrogen bond のみ)	67
20.4.	Get_EleInteractList : 1-5 静電相互作用リスト作成	68
20.5.	Update_InteractTable : 相互作用テーブルの更新	68
20.6.	Create_AmberInteractPairList : Amber 対応相互作用リスト作成.....	69

20.7. Create_OplsInteractList : OPLS 対応相互作用リスト作成	69
20.8. Update_AtomBaseCutoff : 原子距離カットオフでの相互作用テーブル作成	70
20.9. Update_AtomBaseCutoff : 残基中心距離カットオフでの相互作用テーブル作成	70
20.10. Register_ResidueAtoms : 残基内原子の相互作用テーブル登録	70
20.11. Update_ResidueSurfaceCutoff : 残基表面距離カットオフでの相互作用テーブル作成	71
20.12. Register_NearSurfaceAtoms : 残基内原子の相互作用テーブル登録	71
20.13. Reset_FlagFixAtomOrBonded : 相互作用計算候補フラグの設定	72
20.14. Is_InteractionUpdate : 相互作用テーブル更新の必要性チェック	72
20.15. Set_UnitedInteraction : 一様相互作用の設定	72
20.16. Register_Atoms : atom ベースカットオフの相互作用リスト作成	73
20.17. Register_AtomsPeriodic : 周期系での atom ベースカットオフの相互作用リスト作成	74
20.18. Create_PairListWithCellPeriodic : 周期系の空間分割法での相互作用リスト作成	75
20.19. Update_CellIndexPeriodic : 周期系の残基の空間への登録	75
20.20. Create_PairListWithCell : 孤立系の空間分割法での相互作用リスト作成	76
20.21. Update_CellIndex : 孤立系の残基の空間への登録	76
21. Interact_Control	77
21.1. Interact_Control : 相互作用計算主制御	77
22. cosgene	78
22.1. Para_MD : メインプログラム	78
22.2. MD_MainProcess : MD メインプロセス	78
23. Math_Library	79
23.1. Innter_Product : 3 次元ベクトルの内積計算	79
23.2. Add_InnterProduct : 3 次元ベクトルの内積計算と加算	79
23.3. Cross_Product : 3 次元ベクトルの外積計算	79
23.4. ADD_Square_ValueStrict : 3 次元ベクトルの 2 乗の加算	79
23.5. Square_ValueStrict : 3 次元ベクトルの 2 乗の計算	80
23.6. Square_Value : 3 次元ベクトルの 2 乗の計算	80
23.7. Standard_Deviation : 標準偏差と平均値の計算	80
23.8. Determinant_3by3 : 3 次元行列の行列式計算	80
23.9. MatrixProduct_3by3 : 3 次元行列の積の計算	80
23.10. InvertMatrix_3by3 : 3 次元行列の逆行列の計算	81
23.11. Calc_LinearEquation : 連立方程式の解法	81
23.12. Calc_EigenValueProblem : 固有値の計算	81
23.13. Create_RandomNumber : 乱数の生成	82
23.14. Random_Sequence : 乱数列の生成	82
23.15. Exec_QuickSort : クイックソートの実行	82
23.16. Sort_PartialData : 部分データのクイックソート	82
23.17. Exec_SequentialSort : 部分データの逐次ソート	83
24. Minimize_Method	84
24.1. Exec_SteepestDecent : 最急勾配法でのエネルギー最小化	84
24.2. Exec_ConjugateGradient : 共益勾配法でのエネルギー最小化	84
25. Monitoring	85
25.1. Display_Monitor : MD のトラジェクトリー、ログ出力	85
25.2. Monitor_Trajectory : トラジェクトリーの出力	85
25.3. Display_Log : MD のログ情報出力	85
25.4. Open_MonitorFile : モニター用ファイルのオープン	86
25.5. Open_MonitorFile : モニター用ファイルのクローズ	86
25.6. Is_OutputStep : ファイル出力ステップの判定	86
25.7. Display_InputOption : INPUT フェーズ情報の出力	86
25.8. Display_MinimizeOption : MINIMIZE フェーズ情報の出力	87
25.9. Display_MDOption : MD フェーズ情報の出力	87
25.10. Display_OutputOption : OUTPUT フェーズ情報の出力	87
25.11. Display_AnalyzeOption : ANALYZE フェーズ情報の出力	87

25. 12. Displat_MonitorFile : モニター用ファイル情報の出力	88
25. 13. Displat_InOutFile : 入出力ファイル情報の出力	88
25. 14. Monitor_Minimize : MINIMIZE のログ出力	88
25. 15. Display_PotentialEnergy : ポテンシャルエネルギーの出力	88
25. 16. Output_TsallisData : Tsallis 用データの出力	89
26. Fast_Nonbonded	90
26. 1. Calc_Cutoff15InteractEnergy : カットオフでの 1-5 相互作用計算制御	90
26. 2. Calc_VdwCutoff : カットオフでの van der Waals 計算	91
26. 3. Calc_HydCutoff : カットオフでの水素結合計算	91
26. 4. Calc_ConstEleCutoff : カットオフでの距離非依存静電計算	91
26. 5. Calc_DependEleCutoff : カットオフでの距離依存静電計算	92
26. 6. Calc_VdwConstEleCutoff : カットオフでの van der Waals と距離非依存静電計算	92
26. 7. Calc_VdwDependEleCutoff : カットオフでの van der Waals と距離依存静電計算	92
26. 8. Calc_VdwHydCutoff : カットオフでの van der Waals と水素結合計算	93
26. 9. Calc_HydConstEleCutoff : カットオフでの水素結合と距離非依存静電計算	93
26. 10. Calc_HydDependEleCutoff : カットオフでの水素結合と距離依存静電計算	93
26. 11. Calc_VdwHydConstEleCutoff : カットオフでの van der Waals, 水素結合, 距離非依存静電計算	94
26. 12. Calc_VdwHydDependEleCutoff : カットオフでの van der Waals, 水素結合, 距離依存静電計算	94
26. 13. Calc_NoCut15InteractEnergy : カットオフなしの 1-5 相互作用計算	95
26. 14. Calc_VdwNocut : カットオフなしの van der Waals 計算	96
26. 15. Calc_HydNocut : カットオフなしの水素結合計算	96
26. 16. Calc_ConstEleNocut : カットオフなしの距離非依存静電計算	96
26. 17. Calc_DependEleNocut : カットオフなしの距離依存静電計算	97
26. 18. Calc_VdwConstEleNocut : カットオフなしの van der Waals と距離非依存静電計算	97
26. 19. Calc_VdwDependEleNocut : カットオフなしの van der Waals と距離依存静電計算	97
26. 20. Calc_VdwHydNocut : カットオフなしの van der Waals と水素結合計算	97
26. 21. Calc_HydConstEleNocut : カットオフなしの水素結合と距離非依存静電計算	98
26. 22. Calc_HydDependEleNocut : カットオフなしの水素結合と距離依存静電計算	98
26. 23. Calc_VdwHydConstEleNocut : カットオフなし van der Waals, 水素結合, 距離非依存静電計算	98
26. 24. Calc_VdwHydDependEleNocut : カットオフなしの van der Waals, 水素結合, 距離依存静電計算	98
27. Object_Method	99
27. 1. Minimum_ResidueDistance : 残基間の最短距離計算	99
27. 2. Get_Coordinate : 原子位置の取得	99
27. 3. Set_Coordinate : 原子位置の設定	99
27. 4. Get_Gradient : 原子への力の取得	99
27. 5. Set_Gradient : 原子への力の設定	100
27. 6. Get_Mass : 原子質量の取得	100
27. 7. Get_ReferenceCoordinate : 参照位置の取得	100
27. 8. Get_FreeAtomVelocity : 自由原子の速度取得	100
27. 9. Set_FreeAtomVelocity : 自由原子の速度設定	100
27. 10. Get_Cluster : クラスタ ID の取得	101
27. 11. Get_ResidueCenter : 残基重心の取得	101
27. 12. Centering_Translation : 系のセンタリング処理	101
27. 13. Get_ChainCenter : 分子の重心取得	101
27. 14. Get_ReferenceChainCenter : 参照位置での分子の重心取得	102
27. 15. Find_MaxForceAndAtom : 力の最大値の取得	102
27. 16. Calc_TotalFreeAtomForce : 自由原子への力の合計	102
27. 17. Calc_RMSF : RMSF 計算	102
27. 18. Calc_LeastSquareFitting : 非線形最小二乗法での揺らぎ計算	103
27. 19. Update_CoordinateWithBoundary : 周期境界条件での原子位置補正	103

27.20. Calc_Density : 系の密度計算.....	103
27.21. Get_Distance : 2点間の距離計算.....	104
27.22. Get_Angle : 3点間の角度計算.....	104
27.23. Get_Torsion : 4点間の角度計算.....	104
27.24. Calc_ParticleKinetic : 原子(剛体以外)の運動エネルギー計算.....	104
27.25. Calc_ParticleKineticTensor : 原子(剛体以外)の運動エネルギーテンソル計算.....	105
27.26. Calc_RigidTransKinetic : 剛体の並進運動エネルギー計算.....	105
27.27. Calc_RigidTransKineticTensor : 剛体の並進運動エネルギーテンソル計算.....	105
27.28. Calc_RigidRotateKinetic : 剛体の回転運動エネルギー計算.....	105
27.29. Calc_RigidRotateKineticTensor : 剛体の回転運動エネルギーテンソル計算.....	105
27.30. Get_MultiChainCenter : 分子の重心取得(複数チェーン).....	106
27.31. Get_MultiRefChainCenter : 参照位置での分子の重心取得(複数チェーン).....	106
28. Output_Data	107
28.1. Output_TopologyFile : ASCII形式のトポロジーファイル出力.....	107
28.2. Output_BinaryTopologyFile : バイナリ形式のトポロジーファイル出力.....	107
28.3. Output_PDBFile : PDBファイルの出力.....	107
28.4. Output_BinaryCoordinateFile : バイナリー形式の原子位置出力.....	107
28.5. Output_RestartFile : リスタートファイルの出力.....	108
28.6. Output_EnergytFile : エネルギーファイルの出力.....	108
28.7. Output_EnergytForAnalysis : MINIMIZEのエネルギーファイルの出力.....	108
28.8. Output_FPAverageFile : FP平均値ファイルの出力.....	108
28.8. Open_FileWithForm : 各データ形式でのファイルのオープン.....	109
28.9. Scale_RadiusAndCharge : vdWパラメータおよび電荷のスケーリング.....	109
29. Output_MD	110
29.1. Output_MDfile : MD終了結果のファイル出力.....	110
30. Calc_Position.....	111
30.1. Calc_PositionEnergy : 位置拘束のエネルギー計算.....	111
30.2. Calc_PositionEnergy : 位置拘束のクラスタ間エネルギー計算.....	111
31. Calc_Repulsion.....	112
31.1. Calc_RepulsionEnergy : repulsion エネルギー計算.....	112
31.2. Calc_15RepulsionEnergy : 1-5 相互作用原子間の repulsion エネルギー計算.....	112
31.3. Calc_ClusterRepulsionEnergy : クラスタ間 repulsion エネルギー計算.....	113
31.4. Calc_Cluster15Repulsion : 1-5 相互作用原子間のクラスタ間 repulsion エネルギー計算.....	113
32. Restraine_Control	114
32.1. Restraine_Main : 拘束エネルギー計算主制御.....	114
33. Shake_Method.....	115
33.1. Exec_Shake : SHAKE 計算主制御.....	115
33.2. Calc_TwoAtomsHarmonicShake : 収束ループでの2原子間 SHAKE.....	115
33.3. Calc_TwoAtomsBiquadraticShake : 2次方程式での2原子間 SHAKE.....	115
33.4. Calc_ThreeAtomsShake : 3原子間 SHAKE の実行.....	116
33.5. Calc_FourAtomsShake : 4原子間 SHAKE の実行.....	116
34. String	117
34.1. I4_To_String : 整数の文字列変換.....	117
34.2. R4_To_String : 4バイト実数の文字列変換.....	117
34.3. R8_To_String : 8バイト実数の文字列変換.....	117
34.4. R16_To_String : 16バイト実数の文字列変換.....	117
34.5. L4_To_String : 論理型の文字列変換.....	118
34.6. String_ToReal : 文字列の実数変換.....	118
34.7. String_ToInteger : 文字列の整数変換.....	118
34.8. Read_RealValue : 実数の読み込みと当該文字列の削除.....	119
34.9. Read_IntValue : 整数の読み込みと当該文字列の削除.....	119
34.10. Read_String : 文字列の読み込みと当該文字列の削除.....	119
34.11. Change_CapitalWord : 小文字の大文字変換.....	120

34. 12. IsEqual : 文字列比較	120
35. System_Class	121
35. 1. System_Clock : 時間の取得	121
35. 2. Write_Time : 時間の出力	121
35. 3. Inform_Job : ジョブ情報取得	121
36. Temperature_Method	122
36. 1. Calc_KineticAndTemperature : 系の運動エネルギーと温度の取得	122
36. 2. Calc_TemperatureOfMolecules : 各分子の温度の取得	122
36. 3. Set_TemperatureOfMolecules : 各分子の温度の登録	122
36. 4. Calc_KineticOfMolecules : 各分子の運動エネルギーを計算	122
36. 5. Calc_RigidKineitc : 剛体の運動エネルギー計算	123
36. 6. Calc_RigidKineticTensor : 剛体の運動エネルギー行列計算	123
36. 7. Calc_RigidKineticOfMolecules : 剛体の各分子の運動エネルギー計算	123
37. Calc_Torsion	124
37. 1. Calc_TorsionEnergy : torsion のエネルギー計算	124
37. 2. Calc_ClusterTorsionEnergy : torsion のクラスタ間エネルギー計算	124
38. Calc_Torsional	125
38. 1. Calc_TorsionVdwEnergy : 1-4 の van der Waals エネルギー計算	125
38. 2. Calc_TorsionEleEnergy : 1-4 の静電エネルギー計算	125
38. 3. Calc_ClusterTorsionVdwEnergy : 1-4 のクラスタ間 van der Waals エネルギー計算	126
38. 4. Calc_ClusterTorsionEleEnergy : 1-4 のクラスタ間静電エネルギー計算	126
39. Verlet_Method	127
39. 1. Exec_NVELeapFlogVerlet : NVE アンサンブルでの蛙とびベルレ法	127
39. 2. Exec_NVEVelocityVerlet : NVE アンサンブルでの速度ベルレ法	127
39. 3. Exec_NVTLeapFlogVerlet : NVT アンサンブルでの蛙とびベルレ法	128
39. 4. Exec_NVTVelocityVerlet : NVT アンサンブルでの速度ベルレ法	128
39. 5. Scale_GradientForMultiCanonical : マルチカノニカルでの力のスケーリング	128
39. 6. Exec_NPTVelocityVerlet : NPT アンサンブル計算	129
40. Ewald_Method	130
40. 1. Init_Ewald : Ewald 法初期設定	130
40. 2. Calc_DirectEnergy : 実空間の静電エネルギーと力の計算	130
40. 3. Calc_ReciprocalEnergy : 逆空間の静電エネルギーと力の計算	131
40. 4. Calc_SelfEnergy : Ewald の定数項の計算	131
40. 5. Calc_ExcessEnergy : 基本セル内の結合原子間の静電エネルギー計算	132
40. 6. Calc_WaveVector : 周期関数ベクトル計算	132
40. 7. Init_PME : PME 法初期設定	133
40. 8. Calc_PMEForce : PME 法での逆空間エネルギー計算	134
40. 9. Load_BsplineModuli : B スプラインの設定主制御	134
40. 10. Calc_DFTModuli : 各次元の B スプラインの設定	135
40. 11. Get_BSplineCoeffs : B スプライン係数の計算	135
40. 12. Fill_Bspline3Dim : 各原子の B スプライン係数の計算	135
40. 13. Fill_Bspline : B スプライン係数の計算	135
40. 14. Get_ScaledFractionals : 座標断片の計算	136
40. 15. Fill_ChargeGrid : グリッドの電荷設定	136
40. 16. Sum_Energy : PME 法での逆空間静電エネルギー計算	137
40. 17. Sum_Gradient : PME 法での逆空間での原子への力の計算	137
40. 18. Initial_FFT : FFT の初期化	138
40. 19. FFT_Forward : フォワード FFT	138
40. 20. FFT_Backward : バックワード FFT	138
41. Hybrid_Method	139
41. 1. Set_HybridFlag : Hybrid MD/HD 法のフラグ設定	139
41. 2. Allocate_HybridData : Hybrid 法の動的メモリの確保	139
41. 3. Set_HybridParameter : Hybrid MD/HD 法のパラメータ設定	139

41.4.	Init_HybridVelocity : Hybrid MD/HD 法の初期化	140
41.5.	Scale_HybridVelocity : HD 部分の速度スケーリング	140
41.6.	Set_HybridRestartVelocity : リスタート時の HD 部分の速度設定	140
41.7.	Init_HybridVelocity : Hybrid MD/HD 法の初期化	140
41.8.	Calc_HybridEnergy : HD 部分のエネルギー計算	141
41.9.	Calc_HybridVelocity : HD 部分の速度計算	141
41.10.	Calc_HybridCoordinate : HD 部分の座標計算	141
41.11.	Backup_HybridQuantities : HD 部分の物理量のバックアップ	141
41.12.	Rollback_HybridVelocity : HD 部分の速度のロールバック	142
41.13.	Add_HybridMoment : 回転モーメントへの HD 部分の加算	142
41.14.	Remove_HybridTotalVelocity : HD 部分の系全体の回転、並進の削除	142
41.15.	Get_HybridTrans : HD 部分の並進速度計算	142
41.16.	Get_HybridRotate : HD 部分の回転速度計算	143
41.17.	Calc_HybridKinetic : HD 部分の運動エネルギー計算	143
42.	Barostat_Method	144
42.1.	Exec_AndersenMethod : Andersen の圧力制御	144
42.2.	Exec_ParrinelloRahmanMethod : Parrinello-Rahman の圧力制御	144
42.3.	Exec_BerendsenBarostat : Berendsen の圧力制御	145
42.4.	Exec_BerendsenThermostat : Berendsen の温度制御	145
43.	Control_Dynamics	146
43.1.	Exec_Dynamics : MD の 1 ステップ実行	146
43.2.	Backup_PreviousQuantities : 物理量のバックアップ	146
44.	Thermostat_Method	147
44.1.	Scale_Velocity : 温度制御の主制御	147
44.2.	Exec_BerendsenThermostat : Berendsen の温度制御	147
44.3.	Exec_NoseHooverThermostat : 能勢-Hoover の温度制御	147
44.4.	Exec_EachMoleculeNoseHoover : 分子単位の能勢-Hoover の温度制御	148
44.5.	Exec_ShakeConvergeLoop : SHAKE ループ実行	148
44.6.	Exec_HooverEvansThermostat : Hoover-Evans の温度スケーリング	149
44.7.	Scale_VelocityWithShakeGradient : SHAKE 向け温度スケーリング	149
44.8.	Is_ConvergeTemperature : 収束温度判定	149
45.	Rigid_Method	150
45.1.	Calc_RigidForce : 剛体の力の計算 (並進+回転)	150
45.2.	Calc_RigidVelocity : 剛体の速度計算	150
45.3.	Calc_RigidAngularMomentum : 剛体の角運動量補正	150
45.4.	Calc_RigidCoordinate : 剛体の座標計算	151
45.5.	Calc_TorqueAndForce : 剛体への力の分離 (並進+回転)	151
45.6.	Convert_Torque : トルクの補正	151
45.7.	Calc_AngularPhase : 剛体角の計算	151
45.8.	Calc_AngularMomentum : 剛体の角運動量補正	152
45.9.	Calc_EllipticFunction : ヤコビの楕円関数計算	152
45.10.	Calc_RigidVirialTensor : 剛体のビリアル計算	152
46.	FMM_Method	153
46.1.	System_Size : 系の大きさの計算	153
46.2.	Create_CellGeometry : セルの中心座標計算	153
46.3.	Downward_Local : 局所展開係数計算の制御	153
46.4.	Calc_LocalToLocal : 局所展開係数計算	154
46.5.	Upward_Multipole : 多重極展開計算の制御	154
46.6.	Calc_PolynomialToMultipole : 多重極モーメント計算	154
46.7.	Calc_MultipoleToMultipole : 多重極モーメントの展開	155
46.8.	Update_FMMCell : FMM のセル情報の更新	155
46.9.	Calc_LegendrePolynomial : Legendre 陪多項式計算	155
46.10.	Calc_MultipoleToLocal : 多重極局所展開	156

46.11.	Calc_DirectForceInCell : 同一セル間相互作用直接計算	156
46.12.	Calc_DirectForceNeighborCell : 隣接セル間相互作用直接計算	156
46.13.	Calc_LocalFoece : 非隣接セルからの相互作用計算	157
46.14.	Calc_FMMForce : FMM での静電相互作用計算の主制御	157
46.15.	Calc_InteractCellList : 隣接セルリストの作成主制御	157
46.16.	Set_InteractIndex : 隣接セルリストへのセルの登録	158
46.17.	Calc_Coefficients : 多重極展開係数の計算	158
46.18.	Init_FMM : FMM の初期処理	158
46.19.	Final_FMM : FMM の終了処理	159
46.20.	Calc_FMMExcessEnergy : 結合原子間の相互作用計算	159
46.21.	Get_ParentCell : parent セル取得	159
46.22.	Get_ChildCells : child セルの取得	159
46.23.	ID_OfIndex : セル ID 取得	160
46.24.	Index_OfID : セルのインデックス取得	160
47.	Umbrella_Method	161
47.1.	Calc_UmbrellaEnergyH1 : 参照座標数 1 調和型 Umbrella ポテンシャル計算	161
47.2.	Calc_UmbrellaEnergyH2 : 参照座標数 2 調和型 Umbrella ポテンシャル計算	161
47.3.	Calc_UmbrellaEnergyL1 : 参照座標数 1 線形 Umbrella ポテンシャル計算	161
47.4.	Calc_UmbrellaEnergyL2 : 参照座標数 2 線形 Umbrella ポテンシャル計算	162
47.5.	Calc_FilledPotential : 全体の FilledPotential 計算	162
47.6.	Calc_UmbrellaPotential : 対象次元の FilledPotential 計算	162
47.7.	Calc_FlowEnergy : FlowForce 計算主制御	163
47.8.	Calc_FlowForce : FlowForce 計算	163
47.9.	UpdateFPAverage : FP 対象原子の力および座標の平均値の更新	163
48.	Tsallis_Method	164
48.1.	Calc_Kinetic : 系の運動量計算	164
48.2.	Calc_TsallisEquation : 速度と座標の微分値計算	164
48.3.	Exec_RungeKuttaTsallis : 4 次の Runge-Kutta 法での速度、座標の増分計算	165
48.4.	Set_DataAfterIntegration : 積分器後の物理量計算	165
48.5.	Exec_TsallisIntegration : Tsallis での積分器主制御	166
48.6.	Init_Tsallis : Tsallis 法初期処理	166
49.	ReplicaMD_Control	167
49.1.	Init_ReplicaMD : レプリカ法の MD 初期処理	167
49.2.	Exec_ReplicaMD : レプリカ交換ループの制御	167
49.3.	Exec_ReplicaMDLoop : レプリカ法での MD の実行	167
49.4.	Final_ReplicaMD : レプリカ法での MD の終了処理	168
49.5.	ReplicaMD_Main : レプリカ交換 MD の主制御	168
50.	Replica_Method	169
50.1.	Exchange_Replica : レプリカ交換判定	169
50.2.	Init_Replica : レプリカ交換法初期処理	169
50.3.	Set_ReplicaTemperature : レプリカのシミュレーション温度設定	169
50.4.	Final_Replica : レプリカ法終了処理	169
50.5.	Set_ReplicaData : 系の状態の保存	170
50.6.	Set_ReplicaPotential : エネルギーの保存	170
50.7.	Get_ReplicaData : 形の状態の取得	170
50.8.	Get_ReplicaPotential : エネルギーの取得	170
50.9.	Output_ReplicaLog : 各レプリカのログ出力	171
50.10.	Input_ReplicaRestartFile : レプリカ法のリスタートファイル入力	171
50.11.	Output_ReplicaRestartFile : レプリカ法のリスタートファイル出力	171
50.12.	Get_ReplicaTargetTemperature : レプリカのシミュレーション温度取得	171
50.13.	Add_ReplicaInfoToTrajectory : トrajekトリへのレプリカ情報追加	172
51.	EffectiveTemp_Method	173
51.1.	Init_EffectiveTempMCMD : 有効温度マルチカノニカル法初期化	173

51.2. Scale_GradientForEffectiveTemp : 有効温度での力のスケーリング	173
51.3. Calc_DistributionFromLog : 統計データからのスケーリングファクター計算	173
51.4. Output_Historgam : ヒストグラム出力	173
51.5. Read_EffectiveTempRestart : リスタートデータの入力	174
51.6. Write_EffectiveTempRestart : リスタートデータの出力	174
51.7. Rollback_EffectiveTempRestart : データのロールバック	174
52. Periodic_Nonbonded	175
52.1. Calc_Periodic15InteractEnergy : 周期系での 1-5 相互作用計算	175
52.2. Calc_VdwPeriodic : 周期系での van der Waals 計算	176
52.3. Calc_HydPeriodic : 周期系での水素結合計算	176
52.4. Calc_ConstElePeriodic : 周期系での距離非依存静電計算	176
52.5. Calc_DependElePeriodic : 周期系での距離依存静電計算	177
52.6. Calc_VdwConstElePeriodic : 周期系での van der Waals と距離非依存静電計算	177
52.7. Calc_VdwDependElePeriodic : 周期系での van der Waals と距離依存静電計算	177
52.8. Calc_VdwHydPeriodic : 周期系での van der Waals と水素結合計算	178
52.9. Calc_HydConstElePeriodic : 周期系での水素結合と距離非依存静電計算	178
52.10. Calc_HydDependElePeriodic : 周期系での水素結合と距離依存静電計算	178
52.11. Calc_VdwHydConstElePeriodic : 周期系での van der Waals, 水素結合, 距離非依存静電計算	179
52.12. Calc_VdwHydDependElePeriodic : 周期系での van der Waals, 水素結合, 距離依存静電計算	179
52.13. Calc_PmeEwald15InteractEnergy : PME/Ewald での 1-5 相互作用計算	180
52.14. Calc_EleDirect : 実空間の静電計算	181
52.15. Calc_VdwEleDirect : 実空間の van der Waals, 静電計算	181
52.16. Calc_HydEleDirect : 実空間の水素結合, 静電計算	181
52.17. Calc_VdwHydDirect : 実空間の van der Waals, 水素結合, 静電計算	182
53. Rattle_Method	183
53.1. Exec_Rattle : Rattle 法主制御	183
53.2. Calc_TwoAtomsRattle : 速度補正計算 (2 原子間 Rattle 拘束)	183
53.3. Calc_ThreeAtomsRattle : 速度補正計算 (3 原子間 Rattle 拘束)	183
53.4. Calc_FourAtomsRattle : 速度補正計算 (4 原子間 Rattle 拘束)	183
54. Calc_ExtCap	184
54.1. Allocate_ExtCapData : 計算パラメータの設定および計算領域の確保	184
54.2. Allocate_DynamicArray : 計算領域の確保	184
54.3. Set_ExtCapParameter : 計算パラメータの設定	185
54.4. Transfer_ExtCapParameter : 計算パラメータの送信	185
54.5. Calc_ExtCapEnergy : 拡張 CAP 拘束計算	186
55. Calc_Range	187
55.1. Allocate_RangeData : 計算パラメータの設定および計算領域の確保	187
55.2. Allocate_DynamicArrayForRange : 計算領域の確保	187
55.3. Set_RangeParameter : 計算パラメータの設定	188
55.4. Transfer_RangeParameter : 計算パラメータの送信	188
55.5. Calc_RangeEnergy : 範囲付き位置拘束計算	188
56. GeneralizedBorn_Method	189
56.1. Allocate_GeneralizedBornData : GB 法の動的メモリ確保	189
56.2. Set_GeneralizedBornData : GB 法の原子データ設定	189
56.3. Init_GeneralizedBorn : GB 法の初期設定	190
56.4. Create_AtomList : 原子ペアリストの作成	190
56.5. Calc_GeneralizedBornEnergy : GB 法計算主制御	190
56.6. Calc_BornRadius : Born 半径計算	191
56.7. Calc_GPolarization : Born 半径計算	191
56.8. Create_AtomList : Born 半径計算	192
57. AccessibleSurface_Method	193

57.1. Allocate_AccessibleSurfaceData : ASA 法の動的メモリ確保	193
57.2. Set_AccessibleSurfaceData : ASA 法の原子データ設定	193
57.3. Set_AccessibleSurfaceParameter : ASA 法のパラメータ設定	194
57.4. Update_AccessibleTable : ASA 法の相互作用テーブル作成	194
57.5. Calc_AccessibleSurfaceEnergy : ASA 法のエネルギー・力計算	194
58. SimulatedTemp_Method	195
58.1. Init_SimulatedTempMCMD : Simulated-Tempering 法初期化	195
58.2. Scale_GradientForSimulatedTemp : Simulated-Tempering での力のスケーリング	195
58.3. Calc_DistributionFromLog : 統計データからのスケーリングファクター計算	195
58.4. Output_Historgam : ヒストグラム出力	195
58.5. Get_Integration : ヒストグラムからの確率計算	196
58.6. Read_SimulatedTempRestart : リスタートデータの入力	196
58.7. Write_SimulatedTempRestart : リスタートデータの出力	196
58.8. Rollback_SimulatedTempRestart : データのロールバック	196
59. ForceBias_Method	197
59.1. Init_ForceBiasMCMD : Force-Bias 法初期化	197
59.2. Scale_GradientForForceBias : Force-Bias での力のスケーリング	197
59.3. Calc_DistributionFromLog : 統計データからのスケーリングファクター計算	197
59.4. Output_Historgam : ヒストグラム出力	197
59.5. Get_Integration : ヒストグラムからの確率計算	198
59.6. Read_ForceBiasRestart : リスタートデータの入力	198
59.7. Write_ForceBiasRestart : リスタートデータの出力	198
59.8. Rollback_ForceBiasRestart : データのロールバック	198
60. Charmm_Method	199
60.1. Calc_UreyBradlyEnergy : Urey-Bradly 項の計算	199
60.2. Calc_CharmmImproperEnergy : CHARMM 用 Improper-Torsion の計算	199
60.3. Calc_CharmmTorsionalEnergy : CHARMM 用 1-4 相互作用の計算	199
60.4. Calc_Charmm15InteractEnergy : 孤立系の CHARMM 用 1-5 相互作用の計算の主制御	200
60.5. Calc_CharmmVdwCutoff : 孤立系の CHARMM 用 1-5 van der Waals 相互作用の計算	200
60.6. Calc_CharmmVdwEleCutoff : 孤立系の CHARMM 用 1-5 静電、van der Waals 相互作用の計算	201
60.7. Calc_Charmm15PeriodicEnergy : 周期系の CHARMM 用 1-5 相互作用の計算の主制御	201
60.8. Calc_CharmmVdwPeriodic : 周期系の CHARMM 用 1-5 van der Waals 相互作用の計算	202
60.9. Calc_CharmmVdwElePeriodic : 周期系の CHARMM 用 1-5 静電、van der Waals 相互作用の計算	202
60.10. Calc_CharmmPotentialParameter : CHARMM 用ポテンシャルパラメータの設定	203
60.11. Set_CharmmCutoff : CHARMM 用カットオフ関連パラメータの算出	203
60.12. Init_Charmm : CHARMM ポテンシャル計算用初期設定	203
60.13. Calc_CharmmRigidVirialTensor : 剛体のビリアル計算 (Charmm ポテンシャル用)	203
61. GeneralST_Method	204
61.1. Init_GeneralSTMCMC	204
61.2. Allocate_GuideFunctionData	204
61.3. Scale_GradientForGeneralST	204
61.4. Calc_DistributionFromLog	204
61.5. Output_Histogram	205
61.6. Calc_ApproxWeight	205
61.7. Calc_EnsembleWeight	205
61.8. Get_Integration	205
61.9. Read_GeneralSTRestart	206
61.10. Write_GeneralSTRestart	206
61.11. Rollback_GeneralSTRestart	206
62. Calc_Saturation	207
62.1. Allocate_SatData : SAT ポテンシャルエネルギー計算の動的メモリの確保	207
62.2. Allocate_SatTemporaryArray : SAT ポテンシャルエネルギー計算の動的メモリの確保 (配列)	

.....	207
62.3. Init_Saturation : SAT パラメータの初期化.....	207
62.4. Calc_SatParameter : SAT パラメータの算出.....	207
62.5. Calc_SaturationEnergy : SAT ポテンシャルエネルギー計算主制御.....	208
62.6. eig : 固有値および固有ベクトルの計算.....	208
62.7. txt : 行列と行列の積の計算.....	208
62.8. emi : (単位行列一行列) の計算.....	208
62.9. new : 行列とベクトルの積の計算.....	209
62.10. delexp : SAT ポテンシャルエネルギー計算.....	209
62.11. Output_SatFile : SAT データの出力.....	209
63. Calc_ResidueDipolar.....	210
63.1. Allocate_RDCData : RDC ポテンシャルエネルギー計算の動的メモリの確保.....	210
63.2. Set_RDCParameter : RDC ポテンシャルエネルギー計算のパラメータ設定.....	210
63.3. Calc_ResidueDipolarEnergy : RDC ポテンシャルエネルギー計算.....	210
63.3. Output_RDCFile : RDC データの出力.....	210
64. Calc_NuclearEffect.....	211
64.1. Allocate_NOEData : NOE ポテンシャルエネルギー計算の動的メモリの確保.....	211
64.2. Set_NOEParameter : NOE ポテンシャルエネルギー計算のパラメータ設定.....	211
64.3. Calc_NuclearEffectEnergy : NOE ポテンシャルエネルギー計算.....	211
48.6. Calc_Histogram : エネルギーのヒストグラム計算.....	212
64. Tsallis_Function.....	213
64.1. Get_KTValue : kT の取得.....	213
64.2. Calc_LowerParameter : 低エネルギー側パラメータ計算.....	213
64.3. Calc_ScaleFactor : 速度、座標増分のスケールファクター計算.....	213
64.4. Calc_LowerScale : 低エネルギー側スケールファクター計算.....	213
64.5. Calc_UpperScale : 高エネルギー側スケールファクター計算.....	214
64.6. Calc_ZetaDistributionDensity : ζ 分布計算.....	214
64.7. Calc_LogOfR0 : $\ln(\rho)$ 計算.....	214
64.8. Calc_NormalLogOfR0 : 通常のエネルギーでの $\ln(\rho)$ 計算.....	214
64.9. Calc_LowerLogOfR0 : 低エネルギーでの $\ln(\rho)$ 計算.....	215
64.10. Calc_UpperLogOfR0 : 高エネルギーでの $\ln(\rho)$ 計算.....	215
64.11. Init_TsallisFunction : Tsallis 関数の初期化.....	215
64.12. Display_InitTsallis : Tsallis の初期情報表示.....	216
65. Smooth_Method.....	217
65.14. Calc_VdwPeriodicSmooth : 周期系のスプライン補間付き van der Waals 計算.....	224
65.15. Calc_HydPeriodicSmooth : 周期系のスプライン補間付き水素結合計算.....	224
65.16. Calc_ConstElePeriodicSmooth : 周期系のスプライン補間付き距離非依存静電計算.....	224
65.16. Calc_DependElePeriodic : 周期系のスプライン補間付き距離依存静電計算.....	225
65.17. Calc_VdwConstElePeriodicSmooth : 周期系のスプライン補間付き vdw と距離非依存静電計算.....	225
65.18. Calc_VdwDependElePeriodicSmooth : 周期系のスプライン補間付き vdw と距離依存静電計算.....	225
65.19. Calc_VdwHydPeriodicSmooth : 周期系のスプライン補間付き van der Waals と水素結合計算.....	226
65.20. Calc_HydConstElePeriodicSmooth : 周期系スプライン補間付き水素結合+距離非依存静電計算.....	226
65.21. Calc_HydDependElePeriodicSmooth : 周期系スプライン補間付き水素結合と距離依存静電計算.....	227
65.22. Calc_VdwHydConstElePeriodicSmooth : 周期系スプライン補間 vdw, 水素, 距離非依存静電計算.....	228
65.23. Calc_VdwHydDependElePeriodicSm : 周期スプライン補間付 vdw, 水素結合, 距離依存静電計算.....	228
65. ECC_Function.....	232
65.1. Display_ECCSetting : エラー処理付き算術関数の設定情報表示.....	232

65.2.	Set_CommentForECCFunction : エラー時のコメントの設定	232
65.3.	V_EXP : エラー処理付き指数関数	232
65.3.	V_LOG : エラー処理付き対数関数	232
65.4.	V_DIV : エラー処理付き除算	233
65.5.	V_POW : エラー処理付きべき乗	233
66.	Integrate_Method	234
66.1.	Set_IntegratorFreedom : 自由度の設定	234
66.2.	Init_GeneralParameter : 共通項目の設定	234
66.3.	Init_RungeKutta : 4 次 Runge-Kutta 法初期化	234
66.4.	Exec_RungeKutta : 4 次 Runge-Kutta 法の 1 ステップ実行	235
66.5.	Init_EXVPIntegrator : 拡張相空間体積保存積分法初期化	235
66.6.	Exec_EXVPIntegrator : 拡張相空間体積保存積分法の 1 ステップ実行	235
66.7.	Init_PredictorCorrector : Predictor-Corrector 法初期化	236
66.8.	Exec_PredictorCorrector : Predictor-Corrector 法の 1 ステップ実行	236
66.9.	Get_PredictorRestartData : Predictor-Corrector 法のリスタートデータの取得	236
66.10.	Rollback_PredictorRestartData : Predictor-Corrector 法のリスタートデータの取得	236
66.11.	Put_PredictorRestartData : Predictor-Corrector 法のリスタートデータの退避	237
66.12.	Init_Gear : GEAR 法初期化	237
66.13.	Exec_GearIntegration : GEAR 法の 1 ステップ実行	237
66.14.	Get_GearRestartData : GEAR 法のリスタートデータの取得	237
66.15.	Rollback_GearRestartData : GEAR 法のリスタートデータの取得	238
66.16.	Put_GearRestartData : GEAR 法のリスタートデータの退避	238
67.	Data_Stock	239
67.1.	Allocate_StockData : 退避領域の確保	239
67.2.	Deallocate_StockData : 退避領域の解放	239
67.3.	Set_StockIntScalar : 整数型スカラーのデータ追加	239
67.4.	Set_StockInt1Dim : 整数型 1 次元配列のデータ追加	239
67.5.	Set_StockInt2Dim : 整数型 2 次元配列のデータ追加	239
67.6.	Set_StockInt3Dim : 整数型 3 次元配列のデータ追加	240
67.7.	Set_StockRealScalar : 実数型スカラーのデータ追加	240
67.8.	Set_StockReal1Dim : 実数型 1 次元配列のデータ追加	240
67.9.	Set_StockReal2Dim : 実数型 2 次元配列のデータ追加	240
67.10.	Set_StockReal3Dim : 実数型 3 次元配列のデータ追加	240
67.11.	Get_StockIntScalar : 整数型スカラーのデータ取得	241
67.12.	Get_StockInt1Dim : 整数型 1 次元配列のデータ取得	241
67.13.	Get_StockInt2Dim : 整数型 2 次元配列のデータ取得	241
67.14.	Get_StockInt3Dim : 整数型 3 次元配列のデータ取得	241
67.15.	Get_StockRealScalar : 実数型スカラーのデータ取得	241
67.16.	Get_StockReal1Dim : 実数型 1 次元配列のデータ取得	241
67.17.	Get_StockReal2Dim : 実数型 2 次元配列のデータ取得	242
67.18.	Get_StockReal3Dim : 実数型 3 次元配列のデータ取得	242
67.19.	Output_StockData : 退避データのファイル出力	242
67.20.	Input_StockData : 退避データのファイル入力	242
67.18.	Init_StockIndex : 退避領域のインデックス初期化	243
68.	Calc_MVO	244
68.1.	Calc_MVOEnergy : MVO ポテンシャルエネルギー計算	244
69.	Combine_Method	245
69.1.	Combine_Main : Combine 法向けポテンシャルエネルギー計算主制御	245
69.2.	Calc_CombineVdwConstEle : Combine 法向けポテンシャルエネルギー計算	245

各モジュールの仕様は以下の項目で構成する。なお、型の記述は Fortran90 の文法に従う。

項番	項目	概要
#1	名称	当該メソッドの名称を規定する。メソッドは手続きで表現する。
#2	呼び出し形式	メソッドの種別(手続き種別)と引数を規定する。
#3	引数	手続きの仮引数を規定する。
#4	戻り値	当該メソッドが関数である場合の戻り値を規定する。
#5	機能	当該メソッドの機能概要を規定する。

1. Adjust_Velocity

ファイル名 : Adjust_Velocity.f90

1.1. Control_Velocity : 系全体の運動制御

名称 : Control_Velocity

呼び出し形式 : subroutine Control_Velocity(cord, vel, mass, top, last, mode)

引数 :

real*8, intent(in), dimension(3,*)::cord	原子位置
real*8, intent(inout), dimension(3,*)::vel	原子速度
real*8, intent(in), dimension(*)::mass	自由原子の質量
integer, intent(in)::top	先頭原子番号
integer, intent(in)::last	最終原子番号
integer, intent(in)::mode	系の制御モード(回転、平行移動、両方)

戻り値 :

なし

機能 :

系全体の平行移動、回転運動の速度を求め、各原子の速度から系全体の速度を引くことで系全体の速度を抑止する。

処理手順を以下に示す。

- 1) 系全体の質量と重心を計算する
- 2) 系全体の平行移動量を計算する
- 3) 系全体の回転運動を計算する
- 4) 各分子から系全体の運動量を引き、系全体の運動を止める

2. Analyze_Method

ファイル名 : Analyze_Method.f90

2.1. Analyze_Force : 原子間の力の解析

名称 : Analyze_Force

呼び出し形式 : subroutine Analyze_Force

引数 :

なし

戻り値 :

なし

機能 :

原子間の力を求め、データを出力する

処理手順を以下に示す。

- 1) 1-5 相互作用相手を計算する
- 2) 分子の相互作用力と拘束力を計算する
- 3) root-mean square force を計算する
- 4) root-mean square force を標準出力に出力する
- 5) 各原子にかかる力をファイルに出力する

2.2. Analyze_ElectroPotential : 静電ポテンシャルエネルギーの解析

名称 : Analyze_ElectroPotential

呼び出し形式 : subroutine Analyze_ElectroPotential(atom)

引数 :

type(Atom_t), intent(in), dimension(:)::atom 原子情報

戻り値 :

なし

機能 :

静電ポテンシャルエネルギーを求め、データを出力する。

処理の手順を以下に示す。

- 1) 非共有結合原子間の静電ポテンシャルエネルギーを計算する
- 2) 静電ポテンシャルエネルギーを標準出力に出力する
- 3) 各原子の静電ポテンシャルエネルギーをファイルに出力する

2.3. Analyze_Violation : van der Waals 接触の解析

名称 : Analyze_Violation

呼び出し形式 : subroutine Analyze_Violation(atom)

引数 :

type(Atom_t), intent(in), dimension(:)::atom 原子情報

戻り値 :

なし

機能 :

全原子間の van der Waals 接触(*1)を解析する。

処理の手順を以下に示す。

- 1) bond 結合に対する van der Waals 接触数を計算する
- 2) angle に対する van der Waals 接触数を計算する
- 3) 共有結合していない原子間の van der Waals 接触を計算する
- 4) van der Waals 接触数を出力する

(*1) 2 原子の van der Waals 半径の和に対し、2 原子間の距離が短い場合に接触とみなす

2.4. Analyze_MinimizeEnergy : エネルギー最小化フェーズの結果解析

名称 : Analyze_MinimizeEnergy

呼び出し形式 : subroutine Analyze_MinimizeEnergy

引数 :

なし

戻り値 :

なし

機能 :

Minimize のエネルギートラジェクトリーを解析する
処理の手順を以下に示す。

- ・ 解析区間のトラジェクトリーデータを入力
- ・ エネルギーの平均値と標準偏差を計算
- ・ ログ出力
- ・ GNUPLOT 形式のデータ を出力

2.5. Analyze_MDEnergy : MD のエネルギートラジェクトリー解析

名称 : Analyze_MDEnergy

呼び出し形式 : subroutine Analyze_MDEnergy

引数 :

なし

戻り値 :

なし

機能 :

MD のエネルギートラジェクトリーを解析する
処理の手順を以下に示す。

- ・ 解析区間のトラジェクトリーデータを入力
- ・ エネルギーの平均値と標準偏差を計算
- ・ ログ出力
- ・ GNUPLOT 形式のデータ を出力

2.6. Analyze_DistanceTrajectory : 距離のトラジェクトリー解析

名称 : Analyze_DistanceTrajectory

呼び出し形式 : subroutine Analyze_DistanceTrajectory

引数 :

なし

戻り値 :

なし

機能 :

原子間の距離データのトラジェクトリーを解析する
処理の手順を以下に示す。

- ・ 解析区間のトラジェクトリーデータを入力
- ・ 原子間の距離の平均値と標準偏差を計算
- ・ ログ出力
- ・ GNUPLOT 形式のデータ を出力

2.7. Analyze_DihedralTrajectory : 二面角のトラジェクトリー解析

名称 : Analyze_DihedralTrajectory

呼び出し形式 : subroutine Analyze_DihedralTrajectory

引数 :

なし

戻り値 :

なし

機能 :

二面角の角度データのトラジェクトリーを解析する

処理の手順を以下に示す。

- ・ 解析区間の開始時間から終了時間までのトラジェクトリーデータを入力
- ・ 二面角角度の平均値と標準偏差を計算
- ・ ログ出力
- ・ GNUPLOT 形式のデータ を出力

2.8. Analyze_Fluctuation : 原子位置の揺らぎ解析

名称 : Analyze_Fluctuation

呼び出し形式 : subroutine Analyze_Fluctuation

引数 :

なし

戻り値 :

なし

機能 :

原子位置のトラジェクトリーを入力し、揺らぎを解析する。

処理の手順を以下に示す。

- ・ 参照原子位置データを入力
- ・ 解析区間の原子位置データを読み込む
- ・ 位置の標準偏差と平均値を計算する
- ・ 解析区間の原子位置データを読み込み、揺らぎの総和を計算する
- ・ 揺らぎの計算を行う
- ・ calculate fluctuation
- ・ ログ出力
- ・ 揺らぎのファイル出力
- ・ GNUPLOT 形式のファイル出力

2.9. Analyze_Combine : Combine 法エネルギー解析

名称 : Analyze_Combine

呼び出し形式 : subroutine Analyze_Combine

引数 :

なし

戻り値 :

なし

機能 :

原子位置のトラジェクトリーを入力し、Combine 法向けエネルギー計算を行った結果をファイルに出力する。

3. Calc_Angle

ファイル名 : Angle.f90

3.1. Calc_AngleEnergy : angle のエネルギー計算

名称 : Calc_AngleEnergy

呼び出し形式 : function Calc_AngleEnergy(angle, atom, virial, init, last, step)
result (totalEnergy)

引数 :

type(Angle_t), intent(in), dimension(MAX_ANGLE)::angle	angle 情報
type(Atom_t), intent(inout), dimension(MAX_ATOM)::atom	原子情報
type(Rigid_t), intent(in), dimension(MAX_RIGID)::rigid	剛体情報
real*8, dimension(3,3), intent(inout)::virial	ビリアル
integer, intent(in)::init	ループ初期値
integer, intent(in)::last	ループ終値
integer, intent(in)::step	ループ増分値

戻り値 :

real*8::totalEnergy	angle エネルギー
---------------------	-------------

機能 :

系内の angle のポテンシャルエネルギーと原子に加わる力およびビリアルを求める。

3.2. Calc_ClusterAngleEnergy : angle のクラスタ間エネルギー計算

名称 : Calc_ClusterAngleEnergy

呼び出し形式 : function Calc_ClusterAngleEnergy(angle, atom, Cluster, init, last) &
result (totalEnergy)

引数 :

type(Angle_t), intent(in), dimension(MAX_ANGLE)::angle	angle 情報
type(Atom_t), intent(inout), dimension(MAX_ATOM)::atom	原子情報
type(Cluster_t), dimension(MAX_CLUSTER)::Cluster	クラスタ情報
integer::init	ループ初期値
integer::last	ループ終値

戻り値 :

real*8, dimension(MAX_CLUSTER, MAX_CLUSTER)::totalEnergy	クラスタ間 angle エネルギー
--	-------------------

機能 :

系内とクラスタ間の angle のポテンシャルエネルギーと原子に加わる力およびビリアルを求める。

4. Calc_Bond

ファイル名 : Bond.f90

4.1. Calc_BondEnergy : bond のエネルギー計算

名称 : Calc_BondEnergy

呼び出し形式 : function Calc_BondEnergy(bond, atom, rigid, init, last, step)
result (totalEnergy)

引数 :

type(Bond_t), intent(in), dimension(MAX_BOND)::bond	bond 情報
type(Atom_t), intent(inout), dimension(MAX_ATOM)::atom	原子情報
type(Rigid_t), intent(in), dimension(MAX_RIGID)::rigid	剛体情報
real*8, dimension(3,3), intent(inout)::virial	ビリアル
integer::init	ループ初期値
integer::last	ループ終値
integer::step	ループ増分値

戻り値 :

real*8::totalEnergy	総 bond エネルギー
---------------------	--------------

機能 :

系内の bond のポテンシャルエネルギーと原子に加わる力およびビリアルを求める。

4.2. Calc_BondEnergy : bond のクラスタ間エネルギー計算

名称 : Calc_ClusterBondEnergy

呼び出し形式 : function Calc_ClusterBondEnergy(Bond, Atom, Cluster, init, last) &
result (totalEnergy)

引数 :

type(Bond_t), intent(in), dimension(MAX_BOND)::Bond	bond 情報
type(Atom_t), intent(inout), dimension(MAX_ATOM)::Atom	原子情報
type(Cluster_t), intent(inout), dimension(MAX_CLUSTER)::Cluster	クラスタ情報
integer::init	ループ初期値
integer::last	ループ終値

戻り値 :

real*8, dimension(MAX_CLUSTER, MAX_CLUSTER)::totalEnergy	総 bond エネルギー
--	--------------

機能 :

系内とクラスタ間の bond のポテンシャルエネルギーと原子に加わる力およびビリアルを求める。

5. Boundary_Function

ファイル名 : Boundary_Function.f90

5.1. Adjust_EllipsoidBoundary : 楕円体セルの補正

名称 : Adjust_EllipsoidBoundary

呼び出し形式 : subroutine Adjust_EllipsoidBoundary(object, vel, dt)

引数 :

real*8, intent(in)::dt

MD の 1 ステップ当りの経過時間

type(Object_t), intent(inout)::object

オブジェクトクラス

real*8, intent(inout), dimension(:, :)::vel

自由原子の速度

戻り値 :

なし

機能 :

楕円体の境界条件に対し、境界条件を超えた場合は反射するように速度と原子位置を補正する。

6. Boundary_Method

ファイル名 : Boundary_Method.f90

6.1. Adjust_Boundary : 距離ベクトルの補正

名称 : Adjust_Boundary

呼び出し形式 : function Adjust_Boundary(vec) result (adjustVec)

引数 :

real*8, intent(in), dimension(3)::vec 距離ベクトル

戻り値 :

real*8, dimension(3)::adjustVec 補正後の距離ベクトル

機能 :

距離ベクトルを周期境界条件で補正する。

6.2. Replica_Index : レプリカセルのインデックス取得

名称 : Replica_Index

呼び出し形式 : function Replica_Index(vec) result (index)

引数 :

real*8, intent(in), dimension(3)::vec 原子間の距離ベクトル

戻り値 :

real*8, dimension(3)::index レプリカセルの相対位置

機能 :

距離ベクトルに対し、対応するレプリカセルの位置を求める。

ただし、基本セルの位置は (0, 0, 0) とする。

6.3. Adjust_BoundaryWithReplica : 1-5 相互作用テーブルでの距離ベクトル補正

名称 : Adjust_BoundaryWithReplica

呼び出し形式 : function Adjust_BoundaryWithReplica(vec, id, replicaID) result (adjustVec)

引数 :

real*8, intent(in), dimension(3)::vec 原子 1 と原子 2 の距離ベクトル

integer, intent(in)::id 原子 1 の ID

integer, intent(in)::replicaId 原子 2 の ID

戻り値 :

real*8, dimension(3)::adjustVec 補正後の距離ベクトル

機能 :

周期境界条件において 1-5 相互作用テーブルでの原子 1 に対する原子 2 の距離ベクトルを求める。

6.4. Compress_ReplicaIndex : レプリカセルインデックスの一次元化

名称 : Compress_ReplicaIndex

呼び出し形式 : function Compress_ReplicaIndex(index) result (compressIndex)

引数 :

integer, dimension(3), intent(in)::index レプリカセルの 3 次元インデックス

戻り値 :

integer::compressIndex レプリカセルの 1 次元インデックス

機能 :

レプリカセルの 3 次元インデックスを 1 次元インデックスに変換する。

6.5. Create_ReplicaCellCoordinate : レプリカセルの座標設定

名称 : Create_ReplicaCellCoordinate

呼び出し形式 : subroutine Create_ReplicaCellCoordinate

引数 :

なし

戻り値 :

なし

機能 :

レプリカセルのインデックスに対するレプリカセルの座標を設定する。

6.6. Create_BoundaryStatus : セル行列、セル行列の逆行列および行列式の計算

名称 : Create_BoundaryStatus

呼び出し形式 : subroutine Create_BoundaryStatus(hexa)

引数 :

type(Hexahedral_t), intent(inout)::hexa 斜方セル情報

戻り値 :

なし

機能 :

セルの各辺の長さおよび角度を基に、セル行列、セル行列の逆行列および行列式を計算する。

6.7. Calc_LengthAndAngle : セルの各辺の長さおよび角度の計算

名称 : Calc_LengthAndAngle

呼び出し形式 : subroutine Calc_LengthAndAngle(hexa)

引数 :

type(Hexahedral_t), intent(inout)::hexa 斜方セル情報

戻り値 :

なし

機能 :

セル行列を基に、セルの各辺の長さおよび角度を計算する。

7. Calc_Cap

ファイル名 : Cap. f90

7.1. Calc_CapEnergy : CAP エネルギー計算

名称 : Calc_CapEnergy

呼び出し形式 : function Calc_CapEnergy(Cap, Atom, temp, init, last) result (totalEnergy)

引数 :

type(Cap_t), intent(in), dimension(MAX_MOL)::Cap	Cap 情報
type(Atom_t), intent(inout), dimension(MAX_ATOM)::Atom	原子情報
real*8, intent(in)::temp	拘束温度
integer, intent(in)::init	ループ初期値
integer, intent(in)::last	ループ終値

戻り値 :

real*8::totalEnergy	総 Cap エネルギー
---------------------	-------------

機能 :

系内の Cap のポテンシャルエネルギーと原子に加わる力を求める。

- 1) 系内の各分子の重心を求める
- 2) Cap ポテンシャルエネルギーを求める。また、各分子に加わる力を求める。
エネルギーの計算は以下の二種類がある。
 - 2-1) half harmonic

$$\text{energy} = \text{distance} ** 2 * 0.5 * \text{coeff} \quad (r > \text{cap radius})$$

$$\text{energy} = 0 \quad (r \leq \text{cap radius})$$
 - 2-2) biquadratic

$$\text{energy} = ((\text{distance} ** 2 - \text{radius} ** 2) / \text{radius} ** 2) ** 2 * \text{radius} * \text{coeff} * 0.5$$

distance = distance of cap-center and atom
- 3) 各原子への力を加算する。(

7.2. Calc_ClusterCapEnergy : CAP のクラスタ間エネルギー計算

名称 : Calc_ClusterCapEnergy

呼び出し形式 : function Calc_ClusterCapEnergy(Cap, Atom, cluster, temp, &init, last) result (totalEnergy)

引数 :

type(Cap_t), intent(in), dimension(MAX_MOL)::Cap	Cap 情報
type(Atom_t), intent(inout), dimension(MAX_ATOM)::Atom	原子情報
type(Cluster_t), intent(inout), dimension(MAX_CLUSTER)::cluster	クラスタ情報
real*8, intent(in)::temp	拘束温度
integer, intent(in)::init	ループ初期値
integer, intent(in)::last	ループ終値

戻り値 :

real*8, dimension(MAX_CLUSTER, MAX_CLUSTER)::totalEnergy	クラスタ間の Cap エネルギー
--	------------------

機能 :

系内とクラスタ間の Cap のポテンシャルエネルギーと原子に加わる力を求める。
クラスタ間のエネルギー、力を求めること以外は Calc_CapEnergy と同一処理である。

8. Check_Class

ファイル名 : Check_Class.f90

8.1. Check_AtomDistance : 原子間の距離チェック

名称 : Check_AtomDistance

呼び出し形式 : subroutine Check_AtomDistance

引数 :

なし

戻り値 :

なし

機能 :

全ての二原子間の距離がリミット値より近いかをチェックする。

注意点 :

現在のリミット距離は 1.0d05 (Å)

8.2. Check_LinearTorsion : 二面角構成原子の直線チェック

名称 : Check_LinearTorsion

呼び出し形式 : subroutine Check_LinearTorsion

引数 :

なし

戻り値 :

なし

機能 : torsion を構成する原子が直線上に配置しているかをチェックする。

処理の手順を以下に示す。

注意点 :

現在のリミット角度は 1.0d05 (radian)

8.3. Check_ChainInEllipsoid : 楕円体セルの境界チェック

名称 : Check_ChainInEllipsoid

呼び出し形式 : subroutine Check_ChainInEllipsoid

引数 :

なし

戻り値 :

なし

機能 :

分子の重心が楕円体の境界条件内にあるかをチェックする。

8.4. Check_Boundary : 周期境界条件のチェック

名称 : Check_Boundary

呼び出し形式 : subroutine Check_Boundary

引数 :

なし

戻り値 :

なし

機能 :

周期境界条件において、原子間の距離がセルのサイズを越えないかをチェックする。

8.5. Check_ReplicaPosition : 相互作用テーブルでの周期境界条件チェック

名称 : Check_ReplicaPosition

呼び出し形式 : subroutine Check_ReplicaPosition(interact, i, cutoff)

引数 :

integer, intent(in)::i	当該原子 ID
type(Interact15_t), intent(in)::interact	1-5 相互作用情報
real*8, intent(in)::cutoff	カットオフ距離

戻り値 :

なし

機能 :

周期境界条件において、相互作用テーブルを使用して得た原子間の距離が正しいかをチェックする。(分子ごとに決定した仮想セルの位置と原子単位に決定されるセルの位置の一致性)

8.6. Check_AtomBelongCluster : 所属クラスタのチェック

名称 : Check_AtomBelongCluster

呼び出し形式 : subroutine Check_AtomBelongCluster

引数 :

なし

戻り値 :

なし

機能 :

1-2, 1-3, 1-4 間の系の構造が同一クラスタ内に存在するかチェックする。

8.7. Check_OptionCombination : オプションの組み合わせのチェック

名称 : Check_OptionCombination

呼び出し形式 : subroutine Check_OptionCombination

引数 :

なし

戻り値 :

なし

機能 :

オプションの組み合わせで必須あるいは指定不能なものをチェックする。

- ・ STOPCE オプションに対する組み合わせ
- ・ PME/EWALD オプションに対する組み合わせ
- ・ 1-5 静電相互作用オプションに対する組み合わせ
- ・ Hybrid モデルオプションに対する組み合わせ
- ・ Nose-Hoover オプションに対する組み合わせ
- ・ 剛体モデルオプションに対する組み合わせ
- ・ FMM 法オプションに対する組み合わせ
- ・ NPT オプションに対する組み合わせ

9. Analyze_Control

ファイル名 : Control_Analyze.f90

9.1. Analyze_Main : ANALYZE 主制御

名称 : Analyze_Main

呼び出し形式 : subroutine Analyze_Main

引数 :

なし

戻り値

なし

機能 :

ANALYZE の主制御を行う。

処理手順を以下に示す。

0) 初期処理

1) van der Waals 接触の解析

2) 原子間の力の解析

3) 静電ポテンシャル解析

4) エネルギーのトラジェクトリ解析

5) 距離のトラジェクトリ解析

6) 二面角のトラジェクトリ解析

7) 平均構造と揺らぎの解析

8) 終了処理

10. MD_Control

ファイル名 : Control_MD.f90

10.1. MD_Main : MD 主制御

名称 : MD_Main

呼び出し形式 : subroutine MD_Main

引数 :

なし

戻り値 :

なし

機能 :

MD の主制御を行う。

処理手順を以下に示す。

- 1) モニターファイルの open
- 2) MD の初期処理
- 3) MD の実行
- 4) モニターファイルの close
- 5) MD の終了処理
- 6) 実行時間の出力

10.2. Init_MD : MD 初期処理

名称 : Init_MD

呼び出し形式 : subroutine Init_MD

引数 :

なし

戻り値 :

なし

機能 :

MD の初期処理を行う。

処理の手順を以下に示す。

- 1) セル情報を設定する
- 2) AMBER パラメータをチェックする
- 3) 系の自由度、分子の自由度を計算する
- 4) 原子の速度の設定、および 1-5 相互作用テーブルを作成する。

10.3. Exec_MD : MD の実行

名称 : Exec_MD

呼び出し形式 : subroutine Exec_MD

引数 :

なし

戻り値 :

なし

機能 :

MD の計算ループを実行する

10.4. Final_MD : MD 終了処理

名称 : Final_MD

呼び出し形式 : subroutine Final_MD

引数 :

なし

戻り値 :

なし

機能 :

MD の終了処理を行う。

処理手順を以下に示す。

- 1) 現時点の速度の保存 (3) で MD を実行するため
- 2) 相互作用テーブルの更新
- 3) MD を 1 回実行しエネルギーと力を求める (原子の速度と位置が更新される)
- 4) 原子の位置を元に戻す
- 5) 運動量の計算
- 6) リスタートファイルの出力
- 7) ログ、ファイルの出力

11. Minimize_Control

ファイル名 : Control_Min.f90

11.1. Minimize_Main : MINIMIZE 主制御

名称 : Minimize_Main

呼び出し形式 : subroutine Minimize_Main

引数 :

なし

戻り値 :

なし

機能 :

Minimize の主制御を行う

処理の手順を以下に示す。

- 1) minimization 初期処理
- 2) minimization 実行
- 3) minimization 終了処理

11.2. Init_Minimize : MINIMIZE 初期処理

名称 : Init_Minimize

呼び出し形式 : subroutine Init_Minimize

引数 :

なし

戻り値 :

なし

機能 :

Minimize の初期処理を行う

処理手順を以下に示す。

- 1) 境界情報条件を設定する
- 2) 出力ファイルをオープンする
- 3) 初期構造に SHAKE を適用する
- 4) 1-5 相互作用テーブルを作成する
- 5) AMBER パラメータの設定を行う
- 6) 初期構造のエネルギーと力を計算する
- 7) 初期構造のログを出力する

11.3. Exec_Minimize : MINIMIZE の実行

名称 : Exec_Minimize

呼び出し形式 : subroutine Exec_Minimize

引数 :

なし

戻り値 :

なし

機能 :

Minimize の方法に応じて、各方法の手続きを呼び出す。

11.4. Final_Minimize : MINIMIZE 終了処理

名称 : Final_Minimize

呼び出し形式 : subroutine Final_Minimize

引数 :

なし

戻り値 :

なし

機能 :

Minimize の終了処理を行う。

処理手順を以下に示す。

- 1) 1-5 相互作用テーブルを計算する
- 2) エネルギーと力を計算する
- 3) Analyze 用ファイルをクローズする
- 4) Minimize の結果を出力する

11.5. Write_Histogram : 力のヒストグラム出力

名称 : Write_Histogram

呼び出し形式 : subroutine Write_Histogram

引数 :

なし

戻り値 :

なし

機能 :

力のヒストグラムとエネルギーを標準出力に出力する。

処理手順を以下に示す。

- 1) 力の最大値、RMSF、RMSD を計算する
- 2) 力の分布を計算する
- 3) 力の最大値、RMSF、RMSD、力の分布を出力する
- 4) エネルギー情報を出力する

12. Calc_Dihedral

ファイル名 : Dihedral.f90

12.1. Calc_DihedralEnergy : 二面角拘束のエネルギー計算

名称 : Calc_DihedralEnergy

呼び出し形式 : function Calc_DihedralEnergy(Dihedral, Atom, temp, weight, &
init, last) result (totalEnergy)

引数 :

type(Dihedral_t), intent(in), dimension(MAX_DIHEDRAL):: Dihedral	二面角拘束情報
type(Atom_t), intent(inout), dimension(MAX_ATOM):: Atom	原子情報
integer, intent(in):: init	ループ初期値
integer, intent(in):: last	ループ終値
real*8, intent(in):: weight	二面角係数の重み
real*8, intent(in):: temp	拘束温度

戻り値 :

real*8:: totalEnergy	二面角拘束エネルギー
----------------------	------------

機能 :

系内の二面角拘束のポテンシャルエネルギーと原子に加わる力を求める。

12.2. Calc_DihedralEnergy : 二面角拘束のクラスタ間エネルギー計算

名称 : Calc_ClusterDihedralEnergy

呼び出し形式 : function Calc_ClusterDihedralEnergy(Dihedral, Atom, cluster, &
temp, weight, init, last) result (totalEnergy)

引数 :

type(Dihedral_t), intent(in), dimension(MAX_DIHEDRAL):: Dihedral	二面角拘束情報
type(Atom_t), intent(inout), dimension(MAX_ATOM):: Atom	原子情報
type(Cluster_t), intent(inout), dimension(MAX_CLUSTER):: cluster	クラスタ情報
integer, intent(in):: init	ループ初期値
integer, intent(in):: last	ループ終値
real*8, intent(in):: weight	二面角係数の重み
real*8, intent(in):: temp	拘束温度

戻り値 :

real*8, dimension(MAX_CLUSTER, MAX_CLUSTER):: totalEnergy	二面角拘束エネルギー
---	------------

機能 :

系内とクラスタ間の二面角拘束のポテンシャルエネルギーと原子に加わる力を求める。

13. Calc_Distance

ファイル名 : Distance.f90

13.1. Calc_DistanceEnergy : 距離拘束のエネルギー計算

名称 : Calc_DistanceEnergy

呼び出し形式 : function Calc_DistanceEnergy(Distance, Atom, temp, weight, &
init, last) result (totalEnergy)

引数 :

type(Distance_t), intent(in), dimension(MAX_DISTANCE)::Distance	距離拘束情報
type(Atom_t), intent(inout), dimension(MAX_ATOM)::Atom	原子情報
integer, intent(in)::init	ループ初期値
integer, intent(in)::last	ループ終値
real*8, intent(in)::temp	拘束温度
real*8, intent(in)::weight	距離拘束の重み

戻り値 :

real*8::totalEnergy	距離拘束エネルギー
---------------------	-----------

機能 :

系内の距離拘束エネルギーと力を求める。

計算式は2パターンあり、以下に示す

flat-well harmonic

distance < low	: energy = low_coeff *(distance - low_base)**2
low <= distance <= up	: energy = 0
up < distance	: energy = up_coeff * (distance - up_base)**2

other

distance < low	: energy = low_coeff *(diff/low_base**2)**2
low <= distance <= up	: energy = 0
up < distance	: energy = up_coeff * (diff/up_base**2)**2

diff = distance**2 - base**2

注意点 :

本バージョンでは flat-well harmonic を適用する。

13.2. Calc_DistanceEnergy : 距離拘束のクラスタ間エネルギー計算

名称 : Calc_ClusterDistanceEnergy

呼び出し形式 : function Calc_ClusterDistanceEnergy(Distance, Atom, Cluster, temp, weight, & init, last) result (totalEnergy)

引数 :

type(Distance_t), intent(in), dimension(MAX_DISTANCE)::Distance	距離拘束情報
type(Atom_t), intent(inout), dimension(MAX_ATOM)::Atom	原子情報
type(Cluster_t), intent(inout), dimension(MAX_CLUSTER)::Cluster	クラスタ情報
integer, intent(in)::init	ループ初期値
integer, intent(in)::last	ループ終値
real*8, intent(in)::temp	拘束温度
real*8, intent(in)::weight	距離拘束の重み

戻り値 :

real*8, dimension(MAX_CLUSTER, MAX_CLUSTER)::totalEnergy	距離拘束エネルギー
--	-----------

機能 :

系内とクラスタ間の距離拘束エネルギーと力を求める。

flat-well harmonic

distance < low	: energy = low_coeff *(distance - low_base)**2
low <= distance <= up	: energy = 0
up < distance	: energy = up_coeff * (distance - up_base)**2

other

distance < low	: energy = low_coeff *(diff/low_base**2)**2
low <= distance <= up	: energy = 0
up < distance	: energy = up_coeff * (diff/up_base**2)**2

diff = distance**2 - base**2

14. Exception_Routine

ファイル名 : Error.f90

14.1. Stop_Of : 処理の中断

名称 : Stop_Of

呼び出し形式 : subroutine Stop_Of

引数 :

なし

戻り値 :

なし

機能 :

実行を中断する。

14.2. Error_Of : エラー出力と処理の中断

名称 : Error_Of

呼び出し形式 : subroutine Error_Of(string)

引数 :

character(*), intent(in)::string

戻り値 :

なし

機能 :

エラーを出力し、実行を中断する

14.3. Warning_Of : ワーニングの出力

名称 : Warning_Of

呼び出し形式 : subroutine Warning_Of(string)

引数 :

character(*), intent(in)::string

戻り値 :

なし

機能 :

ワーニングメッセージを出力する。

15. File_Class

ファイル名 : File_Class.f90

15.1. File_Open : ファイルオープン

名称 : File_Open

呼び出し形式 : subroutine File_Open(unit, form, access, status, file, blank, recl, comment)

引数 :

```
integer, intent(in)::unit
character(*), intent(in)::form
character(*), intent(in)::access
character(*), intent(in)::status
character(*), intent(in), optional::file
character(*), intent(in), optional::blank
integer, intent(in), optional::recl
character(*), intent(in), optional::comment
```

戻り値 :

なし

機能 :

引数の指定に従ってファイルをオープンする。

15.2. Check_Keyword : 入出力パラメータのチェック

名称 : Check_Keyword

呼び出し形式 : function Check_Keyword(word, keyword) result (isMatch)

引数 :

character(*), intent(in)::word	入力文字列
character(*), intent(in), dimension(:)::keyword	キーワード群

戻り値 :

logical::isMatch	キーワードに存在
------------------	----------

機能 :

入力した文字列がキーワードに一致するか判定する。

15.3. File_Close : ファイルクローズ

名称 : File_Close

呼び出し形式 : subroutine File_Close(unit)

引数 :

integer, intent(in)::unit	論理装置番号
---------------------------	--------

戻り値 :

なし

機能 :

ファイルをクローズする。

16. Input_Data

ファイル名 : Input_Data.f90

16.1. Input_TopologyFile : ASCII 形式のトポロジーファイルの読み込み

名称 : Input_TopologyFile

呼び出し形式 : subroutine Input_TopologyFile

引数 :

なし

戻り値 :

なし

機能 :

トポロジーファイルを読み込む。

処理の手順を以下に示す。

- 1) 原子データの初期化を行う
- 2) トポロジーファイルをオープンする
- 3) 1行読み込み、データを設定する
- 4) トポロジーファイルをクローズする
- 5) 入力値から計算されるデータを設定する
- 6) トポロジーファイルの情報を出力する
- 7) エラーチェックを行う

16.2. Update_MoleculeInformation : 分子情報の更新

名称 : Update_MoleculeInformation

呼び出し形式 : subroutine Update_MoleculeInformation(molID, headStr)

引数 :

integer, intent(in)::molID

分子 ID

character(8), intent(in)::headStr

行の先頭 8 文字

戻り値 :

なし

機能 :

トポロジーファイル読み込み時に当該分子の情報を更新する

注意点 :

当該分子の情報は、系全体の情報と設定済みである分子情報との差分を取ることで得る。

16.3. Copy_MoleculeAsChains : 分子情報のコピー

名称 : Copy_MoleculeAsChains

呼び出し形式 : subroutine Copy_MoleculeAsChains(molID, headStr)

引数 :

integer, intent(in)::molID

分子 ID

character(8), intent(in)::headStr

行の先頭 8 文字

戻り値 :

なし

機能 :

当該分子の情報を当該分子に属するチェーンにコピーする。

cosgene では分子の総称情報としての MolInfo と個々の分子の情報である ChainInfo の二つで分子を表現する。

16.4. Calc_PotentialParameter : ポテンシャル関数の設定

名称 : Calc_PotentialParameter

呼び出し形式 : subroutine Calc_PotentialParameter

引数 :

なし

戻り値 :

なし

機能 :

1-5 の van der Waals ポテンシャル関数の係数、および 1-4 相互作用のスケールファクターを設定する。

a) AMBER の場合

- a-1) 2 原子の van der Waals の半径、井戸の深さから van der Waals 係数を算出する
2 原子の van der Waals 半径の和を radius、2 原子の van der Waals 井戸の深さの相乗平均を depth とした場合、

$$6 \text{ 乗項} = \text{depth} * \text{radius} ** 6 * 2.0$$

$$12 \text{ 乗項} = \text{depth} * \text{radius} ** 12$$

- a-2) 2 原子の 1-4 相互作用のスケールファクターの最小値を 1-4 相互作用テーブルに設定する

b) OPLS の場合

- a-1) 2 原子の van der Waals の半径、井戸の深さから van der Waals 係数を算出する
一方の van der Waals 半径を radius1、他方の van der Waals 半径を radius2、2 原子の van der Waals 井戸の深さの相乗平均を depth とした場合、

$$6 \text{ 乗項} = \text{radius1} ** 3 * \text{radius} ** 3 * \text{depth} * 4.0\text{d}0$$

$$12 \text{ 乗項} = \text{radius1} ** 6 * \text{radius} ** 6 * \text{depth} * 4.0\text{d}0$$

- a-2) 2 原子の 1-4 相互作用のスケールファクターの最小値を 1-4 相互作用テーブルに設定する

c) ECEPP の場合

- c-1) 2 原子の 1-4 相互作用のスケールファクターの最小値を 1-4 相互作用テーブルに設定する (ECEPP は係数を直接入力するため van der Waals 係数の算出は不要)

16.5. Set_ResidueInformation : 残基情報の設定

名称 : Set_ResidueInformation

呼び出し形式 : subroutine Set_ResidueInformation

引数 :

なし

戻り値 :

なし

機能 :

全残基に対し、先頭と最後の原子を設定する。

また、全原子に対して、当該原子が属する残基を設定する。

16.6. Init_Topology : トポロジーデータの初期化

名称 : Init_Topology

呼び出し形式 : subroutine Init_Topology

引数 :

なし

戻り値 :

なし

機能 :

系の Object 情報、Interact 情報をクリアする。

16.7. Input_BinaryTopologyFile : バイナリ形式のトポロジーファイルの読み込み

名称 : Input_BinaryTopologyFile

呼び出し形式 : subroutine Input_BinaryTopologyFile

引数 :

なし

戻り値 :

なし

機能 :

バイナリ形式のトポロジーファイルを読み込む。

処理の手順を以下に示す。

- 1) 初期設定処理
- 2) トポロジーファイルのオープン
- 3) タイトルの読み込み
- 4) 分子情報の読み込み
- 5) 原子情報の読み込み
- 6) 残基情報の読み込み
- 7) チェイン情報の読み込み
- 8) BOND の読み込み
- 9) ANGLE の読み込み
- 10) TORSION の読み込み
- 11) IMPROPER の読み込み
- 12) FUNCTION の読み込み
- 13) 1-5 相互作用係数の読み込み
- 14) トポロジーファイルのクローズ
- 15) その他の値の設定

16.8. Set_AdjustPhase : torsion 角の補正

名称 : Set_AdjustPhase

呼び出し形式 : subroutine Set_AdjustPhase(torsion, num)

引数 :

type(Torsion_t), intent(inout), dimension(:)::torsion
integer, intent(in)::num

入力 TORSION
総 TORSION 数

戻り値 :

なし

機能 :

TORSION の phase 角を $0 \sim 2\pi$ の値に修正する。

16.9. Input_PDBFile : PDB ファイルの読み込み

名称 : Input_PDBFile

呼び出し形式 : subroutine Input_PDBFile(file, unit, cord)

引数 :

character (NAME_SIZE)::file	ファイル名
integer::unit	装置番号
real*8, dimension(3, MAX_ATOM)::cord	原子座標

戻り値 :

なし

機能 :

PDB ファイルから原子座標を読み込む。使用するデータは座標部分のみで、他のデータは読み飛ばす。

16.10. Input_BinaryCoordinateFile : バイナリ形式の原子座標の読み込み

名称 : Input_BinaryCoordinateFile

呼び出し形式 : subroutine Input_BinaryCoordinateFile(file, unit, cord)

引数 :

character (NAME_SIZE)::file	ファイル名
integer::unit	装置番号
real*8, dimension(3, MAX_ATOM)::cord	原子座標

戻り値 :

なし

機能 :

バイナリ形式の座標ファイルから原子座標を読み込む

16.11. Input_VariableFile : 固定原子情報の読み込み

名称 : Input_VariableFile

呼び出し形式 : subroutine Input_VariableFile

引数 :

なし

戻り値 :

なし

機能 :

自由原子/固定原子の情報を読み込む

自由/固定原子の指定は以下の3種類。

(a) 分子番号、残基番号および原子名による指定

(b) 位置による指定

b-1) 中心になる原子の番号とその半径による指定

b-2) 中心座標とその半径による指定

情報種別		キーワード	パラメータ
原子リスト		SETVAR> LIST	FREE FIX チェイン先頭番号 チェイン最終番号 残基先頭番号 残基最終番号 原子名指定 [リスト出力指定]
原子集合	中心座標	SETVAR> RADIUS	FIX FREE COOR x 座標 y 座標 z 座標 半径下限 半径上限 原子名指定 [リスト出力]
	中心原子	SETVAR> RADIUS	FIX FREE ATOM チェイン番号 残基番号 原子名 半径下限 半径上限 原子名指定 [リスト出力]

16.12. Input_ShakeFile : SHAKE ファイルの読み込み

名称 : Input_ShakeFile

呼び出し形式 : subroutine Input_ShakeFile

引数 :

なし

戻り値 :

なし

機能 :

SHAKE ファイルを読み込み、SHAKE 拘束を行う原子を設定する。

SHAKE 情報には構成する原子の個数と拘束距離群があり、個数の対応を以下の表に示す。

SHAKE 指定数	キーワード	原子数	拘束距離数
2 原子 SHAKE	SHAKE> SHAKE	2	2
3 原子 SHAKE	SHAKE> SHAKE	3	3
4 原子 SHAKE	SHAKE> SHAKE	4	6

16.13. Input_CapFile : CAP 情報の読み込み

名称 : Input_CapFile

呼び出し形式 : subroutine Input_CapFile

引数 :

なし

戻り値 :

なし

機能 :

Cap 情報のファイルを入力する。

Cap 情報には対象チェーン、中心、セルサイズ、半径があり、各対応を以下の表に示す。

情報種別	キーワード	パラメータ
CAP 対象チェーン	BOUND> INCLUDE	分子名 先頭チェーン番号 最終チェーン番号 [リスト出力]
CAP 中心	チェーン	BOUND> CENTER
	原子	BOUND> CENTER
	座標	BOUND> CENTER
セルサイズ	BOUND> BOX	X 座標 (Å) y 座標 (Å) z 座標 (Å)
CAP 半径	BOUND> RADIUS	X 成分 (Å) y 成分 (Å) z 成分 (Å)

16.14. Input_PositionFile : 位置拘束ファイルの読み込み

名称 : Input_PositionFile

呼び出し形式 : subroutine Input_PositionFile

引数 :

なし

戻り値 :

なし

機能 :

位置拘束情報のファイルを入力する。

位置拘束では拘束する原子の指定にチェーン・残基・原子を指定するリスト形式と中心座標と半径を指定する境界指定の二種類があり、各対応を以下に示す。

情報種別	キーワード	パラメータ
位置拘束リスト	GROUP> LIST	チェーン先頭番号 チェーン最終番号 残基先頭番号 残基最終番号 原子名指定 残基名指定 推定誤差 mass-wait [リスト出力指定]
位置拘束境界	GROUP> RADIUS	中心チェーン番号 中心残基先頭番号 中心原子名 下限半径 上限 半径 推定誤差 mass-wait [リスト出力指定]

16.15. Input_BestfitFile : BESTFIT ファイルの読み込み

名称 : Input_BestfitFile

呼び出し形式 : subroutine Input_BestfitFile(unit, form, file, isFree)

引数 :

integer, intent(in)::unit	論理装置番号
integer, intent(in)::form	ファイル形式
character (NAME_SIZE), intent(in)::file	ファイル名
logical, intent(out), dimension(MAX_ATOM)::isFree	自由原子フラグ

戻り値 :

なし

機能 :

BEST-FIT 対象原子指定情報のファイルを入力する。

BEST-FIT 対象は自由原子であり、以下の指定により自由 | 固定原子を設定する。

情報種別		キーワード	パラメータ
原子リスト指定		SETBST> LIST	FIX FREE チェイン先頭番号 チェイン最終番号 残基先頭番号 残基最終番号 原子名指定 残基名指定 [リスト出力指定]
境界指定	中心原子指定	SETBST> RADIUS	FIX FREE ATOM 原子番号 半径下限 半径上限 原子名指定 [リスト出力]
	中心座標指定	SETBST> RADIUS	FIX FREE COOR x 座標 y 座標 z 座標 半径下限 半径上限 原子名指定 [リスト出力]

16.16. Input_DistanceFile : 距離拘束ファイルの読み込み

名称 : Input_DistanceFile

呼び出し形式 : subroutine Input_DistanceFile

引数 :

なし

戻り値 :

なし

機能 :

距離拘束情報のファイルを入力する。

距離拘束は2原子間の距離の拘束であり、2つの対象原子を指定するため、原子の所属チェーン、所属残基、原子名を指定する。

情報種別	キーワード	パラメータ
距離拘束指定	RDDSTC> LIST	原子指定1 原子指定2 下界推定誤差 上界推定誤差 下界距離 上界距離[リスト出力指定]

原子指定 ::= チェイン番号 残基番号 残基名称 原子名称

16.17. Input_DihedralFile : 二面角拘束ファイルの読み込み

名称 : Input_DihedralFile

呼び出し形式 : subroutine Input_DihedralFile

引数 :

なし

戻り値 :

なし

機能 :

二面各拘束情報のファイルを入力する。

二面各拘束を構成する4つの対象原子を指定するため、原子の相対位置あるいは原子の番号を指定する。

情報種別	キーワード	パラメータ
二面角リスト指定	CDIHE> LIST	所属チェーン番号 *原子指定1 原子指定2 原子指定3 原子指定4 下界推定誤差 上界推定誤差 下界角 上界角 [リスト出力]
二面角番号指定	CDIHE> NUMBER	原子番号1 原子番号2 原子番号3 原子番号4 下界推定誤差 上界推定誤差 下界角 上界角 [リスト出力]

*原子指定 ::= 残基番号 原子名

16.18. Input_MonitorFile : モニター対象ファイルの読み込み

名称 : Input_MonitorFile

呼び出し形式 : subroutine Input_MonitorFile

引数 :

なし

戻り値 :

なし

機能 :

モニター対象原子情報を入力する。

モニター対象は原子位置、原子距離、変角、二面角であり、各対応を以下に示す。

情報種別	キーワード	パラメータ
原子位置指定	MONI> COOR	原子指定1 [リスト出力]
距離指定	MONI> DIST	原子指定1 原子指定2 [リスト出力]
変角指定	MONI> ANGL	原子指定1 原子指定2 原子指定3 [リスト出力]
二面角指定	MONI> TORS	原子指定1 原子指定2 原子指定3 原子指定4 [リスト出力]

原子指定 ::= チェーン番号 残基番号 原子名

16.19. Input_MultiCanonicalFile : マルチカノニカルパラメータの読み込み

名称 : Input_MultiCanonicalFile

呼び出し形式 : subroutine Input_MultiCanonicalFile

引数 :

なし

戻り値 :

なし

機能 :

マルチカノニカルのパラメータファイルを入力する。

各アンサンブルおよびクラスタ数により、以下のように入力項目が決定される。

アンサンブル	NAMEP1
SELECTIVELY-CANONICAL	高温側温度
MULTI-CANONICAL	次数 係数(次数分) 下界係数 上界係数 下界値 上界値 温度
TSALLIS	次数 係数(次数分) 温度下界 温度上界 エネルギー下界 エネルギー上界 基底エネルギー 基底温度

16.20. Search_Molecule : 対象分子 ID の検索

名称 : Search_Molecule

呼び出し形式 : function Search_Molecule(line) result (molID)

引数 :

character(*), intent(in)::line

検索対象分子の名称

戻り値 :

integer::molID

検索対象の分子 ID

機能 :

入力した文字列に対応する分子の ID を返却する。

16.21. Input_RestartFile : リスタートファイルの読み込み

名称 : Input_RestartFile

呼び出し形式 : subroutine Input_RestartFile

引数 :

なし

戻り値 :

なし

機能 :

MD のリスタートファイルの読み込みを行う。

リスタートファイルはバイナリ形式で、以下の情報をもつ。

項番	内容	型	個数	入力条件
#1	タイトル	Character (*)	1	—
#2	原子数	integer	1	—
#3	自由原子数	integer	1	—
#4	MD ループ回数	integer	1	—
#5	シミュレート時間	real*8	1	—
#6	総エネルギー	real*8	1	—
#7	運動エネルギー	real*8	1	—
#8	ポテンシャルエネルギー	real*8	1	—
#9	原子位置	real*8	原子数*3	—
#10	原子速度	real*8	原子数*3	—
#11	原子への力	real*8	原子数*3	velocity-verlet 指定時
#12	Hybrid モデル初期座標	real*8	Hybrid 内原子数*3	Hybrid モデル指定時
#13	Hybrid モデル仮想座標	real*8	Hybrid 内原子数*3	Hybrid モデル指定時
#14	Hybrid モデル仮想速度	real*8	Hybrid 内原子数*3	Hybrid モデル指定時
#15	熱浴の仮想座標	real*8	分子数	NOSE-HOOVER 指定時
#16	熱浴の仮想速度	real*8	分子数	NOSE-HOOVER 指定時
#17	熱浴の仮想質量	real*8	分子数	NOSE-HOOVER 指定時
#18	剛体速度	real*8	剛体数*3	剛体モデル指定時
#19	剛体座標	real*8	剛体数*3	剛体モデル指定時
#20	剛体角	real*8	剛体数*3	剛体モデル指定時
#21	剛体角運動量	real*8	剛体数*3	剛体モデル指定時
#22	剛体への力	real*8	剛体数*3	剛体モデル指定時
#23	剛体へのトルク	real*8	剛体数*3	剛体モデル指定時
#24	剛体構成原子の相対位置	real*8	総剛体内原子数*3	剛体モデル指定時
#25	zeta 値	real*8	1	Tsallis 指定時
#26	V 値	real*8	1	Tsallis 指定時
#27	立方体セルのサイズ	real*8	3	Andersen 指定時
#28	斜方セルのセル行列	real*8	9	Parrinello-Rahman 指定時
#29	セル行列の逆行列	real*8	9	Parrinello-Rahman 指定時
#30	セル行列の行列式	real*8	1	Parrinello-Rahman 指定時
#31	Andersen 用のピストン	real*8	4	Andersen 指定時
#32	Andersen 用の系に対する熱浴	real*8	4	Andersen 指定時
#33	Andersen 用の volume に対する熱浴	real*8	4	Andersen 指定時
#34	Parrinello-Rahman 用のピストン	real*8	20	Parrinello-Rahman 指定時

#35	Parrinello-Rahman 用の系に対する熱浴	real*8	4	Parrinello-Rahman 指定時
#36	Parrinello-Rahman 用の volume に対する熱浴	real*8	4	Parrinello-Rahman 指定時
#37	ビリアル	real*8	9*MAX_POT	NPT 指定時

16.22. Input_EnergyFile : エネルギーファイルの読み込み

名称 : Input_EnergyFile

呼び出し形式 : function Input_EnergyFile(currentStep, currentTime, cpuTime, &
rmsf, vdw15Num, hyd15Num, rmsd, &
totalEnergy, kineticEnergy, temp, &
potentialEnergy) result (isEnd)

引数 :

integer, intent(out) :: currentStep	MD のステップ数
real*4, intent(out) :: currentTime	MD のシミュレート時間
real*4, intent(out) :: cpuTime	経過 CPU 時間
real*8, intent(out) :: rmsf	root-mean square distance
real*8, intent(out) :: vdw15Num	1-5 van der Waals 数
real*8, intent(out) :: hyd15Num	1-5 hydrogen bond 数
real*8, intent(out) :: rmsd	root-mean square distance
real*8, intent(out), dimension(:) :: potentialEnergy	ポテンシャルエネルギー
real*8, intent(out) :: totalEnergy	総エネルギー
real*8, intent(out) :: kineticEnergy	運動エネルギー
real*8, intent(out) :: temp	系全体の温度

戻り値 :

logical :: isEnd	入力ファイルの終了
------------------	-----------

機能 :

MINIMIZE, MD で作成したエネルギーファイルを読み込み、各パラメータを設定する。

名称：Input_EnergyAndAtomFile

引数：

戻り値:

機能：

名称 : Get_HeadString

引数：

戻り値:

機能：

- 48 -

16.25. Input_HybridFile : Hybrid ファイル読み込み制御

名称 : Input_HybridFile

呼び出し形式 : subroutine Input_HybridFile

引数 :

なし

戻り値 :

なし

機能 :

Hybrid モデルのファイル入力、Hybrid モデルの初期設定を行う。

16.26. Input_BinaryHybridFile : バイナリ形式の Hybrid モデルのファイル入力

名称 : Input_BinaryHybridFile

呼び出し形式 : subroutine Input_BinaryHybridFile(unti, file)

引数 :

integer, intent(in)::unit

論理装置番号

character (NAME_SIZE), intent(in)::file

Hybrid ファイル名

戻り値 :

なし

機能 :

Hybrid モデルのファイル入力を行う。

- 1) Hybrid モデル内原子数入力
- 2) Hybrid モデル内対象原子 ID 入力
- 3) Hybrid モデルばね定数入力
- 4) 原子位置計算用行列入力
- 5) 原子への力の計算用行列入力
- 6) Hybrid モデル振動子行列入力
- 7) 切り出し前の HD 部分の原子座標
- 8) 切り出し前の HD 部分の原子 ID
- 9) 切り出し後の HD 部分の原子座標
- 10) 切り出し後の HD 部分の原子 ID

16.27. Initialize_HybridModel : Hybrid モデル初期設定

名称 : Initialize_HybridModel

呼び出し形式 : subroutine Initialize_HybridModel(unti, file)

引数 :

なし

戻り値 :

なし

機能 :

Hybrid モデルの初期設定を行う。

- 1) 原子情報に Hybrid モデルフラグを設定する。
- 2) Hybrid 内の BOND をダミーの BOND とし、計算しないようにする。
- 3) Hybrid 内の ANGLE をダミーの ANGLE とし、計算しないようにする。
- 4) Hybrid 内の TORSION をダミーの TORSION とし、計算しないようにする。
- 5) Hybrid 内の IMPROPER をダミーの IMPROPER とし、計算しないようにする。
- 6) Hybrid モデルの行列と原子 ID の変換用リストを作成する。
- 7) Hybrid モデル内原子の初期座標を登録する。
- 8) Hybrid モデルの重心と質量を計算する。

16.28. Input_RigidFile : 剛体モデル入力

名称 : Input_RigidFile

呼び出し形式 : subroutine Input_RigidFile

引数 :

なし

戻り値 :

なし

機能 :

剛体ファイルを読み込み、剛体モデルの初期設定を呼び出す。

情報種別	キーワード	パラメータ
番号指定	RIGID> NUM	分子名 原子数 原子 ID...
番号+モデル指定	RIGID> COO	分子名 原子数 [原子 ID x座標 y座標 z座標]...

16.29. Init_RigidModel : 剛体モデル初期設定

名称 : Init_RigidModel

呼び出し形式 : subroutine Init_RigidModel

引数 :

type(RigidInfo_t), intent(inout)::rigidInfo	剛体情報
integer, intent(in)::rigidID	当該剛体の ID
type(Atom_t), intent(inout), dimension(MAX_ATOM)::atom	原子情報
integer, intent(in)::elmNum	剛体内原子数
type(Chain_t), intent(in)::chain	チェーン(分子)情報
integer, dimension(elmNum)::list	剛体内原子の相対 ID
real*8, intent(inout), dimension(3, elmNum)::cord	原子の座標
real*8, intent(inout), dimension(3, elmNum)::model	モデルとなる原子の座標
real*8, intent(in), dimension(elmNum)::mass	剛体内の原子の質量
integer, intent(inout)::index	剛体内原子位置のインデックス

戻り値 :

なし

機能 :

剛体モデルの原子リスト、重心座標、剛体角、慣性モーメント、剛体内原子の相対位置(剛体重心を原点とし、剛体角が0のときの各原子の位置)を求める。

- 1) 構成原子の原子 ID を剛体モデル情報に登録する。
- 2) 剛体の重心を計算し、剛体モデル情報に登録する
- 3) 剛体のテンソルを計算する。
- 4) テンソルの固有値を計算する。
- 5) 固有値から剛体角を計算し、登録する。
- 6) 剛体内原子の相対位置を計算、剛体モデル情報に登録する。
- 7) 慣性モーメントを計算し、剛体モデル情報に登録する

16.30. Remove_InteractionInRigid : 剛体内の結合相互作用削除

名称 : Remove_InteractionInRigid

呼び出し形式 : subroutine Remove_InteractionInRigid

引数 :

なし

戻り値 :

なし

機能 :

結合相互作用リストから同一剛体内に存在するものを削除する。

16.31. Input_FlowForceFile : FlowForce ファイルの入力

名称 : Input_FlowForceFile

呼び出し形式 : subroutine Input_FlowForceFile

引数 :

なし

戻り値 :

なし

機能 :

FlowForce ファイルから、FlowForce のパラメータを読み込む。

FlowForce ファイルの内容は「入出力ファイル仕様書」を参照。

16.32. Input_UmbrellaFile : Filled-Potential ファイルの入力

名称 : Input_UmbrellaFile

呼び出し形式 : subroutine Input_UmbrellaFile

引数 :

なし

戻り値 :

なし

機能 :

Filled-Potential ファイルから、Filled-Potential のパラメータを読み込む。

Filled-Potential ファイルの内容は「入出力ファイル仕様書」を参照。

16.33. Remove_DummyBond : 計算しない BOND の削除

名称 : Remove_DummyBond

呼び出し形式 : subroutine Remove_DummyBond(isDummy, validNum)

引数 :

logical, intent(inout), dimension(MAX_BOND)::isDummy ダミーを示すフラグ

integer, intent(inout)::validNum 有効な BOND の個数

戻り値 :

なし

機能 :

入力したフラグに従い、有効な BOND を Interact_Instance のリストの先頭に集め、無効な BOND をリストの最後に集める。。

有効な BOND 数は引数に設定する。

16.34. Remove_DummyAngle : 計算しない ANGLE の削除

名称 : Remove_DummyAngle

呼び出し形式 : subroutine Remove_DummyAngle (isDummy, validNum)

引数 :

logical, intent(inout), dimension(MAX_ANGLE)::isDummy	ダミーを示すフラグ
integer, intent(inout)::validNum	有効な BOND の個数

戻り値 :

なし

機能 :

入力したフラグに従い、有効な ANGLE を Interact_Instance のリストの先頭に集め、無効な ANGLE をリストの最後に集める。

有効な ANGLE 数は引数に設定する。

16.35. Remove_DummyTorsion : 計算しない TORSION の削除

名称 : Remove_DummyTorsion

呼び出し形式 : subroutine Remove_DummyTorsion (isDummy, validNum)

引数 :

logical, intent(inout), dimension(MAX_TORSION)::isDummy	ダミーを示すフラグ
integer, intent(inout)::validNum	有効な TORSION の個数

戻り値 :

なし

機能 :

入力したフラグに従い、有効な TORSION を Interact_Instance のリストの先頭に集め、無効な TORSION をリストの最後に集める。。

有効な TORSION 数は引数に設定する。

16.36. Remove_DummyImproper : 計算しない IMPROPER の削除

名称 : Remove_DummyImproper

呼び出し形式 : subroutine Remove_DummyImprope (isDummy, validNum)

引数 :

logical, intent(inout), dimension(MAX_IMPROPER)::isDummy	ダミーを示すフラグ
integer, intent(inout)::validNum	有効な IMPROPER の個数

戻り値 :

なし

機能 :

入力したフラグに従い、有効な IMPROPER を Interact_Instance のリストの先頭に集め、無効な IMPROPER をリストの最後に集める。。

有効な IMPROPER 数は引数に設定する。

16.37. Input_ExtCapFile : 拡張 CAP 拘束データの読み込み

名称 : Input_ExtCapFile

呼び出し形式 : subroutine Input_ExtCapFile

引数 :

なし

戻り値 :

なし

機能 :

拡張 CAP 拘束のデータをファイルから読み込む。

16.38. Input_RangeFile : 範囲付き位置拘束データの読み込み

名称 : Input_RangeFile

呼び出し形式 : subroutine Input_RangeFile

引数 :

なし

戻り値 :

なし

機能 :

範囲付き位置拘束のデータをファイルから読み込む。

16.39. Input_GeneralizedBornFile : GB/ASA データの読み込み

名称 : Input_GeneralizedBornFile

呼び出し形式 : subroutine Input_GeneralizedBornFile

引数 :

なし

戻り値 :

なし

機能 :

GB/ASA のデータをファイルから読み込む。

16.40. Input_SatFile : SAT データの読み込み

名称 : Input_SatFile

呼び出し形式 : subroutine Input_SatFile(chainInfo)

引数 :

type(ChainInfo_t), intent(in)::chainInfo

戻り値 :

なし

機能 :

SAT のデータをファイルから読み込む。

16.41. Input_RdcFile : RDC データの読み込み

名称 : Input_RdcFile

呼び出し形式 : subroutine Input_RdcFile

引数 :

なし

戻り値 :

なし

機能 :

RDC のデータをファイルから読み込む。

16.42. Input_Noefile : NOE データの読み込み

名称 : Input_Noefile

呼び出し形式 : subroutine Input_Noefile

引数 :

なし

戻り値 :

なし

機能 :

NOE のデータをファイルから読み込む。

16.43. Input_TplScalingFile : スケーリングデータの読み込み

名称 : Input_TplScalingFile

呼び出し形式 : subroutine Input_TplScalingFile(atomInfo)

引数 :

type (AtomInfo_t), intent(inout)::atomInfo

原子情報

戻り値 :

なし

機能 :

vdW パラメータおよび電荷のスケーリング対象原子およびスケーリングファクターを
ファイルから読み込む。**16.44. Init_FunkKind : 関数種別行列の初期化**

名称 : Init_FunkKind

呼び出し形式 : subroutine Init_FunkKind(num, method, kind)

引数 :

integer, intent(in)::num

原子種別数

integer, intent(in)::method

相互作用計算方法

integer, dimension (MAX_TYPE, MAX_TYPE), intent(in)::kind

関数種別行列

戻り値 :

なし

機能 :

関数種別行列の初期化を行う。

16.45. Set_AutoShake : SHAKE グループの自動解析

名称 : Set_AutoShake

呼び出し形式 : subroutine Set_AutoShake(obj, intr, dyn)

引数 :

type (Object_t), intent(inout)::obj

オブジェクト情報

type (Interact_t), intent(inout)::intr

相互作用情報

type (Dynamics_t), intent(inout)::dyn

MD 情報

戻り値 :

なし

機能 :

水素原子群と他の原子、水分子を構成する原子群をグループ化し、SHAKE 情報を生成する。
また、出力オプションが指定された場合は、指定されたファイルに解析情報を出力する。

16.46. Set_AutoRigid : RIGID の自動解析

名称 : Set_AutoShake

呼び出し形式 : subroutine Set_AutoShake(obj, intr)

引数 :

type(Object_t), intent(inout)::obj

オブジェクト情報

type(Interact_t), intent(inout)::intr

相互作用情報

戻り値 :

なし

機能 :

水素原子群と他の原子、水分子を構成する原子群をグループ化し、剛体情報を生成する。
また、出力オプションが指定された場合は、指定されたファイルに解析情報を出力する。

16.47. Input_MV0File : MV0 データの読み込み

名称 : Input_MV0File

呼び出し形式 : subroutine Input_MV0File(MV0)

引数 :

type(MV0_t), intent(inout)::MV0

MV0 情報

戻り値 :

なし

機能 :

MV0 対象チェイン番号および対象チェインの座標移動の有無をファイルから読み込み、固定チェインの所属原子を固定原子に設定する。

17. Input_Common

ファイル名 : Input_Common. f90

17.1. Input_CommonFile : cosgene 用共通ファイルの読み込み

名称 : Input_CommonFile

呼び出し形式 : subroutine Input_CommonFile

引数 :

なし

戻り値 :

なし

機能 :

cosgene の各フェーズ (MINIMIZE, MD, ANALYZE) 共通入力ファイルを読み込む。
共通入力ファイルを以下に示す。

項番	ファイル名
#1	トポロジーファイル
#2	コーディネートファイル
#3	自由/固定原子ファイル
#4	SHAKE ファイル
#5	CAP ファイル
#6	位置拘束ファイル
#7	距離拘束ファイル
#8	二面角拘束ファイル

17.2. Calc_InteractNumber : 相互作用数の計算

名称 : Calc_InteractNumber

呼び出し形式 : subroutine Calc_InteractNumber

引数 :

なし

戻り値 :

なし

機能 :

系全体の電荷および相互作用数を計算する。

18. Input_Option

ファイル名 : Input_Option.f90

18.1. Decide_Phase : 当該フェーズの決定

名称 : Decide_Phase

呼び出し形式 : function Decide_Phase () result (phase)

引数 :

なし

戻り値 :

integer::phase

フェーズ ID

機能 :

制御ファイルを読み込み、当該フェーズを決定する。

18.2. Input_InputOption : INPUT フェーズのオプション入力

名称 : Input_InputOption

呼び出し形式 : subroutine Input_InputOption

引数 :

なし

戻り値 :

なし

機能 :

INPUT フェーズのオプションを読み込む。

キーワードは種別に対応したテーブルで管理され、このテーブルに対応したワーク変数を介して入力される。

ワーク変数は Init_Value で初期化され、Set_Value でワーク変数から実体に定義する。

項番	内容	型	ワーク変数
#1	選択型キーワード	Select_t(キーワード、選択肢数、選択肢群)	sopt
#2	整数型キーワード	character*6	iopt
#3	実数型キーワード	character*6	ropt
#4	文字型キーワード	character*80	copt

18.3. Input_OutputOption : OUTPUT フェーズのオプション入力

名称 : Input_OutputOption

呼び出し形式 : subroutine Input_OutputOption

引数 :

なし

戻り値 :

なし

機能 :

OUTPUT フェーズのオプションを読み込む。

キーワードは種別に対応したテーブルで管理され、このテーブルに対応したワーク変数を介して入力される。

ワーク変数は Init_Value で初期化され、Set_Value でワーク変数から実体に定義する。

項番	内容	型	ワーク変数
#1	選択型キーワード	Select_t(キーワード、選択肢数、選択肢群)	sopt
#2	整数型キーワード	character*6	iopt
#3	実数型キーワード	character*6	ropt
#4	文字型キーワード	character*80	copt

18.4. Input_MinimizeOption : MINIMIZE フェーズのオプション入力

名称 : Input_MinimizeOption

呼び出し形式 : subroutine Input_MinimizeOption

引数 :

なし

戻り値 :

なし

機能 :

MINIMIZE フェーズのオプションを読み込む。

キーワードは種別に対応したテーブルで管理され、このテーブルに対応したワーク変数を介して入力される。

ワーク変数は Init_Value で初期化され、Set_Value でワーク変数から実体に定義する。

項番	内容	型	ワーク変数
#1	選択型キーワード	Select_t(キーワード、選択肢数、選択肢群)	sopt
#2	整数型キーワード	character*6	iopt
#3	実数型キーワード	character*6	ropt
#4	文字型キーワード	character*80	copt

18.5. Input_MDOption : MD フェーズのオプション入力

名称 : Input_MDOption

呼び出し形式 : subroutine Input_MDOption

引数 :

なし

戻り値 :

なし

機能 :

MD フェーズのオプションを読み込む。

キーワードは種別に対応したテーブルで管理され、このテーブルに対応したワーク変数を介して入力される。

ワーク変数は Init_Value で初期化され、Set_Value でワーク変数から実体に定義する。

項番	内容	型	ワーク変数
#1	選択型キーワード	Select_t(キーワード、選択肢数、選択肢群)	sopt
#2	整数型キーワード	character*6	iopt
#3	実数型キーワード	character*6	ropt
#4	文字型キーワード	character*80	copt

18.6. Input_AnalyzeOption : ANALYZE フェーズのオプション入力

名称 : Input_AnalyzeOption

呼び出し形式 : subroutine Input_AnalyzeOption

引数 :

なし

戻り値 :

なし

機能 :

機能 :

ANALYZE フェーズのオプションを読み込む。

キーワードは種別に対応したテーブルで管理され、このテーブルに対応したワーク変数を介して入力される。

ワーク変数は Init_Value で初期化され、Set_Value でワーク変数から実体に定義する。

項番	内容	型	ワーク変数
#1	選択型キーワード	Select_t(キーワード、選択肢数、選択肢群)	sopt
#2	整数型キーワード	character*6	iopt
#3	実数型キーワード	character*6	ropt
#4	文字型キーワード	character*80	copt

18.10. Get_RightSideAssignment : 右辺値文字列の取得

名称 : Get_RightSideAssignment

呼び出し形式 : function Get_RightSideAssignment(line) result (right)

引数 :

character (NAME_SIZE), intent(in)::line 読み込み行

戻り値 :

character (NAME_SIZE)::right 右辺文字列

機能 :

読み込み行の右辺を取り出す。

18.11. Is_LineEnd : 行終了チェック

名称 : Is_LineEnd

呼び出し形式 : function Is_LineEnd(line) result (isEnd)

引数 :

character (NAME_SIZE), intent(in)::line 読み込み行

戻り値 :

logical::isEnd 行の終了

機能 :

行の終了をチェックする。

18.12. Error_Line : 読み込み行のエラー出力

名称 : Error_Line

呼び出し形式 : subroutine Error_Line(line, position, message)

引数 :

character (NAME_SIZE), intent(in)::line	読み込み行
integer, intent(in)::position	エラー位置
character (*), intent(in)::message	エラーの内容

戻り値 :

なし

機能 :

読み込み行のエラーを標準出力に出力する。

18.13. Get_Value_I4 : 整数型キーワード値の取得

名称 : Get_Value_I4

呼び出し形式 : function Get_Value_I4(iopt, ioption, numStr) result (value)

引数 :

integer, dimension(:), intent(in)::iopt	整数型キーワード値
character (TOKEN_SIZE), intent(in)::numStr	入力キーワード
character (TOKEN_SIZE), dimension(:), intent(in)::ioption	キーワード配列

戻り値 :

integer::value キーワード値

機能 :

入力キーワードに対する整数キーワードの値を得る。

18.14. Get_Value_C : 文字型キーワード値の取得

名称 : Get_Value_C

呼び出し形式 : function Get_Value_C(copt, coption, numStr) result (value)

引数 :

character (TOKEN_SIZE), dimension(:), intent(in)::copt	文字型キーワード値
character (TOKEN_SIZE), intent(in)::numStr	入力キーワード
character (TOKEN_SIZE), dimension(:), intent(in)::coption	キーワード配列

戻り値 :

character (NAME_SIZE)::value	キーワード値
------------------------------	--------

機能 :

入力キーワードに対する文字キーワードの値を得る。

18.15. Get_Value_R8 : 実数型キーワード値の取得

名称 : Get_Value_R8

呼び出し形式 : function Get_Value_R8(ropt, roption, numStr) result (value)

引数 :

real*8, dimension(:), intent(in)::ropt	実数型キーワード値
character (TOKEN_SIZE), intent(in)::numStr	入力キーワード
character (TOKEN_SIZE), dimension(:), intent(in)::roption	キーワード配列

戻り値 :

real*8::value	キーワード値
---------------	--------

機能 :

入力キーワードに対する実数型キーワードの値を得る。

18.16. Get_Value_Select : 選択型キーワード値の取得

名称 : Get_Value_Select

呼び出し形式 : function Get_Value_Select(sopt, soption, numStr) result (value)

引数 :

integer, dimension(:), intent(in)::sopt	選択型キーワード値
character (TOKEN_SIZE), intent(in)::numStr	入力キーワード
type(Select_t), dimension(:), intent(in)::soption	選択型キーワードテーブル

戻り値 :

integer::value	キーワード値
----------------	--------

機能 :

入力キーワードに対する選択型キーワードの値を得る。

18.17. Init_Value_I4 : 整数型キーワードの初期設定

名称 : Init_Value_I4

呼び出し形式 : subroutine Init_Value_I4(iopt, ioption, numStr, value)

引数 :

integer, dimension(:), intent(out)::iopt	整数型キーワード値
character (TOKEN_SIZE), intent(in)::numStr	入力キーワード
character (TOKEN_SIZE), dimension(:), intent(in)::ioption	キーワード配列
integer, intent(in)::value	設定値

戻り値 :

なし

機能 :

入力キーワードに対応する整数型キーワードの値を初期化する。

18.18. Init_Value_C : 文字型キーワード値の初期設定

名称 : Init_Value_C

呼び出し形式 : subroutine Init_Value_C(copt, coption, numStr, value)

引数 :

character (NAME_SIZE), dimension(:), intent(out)::copt	文字型キーワード値
character (TOKEN_SIZE), intent(in)::numStr	入力キーワード
character (TOKEN_SIZE), dimension(:), intent(in)::coption	キーワード配列
character(*), intent(in)::value	設定値

戻り値 :

なし

機能 :

入力キーワードに対応する文字型キーワードの値を初期化する。

18.19. Init_Value_R8 : 実数型キーワード値の初期設定

名称 : Init_Value_R8

呼び出し形式 : subroutine Init_Value_R8(ropt, roption, numStr, value)

引数 :

real*8, dimension(:), intent(out)::ropt	実数型キーワード値
character (TOKEN_SIZE), intent(in)::numStr	入力キーワード
character (TOKEN_SIZE), dimension(:), intent(in)::roption	キーワード配列
real*8, intent(in)::value	設定値

戻り値 :

なし

機能 :

入力キーワードに対応する実数型キーワードの値を初期化する。

18.20. Init_Value_Select_Char : 選択型キーワード値の初期設定

名称 : Init_Value_Select_Char

呼び出し形式 : subroutine Init_Value_Select_Char(sopt, soption, numStr, char)

引数 :

integer, dimension(:), intent(out)::sopt	選択型キーワード値
character (TOKEN_SIZE), intent(in)::numStr	入力キーワード
type(Select_t), dimension(:), intent(in)::soption	選択キーワードテーブル
character (KIND_SIZE), intent(in)::char	設定文字

戻り値 :

なし

機能 :

入力キーワードに対応する選択型キーワードの値を初期化する。

18.21. Init_Value_Select_Val : 選択型キーワード値の初期設定

名称 : Init_Value_Select_Val

呼び出し形式 : subroutine Init_Value_Select_Val(sopt, soption, numStr, value)

引数 :

integer, dimension(:), intent(out)::sopt	選択型キーワード値
character (TOKEN_SIZE), intent(in)::numStr	入力キーワード
type(Select_t), dimension(:), intent(in)::soption	選択キーワードテーブル
integer, intent(in)::value	設定値

戻り値 :

なし

機能 :

入力キーワードに対応する選択型キーワードの値を初期化する。

18.22. Init_Value_Select_Logical : 選択型キーワード値の初期設定

名称 : Init_Value_Select_Logical

呼び出し形式 : subroutine Init_Value_Select_Logical(sopt, soption, numStr, value)

引数 :

integer, dimension(:), intent(out)::sopt	選択型キーワード値
character (TOKEN_SIZE), intent(in)::numStr	入力キーワード
type(Select_t), dimension(:), intent(in)::soption	選択キーワードテーブル
logical, intent(in)::value	設定値

戻り値 :

なし

機能 :

入力キーワードに対応する選択型キーワードの値を初期化する。

注意点 :

本手続きの設定値は論理型である。内部的には 1 が偽を示し、2 が真を示す。

18.23. Set_Value_I4 : 整数型キーワードの設定

名称 : Set_Value_I4

呼び出し形式 : subroutine Set_Value_I4(iopt, ioption, numStr, object)

引数 :

integer, dimension(:), intent(in)::iopt	整数型キーワード値
character (TOKEN_SIZE), intent(in)::numStr	入力キーワード
character (TOKEN_SIZE), dimension(:), intent(in)::ioption	キーワード配列
integer, intent(out)::object	実体

戻り値 :

なし

機能 :

入力キーワードに対応するキーワード値を実体に代入する。

18.24. Set_Value_C : 文字型キーワードの設定

名称 : Set_Value_C

呼び出し形式 : subroutine Set_Value_C(copt, coption, numStr, object)

引数 :

character (NAME_SIZE), dimension(:), intent(in)::copt	文字型キーワード値
character (TOKEN_SIZE), intent(in)::numStr	入力キーワード
character (TOKEN_SIZE), dimension(:), intent(in)::coption	キーワード配列
character (*), intent(out)::object	実体

戻り値 :

なし

機能 :

入力キーワードに対応するキーワード値を実体に代入する。

18.25. Set_Value_R8 : 実数型キーワードの設定

名称 : Set_Value_R8

呼び出し形式 : subroutine Set_Value_R8(ropt, roption, numStr, object)

引数 :

real*8, dimension(:), intent(in)::ropt	実数型キーワード値
character (TOKEN_SIZE), intent(in)::numStr	入力キーワード
character (TOKEN_SIZE), dimension(:), intent(in)::roption	キーワード配列
real*8, intent(out)::object	実体

戻り値 :

なし

機能 :

入力キーワードに対応するキーワード値を実体に代入する。

18.26. Set_Value_Select : 選択型キーワードの設定

名称 : Set_Value_Select

呼び出し形式 : subroutine Set_Value_Select(sopt, soption, numStr, object)

引数 :

integer, dimension(:), intent(in)::sopt	選択型キーワード値
character (TOKEN_SIZE), intent(in)::numStr	入力キーワード
type(Select_t), dimension(:), intent(in)::soption	選択型キーワードテーブル
integer, intent(out)::object	実体

戻り値 :

なし

機能 :

入力キーワードに対応するキーワード値を実体に代入する。

18.27. Set_Value_Select_Logical : 選択型キーワードの設定

名称 : Set_Value_Select_Logical

呼び出し形式 : subroutine Set_Value_Select_Logical(sopt, soption, numStr, object)

引数 :

integer, dimension(:), intent(in)::sopt	選択型キーワード値
character (TOKEN_SIZE), intent(in)::numStr	入力キーワード
type(Select_t), dimension(:), intent(in)::soption	選択型キーワードテーブル
logical, intent(out)::object	実体

戻り値 :

なし

機能 :

入力キーワードに対応するキーワード値を実体に代入する。

18.28. Chekc_Option : キーワード重複チェック

名称 : Check_Option

呼び出し形式 : subroutine Check_Option(kind, option)

引数 :

character (4), intent(in)::kind	キーワード種別文字 ('real' 'int' 'char')
character (TOKEN_SIZE), dimension(:), intent(in)::option	キーワード配列

戻り値 :

なし

機能 :

同一種類のキーワードに重複がないかをチェックする。

18.29. Check_SelectOption : 選択型キーワードのチェック

名称 : Check_SelectOption

呼び出し形式 : subroutine Check_SelectOption(soption)

引数 :

type(Select_t), dimension(:), intent(in)::soption

選択型キーワードテーブル

戻り値 :

なし

機能 :

選択型キーワードテーブルの以下のチェックを行う。

- (1) 選択型キーワード内でのキーワードの重複
- (2) 選択肢の個数(2以上8以下)
- (3) 選択肢の重複
- (4) 選択肢の個数と文字列の一致

18.30. Check_DuplicateOption : 重複オプションのチェック

名称 : Check_DuplicateOption

呼び出し形式 : subroutine Check_DuplicateOption(ioption, soption, roption, coption)

引数 :

character(TOKEN_SIZE), dimension(:), intent(in)::ioption

整数型キーワード配列

type(Select_t), dimension(:), intent(in)::soption

選択型キーワードテーブル

character(TOKEN_SIZE), dimension(:), intent(in)::roption

実数型キーワード配列

character(TOKEN_SIZE), dimension(:), intent(in)::coption

文字型キーワード配列

戻り値 :

なし

機能 :

異なる種別のキーワードに重複がないかチェックする。

19. Interact_Method

本メソッドはインライン展開により、削除する。

19.1. Get_InteractType

本手続きはインライン展開により、削除する。

20. InteractTable_Method

ファイル名 : InteractTable.f90

20.1. Get_InteractPairList : 1-5 相互作用リスト作成 (van der Waals + hydrogen bond)

名称 : Get_InteractPairList

呼び出し形式 : subroutine Get_InteractPairList(atomID, vdwList, vdwNum, hydList, hydNum, & isFreeAtom, type, funcKind)

引数 :

integer, intent(in)::atomID	当該原子 ID
integer, dimension(MAX_ATOM), intent(out)::vdwList	1-5 van der Waals のリスト
integer, intent(out)::vdwNum	1-5 van der Waals 数
integer, dimension(MAX_ATOM), intent(out)::hydList	水素結合のリスト
integer, intent(out)::hydNum	水素結合数
logical, dimension(MAX_ATOM)::isFreeAtom	自由原子フラグ
integer, dimension(MAX_ATOM)::type	原子の型
integer, dimension(MAX_TYPE, MAX_TYPE)::funcKind	原子間の関数種別

戻り値 :

なし

注意 :

本機能は cut-off なしの場合にのみ使用される。

機能 :

当該原子にかかる 1-5 相互作用の原子リストを作成する。

20.2. Get_VdwInteractList : 1-5 相互作用リスト作成 (van der Waals のみ)

名称 : Get_VdwInteractList

呼び出し形式 : subroutine Get_VdwInteractList(atomID, vdwList, vdwNum)

引数 :

integer, intent(in)::atomID	当該原子 ID
integer, dimension(MAX_ATOM), intent(out)::vdwList	1-5 van der Waals のリスト
integer, intent(out)::vdwNum	1-5 van der Waals 数

戻り値 :

なし

注意 :

本機能は cut-off なしの場合にのみ使用される。

機能 :

当該原子にかかる 1-5 の van der Waals 相互作用の原子リストを作成する。

20.3. Get_HydInteractList : 1-5 相互作用リスト作成 (hydrogen bond のみ)

名称 : Get_HydInteractList

呼び出し形式 : subroutine Get_HydInteractList(atomID, hydList, hydNum)

引数 :

integer, intent(in)::atomID	当該原子 ID
integer, dimension(MAX_ATOM), intent(out)::hydList	水素結合のリスト
integer, intent(out)::hydNum	水素結合数

戻り値 :

なし

注意 :

本機能は cut-off なしの場合にのみ使用される。

機能 :

当該原子にかかる 1-5 の水素結合相互作用の原子リストを作成する。

20.4. Get_EleInteractList : 1-5 静電相互作用リスト作成

名称 : Get_EleInteractList

呼び出し形式 : subroutine Get_EleInteractList(atomID, eleList, eleNum)

引数 :

integer, intent(in)::atomID

当該原子 ID

integer, dimension(MAX_ATOM), intent(out)::eleList

静電相互作用相手のリスト

integer, intent(out)::eleNum

静電相互作用数

戻り値 :

なし

注意 :

本機能は cut-off なしの場合にのみ使用される。

機能 :

当該原子にかかる 1-5 の静電相互作用の原子リストを作成する。

20.5. Update_InteractTable : 相互作用テーブルの更新

名称 : Update_InteractTable

呼び出し形式 : subroutine Update_InteractTable(outLog)

引数 :

logical, intent(in)::outLog

ログの出力

戻り値 :

なし

機能 :

相互作用テーブルを更新する。

20.6. Create_AmberInteractPairList : Amber 対応相互作用リスト作成

名称 : Create_AmberInteractPairList

呼び出し形式 : subroutine Create_AmberInteractPairList (atomID, &
vdwList, vdwNum, &
hydList, hydNum, &
Atom, type, &
isFreeAtom, funcKind, &
num)

引数 :

integer, intent(in)::atomID	当該原子 ID
integer, dimension(MAX_ATOM), intent(out)::vdwList	van der Waals 原子リスト
integer, dimension(MAX_ATOM), intent(out)::hydList	水素結合原子リスト
integer, intent(out)::vdwNum	van der Waals 原子数
integer, intent(out)::hydNum	水素結合原子数
integer, intent(in)::num	全原子数
type (Atom_t), dimension(num), intent(in)::Atom	原子情報
integer, dimension(MAX_ATOM)::type	原子の型
logical, dimension(MAX_ATOM)::isFreeAtom	自由原子フラグ
integer, dimension(MAX_TYPE, MAX_TYPE)::funcKind	原子間の関数種別

戻り値 :

なし

機能 :

AMBER 向けの相互作用リストを作成する。

20.7. Create_OplsInteractList : OPLS 対応相互作用リスト作成

名称 : Create_OplsInteractList

呼び出し形式 : subroutine Create_OplsInteractList (atomID, &
vdwList, vdwNum, eleList, eleNum, &
hydList, hydNum, Atom, num)

引数 :

integer, intent(in)::atomID	当該原子 ID
integer, dimension(MAX_ATOM), intent(out)::vdwList	van der Waals 原子リスト
integer, dimension(MAX_ATOM), intent(out)::hydList	水素結合原子リスト
integer, intent(out)::vdwNum	van der Waals 原子数
integer, intent(out)::hydNum	水素結合原子数
integer, intent(in)::num	全原子数
type (Atom_t), dimension(num), intent(in)::Atom	原子情報

戻り値 :

なし

機能 :

OPLS 向けの相互作用リストを作成する。(OPLS では水素結合は計算しない)

20.8. Update_AtomBaseCutoff : 原子距離カットオフでの相互作用テーブル作成

名称 : Update_AtomBaseCutoff

呼び出し形式 : subroutine Update_AtomBaseCutoff (Atom, NonBonded, num)

引数 :

type (Atom_t), intent (inout), dimension (MAX_ATOM) :: Atom	原子情報
type (NonBondedInfo_t), intent (out) :: NonBonded	非結合相互作用情報
integer, intent (in) :: num	原子数

戻り値 :

なし

機能 :

原子間の距離に対するカットオフを行い、相互参照テーブルを更新する。

登録する条件は以下の全てを満たす場合である。

- (a) 原子の中心座標が当該原子の中心座標に対し、カットオフ半径内に存在する。
- (b) (a) の原子は当該原子と 1-5 以上離れている。
- (c) (a) の原子 ID は当該原子の ID よりも大きい。

20.9. Update_AtomBaseCutoff : 残基中心距離カットオフでの相互作用テーブル作成

名称 : Update_ResidueCenterCutoff

呼び出し形式 : subroutine Update_ResidueCenterCutoff (Atom, NonBonded, Resid, &
atomNum, residNum)

引数 :

type (Atom_t), intent (inout), dimension (MAX_ATOM) :: Atom	原子情報
type (NonBondedInfo_t), intent (inout) :: NonBonded	非結合相互作用情報
type (Resid_t), intent (in), dimension (MAX_RESID) :: Resid	残基情報
integer, intent (in) :: atomNum	原子数
integer, intent (in) :: residNum	残基数

戻り値 :

なし

機能 :

残基中心間に対するカットオフを行い、相互参照テーブルを更新する。

登録する条件は以下の全てを満たす場合である。

- (a) 原子の属する残基の重心座標と当該原子の属する残基の重心座標がカットオフ半径内に存在する。
- (b) (a) の原子は当該原子と 1-5 以上離れている。
- (c) (a) の原子 ID は当該原子の ID よりも大きい。

20.10. Register_ResidueAtoms : 残基内原子の相互作用テーブル登録

名称 : Register_ResidueAtoms

呼び出し形式 : subroutine Register_ResidueAtoms (atomID, resid, interact, &
replicaIndex, index, kind, tableMax)

引数 :

integer, intent (in) :: atomID	原子 ID
type (Resid_t), intent (in) :: resid	残基情報
type (Interact15_t), intent (out) :: interact	相互作用リスト
integer, intent (in), dimension (3) :: replicaIndex	レプリカセル位置
integer, intent (inout) :: index	相互作用インデックス
integer, intent (in) :: kind	当該原子の型
integer, intent (in) :: tableMax	相互作用リスト数の上限値

戻り値 :

なし

機能 :

当該原子と相互作用を計算する残基内の原子を相互作用リストに登録する。

また、この際にレプリカセルの位置を設定する。

20. 11. Update_ResidueSurfaceCutoff : 残基表面距離カットオフでの相互作用テーブル作成

名称 : Update_ResidueSurfaceCutoff

呼び出し形式 : subroutine Update_ResidueSurfaceCutoff (Atom, NonBonded, Resid, &
atomNum, residNum)

引数 :

type (Atom_t), intent (inout), dimension (MAX_ATOM) :: Atom	aggregate of atom
type (NonBondedInfo_t), intent (inout) :: NonBonded	aggregate of non-bond
type (Resid_t), intent (in), dimension (MAX_RESID) :: Resid	aggregate of resid
integer, intent (in) :: atomNum	count of atoms
integer, intent (in) :: residNum	count of residues

戻り値 :

なし

機能 :

アクセシビリサーファスに対するカットオフを行い、相互参照テーブルを更新する。

登録する条件は以下の全てを満たす場合である。

- (a) 原子の属する残基の全原子と当該原子の属する残基の全原子との組合せのいずれかがカットオフ半径内に存在する。
- (b) (a) の原子は当該原子と 1-5 以上離れている。
- (c) (a) の原子 ID は当該原子の ID よりも大きい。

20. 12. Register_NearSurfaceAtoms : 残基内原子の相互作用テーブル登録

名称 : Register_NearSurfaceAtoms

呼び出し形式 : subroutine Register_NearSurfaceAtoms (atomID, resid, interact, &
replicaIndex, index, kind, tableMax)

引数 :

引数 :

integer, intent (in) :: atomID	原子 ID
type (Resid_t), intent (in) :: resid	残基情報
type (Interact15_t), intent (out) :: interact	相互作用リスト
integer, intent (in), dimension (3) :: replicaIndex	レプリカセル位置
integer, intent (inout) :: index	相互作用インデックス
integer, intent (in) :: kind	当該原子の型
integer, intent (in) :: tableMax	相互作用リスト数の上限値

戻り値 :

なし

機能 :

当該原子と相互作用を計算する残基内の原子を相互作用リストに登録する。

また、この際にレプリカセルの位置を設定する。

20.13. Reset_FlagFixAtomOrBonded : 相互作用計算候補フラグの設定

名称 : Reset_FlagFixAtomOrBonded

呼び出し形式 : subroutine Reset_FlagFixAtomOrBonded(atomID, init, last, isCalc)

引数 :

integer, intent(in)::atomID	当該原子 ID
integer, intent(in)::init	相互作用候補原子の先頭
integer, intent(in)::last	相互作用候補原子の最後
logical, dimension(*), intent(inout)::isCalc	相互作用候補原子フラグ

戻り値 :

なし

機能 :

当該原子の相互作用候補原子のフラグを設定する。

当該原子が固定原子でかつ相手の原子が固定原子の場合、候補フラグを偽とする。

20.14. Is_InteractionUpdate : 相互作用テーブル更新の必要性チェック

名称 : Is_InteractionUpdate

呼び出し形式 : function Is_InteractionUpdate() result (update)

引数 :

なし

戻り値 :

logical::update 相互作用テーブルの更新フラグ

機能 :

1-5 相互作用テーブルの更新が必要かを調べる。

1-5 相互作用テーブルの個数が系内の全相互作用数と一致すれば、テーブルの更新は不要となる。

20.15. Set_UnitedInteraction : 一様相互作用の設定

名称 : Set_UnitedInteraction

呼び出し形式 : subroutine Set_UnitedInteraction

引数 :

なし

戻り値 :

なし

機能 :

1-5 相互作用が一様であるかを設定する。

1-5 相互作用が一様である条件は以下の全てを満たす場合である。

- A) 1つのチェーン(分子)は1つの残基で構成される。
- B) 当該分子を構成する原子は、全て自由原子か全て固定原子である。
- C) 分子内には 1-5 相互作用が存在しない。
- D) 原子間のポテンシャル関数は OPLS を使用する。

名称: Register_Atoms

引数：

戻り値:

機能：

当該原子に対し、後続 ID でかつカットオフ距離内に存在する原子を相互作用テーブルに登録する。

名称: Register_AtomsPeriodic

引数：

戻り値：

機能：

当該原子に対し、後続 ID でかつカットオフ距離内に存在する原子を相互作用テーブルに登録する。
本モジュールは周期系で原子間距離を計算する。

20. 18. Create_PairListWithCellPeriodic : 周期系の空間分割法での相互作用リスト作成

名称 : Create_PairListWithCellPeriodic

呼び出し形式 : subroutine Create_PairListWithCellPeriodic(atomID, &
vdwList, vdwNum, &
hydList, hydNum, &
vdwReplica, &
hydReplica, &
Atom, type, &
isFreeAtom, &
num)

引数 :

integer, intent(in)::atomID ! current atom
integer, dimension(MAX_ATOM), intent(out)::vdwList ! vdw atom list
integer, dimension(MAX_ATOM), intent(out)::hydList ! hyd atom list
integer, dimension(MAX_ATOM), intent(out)::vdwReplica ! vdw replica index
integer, dimension(MAX_ATOM), intent(out)::hydReplica ! hyd replica index
integer, intent(out)::vdwNum ! count of vdwList
integer, intent(out)::hydNum ! count of hydList

integer, intent(in)::num ! number of atoms
type(Atom_t), dimension(MAX_ATOM), intent(in)::Atom ! atom information
integer, dimension(MAX_ATOM)::type
logical, dimension(MAX_ATOM)::isFreeAtom

戻り値 :

なし

機能 :

残基中心間に対するカットオフを行い、相互参照テーブルを更新する。

登録する条件は以下の全てを満たす場合である。

- (a) 当該原子が所属する原子の空間に対し、対象残基は隣接セル内に存在し、残基間の距離はカットオフ距離よりも短い。
- (b) (a) の原子は当該原子と 1-5 以上離れている。
- (c) (a) の原子 ID は当該原子の ID よりも大きい。

20. 19. Update_CellIndexPeriodic : 周期系の残基の空間への登録

名称 : Update_CellIndexPeriodic

呼び出し形式 : subroutine Update_CellIndexPeriodic

引数 :

なし

戻り値 :

なし

機能 :

カットオフ長を基準とした空間に系を分割し、各空間に所属する残基を登録する。

20.20. Create_PairListWithCell : 孤立系の空間分割法での相互作用リスト作成

名称 : Create_PairListWithCell

呼び出し形式 : subroutine Create_PairListWithCellPeriodic(atomID, &
vdwList, vdwNum, &
hydList, hydNum, &
Atom, type, &
isFreeAtom, &
num)

引数 :

integer, intent(in)::atomID ! current atom
integer, dimension(MAX_ATOM), intent(out)::vdwList ! vdw atom list
integer, dimension(MAX_ATOM), intent(out)::hydList ! hyd atom list
integer, intent(out)::vdwNum ! count of vdwList
integer, intent(out)::hydNum ! count of hydList
integer, intent(in)::num ! number of atoms
type (Atom_t), dimension(MAX_ATOM), intent(in)::Atom ! atom information
integer, dimension(MAX_ATOM)::type
logical, dimension(MAX_ATOM)::isFreeAtom

戻り値 :

なし

機能 :

残基中心間に対するカットオフを行い、相互参照テーブルを更新する。

登録する条件は以下の全てを満たす場合である。

- (a) 当該原子が所属する原子の空間に対し、対象残基は隣接セル内に存在し、残基間の距離はカットオフ距離よりも短い。
- (b) (a) の原子は当該原子と 1-5 以上離れている。
- (c) (a) の原子 ID は当該原子の ID よりも大きい。

20.21. Update_CellIndex : 孤立系の残基の空間への登録

名称 : Update_CellIndex

呼び出し形式 : subroutine Update_CellIndex

引数 :

なし

戻り値 :

なし

機能 :

カットオフ長を基準とした空間に系を分割し、各空間に所属する残基を登録する。

21. Interact_Control

ファイル名 : Interact_Control.f90

21.1. Interact_Control : 相互作用計算主制御

名称 : Interact_Control

呼び出し形式 : subroutine Interact_Main(object, interact, dynamics, cluster, boundary, option)

引数 :

type(Object_t), intent(inout)::object	オブジェクト情報
type(Interact_t), intent(in)::interact	相互作用情報
type(Dynamics_t), intent(inout)::dynamics	分子動力学情報
type(MultiCluster_t), intent(inout)::cluster	クラスタ情報
type(Option_t), intent(in)::option	オプション情報
type(Boundary_t), intent(in)::boundary	バウンダリ情報

戻り値 :

なし

機能 :

相互作用計算処理を制御する。

相互作用計算は以下の順に行う。

- 1) bond
- 2) angle
- 3) torsion
- 4) improper
- 5) 1-4 van der Waals
- 6) 1-4 static electric
- 7) 1-5 van der Waals
- 8) 1-5 static electric
- 9) 1-5 hydrogen-bond

22. cosgene

ファイル名 : cosgene.f90

22.1. Para_MD : メインプログラム

名称 : Para_MD

呼び出し形式 : メインプログラムのためなし

引数 :

なし

戻り値 :

なし

機能 :

並列実行時に、0 番 CPU をメインプロセスとし、他の CPU を child プロセスとして、それぞれの処理を実行する。

22.2. MD_MainProcess : MD メインプロセス

名称 : MD_MainProcess

呼び出し形式 : subroutine MD_MainProcess

引数 :

なし

戻り値 :

なし

機能 :

Cosgene の初期設定を実行する。

また、制御ファイルを読み込みながら、内容に応じて各フェーズの呼び出しを行う。

23. Math_Library

ファイル名 : Math_Library.f90

23.1. Innter_Product : 3 次元ベクトルの内積計算

名称 : Inner_Product

呼び出し形式 : function Inner_Product(vec1, vec2) result (innerProduct)

引数 :

real*8, intent(in), dimension(3)::vec1, vec2 内積計算対象のベクトル

戻り値 :

real*8::innerProduct 内積

機能 :

2つのベクトルの内積を計算する。

23.2. Add_InnterProduct : 3 次元ベクトルの内積計算と加算

名称 : Add_InnerProduct

呼び出し形式 : function Add_InnerProduct(inner, vec1, vec2) result (addInner)

引数 :

real*8, intent(in), dimension(3)::vec1, vec2 内積計算対象のベクトル

real*8, intent(in)::inner 中間結果

戻り値 :

real*8::addInner 内積

機能 :

2つのベクトルの内積を計算し、中間結果に加算する。

23.3. Cross_Product : 3 次元ベクトルの外積計算

名称 : Cross_Product

呼び出し形式 : function Cross_Product(vec1, vec2) result (crossProduct)

引数 :

real*8, intent(in), dimension(3)::vec1, vec2 外積計算対象のベクトル

戻り値 :

real*8, dimension(3)::crossProduct 外積

機能 :

2つのベクトルの外積を計算する。

23.4. ADD_Square_ValueStrict : 3 次元ベクトルの 2 乗の加算

名称 : ADD_Square_ValueStrict

呼び出し形式 : function Add_Square_ValueStrict(value, vec) result (result)

引数 :

real*8, intent(in)::value 中間結果

real*8, intent(in), dimension(3)::vec ベクトル

戻り値 :

real*8::result

機能 :

中間結果にベクトルの大きさの 2 乗を加算する。(精度を高くするため、べき乗関数を使用)

23.5. Square_ValueStrict : 3次元ベクトルの2乗の計算

名称 : Square_ValueStrict

呼び出し形式 : function Square_ValueStrict(vec) result (value)

引数 :

real*8, intent(in), dimension(3)::vec

戻り値 :

real*8::value

機能 :

ベクトルの大きさの2乗を計算する。(精度を高くするため、べき乗関数を使用)

23.6. Square_Value : 3次元ベクトルの2乗の計算

名称 : Square_Value

呼び出し形式 : calculate square value of vector (use multiplication)

引数 :

real*8, intent(in), dimension(3)::vec

戻り値 :

real*8::value

機能 :

ベクトルの大きさの2乗を計算する。(乗算を使用)

23.7. Standard_Deviation : 標準偏差と平均値の計算

名称 : Standard_Deviation

呼び出し形式 : function Standard_Deviation(data, average) result (sd)

引数 :

real*8, intent(in), dimension(:)::data

データ

real*8, intent(out)::average

平均値

戻り値 :

real*8::sd

標準偏差

機能 :

データの標準偏差と平均値を計算する。

23.8. Determinant_3by3 : 3次元行列の行列式計算

名称 : Determinant_3by3

呼び出し形式 : function Determinant_3by3(matrix) result (determinant)

引数 :

real*8, dimension(3,3), intent(in)::matrix

3次元行列

戻り値 :

real*8::determinant

行列式

機能 :

3次元行列の行列式を計算する。

23.9. MatrixProduct_3by3 : 3次元行列の積の計算

名称 : MatrixProduct_3by3

呼び出し形式 : function MatrixProduct_3by3(matrix1, matrix2) result (resultMatrix)

引数 :

real*8, dimension(3,3), intent(in)::matrix1

3次元行列

real*8, dimension(3,3), intent(in)::matrix2

3次元行列

戻り値 :

real*8, dimension(3,3)::resultMatrix

行列の積

機能 :

3次元行列間の積を計算する。

23. 10. InvertMatrix_3by3 : 3次元行列の逆行列の計算

名称 : InvertMatrix_3by3

呼び出し形式 : function InvertMatrix_3by3(matrix1) result (resultMatrix)

引数 :

real*8, dimension(3,3), intent(in)::matrix 3次元行列

戻り値 :

real*8, dimension(3,3)::resultMatrix 逆行列

機能 :

3次元の逆行列を計算する。

23. 11. Calc_LinearEquation : 連立方程式の解法

名称 : Calc_LinearEquation

呼び出し形式 : subroutine Calc_LinearEquation(a, x, b, declare, size, eps)

引数 :

integer, intent(in)::declare	配列宣言用
real*8, dimension(declare, declare), intent(inout)::a	連立方程式の係数行列
real*8, dimension(declare), intent(in)::b	連立方程式の結果
real*8, dimension(declare), intent(out)::x	連立方程式の解
integer, intent(in)::size	連立方程式の次数
real*8, intent(in)::eps	収束判定境界

戻り値 :

なし

機能 :

ガウスの消去法(LU法)を使用して、連立方程式の解を計算する。

$$a(1:size, 1:size) * x(1:size) = b(1:size)$$
23. 12. Calc_EigenValueProblem : 固有値の計算

名称 : Calc_EigenValueProblem

呼び出し形式 : function Calc_EigenValueProblem(matrix, loop, converge) result (eigen)

引数 :

real*8, intent(inout), dimension(:, ::) :: matrix	固有値行列
integer, intent(in)::loop	収束ループ回数上限
real*8, intent(in)::converge	収束半定値

戻り値 :

real*8, dimension(SIZE(matrix, 1))::eigen 固有値

機能 :

ヤコビ法を使用して固有値を計算する。

23. 13. Create_RandomNumber : 乱数の生成

名称 : Create_RandomNumber

呼び出し形式 : subroutine Create_RandomNumber (size, seed, randomList)

引数 :

integer, intent(in)::seed	乱数の種
integer, intent(in)::size	生成する乱数の個数
real*4, intent(inout), dimension(size)::randomList	乱数リスト

戻り値 :

なし

機能 :

正規化した乱数を生成する。

23. 14. Random_Sequence : 乱数列の生成

名称 : Random_Sequence

呼び出し形式 : subroutine Random_Sequence (size, sequence, randomList)

引数 :

integer, intent(in)::size	生成する乱数の個数
real*4, intent(inout), dimension(size)::randomList	乱数リスト
integer, intent(inout)::sequence	乱数の種

戻り値 :

なし

機能 :

乱数列を生成する。

23. 15. Exec_QuickSort : クイックソートの実行

名称 : Exec_QuickSort

呼び出し形式 : subroutine Exec_QuickSort (data, tag, num)

引数 :

real*8::data(*)	ソートデータ
integer::tag(*)	ソート前の位置
integer::num	ソートデータ数

戻り値 :

なし

機能 :

クイックソートの初期処理と開始を行う。

23. 16. Sort_PartialData : 部分データのクイックソート

名称 : Sort_PartialData

呼び出し形式 : recursive subroutine Sort_PartialData (data, tag, low, high)

引数 :

real*8::data(*)	ソートデータ
integer::tag(*)	ソート前の位置
integer::low	ソート開始添字
integer::high	ソート最終添字

戻り値 :

なし

機能 :

ソート開始添え字から、ソート最終添字までのデータのクイックソートを行う。

要素数が少ない場合、クイックソートよりも逐次ソートのほうが高速であるため、ソートする要素が 14 以下の場合、逐次ソートを実行する。

ソート終了後、最小値 : 中間値-1、中間値+1 : 最大値までの要素に対し、再帰的にクイックソートを実行する。

23.17. Exec_SequentialSort : 部分データの逐次ソート

名称 : Exec_SequentialSort

呼び出し形式 : subroutine Exec_SequentialSort (data, tag, low, high)

引数 :

real*8::data(*)	ソートデータ
integer::tag(*)	ソート前の位置
integer::low	ソート開始添字
integer::high	ソート最終添字

戻り値 :

なし

機能 :

ソート開始添字から、ソート最終添字までのデータの逐次ソートを行う。

24. Minimize_Method

ファイル名 : Minimize_Method.f90

24.1. Exec_SteepestDecent : 最急勾配法でのエネルギー最小化

名称 : Exec_SteepestDecent

呼び出し形式 : subroutine Exec_SteepestDecent(object, interact, restrain, condition, &
dynamics, option, boundary, cluster)

引数 :

type(Object_t), intent(inout)::object	オブジェクト情報
type(Interact_t), intent(in)::interact	相互作用情報
type(Restrain_t), intent(in)::restrain	拘束情報
type(Condition_t), intent(in)::condition	シミュレート条件情報
type(Dynamics_t), intent(inout)::dynamics	ダイナミクス情報
type(Option_t), intent(inout)::option	オプション情報
type(Boundary_t), intent(in)::boundary	バウンダリ情報
type(MultiCluster_t), intent(inout)::cluster	クラスタ情報

戻り値 :

なし

機能 :

最急勾配法を使用し、系全体のポテンシャルエネルギーが減少するように原子位置を変更する。

24.2. Exec_ConjugateGradient : 共益勾配法でのエネルギー最小化

名称 : Exec_ConjugateGradient

呼び出し形式 : subroutine Exec_ConjugateGradient(object, interact, restrain, &
condition, dynamics, option, boundary, cluster)

引数 :

type(Object_t), intent(inout)::object	オブジェクト情報
type(Interact_t), intent(in)::interact	相互作用情報
type(Restrain_t), intent(in)::restrain	拘束情報
type(Condition_t), intent(in)::condition	シミュレート条件情報
type(Dynamics_t), intent(inout)::dynamics	ダイナミクス情報
type(Option_t), intent(inout)::option	オプション情報
type(Boundary_t), intent(in)::boundary	バウンダリ情報
type(MultiCluster_t), intent(inout)::cluster	クラスタ情報

戻り値 :

なし

機能 :

共益勾配法を使用し、系全体のポテンシャルエネルギーが減少するように原子位置を変更する。

25. Monitoring

ファイル名 : Monitor.f90

25.1. Display_Monitor : MD のトラジェクトリー、ログ出力

名称 : Display_Monitor

呼び出し形式 : subroutine Display_Monitor(vel, simulateTime)

引数 :

real*8, dimension(:, :)::vel	自由原子の速度
real*8, intent(in)::simulateTime	シミュレート時間

戻り値 :

なし

機能 :

トラジェクトリー出力およびログの出力の呼び出しを行う。

25.2. Monitor_Trajectory : トラジェクトリーの出力

名称 : Monitor_Trajectory

呼び出し形式 : subroutine Monitor_Trajectory(Atom, vel, simulateTime)

引数 :

type(Atom_t), intent(in), dimension(:)::Atom	原子情報
real*8, intent(in), dimension(:, :)::vel	自由原子の速度
real*8, intent(in)::simulateTime	シミュレート時間

戻り値 :

なし

機能 :

モニターファイルで指定されたトラジェクトリーの出力を行う。

トラジェクトリーの種別を以下に示す。

- 1) 全原子の位置
- 2) 全自由原子の速度
- 3) ポテンシャルエネルギー
- 4) モニター指定された構造
 - 4-1) 指定した原子の位置
 - 4-2) 指定した原子間の距離
 - 4-3) 指定した angle の角度
 - 4-4) 指定した torsion の角度

25.3. Display_Log : MD のログ情報出力

名称 : Display_Log

呼び出し形式 : subroutine Display_Log(totalEnergy, potentialEnergy, kineticEnergy, &
rmsf, density, temp, simulateTime, loopNum, format)

引数 :

real*8, intent(in)::totalEnergy	総エネルギー
real*8, intent(in)::potentialEnergy	ポテンシャルエネルギー
real*8, intent(in)::kineticEnergy	運動エネルギー
real*8, intent(in)::temp	系の温度
real*8, intent(in)::simulateTime	シミュレート時間
real*8, intent(in)::rmsf	root-mean square force
real*8, intent(in)::density	系の密度
integer, intent(in)::loopNum	MD ステップ数
integer, intent(in)::format	ログの形式

戻り値 :

なし

機能 :

MD のログを出力する。

25.4. Open_MonitorFile : モニター用ファイルのオープン

名称 : Open_MonitorFile

呼び出し形式 : subroutine Open_MonitorFile

引数 :

なし

戻り値 :

なし

機能 :

モニター用のファイルをオープンする。

モニターの対象を以下に示す。

- 1) 全原子の位置
- 2) 全自由原子の速度
- 3) ポテンシャルエネルギー
- 4) モニター指定された構造

25.5. Open_MonitorFile : モニター用ファイルのクローズ

名称 : Close_MonitorFile

呼び出し形式 : subroutine Close_MonitorFile

引数 :

なし

戻り値 :

なし

機能 :

モニター用のファイルをクローズする。

25.6. Is_OutputStep : ファイル出力ステップの判定

名称 : Is_OutputStep

呼び出し形式 : function Is_OutputStep(info, i) result (output)

引数 :

type(FileMonitor_t), intent(in)::info

ファイル情報

integer, intent(in)::i

当該 MD ループ回数

戻り値 :

logical::output

出力フラグ

機能 :

当該 MD 回数がファイル出力のタイミングかをチェックする。

25.7. Display_InputOption : INPUT フェーズ情報の出力

名称 : Display_InputOption

呼び出し形式 : subroutine Display_InputOption

引数 :

なし

戻り値 :

なし

機能 :

INPUT フェーズ情報を標準出力に表示する。

25.8. Display_MinimizeOption : MINIMIZE フェーズ情報の出力

名称 : Display_MinimizeOption

呼び出し形式 : subroutine Display_MinimizeOption

引数 :

なし

戻り値 :

なし

機能 :

MINIMIZE フェーズ情報を標準出力に表示する。

25.9. Display_MDOption : MD フェーズ情報の出力

名称 : Display_MDOption

呼び出し形式 : subroutine Display_MDOption

引数 :

なし

戻り値 :

なし

機能 :

MD フェーズ情報を標準出力に表示する。

25.10. Display_OutputOption : OUTPUT フェーズ情報の出力

名称 : Display_OutputOption

呼び出し形式 : subroutine Display_OutputOption

引数 :

なし

戻り値 :

なし

機能 :

OUTPUT フェーズ情報を標準出力に表示する。

25.11. Display_AnalyzeOption : ANALYZE フェーズ情報の出力

名称 : Display_AnalyzeOption

呼び出し形式 : subroutine Display_AnalyzeOption

引数 :

なし

戻り値 :

なし

機能 :

ANALYZE フェーズ情報を標準出力に表示する。

25. 12. Displat_MonitorFile : モニター用ファイル情報の出力

名称 : Display_MonitorFile

呼び出し形式 : subroutine Display_MonitorFile(title, monitor)

引数 :

character(*), intent(in)::title

モニタータイトル名

type(FileMonitor_t), intent(inout)::monitor

モニター情報

戻り値 :

なし

機能 :

モニターのファイル情報を標準出力に出力する。

- 1) 装置番号
- 2) 出力間隔
- 3) ファイル名称
- 4) ファイルフォーマット

25. 13. Displat_InOutFile : 入出力ファイル情報の出力

名称 : Display_InOutFile

呼び出し形式 : subroutine Display_InOutFile(title, input)

引数 :

character(*), intent(in)::title

タイトル名

type(File_t), intent(in)::input

ファイル情報

戻り値 :

なし

機能 :

入出力ファイルの情報を標準出力に出力する。

- 1) 装置番号
- 2) ファイル名称
- 3) ファイルフォーマット

25. 14. Monitor_Minimize : MINIMIZE のログ出力

名称 : Monitor_Minimize

呼び出し形式 : subroutine Monitor_Minimize(loop, stepLength)

引数 :

integer, intent(in)::loop

MINIMIZE の回数

real*8, intent(in)::stepLength

ステップ長

戻り値 :

なし

機能 :

MINIMIZE のログを出力する。

25. 15. Display_PotentialEnergy : ポテンシャルエネルギーの出力

名称 : Display_PotentialEnergy

呼び出し形式 : subroutine Display_PotentialEnergy

引数 :

なし

戻り値 :

なし

機能 :

各ポテンシャルエネルギーを標準出力に出力する。

25. 16. Output_TsallisData : Tsallis 用データの出力

名称 : Display_TsallisData

呼び出し形式 : subroutine Output_TsallisData(monMD, dyn)

引数 :

type(MD_t), intent(in) :: monMD

type(Dynamics_t), intent(in) :: dyn

戻り値 :

なし

機能 :

Tsallis の ξ 値、積分チェック値、 ρ の値をファイルに出力する。

26. Fast_Nonbonded

ファイル名 : Fast_NonBonded.f90

26.1. Calc_Cutoff15InteractEnergy : カットオフでの 1-5 相互作用計算制御

名称 : Calc_Cutoff15InteractEnergy

呼び出し形式 : subroutine Calc_Cutoff15InteractEnergy(nonBonded, Atom, loopInit, loopLast,
dyn, option, updatePeriod)

引数 :

type(NonBondedInfo_t), intent(in)::nonBonded	1-5 相互作用情報
type(Atom_t), intent(inout), dimension(MAX_ATOM)::Atom	原子情報
integer, intent(in)::loopInit	ループ初期値
integer, intent(in)::loopLast	ループ終値
type(Dynamics_t), intent(inout)::dyn	MD 情報
type(Option_t), intent(in)::option	オプション情報
integer, intent(in)::updatePeriod	更新間隔

戻り値 :

なし

機能 :

孤立系でカットオフを使用した場合の 1-5 相互作用を計算する。
FMM を使用する場合、この処理で 1-5 van der Waals を計算する。

注 26.2~26.12 に記述する相互作用計算処理の引数は共通であるため、ここで記述する。

real*8, dimension(3, MAX_ATOM)::cord	原子座標
real*8, dimension(3, MAX_ATOM)::grad	原子への力
real*8, dimension(MAX_ATOM)::ele	原子電荷
real*8::eleCoeff	静電係数
real*8, dimension(MAX_TYPE, MAX_TYPE)::hyd10Coeff	水素結合 10 乗項係数
real*8, dimension(MAX_TYPE, MAX_TYPE)::hyd12Coeff	水素結合 12 乗項係数
real*8, dimension(MAX_TYPE, MAX_TYPE)::vdw6Coeff	van der Waals 6 乗項係数
real*8, dimension(MAX_TYPE, MAX_TYPE)::vdw12Coeff	van der Waals 12 乗項係数
integer, intent(in)::init	ループ初期値 (処理対象原子先頭)
integer, intent(in)::last	ループ終値 (処理対象原子の最後)
integer, intent(in)::step	ループ増分値
real*8, dimension(MAX_POT)::energy	ポテンシャルエネルギー
integer, dimension(MAX_ATOM)::vdwTopAtom	van der Waals の先頭インデックス配列
integer, dimension(MAX_ATOM)::vdwLastAtom	van der Waals の最後インデックス配列
integer, dimension(MAX_VDW)::vdwAtomID	van der Waals の相手の原子 ID
integer, dimension(MAX_ATOM)::hydTopAtom	水素結合の先頭インデックス配列
integer, dimension(MAX_ATOM)::hydLastAtom	水素結合の最後インデックス配列
integer, dimension(MAX_HYD)::hydAtomID	水素結合の相手の原子 ID
integer, dimension(MAX_ATOM)::type	原子タイプ

26.2. Calc_VdwCutoff : カットオフでの van der Waals 計算

名称 : Calc_VdwCutoff

呼び出し形式 : subroutine Calc_VdwCutoff(cord, grad, ele, type, energy, eleCoeff,
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID,
hydTopAtom, hydLastAtom, hydAtomID)

引数 :

26.1. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の van der Waals の力とポテンシャルの計算を行う。

26.3. Calc_HydCutoff : カットオフでの水素結合計算

名称 : Calc_HydCutoff

呼び出し形式 : subroutine Calc_HydCutoff(cord, grad, ele, type, energy, eleCoeff,
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID,
hydTopAtom, hydLastAtom, hydAtomID)

引数 :

26.1. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の水素結合の力とポテンシャルの計算を行う。

26.4. Calc_ConstEleCutoff : カットオフでの距離非依存静電計算

名称 : Calc_ConstEleCutoff

呼び出し形式 : subroutine Calc_ConstEleCutoff(cord, grad, ele, type, energy, eleCoeff,
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID,
hydTopAtom, hydLastAtom, hydAtomID)

引数 :

26.1. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の距離非依存静電の力とポテンシャルの計算を行う。

26.5. Calc_DependEleCutoff : カットオフでの距離依存静電計算

名称 : Calc_DependEleCutoff

呼び出し形式 : subroutine Calc_DependEleCutoff (cord, grad, ele, type, energy, eleCoeff,
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID,
hydTopAtom, hydLastAtom, hydAtomID)

引数 :

26.1. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の距離依存静電の力とポテンシャルの計算を行う。

26.6. Calc_VdwConstEleCutoff : カットオフでの van der Waals と距離非依存静電計算

名称 : Calc_VdwConstEleCutoff

呼び出し形式 : subroutine Calc_VdwConstEleCutoff(cord, grad, ele, type, energy, eleCoeff,
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID,
hydTopAtom, hydLastAtom, hydAtomID)

引数 :

26.1. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の van der Waals と距離非依存静電の力とポテンシャルの計算を行う。

26.7. Calc_VdwDependEleCutoff : カットオフでの van der Waals と距離依存静電計算

名称 : Calc_VdwDependEleCutoff

呼び出し形式 : subroutine Calc_VdwDependEleCutoff(cord, grad, ele, type, energy, eleCoeff,
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID,
hydTopAtom, hydLastAtom, hydAtomID)

引数 :

26.1. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の van der Waals と距離依存静電の力とポテンシャルの計算を行う。

26.8. Calc_VdwHydCutoff : カットオフでの van der Waals と水素結合計算

名称 : Calc_VdwHydCutoff

呼び出し形式 : subroutine Calc_VdwHydCutoff(cord, grad, ele, type, energy, eleCoeff,
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID,
hydTopAtom, hydLastAtom, hydAtomID)

引数 :

26.1. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の van der Waals と水素結合の力とポテンシャルの計算を行う。

26.9. Calc_HydConstEleCutoff : カットオフでの水素結合と距離非依存静電計算

名称 : Calc_HydConstEleCutoff

呼び出し形式 : subroutine Calc_HydConstEleCutoff(cord, grad, ele, type, energy, eleCoeff,
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID,
hydTopAtom, hydLastAtom, hydAtomID)

引数 :

26.1. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の水素結合と距離非依存静電の力とポテンシャルの計算を行う。

26.10. Calc_HydDependEleCutoff : カットオフでの水素結合と距離依存静電計算

名称 : Calc_HydDependEleCutoff

呼び出し形式 : subroutine Calc_HydDependEleCutoff(cord, grad, ele, type, energy, eleCoeff,
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID,
hydTopAtom, hydLastAtom, hydAtomID)

引数 :

26.1. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の水素結合と距離依存静電の力とポテンシャルの計算を行う。

26.11. Calc_VdwHydConstEleCutoff : カットオフでの van der Waals, 水素結合, 距離非依存静電計算

名称 : Calc_VdwHydConstEleCutoff

呼び出し形式 : subroutine Calc_VdwHydConstEleCutoff(cord, grad, ele, type, energy, eleCoeff,
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID,
hydTopAtom, hydLastAtom, hydAtomID)

引数 :

26.1. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の van der Waals, 水素結合, 距離非依存静電の力とポテンシャルの計算を行う。

26.12. Calc_VdwHydDependEleCutoff : カットオフでの van der Waals, 水素結合, 距離依存静電計算

名称 : Calc_VdwHydDependEleCutoff

呼び出し形式 : subroutine Calc_VdwHydDependEleCutoff(cord, grad, ele, type, energy, eleCoeff,
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID,
hydTopAtom, hydLastAtom, hydAtomID)

引数 :

26.1. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の van der Waals, 水素結合, 距離依存静電の力とポテンシャルの計算を行う。

26. 13. Calc_NoCut15InteractEnergy : カットオフなしの 1-5 相互作用計算

名称 : Calc_NoCut15InteractEnergy

呼び出し形式 : subroutine Calc_NoCut15InteractEnergy(nonBonded, Atom, loopInit, loopLast,
dyn, option)

引数 :

type(NonBondedInfo_t), intent(in)::nonBonded	1-5 相互作用情報
type(Atom_t), intent(inout), dimension(MAX_ATOM)::Atom	原子情報
integer, intent(in)::loopInit	ループ初期値
integer, intent(in)::loopLast	ループ終値
type(Dynamics_t), intent(inout)::dyn	MD 情報
type(Option_t), intent(in)::option	オプション情報

戻り値 :

なし

機能 :

孤立系でカットオフを使用しない場合の 1-5 相互作用を計算する。

注 26. 14~26. 24 に記述する相互作用計算処理の引数は共通であるため、ここで記述する。

real*8, dimension(3, MAX_ATOM)::cord	原子座標
real*8, dimension(3, MAX_ATOM)::grad	原子への力
real*8, dimension(MAX_ATOM)::ele	原子電荷
real*8::eleCoeff	静電係数
real*8, dimension(MAX_TYPE, MAX_TYPE)::hyd10Coeff	水素結合 10 乗項係数
real*8, dimension(MAX_TYPE, MAX_TYPE)::hyd12Coeff	水素結合 12 乗項係数
real*8, dimension(MAX_TYPE, MAX_TYPE)::vdw6Coeff	van der Waals 6 乗項係数
real*8, dimension(MAX_TYPE, MAX_TYPE)::vdw12Coeff	van der Waals 12 乗項係数
integer, intent(in)::init	ループ初期値(処理対象原子先頭)
integer, intent(in)::last	ループ終値(処理対象原子の最後)
integer, intent(in)::step	ループ増分値
real*8, dimension(MAX_POT)::energy	ポテンシャルエネルギー
logical, dimension(MAX_ATOM)::isFreeAtom	自由原子フラグ
integer, dimension(MAX_TYPE, MAX_TYPE)::funkKind	ポテンシャル関数種別マトリックス

26. 14. Calc_VdwNocut : カットオフなしの van der Waals 計算

名称 : Calc_VdwNocut

呼び出し形式 : subroutine Calc_VdwNocut(cord, grad, ele, type, energy, eleCoeff,
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
isFreeAtom, funcKind)

引数 :

26. 13. を参照

戻り値 :

なし

機能 :

相互作用リストを作成し 1-5 の van der Waals の力とポテンシャルの計算を行う。

26. 15. Calc_HydNocut : カットオフなしの水素結合計算

名称 : Calc_HydNocut

呼び出し形式 : subroutine Calc_HydNocut(cord, grad, ele, type, energy, eleCoeff,
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
isFreeAtom, funcKind)

引数 :

26. 13. を参照

戻り値 :

なし

機能 :

相互作用リストを作成し 1-5 の水素結合の力とポテンシャルの計算を行う。

26. 16. Calc_ConstEleNocut : カットオフなしの距離非依存静電計算

名称 : Calc_ConstEleNocut

呼び出し形式 : subroutine Calc_ConstEleNocut(cord, grad, ele, type, energy, eleCoeff,
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
isFreeAtom, funcKind)

引数 :

26. 13. を参照

戻り値 :

なし

機能 :

相互作用リストを作成し 1-5 の距離非依存静電の力とポテンシャルの計算を行う。

26.17. Calc_DependEleNocut : カットオフなしの距離依存静電計算

名称 : Calc_DependEleNocut

呼び出し形式 : subroutine Calc_DependEleNocut (cord, grad, ele, type, energy, eleCoeff,
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
isFreeAtom, funcKind)

引数 :

26.13. を参照

戻り値 :

なし

機能 :

相互作用リストを作成し 1-5 の距離依存静電の力とポテンシャルの計算を行う。

26.18. Calc_VdwConstEleNocut : カットオフなしの van der Waals と距離非依存静電計算

名称 : Calc_VdwConstEleNocut

呼び出し形式 : subroutine Calc_VdwConstEleNocut(cord, grad, ele, type, energy, eleCoeff,
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
isFreeAtom, funcKind)

引数 :

26.13. を参照

戻り値 :

なし

機能 :

相互作用リストを作成し 1-5 の van der Waals と距離非依存静電の力とポテンシャルの計算を行う。

26.19. Calc_VdwDependEleNocut : カットオフなしの van der Waals と距離依存静電計算

名称 : Calc_VdwDependEleNocut

呼び出し形式 : subroutine Calc_VdwDependEleNocut(cord, grad, ele, type, energy, eleCoeff,
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
isFreeAtom, funcKind)

引数 :

26.13. を参照

戻り値 :

なし

機能 :

相互作用リストを作成し 1-5 の van der Waals と距離依存静電の力とポテンシャルの計算を行う。

26.20. Calc_VdwHydNocut : カットオフなしの van der Waals と水素結合計算

名称 : Calc_VdwHydNocut

呼び出し形式 : subroutine Calc_VdwHydNocut(cord, grad, ele, type, energy, eleCoeff,
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
isFreeAtom, funcKind)

引数 :

26.13. を参照

戻り値 :

なし

機能 :

相互作用リストを作成し 1-5 の van der Waals と水素結合の力とポテンシャルの計算を行う。

26.21. Calc_HydConstEleNocut : カットオフなしの水素結合と距離非依存静電計算

名称 : Calc_HydConstEleNocut

呼び出し形式 : subroutine Calc_HydConstEleNocut(cord, grad, ele, type, energy, eleCoeff,
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
isFreeAtom, funcKind)

引数 :

26.13. を参照

戻り値 :

なし

機能 :

相互作用リストを作成し 1-5 の水素結合と距離非依存静電の力とポテンシャルの計算を行う。

26.22. Calc_HydDependEleNocut : カットオフなしの水素結合と距離依存静電計算

名称 : Calc_HydDependEleNocut

呼び出し形式 : subroutine Calc_HydDependEleNocut(cord, grad, ele, type, energy, eleCoeff,
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
isFreeAtom, funcKind)

引数 :

26.13. を参照

戻り値 :

なし

機能 :

相互作用リストを作成し 1-5 の水素結合と距離依存静電の力とポテンシャルの計算を行う。

26.23 Calc_VdwHdConstEleNocut : カットオフなし van der Waals, 水素結合, 距離非依存静電計算

名称 : Calc_VdwHdConstEleNocut

呼び出し形式 : subroutine Calc_VdwHdConstEleNocut(cord, grad, ele, type, energy, eleCoeff,
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
isFreeAtom, funcKind)

引数 :

26.13. を参照

戻り値 :

なし

機能 :

相互作用リストを作成し 1-5 の van der Waals, 水素結合, 距離非依存静電の力とポテンシャルの計算を行う。

26.24 Calc_VdwHdDependEleNocut : カットオフなしの van der Waals, 水素結合, 距離依存静電計算

名称 : Calc_VdwHdDependEleNocut

呼び出し形式 : subroutine Calc_VdwHdDependEleNocut(cord, grad, ele, type, energy, eleCoeff,
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
isFreeAtom, funcKind)

引数 :

26.13. を参照

戻り値 :

なし

機能 :

相互作用リストを作成し 1-5 の van der Waals, 水素結合, 距離依存静電の力とポテンシャルの計算を行う。

27. Object_Method

ファイル名 : Object.f90

27.1. Minimum_ResidueDistance : 残基間の最短距離計算

名称 : Minimum_ResidueDistance

呼び出し形式 : function Minimum_ResidueDistance(resid1ID, resid2ID, replica) result (minimum)

引数 :

integer, intent(in)::resid1ID	残基 1 の ID
integer, intent(in)::resid2ID	残基 2 の ID
integer, dimension(3), intent(out)::replica	レプリカセルの相対位置

戻り値 :

real*8::minimum	残基間の最小距離の 2 乗
-----------------	---------------

機能 :

2 つの残基間の最短距離を求める。

残基 1 の原子と残基 2 の原子との組合せで、もっとも距離が近いものが残基間の最小距離となる。

注意点 :

戻り値は距離の 2 乗である。

27.2. Get_Coordinate : 原子位置の取得

名称 : Get_Coordinate

呼び出し形式 : subroutine Get_Coordinate(array)

引数 :

real*8, dimension(3, MAX_ATOM)::array	原子位置配列
---------------------------------------	--------

戻り値 :

なし

機能 :

原子位置を引数の配列に代入する。

27.3. Set_Coordinate : 原子位置の設定

名称 : Set_Coordinate

呼び出し形式 : subroutine Set_Coordinate(array)

引数 :

real*8, dimension(3, MAX_ATOM)::array	原子位置配列
---------------------------------------	--------

戻り値 :

なし

機能 :

引数の配列の値を原子位置に登録する。

27.4. Get_Gradient : 原子への力の取得

名称 : Get_Gradient

呼び出し形式 : subroutine Get_Gradient(array)

引数 :

real*8, dimension(3, MAX_ATOM)::array	原子に加わる力の配列
---------------------------------------	------------

戻り値 :

なし

機能 :

原子に加わる力を引数の配列に代入する。

27.5. Set_Gradient : 原子への力の設定

名称 : Set_Gradient

呼び出し形式 : subroutine Set_Gradient(array)

引数 :

real*8, dimension(3, MAX_ATOM)::array

原子に加わる力の配列

戻り値 :

なし

機能 :

引数の配列を原子に加わる力に登録する。

27.6. Get_Mass : 原子質量の取得

名称 : Get_Mass

呼び出し形式 : subroutine Get_Mass(array)

引数 :

real*8, dimension(MAX_ATOM)::array

原子質量の配列

戻り値 :

なし

機能 :

原子の質量を引数の配列に代入する。

27.7. Get_ReferenceCoordinate : 参照位置の取得

名称 : Get_ReferenceCoordinate

呼び出し形式 : subroutine Get_ReferenceCoordinate(array)

引数 :

real*8, dimension(3, MAX_ATOM)::array

参照位置の配列

戻り値 :

なし

機能 :

原子の参照位置を引数の配列に代入する。

27.8. Get_FreeAtomVelocity : 自由原子の速度取得

名称 : Get_FreeAtomVelocity

呼び出し形式 : subroutine Get_FreeAtomVelocity(array)

引数 :

real*8, dimension(3, MAX_ATOM)::array

自由原子の速度の配列

戻り値 :

なし

機能 :

自由原子の速度を引数の配列に代入する。

27.9. Set_FreeAtomVelocity : 自由原子の速度設定

名称 : Set_FreeAtomVelocity

呼び出し形式 : subroutine Set_FreeAtomVelocity(array)

引数 :

real*8, dimension(3, MAX_ATOM)::array

自由原子の速度の配列

戻り値 :

なし

機能 :

引数の配列を自由原子の速度に登録する。

27. 10. Get_Cluster : クラスタ ID の取得

名称 : Get_Cluster

呼び出し形式 : function Get_Cluster(i) result (cluster)

引数 :

integer, intent(in)::i 原子 ID

戻り値 :

integer::cluster クラスタ ID

機能 :

当該原子の属するクラスタの ID を返す。

27. 11. Get_ResidueCenter : 残基重心の取得

名称 : Get_ResidueCenter

呼び出し形式 : function Get_ResidueCenter(num) result (center)

引数 :

integer, intent(in)::num 重心を求める残基数

戻り値 :

real*8, dimension(3, MAX_RESID)::center 残基の重心

機能 :

残基の重心を計算する。

27. 12. Centering_Translation : 系のセンタリング処理

名称 : Centering_Translation

呼び出し形式 : subroutine Centering_Translation

引数 :

なし

戻り値 :

なし

機能 :

系の重心を基本セルの原点に設定する。この際、参照位置の補正も行う。

27. 13. Get_ChainCenter : 分子の重心取得

名称 : Get_ChainCenter

呼び出し形式 : function Get_ChainCenter(chainID, cord, mass) result (center)

引数 :

integer, intent(in)::chainID チェイン ID

real*8, intent(in), dimension(:, ::) :: cord 原子位置配列

real*8, intent(in), dimension(:, ::) :: mass 原子の質量の配列

戻り値 :

real*8, dimension(3) :: center 当該チェインの重心

機能 :

チェイン(分子)の重心を求める。

27.14. Get_ReferenceChainCenter : 参照位置での分子の重心取得

名称 : Get_ReferenceChainCenter

呼び出し形式 : function Get_ReferenceChainCenter(chainID, mass) result (center)

引数 :

integer, intent(in)::chainID	チェーン ID
real*8, intent(in), dimension(:)::mass	原子の質量の配列

戻り値 :

real*8, dimension(3)::center	当該チェーンの重心
------------------------------	-----------

機能 :

参照位置に対するチェーン(分子)の重心を求める。

27.15. Find_MaxForceAndAtom : 力の最大値の取得

名称 : Find_MaxForceAndAtom

呼び出し形式 : function Find_MaxForceAndAtom(atomID) result (maxForce)

引数 :

integer, intent(out)::atomID	最大の力が加わる原子の ID
------------------------------	----------------

戻り値 :

real*8::maxForce	原子に加わる力の最大値
------------------	-------------

機能 :

最大の力が加わっている原子と、その力の大きさを求める。

27.16. Calc_TotalFreeAtomForce : 自由原子への力の合計

名称 : Calc_TotalFreeAtomForce

呼び出し形式 : function Calc_TotalFreeAtomForce() result (force)

引数 :

なし

戻り値 :

real*8::force	力の合計
---------------	------

機能 :

自由原子に加わる力の合計を計算する。

27.17. Calc_RMSF : RMSF 計算

名称 : Calc_RMSF

呼び出し形式 : function Calc_RMSF() result (rmsf)

引数 :

なし

戻り値 :

real*8::rmsf	root-mean square of force
--------------	---------------------------

機能 :

自由原子の root-mean square of force を計算する。

27. 18. Calc_LeastSquareFitting : 非線形最小二乗法での揺らぎ計算**(1) Calc_LeastSquareFitting**

名称 : Calc_LeastSquareFitting

呼び出し形式 : function Calc_LeastSquareFitting(topChainID, &
 lastChainID, &
 refCord, &
 cord, &
 mass, &
 rmsdWithoutH, &
 rmsdCAlpha, &
 rmsdMainChain, &
 isChangeCord) result (rmsd)

引数 :

integer, intent(in)::topChainID	対象先頭チェーンの ID
integer, intent(in)::lastChainID	対象最終チェーンの ID
real*8, intent(in), dimension(:,:)::refCord	参照位置
real*8, intent(inout), dimension(:,:)::cord	原子位置
real*8, dimension(:)::mass	原子の質量の配列
real*8, intent(out)::rmsdWithoutH	水素原子以外の RMSD
real*8, intent(out)::rmsdCAlpha	C α の RMSD
real*8, intent(out)::rmsdMainChain	主鎖の RMSD
logical, intent(in)::isChangeCord	原子位置の変更フラグ

戻り値 :

real*8::rmsd	全原子の RMSD
--------------	-----------

機能 :

原子の位置と参照位置から構造の揺らぎの大きさ (root-mean square distance) を非線形最小二乗法を用いて計算する。topChainID から lastChainID までのチェーンを計算対象とし、全原子、水素原子以外、C α 、主鎖の揺らぎの大きさをそれぞれ計算する。

原子位置の変更フラグが真の場合、原子位置の変更を行う。

27. 19. Update_CoordinateWithBoundary : 周期境界条件での原子位置補正

名称 : Update_CoordinateWithBoundary

呼び出し形式 : subroutine Update_CoordinateWithBoundary

引数 :

なし

戻り値 :

なし

機能 :

重心が基本セル外にあるチェーンの位置を基本セル内に移動する。

注意点 :

周期境界条件にのみ有効である。

27. 20. Calc_Density : 系の密度計算

名称 : Calc_Density

呼び出し形式 : function Calc_Density() result (density)

引数 :

なし

戻り値 :

real*8::density	系の密度
-----------------	------

機能 :

系の密度を計算する

注意点 :

周期境界条件にのみ有効である。

27.21. Get_Distance : 2 点間の距離計算

名称 : Get_Distance

呼び出し形式 : Get_Distance(cord1, cord2) result (result)

引数 :

real*8, intent(in), dimension(3)::cord1	座標 1
real*8, intent(in), dimension(3)::cord2	座標 2

戻り値 :

real*8::result 座標間距離

機能 :

2 点間の距離を計算する。

27.22. Get_Angle : 3 点間の角度計算

名称 : Get_Angle

呼び出し形式 : Get_Angle(cord1, cord2, cord3) result (result)

引数 :

real*8, intent(in), dimension(3)::cord1	座標 1
real*8, intent(in), dimension(3)::cord2	座標 2
real*8, intent(in), dimension(3)::cord3	座標 3

戻り値 :

real*8::result 角度

機能 :

(cord1-cord2 のベクトル) と (cord3-cord2 のベクトル) の角度を計算する。

27.23. Get_Torsion : 4 点間の角度計算

名称 : Get_Torsion

呼び出し形式 : Get_Torsion(cord1, cord2, cord3, cord4) result (result)

引数 :

real*8, intent(in), dimension(3)::cord1	座標 1
real*8, intent(in), dimension(3)::cord2	座標 2
real*8, intent(in), dimension(3)::cord3	座標 3
real*8, intent(in), dimension(3)::cord4	座標 4

戻り値 :

real*8::result 角度

機能 :

(cord1-cord2-cord3 の平面) と (cord2-cord3-cord4 の平面) の角度を計算する。

27.24. Calc_ParticleKinetic : 原子 (剛体以外) の運動エネルギー計算

名称 : Calc_ParticleKinetic

呼び出し形式 : function Calc_ParticleKinetic(vel, num) result (kinetic)

引数 :

real*8, dimension(3, MAX_ATOM), intent(in)::vel	原子速度
integer, intent(in)::num	自由原子数

戻り値 :

real*8::kinetic 原子 (剛体以外) の運動エネルギー

機能 :

剛体に所属しない原子のみの運動エネルギーを計算する。

27.25. Calc_ParticleKineticTensor : 原子（剛体以外）の運動エネルギーテンソル計算

名称 : Calc_ParticleKineticTensor

呼び出し形式 : function Calc_ParticleKineticTensor(vel, num) result (ktensor)

引数 :

real*8, dimension(3, MAX_ATOM), intent(in)::vel	原子速度
integer, intent(in)::num	自由原子数

戻り値 :

real*8, dimension(3, 3)::ktensor	原子（剛体以外）の運動エネルギーテンソル
----------------------------------	----------------------

機能 :

剛体に所属しない原子のみの運動エネルギーテンソルを計算する。

27.26. Calc_RigidTransKinetic : 剛体の並進運動エネルギー計算

名称 : Calc_RigidTransKinetic

呼び出し形式 : function Calc_RigidTransKinetic(rigidInfo) result (kinetic)

引数 :

type(RigidInfo_t), intent(in)::rigidInfo	剛体情報
--	------

戻り値 :

real*8::kinetic	剛体の並進運動エネルギー
-----------------	--------------

機能 :

剛体の並進運動エネルギーを計算する。

27.27. Calc_RigidTransKineticTensor : 剛体の並進運動エネルギーテンソル計算

名称 : Calc_RigidTransKineticTensor

呼び出し形式 : function Calc_RigidTransKineticTensor(rigidInfo) result (ktensor)

引数 :

type(RigidInfo_t), intent(in)::rigidInfo	剛体情報
--	------

戻り値 :

real*8, dimension(3, 3)::ktensor	剛体の並進運動エネルギーテンソル
----------------------------------	------------------

機能 :

剛体の並進運動エネルギーテンソルを計算する。

27.28. Calc_RigidRotateKinetic : 剛体の回転運動エネルギー計算

名称 : Calc_RigidRotateKinetic

呼び出し形式 : function Calc_RigidTRotateKinetic(rigidInfo) result (kinetic)

引数 :

type(RigidInfo_t), intent(in)::rigidInfo	剛体情報
--	------

戻り値 :

real*8::kinetic	剛体の回転運動エネルギー
-----------------	--------------

機能 :

剛体の回転運動エネルギーを計算する。

27.29. Calc_RigidRotateKineticTensor : 剛体の回転運動エネルギーテンソル計算

名称 : Calc_RigidRotateKineticTensor

呼び出し形式 : function Calc_RigidTRotateKineticTensor(rigidInfo) result (kinetic)

引数 :

type(RigidInfo_t), intent(in)::rigidInfo	剛体情報
--	------

戻り値 :

real*8::kinetic	剛体の回転運動エネルギーテンソル
-----------------	------------------

機能 :

剛体の回転運動エネルギーテンソルを計算する。

27.30. Get_MultiChainCenter : 分子の重心取得 (複数チェーン)

名称 : Get_MultiChainCenter

呼び出し形式 : function Get_MultiChainCenter (topChainID, lastChainID, cord, mass) &
result (center)

引数 :

integer, intent(in) :: topChainID	先頭チェーン ID
integer, intent(in) :: lastChainID	最終チェーン ID
real*8, intent(in), dimension(:, :) :: cord	原子位置配列
real*8, intent(in), dimension(:) :: mass	原子の質量の配列

戻り値 :

real*8, dimension(3) :: center	重心座標
--------------------------------	------

機能 :

先頭チェーンから最終チェーンまで全体の重心を求める。

27.31. Get_MultiRefChainCenter : 参照位置での分子の重心取得 (複数チェーン)

名称 : Get_MultiRefChainCenter

呼び出し形式 : function Get_MultiRefChainCenter (topChainID, lastChainID, mass) &
result (center)

引数 :

integer, intent(in) :: topChainID	先頭チェーン ID
integer, intent(in) :: lastChainID	最終チェーン ID
real*8, intent(in), dimension(:) :: mass	原子の質量の配列

戻り値 :

real*8, dimension(3) :: center	当該チェーンの重心
--------------------------------	-----------

機能 :

参照位置に対する先頭チェーンから最終チェーンまで全体の重心を求める。

28. Output_Data

ファイル名 : Output_Data.f90

28.1. Output_TopologyFile : ASCII 形式のトポロジーファイル出力

名称 : Output_TopologyFile

呼び出し形式 : subroutine Output_TopologyFile

引数 :

なし

戻り値 :

なし

機能 :

現在の系の状態を ASCII 形式のトポロジーファイルとして出力する。

28.2. Output_BinaryTopologyFile : バイナリ形式のトポロジーファイル出力

名称 : Output_BinaryTopologyFile

呼び出し形式 : subroutine Output_BinaryTopologyFile

引数 :

なし

戻り値 :

なし

機能 :

現在の系の状態をバイナリー形式のトポロジーファイルとして出力する。

28.3. Output_PDBFile : PDB ファイルの出力

名称 : Output_PDBFile

呼び出し形式 : subroutine Output_PDBFile(file, unit, cord)

引数 :

character (NAME_SIZE)::file

ファイル名

integer::unit

出力装置番号

real*8, dimension (3, MAX_ATOM)::cord

原子座標配列

戻り値 :

なし

機能 :

現在の系の状態を PDB 形式のファイルとして出力する。

28.4. Output_BinaryCoordinateFile : バイナリー形式の原子位置出力

名称 : Output_BinaryCoordinateFile

呼び出し形式 : subroutine Output_BinaryCoordinateFile(file, unit, cord)

引数 :

character (NAME_SIZE)::file

ファイル名

integer::unit

出力装置番号

real*8, dimension (3, MAX_ATOM)::cord

原子座標配列

戻り値 :

なし

機能 :

現在の系内の原子位置をバイナリ形式のファイルとして出力する。

28. 5. Output_RestartFile : リスタートファイルの出力

名称 : Output_RestartFile

呼び出し形式 : subroutine Output_RestartFile(vel)

引数 :

real*8, dimension(:,:), intent(in)::vel

現時点の原子の速度

戻り値 :

なし

機能 :

MD のリスタートファイルを出力する。

リスタートファイルの内容は「16. 21. Input_RestartFile : リスタートファイルの読み込み」を参照。

28. 6. Output_EnergytFile : エネルギーファイルの出力

名称 : Output_EnergyFile

呼び出し形式 : subroutine Output_EnergyFile

引数 :

なし

戻り値 :

なし

機能 :

MD のエネルギーをファイルに出力する。

28. 7. Output_EnergytForAnalysis : MINIMIZE のエネルギーファイルの出力

名称 : Output_EnergyForAnalysis

呼び出し形式 : subroutine Output_EnergyForAnalysis(stepLength, rmse, rmsf, rmsd)

引数 :

real*8, intent(in)::stepLength

real*8, intent(in)::rmse

real*8, intent(in)::rmsf

real*8, intent(in)::rmsd

戻り値 :

なし

機能 :

MINIMIZE で計算したエネルギーを ANALYZE 向けのファイルに出力する。

28. 8. Output_FPAverageFile : FP 平均値ファイルの出力

名称 : Output_FPAverageFile

呼び出し形式 : subroutine Output_FPAverageFile(file, rest)

引数 :

type(File_t), intent(in)::file FP 平均値ファイル情報

type(Restrain_t), intent(inout)::rest 拘束情報

戻り値 :

なし

機能 :

FP 対象原子にかかる力（アンブレラポテンシャルを除く）と座標の平均値を算出し、FP 平均値ファイルに出力する。

28. 8. Open_FileWithForm : 各データ形式でのファイルのオープン

名称 : Open_FileWithForm

呼び出し形式 : subroutine Open_FileWithForm(unit, file, form, comment)

引数 :

integer, intent(in)::unit	装置番号
integer, intent(in)::form	データ形式
character(NAME_SIZE), intent(in)::file	ファイル名
character(*), intent(in)::comment	当該ファイルのコメント

戻り値 :

なし

機能 :

指定されたデータ形式のファイルをオープンする。

28. 9. Scale_RadiusAndCharge : vdW パラメータおよび電荷のスケーリング

名称 : Scale_RadiusAndCharge

呼び出し形式 : subroutine Scale_RadiusAndCharge(atomInfo, molInfo, nonBondedInfo)

引数 :

type (AtomInfo_t)::atomInfo	原子情報
type (MolInfo_t)::molInfo	分子情報
type (NonBondedInfo_t)::nonBondedInfo	非共有結合情報

戻り値 :

なし

機能 :

スケーリング対象原子に対して VdW パラメータおよび電荷のスケーリングを行い、スケーリングに伴うトポロジーデータの更新を行う。

29. Output_MD

ファイル名 : Output_MD.f90

29.1. Output_MDfile : MD 終了結果のファイル出力

名称 : Output_MDfile

呼び出し形式 : subroutine Output_MDfile

引数 :

戻り値 :

機能 :

MD 終了時のトポロジーファイルと PDB ファイルを出力する。

30. Calc_Position

ファイル名 : Position.f90

30.1. Calc_PositionEnergy : 位置拘束のエネルギー計算

名称 : Calc_PositionEnergy

呼び出し形式 : function Calc_PositionEnergy(Position, Atom, temp, weight, &
init, last) result (totalEnergy)

引数 :

type(Position_t), intent(in), dimension(MAX_POSITION)::position	位置拘束情報
type(Atom_t), intent(inout), dimension(MAX_ATOM)::Atom	原子情報
integer, intent(in)::init	ループ初期値
integer, intent(in)::last	ループ終値
real*8, intent(in)::weight	位置拘束重み
real*8, intent(in)::temp	拘束温度

戻り値 :

real*8::totalEnergy	総位置拘束エネルギー
---------------------	------------

機能 :

系内の位置拘束エネルギーと原子に加わる力を計算する。

30.2. Calc_PositionEnergy : 位置拘束のクラスタ間エネルギー計算

名称 : Calc_ClusterPositionEnergy

呼び出し形式 : function Calc_ClusterPositionEnergy(Position, Atom, cluster, temp, &
weight, init, last) result (totalEnergy)

引数 :

type(Position_t), intent(in), dimension(MAX_POSITION)::position	位置拘束情報
type(Atom_t), intent(inout), dimension(MAX_ATOM)::Atom	原子情報
type(Cluster_t), intent(inout), dimension(MAX_CLUSTER)::cluster	クラスタ情報
integer, intent(in)::init	ループ初期値
integer, intent(in)::last	ループ終値
real*8, intent(in)::weight	位置拘束重み
real*8, intent(in)::temp	拘束温度

戻り値 :

real*8, dimension(MAX_CLUSTER, MAX_CLUSTER)::totalEnergy	総位置拘束エネルギー
--	------------

機能 :

系内とクラスタ間の位置拘束エネルギーと原子に加わる力を計算する。

31. Calc_Repulsion

ファイル名 : Repulsion.f90

31.1. Calc_RepulsionEnergy : repulsion エネルギー計算

名称 : Calc_RepulsionEnergy

呼び出し形式 : function Calc_RepulsionEnergy (Atom, temp, weight) result (totalEnergy)

引数 :

type (Atom_t), intent (inout), dimension (MAX_ATOM) :: Atom	原子情報
real*8, intent (in) :: temp	拘束温度
real*8, intent (in) :: weight	Repulsion 重み

戻り値 :

real*8 :: totalEnergy	総 repulsion エネルギー
-----------------------	-------------------

機能 :

系内の repulsion (軟体球での van der Waals 衝突) のエネルギーと原子に加わる力を計算する。
repulsion は 1-4 および 1-5 の関係にある原子に対して計算する。

31.2. Calc_15RepulsionEnergy : 1-5 相互作用原子間の repulsion エネルギー計算

名称 : Calc_15Repulsion

呼び出し形式 : function Calc_15Repulsion (Atom, interact, energy) result (totalEnergy)

引数 :

type (Atom_t), intent (inout), dimension (MAX_ATOM) :: Atom	原子情報
type (Interact15_t), intent (in) :: interact	1-5 相互作用テーブル

戻り値 :

real*8 :: totalEnergy	1-5 repulsion エネルギー
-----------------------	---------------------

機能 :

系内の 1-5 相互作用のある原子に対して repulsion (軟体球での van der Waals 衝突) のエネルギーと原子に加わる力を計算する。

31.3. Calc_ClusterRepulsionEnergy : クラスタ間 repulsion エネルギー計算

名称 : Calc_ClusterRepulsionEnergy

呼び出し形式 : function Calc_ClusterRepulsionEnergy (Atom, Cluster, temp, weight) &
result (totalEnergy)

引数 :

type (Atom_t), intent (inout), dimension (MAX_ATOM) :: Atom	原子情報
type (Cluster_t), intent (inout), dimension (MAX_CLUSTER) :: Cluster	クラスタ情報
real*8, intent (in) :: temp	拘束温度
real*8, intent (in) :: weight	Repulsion 重み

戻り値 :

real*8, dimension (MAX_CLUSTER, MAX_CLUSTER) :: totalEnergy 総 repulsion エネルギー

機能 :

系内とクラスタ間の repulsion (軟体球での van der Waals 衝突) のエネルギーと原子に加わる力を計算する。

repulsion は 1-4 および 1-5 の関係にある原子に対して計算する。

31.4. Calc_Cluster15Repulsion : 1-5 相互作用原子間のクラスタ間 repulsion エネルギー計算

名称 : Calc_Cluster15Repulsion

呼び出し形式 : function Calc_Cluster15Repulsion (Atom, interact, cluster) result (energy)

引数 :

type (Atom_t), intent (inout), dimension (MAX_ATOM) :: Atom	原子情報
type (Interact15_t), intent (in) :: interact	1-5 相互作用テーブル
type (Cluster_t), intent (out), dimension (MAX_CLUSTER) :: cluster	クラスタ情報

戻り値 :

real*8, dimension (MAX_CLUSTER, MAX_CLUSTER) :: energy 1-5 repulsion エネルギー

機能 :

系内とクラスタ間の 1-5 相互作用のある原子に対して repulsion (軟体球での van der Waals 衝突) のエネルギーと原子に加わる力を計算する。

32. Restrain_Control

ファイル名 : Restrain_Control.f90

32.1. Restrain_Main : 拘束エネルギー計算主制御

名称 : Restrain_Main

呼び出し形式 : subroutine Restrain_Main(
object, interact, restrain, cluster, dynamics, option) &

引数 :

type(Object_t), intent(inout)::object	オブジェクト情報
type(Interact_t), intent(in)::interact	相互作用情報
type(Restrain_t), intent(in)::restrain	拘束力情報
type(Dynamics_t), intent(inout)::dynamics	MD 情報
type(MultiCluster_t), intent(inout)::cluster	クラスター情報
type(Option_t), intent(in)::option	オプション情報

戻り値 :

なし

機能 :

拘束エネルギー、拘束力計算の主制御を行う。

拘束エネルギーは以下の順で計算する。

- 1) 位置拘束
- 2) 距離拘束
- 3) 二面各拘束
- 4) CAP
- 5) repulsion

33. Shake_Method

ファイル名 : Shake.f90

33.1. Exec_Shake : SHAKE 計算主制御

名称 : Exec_Shake

呼び出し形式 : subroutine Exec_Shake(cord, preCord)

引数 :

real*8::cord(3, MAX_ATOM)

現在時刻の原子位置

real*8::preCord(3, MAX_ATOM)

直前の時刻の原子位置

戻り値 :

なし

機能 :

SHAKE(原子間の距離を一定に保つために原子位置と原子速度を補正する処理)の実行を制御する。

33.2. Calc_TwoAtomsHarmonicShake : 収束ループでの 2 原子間 SHAKE

名称 : Calc_TwoAtomsHarmonicShake

呼び出し形式 : subroutine Calc_TwoAtomsHarmonicShake(cord, preCord)

引数 :

real*8, intent(inout)::cord(3, MAX_ATOM)

現在時刻の原子位置

real*8, intent(in)::preCord(3, MAX_ATOM)

直前の時刻の原子位置

戻り値 :

なし

機能 :

収束判定ループを用いた 2 原子間の SHAKE を行う。

33.3. Calc_TwoAtomsBiquadraticShake : 2 次方程式での 2 原子間 SHAKE

名称 : Calc_TwoAtomsBiquadraticShake

呼び出し形式 : subroutine Calc_TwoAtomsBiquadraticShake(cord, preCord)

引数 :

real*8, intent(inout)::cord(3, MAX_ATOM)

現在時刻の原子位置

real*8, intent(in)::preCord(3, MAX_ATOM)

直前の時刻の原子位置

戻り値 :

なし

機能 :

2 次方程式の解法を用いた 2 原子間の SHAKE を行う。

33. 4. Calc_ThreeAtomsShake : 3 原子間 SHAKE の実行

名称 : Calc_ThreeAtomsShake

呼び出し形式 : subroutine Calc_ThreeAtomsShake(cord, preCord)

引数 :

real*8, intent(inout)::cord(3, MAX_ATOM)

現在時刻の原子位置

real*8, intent(in)::preCord(3, MAX_ATOM)

直前の時刻の原子位置

戻り値 :

なし

機能 :

3 原子間の SHAKE を行う。

33. 5. Calc_FourAtomsShake : 4 原子間 SHAKE の実行

名称 : Calc_FourAtomsShake

呼び出し形式 : subroutine Calc_FourAtomsShake(cord, preCord)

引数 :

real*8, intent(inout)::cord(3, MAX_ATOM)

現在時刻の原子位置

real*8, intent(in)::preCord(3, MAX_ATOM)

直前の時刻の原子位置

戻り値 :

なし

機能 :

4 原子間の SHAKE を行う。

34.5. L4_To_String : 論理型の文字列変換

名称 : L4_To_String

呼び出し形式 : function L4_To_String(l4) result (string)

引数 :

logical, intent(in)::l4

変換対象論理型値

戻り値 :

character(6)::string

変換結果文字列

機能 :

論理値を文字列に変換する。

34.6. String_ToReal : 文字列の実数変換

名称 : String_ToReal

呼び出し形式 : function String_ToReal(str, line) result (r8)

引数 :

character(*), intent(in)::str

入力文字列

character(*), intent(in)::line

当該行

戻り値 :

real*8::r8

変換結果

機能 :

文字列を実数型 8 バイトの値に変換する。

34.7. String_ToInteger : 文字列の整数変換

名称 : String_ToInteger

呼び出し形式 : function String_ToInteger(str, line) result (i4)

引数 :

character(*), intent(in)::str

入力文字列

character(*), intent(in)::line

当該行

戻り値 :

integer::i4

変換結果

機能 :

文字列を整数型 4 バイトの値に変換する。

34. 8. Read_RealValue : 実数の読み込みと当該文字列の削除

名称 : Read_RealValue

呼び出し形式 : function Read_RealValue(str) result (r8)

引数 :

character(*), intent(inout)::str 入力文字列

戻り値 :

real*8::r8 変換結果

機能 :

実数の文字列を読み込み、値を得る。この際に、入力文字列から実数の字句を削除する。

34. 9. Read_IntValue : 整数の読み込みと当該文字列の削除

名称 : Read_IntValue

呼び出し形式 : function Read_IntValue(str) result (i4)

引数 :

character(*), intent(inout)::str 入力文字列

戻り値 :

integer::i4 変換結果

機能 :

実数の文字列を読み込み、値を得る。この際に、入力文字列から整数の字句を削除する。

34. 10. Read_String : 文字列の読み込みと当該文字列の削除

名称 : Read_String

呼び出し形式 : function Read_String(str, target) result (name)

引数 :

character(*), intent(inout)::str 入力文字列

character(*), intent(in)::target 戻り値を示す文字列

戻り値 :

character(len(target))::name 文字列

機能 :

文字列を読み込む。この際に入力文字列から文字列の字句を削除する。

34. 11. Change_CapitalWord : 小文字の大文字変換

名称 : Change_CapitalWord

呼び出し形式 : subroutine Change_CapitalWord(str)

引数 :

character(*), intent(inout)::str

入力文字列および変換後文字列

戻り値 :

なし

機能 :

入力文字列の英字の小文字を大文字に変換する。

34. 12. IsEqual : 文字列比較

名称 : Is_Equal

呼び出し形式 : function Is_Equal(str1, str2) result (isEqual)

引数 :

character(*), intent(in)::str1

比較文字列 1

character(*), intent(in)::str2

比較文字列 2

戻り値 :

logical::isEqual

一致フラグ

機能 :

2つの文字列を比較する。

35. System_Class

ファイル名 : System_Class.f90

35.1. System_Clock : 時間の取得

名称 : System_Clock

呼び出し形式 : function System_Clock() result (second)

引数 :

なし

戻り値 :

real*8::second

当該時間

機能 :

システムライブラリを使用して、現在の時間を得る。

35.2. Write_Time : 時間の出力

名称 : Write_Time

呼び出し形式 : subroutine Write_Time(title, sec)

引数 :

character(6)::title

タイトル名

real*8::sec

当該時間

戻り値 :

なし

機能 :

当該時間を標準出力に出力する。

35.3. Inform_Job : ジョブ情報取得

名称 : Inform_Job

呼び出し形式 : subroutine Inform_Job(time, user, leng)

引数 :

character(23), intent(out)::time

時間を示す文字列

character(*), intent(out)::user

ユーザ名

integer, intent(out)::leng

文字列長

戻り値 :

なし

機能 :

現在時刻と当該ユーザ名を得る。

36. Temperature_Method

ファイル名 : Temperature.f90

36.1. Calc_KineticAndTemperature : 系の運動エネルギーと温度の取得

名称 : Calc_KineticAndTemperature

呼び出し形式 : function Calc_KineticAndTemperature(vel, num, kinetic, ktensor) result (temp)

引数 :

integer, intent(in)::num	自由原子の個数
real*8, intent(in)::vel(3, num)	自由原子の速度
real*8, intent(out):: kinetic	系の運動エネルギー

戻り値 :

real*8:: temp	系の温度
---------------	------

機能 :

系の運動エネルギー、運動エネルギー行列および温度を計算する。温度は原子全体の運動エネルギーを系の自由度で割り、この値を温度に変換することで求める。

36.2. Calc_TemperatureOfMolecules : 各分子の温度の取得

名称 : Calc_TemperatureOfMolecules

呼び出し形式 : function Calc_TemperatureOfMolecules(vel, num) result (temp)

引数 :

integer, intent(in)::num	自由原子の個数
real*8, intent(in), dimension(3, num)::vel	自由原子の速度

戻り値 :

real*8, dimension(MAX_MOL)::temp	分子ごとの温度
----------------------------------	---------

機能 :

分子ごとの温度を計算する。温度は各分子を構成する原子の運動エネルギーを各分子の自由度で割り、この値を温度に変換することで求める。

36.3. Set_TemperatureOfMolecules : 各分子の温度の登録

名称 : Set_TemperatureOfMolecules

呼び出し形式 : subroutine Set_TemperatureOfMolecules

引数 :

なし

戻り値 :

なし

機能 :

現在時間の分子ごとの温度を MD 情報に登録する。

36.4. Calc_KineticOfMolecules : 各分子の運動エネルギーを計算

名称 : Calc_KineticOfMolecules

呼び出し形式 : function Calc_KineticOfMolecules(vel, num) result (kinetic)

引数 :

integer, intent(in)::num	自由原子の個数
real*8, intent(in), dimension(3, num)::vel	自由原子の速度

戻り値 :

real*8, dimension(MAX_MOL)::kinetic	分子ごとの運動エネルギー
-------------------------------------	--------------

機能 :

分子ごとの運動エネルギーを計算する。

36.5. Calc_RigidKinetic : 剛体の運動エネルギー計算

名称 : Calc_RigidKinetic

呼び出し形式 : function Calc_RigidKinetic() result (kinetic)

引数 :

なし

戻り値 :

real*8::kinetic

剛体の運動エネルギー

機能 :

剛体の運動エネルギーを計算する。

36.6. Calc_RigidKineticTensor : 剛体の運動エネルギー行列計算

名称 : Calc_RigidKineticTensor

呼び出し形式 : function Calc_RigidKineticTensor() result (ktensor)

引数 :

なし

戻り値 :

real*8, dimension(3,3)::ktensor

剛体の運動エネルギー行列

機能 :

剛体の運動エネルギー行列を計算する。

36.7. Calc_RigidKineticOfMolecules : 剛体の各分子の運動エネルギー計算

名称 : Calc_RigidKineticOfMolecules

呼び出し形式 : function Calc_RigidKineticOfMolecules() result (kinetic)

引数 :

なし

戻り値 :

function Calc_RigidKineticOfMolecules() result (kinetic)

機能 :

剛体の各分子の運動エネルギーを計算する。

37. Calc_Torsion

ファイル名 : Torsion.f90

37.1. Calc_TorsionEnergy : torsion のエネルギー計算

名称 : Calc_TorsionEnergy

呼び出し形式 : function Calc_TorsionEnergy(Torsion, Atom, init, last) result (totalEnergy)

引数 :

type(Torsion_t), intent(in), dimension(MAX_TORSION)::Torsion	torsion 情報
type(Atom_t), intent(inout), dimension(MAX_ATOM)::Atom	原子情報
type(Rigid_t), intent(in), dimension(MAX_RIGID)::rigid	剛体情報
integer, intent(in)::init	ループ初期値
integer, intent(in)::last	ループ終値
integer, intent(in)::step	ループ増分値

戻り値 :

real*8::totalEnergy	torsion のエネルギー
---------------------	----------------

機能 :

系内の torsion のエネルギーと原子に加わる力およびビリアルを計算する。

37.2. Calc_ClusterTorsionEnergy : torsion のクラスタ間エネルギー計算

名称 : Calc_ClusterTorsionEnergy

呼び出し形式 : function Calc_ClusterTorsionEnergy(Torsion, Atom, Cluster, init, last) &
result (totalEnergy)

引数 :

type(Torsion_t), intent(in), dimension(MAX_TORSION)::Torsion	torsion 情報
type(Atom_t), intent(inout), dimension(MAX_ATOM)::Atom	原子情報
type(Cluster_t), intent(inout), dimension(MAX_Cluster)::Cluster	クラスタ情報
integer, intent(in)::init	ループ初期値
integer, intent(in)::last	ループ終値

戻り値 :

real*8, dimension(MAX_CLUSTER, MAX_CLUSTER)::totalEnergy	torsion のエネルギー
--	----------------

機能 :

系内とクラスタ間の torsion のエネルギーと原子に加わる力およびビリアルを計算する。

38. Calc_Torsional

ファイル名 : Torsional.f90

38.1. Calc_TorsionVdwEnergy : 1-4 の van der Waals エネルギー計算

名称 : Calc_TorsionVdwEnergy

呼び出し形式 : function Calc_TorsionVdwEnergy(Torsion, Atom, init, last) result &
(totalEnergy)

引数 :

type(Torsion_t), intent(in), dimension(MAX_TORSION)::Torsion	torsion 情報
type(Atom_t), intent(inout), dimension(MAX_ATOM)::Atom	原子情報
type(Rigid_t), intent(in), dimension(MAX_RIGID)::rigid	剛体情報
integer, intent(in)::init	ループ初期値
integer, intent(in)::last	ループ終値
integer, intent(in)::step	ループ増分値

戻り値 :

real*8::totalEnergy	1-4 の van der Waals エネルギー
---------------------	---------------------------

機能 :

1-4 の van der Waals エネルギーと系に加わる力およびビリアルを計算する。

38.2. Calc_TorsionEleEnergy : 1-4 の静電エネルギー計算

名称 : Calc_TorsionEleEnergy

呼び出し形式 : function Calc_TorsionEleEnergy(Torsion, Atom, init, last) result &
(totalEnergy)

引数 :

type(Torsion_t), intent(in), dimension(MAX_TORSION)::Torsion	torsion 情報
type(Atom_t), intent(inout), dimension(MAX_ATOM)::Atom	原子情報
integer, intent(in)::init	ループ初期値
integer, intent(in)::last	ループ終値

戻り値 :

real*8::totalEnergy	1-4 の静電エネルギー
---------------------	--------------

機能 :

1-4 の静電エネルギーと系に加わる力およびビリアルを計算する。

38.3. Calc_ClusterTorsionVdwEnergy : 1-4 のクラスタ間 van der Waals エネルギー計算

名称 : Calc_ClusterTorsionVdwEnergy

呼び出し形式 : function Calc_ClusterTorsionVdwEnergy(Torsion, Atom, Cluster, init, last) &
result (totalEnergy)

引数 :

type(Torsion_t), intent(in), dimension(MAX_TORSION)::Torsion	torsion 情報
type(Atom_t), intent(inout), dimension(MAX_ATOM)::Atom	原子情報
type(Cluster_t), intent(inout), dimension(MAX_CLUSTER)::Cluster	クラスタ情報
integer, intent(in)::init	ループ初期値
integer, intent(in)::last	ループ終値

戻り値 :

real*8, dimension(MAX_CLUSTER, MAX_CLUSTER)::totalEnergy
クラスタ毎の 1-4 の van der Waals エネルギー

機能 :

系内とクラスタ間の 1-4 の van der Waals エネルギーと系に加わる力を計算する。

38.4. Calc_ClusterTorsionEleEnergy : 1-4 のクラスタ間静電エネルギー計算

名称 : Calc_ClusterTorsionEleEnergy

呼び出し形式 : function Calc_ClusterTorsionEleEnergy(Torsion, Atom, Cluster, init, last) &
result (totalEnergy)

機能 :

引数 :

type(Torsion_t), intent(in), dimension(MAX_TORSION)::Torsion	torsion 情報
type(Atom_t), intent(inout), dimension(MAX_ATOM)::Atom	原子情報
type(Cluster_t), intent(inout), dimension(MAX_CLUSTER)::Cluster	クラスタ情報
integer, intent(in)::init	ループ初期値
integer, intent(in)::last	ループ終値

戻り値 :

real*8, dimension(MAX_CLUSTER)::totalEnergy
クラスタ毎の 1-4 の静電エネルギー

機能 :

系内とクラスタ間の 1-4 の静電エネルギーと系に加わる力を計算する。

39. Verlet_Method

ファイル名 : Verlet.f90

39.1. Exec_NVELeapFlogVerlet : NVE アンサンブルでの蛙とびベルレ法

名称 : Exec_NVELeapFlogVerlet

呼び出し形式 : subroutine Exec_NVELeapFlogVerlet(object, interact, restrain, &
condition, dynamics, option, &
boundary, cluster, isFirstStep)

引数 :

type(Object_t), intent(inout)::object	オブジェクト情報
type(Interact_t), intent(in)::interact	相互作用情報
type(Restrain_t), intent(inout)::restrain	拘束力情報
type(Condition_t), intent(in)::condition	実験条件情報
type(Dynamics_t), intent(inout)::dynamics	力学情報
type(Option_t), intent(in)::option	オプション情報
type(Boundary_t), intent(in)::boundary	バウンダリ情報
type(MultiCluster_t), intent(inout)::cluster	クラスタ情報
logical, intent(in)::isFirstStep	初回の Verlet の実行

戻り値 :

なし

機能 :

定温および NVE アンサンブルでの蛙とびベルレ法を 1 回実行する。

39.2. Exec_NVEVelocityVerlet : NVE アンサンブルでの速度ベルレ法

名称 : Exec_NVELeapFlogVerlet

呼び出し形式 : subroutine Exec_NVELeapFlogVerlet(object, interact, restrain, &
condition, dynamics, option, &
boundary, cluster, isFirstStep)

引数 :

type(Object_t), intent(inout)::object	オブジェクト情報
type(Interact_t), intent(in)::interact	相互作用情報
type(Restrain_t), intent(inout)::restrain	拘束力情報
type(Condition_t), intent(in)::condition	実験条件情報
type(Dynamics_t), intent(inout)::dynamics	力学情報
type(Option_t), intent(in)::option	オプション情報
type(Boundary_t), intent(in)::boundary	バウンダリ情報
type(MultiCluster_t), intent(inout)::cluster	クラスタ情報
logical, intent(in)::isFirstStep	初回の Verlet の実行

戻り値 :

なし

機能 :

定温および NVE アンサンブルでの速度ベルレ法を 1 回実行する。

39. 3. Exec_NVTLeapFlogVerlet : NVT アンサンブルでの蛙とびベルレ法

名称 : Exec_NVTLeapFlogVerlet

呼び出し形式 : subroutine Exec_NVTLeapFlogVerlet (object, interact, restrain, &
condition, dynamics, option, &
boundary, cluster, isMulti)

引数 :

type(Object_t), intent(inout)::object	オブジェクト情報
type(Interact_t), intent(in)::interact	相互作用情報
type(Restrain_t), intent(inout)::restrain	拘束力情報
type(Condition_t), intent(in)::condition	実験条件情報
type(Dynamics_t), intent(inout)::dynamics	力学情報
type(Option_t), intent(in)::option	オプション情報
type(Boundary_t), intent(in)::boundary	バウンダリ情報
type(MultiCluster_t), intent(inout)::cluster	クラスタ情報
logical, intent(in)::isMulti	マルチカノニカルのフラグ

戻り値 :

なし。

機能 :

NVT アンサンブルおよびマルチカノニカルでの蛙とびベルレ法を 1 回実行する。

39. 4. Exec_NVTVelocityVerlet : NVT アンサンブルでの速度ベルレ法

名称 : Exec_NVTVelocityVerlet

呼び出し形式 : subroutine Exec_NVTVelocityVerlet (object, interact, restrain, &
condition, dynamics, option, &
boundary, cluster, isMulti)

引数 :

type(Object_t), intent(inout)::object	オブジェクト情報
type(Interact_t), intent(in)::interact	相互作用情報
type(Restrain_t), intent(inout)::restrain	拘束力情報
type(Condition_t), intent(in)::condition	実験条件情報
type(Dynamics_t), intent(inout)::dynamics	力学情報
type(Option_t), intent(in)::option	オプション情報
type(Boundary_t), intent(in)::boundary	バウンダリ情報
type(MultiCluster_t), intent(inout)::cluster	クラスタ情報
logical, intent(in)::isMulti	マルチカノニカルのフラグ

戻り値 :

機能 :

NVT アンサンブルおよびマルチカノニカルでの速度ベルレ法を 1 回実行する。

39. 5. Scale_GradientForMultiCanonical : マルチカノニカルでの力のスケーリング

名称 : Scale_GradientForMultiCanonical

呼び出し形式 : subroutine Scale_GradientForMultiCanonical (object, dynamics)

引数 :

type(Object_t), intent(inout)::object	オブジェクト情報
type(Dynamics_t), intent(in)::dynamics	MD 情報

戻り値 :

なし

機能 :

マルチカノニカル向けに力をスケーリングする。

39.6. Exec_NPTVelocityVerlet : NPT アンサンブル計算

名称 : Exec_NPTVelocityVerlet

呼び出し形式 : subroutine Exec_NPTVelocityVerlet(obj, intr, rest, cond, dyn, option, &
bound, cluster)

引数 :

type(Object_t), intent(inout)::obj	オブジェクト情報
type(Interact_t), intent(inout)::intr	相互作用情報
type(Restrain_t), intent(inout)::rest	拘束力情報
type(Condition_t), intent(in)::cond	シミュレート条件情報
type(Dynamics_t), intent(inout)::dyn	ダイナミクス情報
type(Option_t), intent(in)::option	オプション情報
type(Boundary_t), intent(inout)::bound	バウンダリ情報
type(MultiCluster_t), intent(inout)::cluster	クラスタ情報

戻り値 :

なし

機能 :

NPTアンサンブル計算を 1 回実行する

40. Ewald_Method

ファイル名 : Ewald_Method.f90

40.1. Init_Ewald : Ewald 法初期設定

名称 : Init_Ewald

引数 :

real*8, intent(out)::maxExponent	指数部
real*8, intent(in)::realTolerance	実空間許容値
real*8, intent(inout)::recipTolerance	逆空間許容値
real*8, intent(in)::alphaEwald	Ewald パラメータ
real*8, intent(out), dimension(3,3)::lattice	逆空間格子ベクトル
real*8, intent(out)::volume	体積
integer, intent(out)::maxDimSize	最大次数
type(Boundary_t), intent(in)::boundary	バウンダリ情報

戻り値 :

なし

機能 :

Ewald 法の初期設定を行う。

- 1) セルベクトルとセル体積の算出
- 2) 最大指数の算出
- 3) 逆空間行列の算出

40.2. Calc_DirectEnergy : 実空間の静電エネルギーと力の計算

名称 : Calc_DirectEnergy

 呼び出し形式 : subroutine Init_Ewald(maxExponent, realTolerance, recipTolerance, &
 alphaEwald, lattice, volume, maxDimSize, boundary)

引数 :

real*8, dimension(3, MAX_ATOM)::cord	原子位置
real*8, dimension(3, MAX_ATOM)::grad	原子への力
real*8, dimension(MAX_ATOM)::charge	電荷
real*8::alphaEwald	Ewald パラメータ
real*8, dimension(3, 3)::virial	ビリアル
real*8::eleCoeff	静電係数
integer, dimension(MAX_ATOM)::vdwTopAtom	van der Waals 先頭原子番号
integer, dimension(MAX_ATOM)::vdwLastAtom	van der Waals 最終原子番号
integer, dimension(MAX_VDW)::vdwAtomID	van der Waals 相手原子リスト
integer, dimension(MAX_VDW)::vdwReplica	van der Waals レプリカインデックス
integer, dimension(MAX_ATOM)::hydTopAtom	水素結合最終原子番号
integer, dimension(MAX_ATOM)::hydLastAtom	水素結合先頭原子番号
integer, dimension(MAX_HYD)::hydAtomID	水素結合相手原子リスト
integer, dimension(MAX_HYD)::hydReplica	水素結合レプリカインデックス
real*8::energy	総静電エネルギー
real*8, dimension(3, -REPLICA_SIZE**3/2:REPLICA_SIZE**3/2)::replicaCord	レプリカセル座標
integer::init	ループ初期値
integer::last	ループ終値
integer::step	ループステップ

戻り値 :

なし

機能 :

相互作用テーブルを使用し、実空間での静電エネルギーと原子に加わる力を計算する。

40. 3. Calc_ReciprocalEnergy : 逆空間の静電エネルギーと力の計算

名称 : Calc_ReciprocalEnergy

呼び出し形式 : subroutine Calc_DirectEnergy(cord, grad, charge, energy, init, last, step, &
 replicaCord, eleCoeff, alphaEwald, virial, &
 vdWTopAtom, vdWLastAtom, vdWAtomID, vdWReplica, &
 hydTopAtom, hydLastAtom, hydAtomID, hydReplica)

引数 :

real*8, dimension(3, MAX_ATOM)::cord	原子座標
real*8, dimension(3, MAX_ATOM)::grad	原子への力
real*8, dimension(MAX_ATOM)::charge	電荷
real*8::alphaEwald	Ewald パラメータ
real*8::eleCoeff	静電係数
real*8, dimension(3, 3)::virial	ビリアル
integer::vecNum	ベクトル数
real*8, dimension(3, vecNum)::waveVector	周期関数ベクトル
real*8, dimension(vecNum)::cosTerm	cos 係数
real*8, dimension(vecNum)::sinTerm	sin 係数
real*8, dimension(vecNum)::expTerm	exp 係数
real*8::energy	エネルギー
real*8::volume	体積
integer::totalNum	総原子数

戻り値 :

なし

機能 :

逆空間での静電エネルギーと原子に加わる力およびビリアルを計算する。

本モジュールでは、基本セル内での 1-2, 1-3, 1-4 の静電相互作用も計算するため、(5) の Calc_ExcessEnergy で得られるエネルギーと力を減算することで補正する。

40. 4. Calc_SelfEnergy : Ewald の定数項の計算

名称 : Calc_SelfEnergy

呼び出し形式 : function Calc_SelfEnergy(totalNum, charge, eleCoeff, alphaEwald)
 result (selfEnergy)

引数 :

real*8, intent(in)::alphaEwald	Ewald パラメータ
real*8, intent(in), dimension(MAX_ATOM)::charge	電荷
real*8, intent(in)::eleCoeff	静電係数
integer, intent(in)::totalNum	総原子数

戻り値 :

real*8::selfEnergy 定数項

機能 :

Ewald 和の定数項を求める。

40. 5. Calc_ExcessEnergy : 基本セル内の結合原子間の静電エネルギー計算

名称 : Calc_ExcessEnergy

呼び出し形式 : subroutine Calc_ExcessEnergy(cord, grad, charge, energy, init, last, step, &
eleCoeff, atom, alphaEwald, virial)

引数 :

real*8, dimension(3, MAX_ATOM)::cord	原子座標
real*8, dimension(3, MAX_ATOM)::grad	原子への力
real*8, dimension(MAX_ATOM)::charge	電荷
real*8::eleCoeff	静電係数
real*8::energy	エネルギー
integer::init	ループ初期値
integer::last	ループ終値
integer::step	ループステップ
type(Atom_t), dimension(MAX_ATOM)::atom	原子情報
real*8::alphaEwald	Ewald パラメータ
real*8, dimension(3, 3)::virial	ビリアル

戻り値 :

なし

機能 :

基本セル内の 1-2, 1-3, 1-4 相互作用での静電エネルギーと力およびビリアルを計算する。

本エネルギーは基本セルに対する Ewald 和であり、このエネルギーと力を減算することで正しい結果が得られる。

40. 6. Calc_WaveVector : 周期関数ベクトル計算

名称 : Calc_WaveVector

呼び出し形式 : subroutine Calc_WaveVector(waveVector, expTerm, alphaEwald, &
maxExponent, maxDimSize, totalNum, lattice)

引数 :

real*8, intent(out), dimension(3, MAX_EWALD_3)::waveVector	周期関数ベクトル
real*8, intent(out), dimension(MAX_EWALD_3)::expTerm	指数項
real*8, intent(in)::alphaEwald	Ewald パラメータ
real*8, intent(in)::maxExponent	最大指数
integer, intent(in)::maxDimSize	格子の最大数 (x, y, z)
integer, intent(out)::totalNum	波動関数の数
real*8, intent(in), dimension(3, 3)::lattice	基本セルの逆行列

戻り値 :

なし

機能 :

Ewald の周期関数ベクトル、指数項を計算する。

40. 7. Init_PME : PME 法初期設定

名称 : Init_PME

呼び出し形式 : subroutine Init_PME (fftSize, gridSize, order, bSpline, meshIndex, tolerance, & alphaEwald, lattice, volume, boundary)

引数

integer, intent(in), dimension(3)::fftSize	FFT サイズ
integer, intent(out), dimension(3)::gridSize	グリッドサイズ
integer, intent(in)::order	B スプラインの次数
real*8, intent(inout), dimension(3, MAX_GRID)::bSpline	B スプライン
integer, intent(inout), dimension(3, MAX_GRID+MAX_ORDER+1)::meshIndex	メッシュインデックス
real*8, intent(out), dimension(3, 3)::lattice	逆空間行列ベクトル
real*8, intent(out)::volume	セル体積
real*8, intent(in)::tolerance	許容値
real*8, intent(in)::alphaEwald	Ewald パラメータ
type(Boundary_t), intent(in)::boundary	バウンダリ情報

戻り値 :

なし

機能 :

PME 法の逆空間計算の初期設定を行う。

処理を以下に示す。

- 1) セルベクトル、セル体積、逆空間ベクトルを算出する。
- 2) グリッドサイズを算出する。
- 3) B-スプライン関数の初期設定を行う。
- 4) メッシュインデックスを計算する。
- 5) FFT の初期設定を行う。

40. 8. Calc_PMEForce : PME 法での逆空間エネルギー計算

名称 : Calc_PMEForce

呼び出し形式 : subroutine Calc_PMEForce(totalNum, cord, charge, grad, energy,
 virial, eleCoeff, fraction, order,
 theta, dtheta, fftSize, gridSize, chargeGrid, bSpline,
 meshIndex, volume, alphaEwald, lattice)

引数 :

real*8, intent(in), dimension(3, MAX_ATOM)::cord	原子座標
real*8, intent(inout), dimension(3, MAX_ATOM)::grad	原子に加わる力
real*8, intent(in), dimension(MAX_ATOM)::charge	電荷
real*8, intent(in)::eleCoeff	静電係数
real*8, intent(inout), dimension(3, MAX_ATOM)::fraction	scaled fractional coordinate
real*8, intent(in), dimension(3, 3)::lattice	逆空間行列ベクトル
integer, intent(in)::totalNum	原子数
integer, intent(in), dimension(3)::fftSize	FFT サイズ
integer, intent(in), dimension(3)::gridSize	グリッドサイズ
integer, intent(in)::order	B スプラインの次数
integer, intent(in), dimension(3, fftSize(1)+order+1)::meshIndex	メッシュインデックス
real*8, intent(in)::volume	セル体積
real*8, intent(in)::alphaEwald	Ewald パラメータ
real*8, intent(out)::energy	静電エネルギー
real*8, intent(out), dimension(3, 3)::virial	ビリアル
real*8, intent(inout), dimension(3, order, MAX_ATOM)::theta	スプライン係数
real*8, intent(inout), dimension(3, order, MAX_ATOM)::dtheta	一次微分スプライン係数
real*8, intent(inout), dimension(2, gridSize(1), gridSize(2), gridSize(3))::chargeGrid	グリッドの電荷
real*8, dimension(3, fftSize(1))::bSpline	B スプラインの絶対値和

戻り値 :

なし

機能 :

PME 法で逆空間行列での静電エネルギー、原子に加わる力およびビリアルを計算する。

処理を以下に示す。

- (1) 原子への力の初期化
- (2) fraction、スプライン係数、一回微分値の算出
- (3) グリッドの電荷設定
- (4) forward FFT
- (5) エネルギー計算
- (6) backward FFT
- (7) 原子への力の計算

40. 9. Load_BSplineModuli : B スプラインの設定主制御

名称 : Load_BSplineModuli

呼び出し形式 : subroutine Load_BSplineModuli(bSpline, fftSize, order)

引数 :

integer, intent(in), dimension(3)::fftSize	FFT サイズ
integer, intent(in)::order	B スプラインの次数
real*8, intent(out)::bSpline(3, fftSize(1))	B スプラインの絶対値和

戻り値 :

なし

機能 :

B スプライン関数の絶対値和の値を設定する。

40. 10. Calc_DFTModuli : 各次元の B スプラインの設定

名称 : Calc_DFTModuli

呼び出し形式 : subroutine Calc_DFTModuli(bSpline, bsp_arr, dimSize, dim)

引数 :

integer, intent(in)::dimSize	当該次元のサイズ
real*8, intent(out)::bSpline(3, dimSize)	B スプラインの絶対値和
real*8, intent(in)::bsp_arr(dimSize)	B スプライン関数
integer, intent(in)::dim	当該次元

戻り値 :

なし

機能 : (bSpline, bsp_arr, dimSize, dim)

Distance Fourier Transform により指定次元の B スプライン関数の値を設定する。

40. 11. Get_BSplineCoeffs : B スプライン係数の計算

名称 : Get_BSplineCoeffs

呼び出し形式 : Get_BSplineCoeffs(totalNum, fraction, order, theta, dtheta)

引数 :

integer, intent(in)::totalNum	総原子数
integer, intent(in)::order	B スプラインの次数
real*8, intent(in), dimension(3, MAX_ATOM)::fraction	座標断片
real*8, intent(inout)::theta(3, order, MAX_ATOM)	B スプライン係数
real*8, intent(inout)::dtheta(3, order, MAX_ATOM)	B スプラインの一次微分

戻り値 :

なし

機能 :

Fill_Bspline3dim を呼び出し、全原子の B スプライン係数を計算する。

40. 12. Fill_Bspline3Dim : 各原子の B スプライン係数の計算

名称 : Fill_Bspline3Dim

呼び出し形式 : Fill_Bspline3Dim(floatPart, order, array, darray, atomID)

引数 :

integer, intent(in)::order	B スプラインの次数
real*8, intent(in), dimension(3)::floatPart	座標断片の小数部
real*8, intent(out), dimension(1:3, order, *)::array	B スプライン係数
real*8, intent(out), dimension(1:3, order, *)::darray	B スプラインの一次微分
integer, intent(in)::atomID	原子 ID

戻り値 :

なし

機能 :

当該原子の B スプライン係数を計算する。

40. 13. Fill_BSpline : B スプライン係数の計算

名称 : Fill_BSpline

呼び出し形式 : Fill_BSpline(floatPart, order, array, darray)

引数 :

integer, intent(in)::order	B スプラインの次数
real*8, intent(in), dimension(3)::floatPart	座標断片の小数部
real*8, intent(out), dimension(1:3, order, *)::array	B スプライン係数
real*8, intent(out), dimension(1:3, order, *)::darray	B スプラインの一次微分

戻り値 :

なし

機能 :

B スプライン係数を計算する。

40.14. Get_ScaledFractionals : 座標断片の計算

名称 : Get_ScaledFractionals

呼び出し形式 : Get_ScaledFractionals(totalNum, cord, lattice, fftSize, fraction)

引数 :

integer, intent(in)::totalNum	総原子数
integer, intent(in), dimension(3)::fftSize	FFT のサイズ
real*8, intent(in), dimension(3, MAX_ATOM)::cord	原子座標
real*8, intent(in), dimension(3, 3)::lattice	基本セルの逆行列
real*8, intent(out), dimension(3, MAX_ATOM)::fracion	座標断片

戻り値 :

なし

機能 :

原子の座標断片を計算する。

40.15. Fill_ChargeGrid : グリッドの電荷設定

名称 : Fill_ChargeGrid

呼び出し形式 : subroutine Fill_ChargeGrid(totalNum, charge, eleCoeff, theta, fraction, & order, fftSize, gridSize, chargeGrid, meshIndex)

引数 :

integer, intent(in)::totalNum	総原子数
real*8, intent(in), dimension(MAX_ATOM)::charge	原子の電荷
real*8, intent(in)::eleCoeff	静電係数
integer, intent(in)::order	B スプラインの次数
integer, intent(in), dimension(3)::fftSize	FFT のサイズ
integer, intent(in), dimension(3)::gridSize	グリッドのサイズ
real*8, intent(in), dimension(3, MAX_ATOM)::fraction	座標断片
real*8, intent(in), dimension(3, order, MAX_ATOM)::theta	B スプラインの係数
real*8, intent(out), dimension(2, gridSize(1), gridSize(2), gridSize(3))::chargeGrid	グリッドの電荷
integer, intent(in), dimension(3, fftSize(1)+order+1)::meshIndex	グリッドのインデックス

戻り値 :

なし

機能 :

グリッドの電荷を設定する。

40. 16. Sum_Energy : PME 法での逆空間静電エネルギー計算

名称 : Sum_Energy

呼び出し形式 : subroutine Sum_Energy(chargeGrid, ewaldCoeff, eleCoeff, volume, lattice, &
bSpline, fftSize, gridSize, energy, virial)

引数 :

integer, intent(in), dimension(3)::fftSize	FFT のサイズ
integer, intent(in), dimension(3)::gridSize	グリッドのサイズ
real*8, intent(inout), dimension(2, gridSize(1), gridSize(2), gridSize(3))::chargeGrid	グリッドの電荷
real*8, intent(in), dimension(3, fftSize(1))::bSpline	B スプライン
real*8, intent(in)::ewaldCoeff	Ewald パラメータ
real*8, intent(in)::eleCoeff	静電係数
real*8, intent(in)::volume	基本セルの体積
real*8, intent(in)::lattice(3, 3)	基本セルの逆行列
real*8, intent(out)::energy	静電エネルギー
real*8, intent(inout), dimension(3, 3)::virial	ビリアル

戻り値 :

なし

機能 :

PME での逆空間での静電エネルギーおよびビリアルを計算する。

40. 17. Sum_Gradient : PME 法での逆空間での原子への力の計算

名称 : Sum_Gradient

呼び出し形式 : subroutine Sum_Gradient(totalNum, charge, eleCoeff, lattice, theta, dtheta, &
grad, fraction, order, fftSize, gridSize, &
chargeGrid, meshIndex)

引数 :

integer, intent(in)::totalNum	総原子数
real*8, intent(in), dimension(MAX_ATOM)::charge	原子の電荷
real*8, intent(in)::eleCoeff	静電係数
real*8, intent(in), dimension(3, 3)::lattice	基本セルの逆行列
integer, intent(in)::order	B スプラインの次数
real*8, intent(in), dimension(3, order, MAX_ATOM)::theta	B スプラインの係数
real*8, intent(inout), dimension(3, order, MAX_ATOM)::dtheta	B スプラインの一次微分
real*8, intent(inout), dimension(3, MAX_ATOM)::grad	逆空間行列での原子への力
real*8, intent(inout), dimension(3, MAX_ATOM)::fraction	座標断片
integer, intent(in), dimension(3)::fftSize	FFT サイズ
integer, intent(in), dimension(3)::gridSize	グリッドサイズ
real*8, intent(inout), dimension(2, gridSize(1), gridSize(2), gridSize(3))::chargeGrid	グリッドの電荷
integer, intent(in), dimension(3, fftSize(1)+order+1)::meshInd	グリッドのインデックス

戻り値 :

なし

機能 :

PME での逆空間での原子への力を計算する。

40. 18. Initial_FFT : FFT の初期化

名称 : Initial_FFT

呼び出し形式 : subroutine Initial_FFT(fftSize, gridSize)

引数 :

integer, intent(in), dimension(3) :: fftSize	FFT のサイズ
integer, dimension(3) :: gridSize	グリッドサイズ

戻り値 :

なし

注意 :

FFT のライブラリの種別により、各々対応するライブラリを呼び出す。

機能 :

FFT の初期化を行う。

40. 19. FFT_Forward : フォワード FFT

名称 : FFT_Forward

呼び出し形式 : subroutine FFT_Forward(array, fftSize, gridSize)

引数 :

real*8, intent(inout), dimension(*) :: array	FFT 行列
integer, intent(in), dimension(3) :: fftSize	FFT のサイズ
integer, dimension(3) :: gridSize	グリッドサイズ

戻り値 :

なし

注意 :

FFT のライブラリの種別により、各々対応するライブラリを呼び出す。

機能 :

forward FFT を行う。

40. 20. FFT_Backward : バックワード FFT

名称 : FFT_Backward

呼び出し形式 : subroutine FFT_Backward(array, fftSize, gridSize)

引数 :

real*8, intent(inout), dimension(*) :: array	FFT 行列
integer, intent(in), dimension(3) :: fftSize	FFT のサイズ
integer, dimension(3) :: gridSize	グリッドサイズ

戻り値 :

なし

注意 :

FFT のライブラリの種別により、各々対応するライブラリを呼び出す。

機能 :

backward FFT を行う。

41.4. Init_HybridVelocity : Hybrid MD/HD 法の初期化

名称 : Init_HybridVelocity

呼び出し形式 : subroutine Init_HybridVelocity (hybrid, atom, random, initKinetic)

引数 :

type (Hybrid_t), intent (inout) :: hybrid	Hybrid 情報
type (Atom_t), intent (inout), dimension (*) :: atom	原子情報
real*4, intent (in), dimension (3, *) :: random	初期化用乱数
real*8, intent (in) :: initKinetic	初期運動エネルギー

戻り値 :

なし

機能 :

HD 部分の速度と座標の初期化を行う。

41.5. Scale_HybridVelocity : HD 部分の速度スケーリング

名称 : Scale_HybridVelocity

呼び出し形式 : subroutine Scale_HybridVelocity (rate)

引数 :

real*8, intent (in) :: rate	速度のスケールファクター
-----------------------------	--------------

戻り値 :

なし

機能 :

HD 部分の速度のスケーリングを入力した値で行う。

41.6. Set_HybridRestartVelocity : リスタート時の HD 部分の速度設定

名称 : Set_HybridRestartVelocity

呼び出し形式 : subroutine Set_HybridRestartVelocity (hybrid)

引数 :

type (Hybrid_t), intent (inout) :: hybrid	Hybrid 情報
---	-----------

戻り値 :

なし

機能 :

リスタート時の HD 部分の速度を設定する。

41.7. Init_HybridVelocity : Hybrid MD/HD 法の初期化

名称 : Init_HybridVelocity

呼び出し形式 : subroutine Init_HybridVelocity (hybrid, atom, random, initKinetic)

引数 :

type (Hybrid_t), intent (inout) :: hybrid	Hybrid 情報
type (Atom_t), intent (inout), dimension (*) :: atom	原子情報
real*4, intent (in), dimension (3, *) :: random	初期化用乱数列
real*8, intent (in) :: initKinetic	初期運動エネルギー

戻り値 :

なし

機能 :

HD 部分の速度と座標の初期化を行う。

41.8. Calc_HybridEnergy : HD 部分のエネルギー計算

名称 : Calc_HybridEnergy

呼び出し形式 : function Calc_HybridEnergy (hybrid) result(energy)

引数 :

type(Hybrid_t), intent(inout)::hybrid Hybrid 情報

戻り値 :

なし

機能 :

HD 部分のポテンシャルエネルギーを計算する。

41.9. Calc_HybridVelocity : HD 部分の速度計算

名称 : Calc_HybridVelocity

呼び出し形式 : subroutine Calc_HybridVelocity(vel, dt, scale, integration, atom, hybrid)

引数 :

real*8, intent(out), dimension(3, MAX_ATOM)::vel	MD 部分の原子の速度
real*8, intent(in)::dt	タイムステップ
real*8, intent(in)::scale	NVT での速度スケールファクター
integer, intent(in)::integration	積分器の種別
type(Atom_t), intent(in), dimension(MAX_ATOM)::atom	MD 部分の原子情報
type(Hybrid_t), intent(inout)::hybrid	Hybrid 情報

戻り値 :

なし

機能 :

HD 部分の速度を計算する。

温度一定の場合は HD 部分の速度のスケーリングを行う。

41.10. Calc_HybridCoordinate : HD 部分の座標計算

名称 : Calc_HybridCoordinate

呼び出し形式 : subroutine Calc_HybridCoordinate (cord, dt, hybrid)

引数 :

real*8, intent(inout), dimension(3, MAX_ATOM)::cord	MD 部分の原子の座標
real*8, intent(in)::dt	タイムステップ
type(Hybrid_t), intent(inout)::hybrid	Hybrid 情報

戻り値 :

なし

機能 :

HD 部分の座標を計算する。

41.11. Backup_HybridQuantities : HD 部分の物理量のバックアップ

名称 : Backup_HybridQuantities

呼び出し形式 : subroutine Backup_HybridQuantities

引数 :

なし

戻り値 :

なし

機能 :

HD 部分の速度のバックアップを行う。

41.12. Rollback_HybridVelocity : HD 部分の速度のロールバック

名称 : Rollback_HybridQuantities

呼び出し形式 : subroutine Rollback_HybridQuantities

引数 :

なし

戻り値 :

なし

機能 :

HD 部分の速度を、1 ステップ前の状態に戻す。

41.13. Add_HybridMoment : 回転モーメントへの HD 部分の加算

名称 : Add_HybridMomen

呼び出し形式 : subroutine Add_HybridMomen(ixx, iyy, izz, ixy, iyz, izx, massCenter)

引数 :

real*8, intent(inout)::ixx	回転モーメントの xy 成分
real*8, intent(inout)::iyy	回転モーメントの yy 成分
real*8, intent(inout)::izz	回転モーメントの zz 成分
real*8, intent(inout)::ixy	回転モーメントの xy 成分
real*8, intent(inout)::iyz	回転モーメントの yz 成分
real*8, intent(inout)::izx	回転モーメントの zx 成分
real*8, intent(in), dimension(3)::massCenter	系の重心

戻り値 :

なし

機能 :

入力した回転モーメントに HD 部分のモーメントを加算する。

41.14. Remove_HybridTotalVelocity : HD 部分の系全体の回転、並進の削除

名称 : Remove_HybridTotalVelocity

呼び出し形式 : subroutine Remove_HybridTotalVelocity(trans, rotate, massCenter)

引数 :

real*8, intent(in), dimension(3)::trans	並進運動の補正值
real*8, intent(in), dimension(3)::rotate	回転運動の補正值
real*8, intent(in), dimension(3)::massCenter	系の重心

戻り値 :

なし

機能 :

入力した値で HD 部分の系全体の回転、並進の削除を行う。

41.15. Get_HybridTrans : HD 部分の並進速度計算

名称 : Get_HybridTrans

呼び出し形式 : function Get_HybridTrans() result(trans)

引数 :

なし

戻り値 :

real*8, dimension(3)::trans HD 部分の系の並進運動

機能 :

HD 部分の並進運動を計算する。

41.16. Get_HybridRotate : HD 部分の回転速度計算

名称 : Get_HybridRotate

呼び出し形式 : function Get_HybridRotate (massCenter, matrix) result(trans)

引数 :

real*8, intent(in), dimension(3)::massCenter 系の重心

real*8, intent(in), dimension(3,3)::matrix 回転運動のマトリックス

戻り値 :

real*8, dimension(3)::rotate 系全体の回転運動

機能 :

MD 部分の回転運動に HD 部分の回転運動を加算する。

41.17. Calc_HybridKinetic : HD 部分の運動エネルギー計算

名称 : Get_HybridTrans

呼び出し形式 : function Get_HybridTrans(integration) result(kinetic)

引数 :

integer, intent(in)::integration 積分器種別

戻り値 :

real*8::kinetic HD 部分の系の運動エネルギー

機能 :

HD 部分の運動エネルギーを計算する。

42. Barostat_Method

ファイル名 : Barostat.f90

42. 1. Exec_AndersenMethod : Andersen の圧力制御

名称 : Exec_AndersenMethod

呼び出し形式 : subroutine Exec_AndersenMethod(vel, dyn, cond, atomInfo, size, rigidInfo)

引数 :

real*8, dimension(3, MAX_ATOM), intent(inout)::vel	自由原子の速度
type(Dynamics_t), intent(inout)::dyn	ダイナミクス情報
type(Condition_t), intent(in)::cond	シミュレート条件情報
type AtomInfo_t, intent(in)::atomInfo	原子情報
real*8, dimension(3), intent(in)::size	セルサイズ
type(RigidInfo_t), intent(inout)::rigidInfo	剛体情報

戻り値 :

なし

機能 :

Andersen の圧力制御を行い、自由原子の速度をスケーリングする。
処理を以下に示す。

- (1) 仮想粒子の質量、力、速度および座標を求める。
- (2) 仮想系の運動方程式を解き、自由原子の速度をスケーリングする。
- (3) 仮想粒子のハミルトニアンを計算する。

42. 2. Exec_ParrinelloRahmanMethod : Parrinello-Rahman の圧力制御

名称 : Exec_ParrinelloRahmanMethod

呼び出し形式 : subroutine Exec_ParrinelloRahmanMethod(vel, dyn, cond, atomInfo, edge, &
rigidInfo)

引数 :

real*8, dimension(3, MAX_ATOM), intent(inout)::vel	自由原子の速度
type(Dynamics_t), intent(inout)::dyn	ダイナミクス情報
type(Condition_t), intent(in)::cond	シミュレート条件情報
type AtomInfo_t, intent(in)::atomInfo	原子情報
real*8, dimension(3, 3), intent(inout)::edge	セル行列
type(RigidInfo_t), intent(inout)::rigidInfo	剛体情報

戻り値 :

なし

機能 :

Parrinello-Rahman の圧力制御を行い、自由原子の速度をスケーリングする。
処理を以下に示す。

- (1) 仮想粒子の質量、力、速度および座標を求める。
- (2) 仮想系の運動方程式を解き、自由原子の速度をスケーリングする。
- (3) 仮想粒子のハミルトニアンを計算する。

42.3. Exec_BerendsenBarostat : Berendsen の圧力制御

名称 : Exec_BerendsenBarostat

呼び出し形式 : subroutine Exec_BerendsenBarostat(cord, mass, obj, dyn, ktensor, &
bound, nonBondedInfo)

引数 :

real*8, dimension(3, MAX_ATOM), intent(inout)::cord	原子座標
real*8, dimension(MAX_ATOM), intent(in)::mass	質量
type(Object_t), intent(inout)::obj	オブジェクト情報
type(Dynamics_t), intent(inout)::dyn	ダイナミクス情報
real*8, dimension(3, 3), intent(in)::ktensor	運動エネルギーテンソル
type(Boundary_t), intent(inout)::bound	境界条件情報
type(NonBondedInfo_t), intent(inout)::nonBondedInfo	非結合相互作用情報

戻り値 :

なし

機能 :

Berendsen の圧力制御を行い、原子座標およびセル情報の更新を行う。
処理を以下に示す。

- (1) 圧力およびスケールリングファクターを算出する。
- (2) セル情報の更新を行う。
- (3) 原子座標の更新を行う。

42.4. Exec_BerendsenThermostat : Berendsen の温度制御

名称 : Exec_BerendsenMethod

呼び出し形式 : subroutine Exec_BerendsenThermostat (vel, settingTemp, dyn, obj)

引数 :

real*8, intent(inout), dimension(3, MAX_ATOM)::vel	自由原子速度
real*8, intent(in)::settingTemp	設定温度
type(Dynamics_t), intent(inout)::dyn	ダイナミクス情報
type(Object_t), intent(inout)::obj	オブジェクト情報

戻り値 :

なし

機能 :

Berendsen の温度制御を行い、原子速度の更新を行う。
処理を以下に示す。

- (1) スケールリングファクターを算出する。
- (2) 原子速度の更新を行う。

43. Control_Dynamics

ファイル名 : Control_Dynamics.f90

43.1. Exec_Dynamics : MD の 1 ステップ実行

名称 : Exec_Dynamics

呼び出し形式 : subroutine Exec_Dynamics(obj, intr, rest, cond, dyn, optn,
bound, cluster)

引数 :

type(Object_t), intent(inout)::obj	! オブジェクト情報
type(Interact_t), intent(inout)::intr	! 相互作用情報
type(Restraining_t), intent(inout)::rest	! 拘束力情報
type(Condition_t), intent(inout)::cond	! シミュレート条件情報
type(Dynamics_t), intent(inout)::dyn	! MD 情報

type(Option_t), intent(in)::optn	! オプション情報
type(Boundary_t), intent(in)::bound	! バウンダリ情報
type(MultiCluster_t), intent(inout)::cluster	! クラスタ情報

戻り値 :

なし

機能 :

アンサンブル、積分器の組み合わせに従い、それぞれの MD1 ステップを実行する手続きを呼び出す。

43.2. Backup_PreviousQuantities : 物理量のバックアップ

名称 : Backup_PreviousQuantities

呼び出し形式 : subroutine Backup_PreviousQuantities (obj, dyn)

引数 :

type(Dynamics_t), intent(inout)::dyn	! MD 情報
type(Object_t), intent(inout)::obj	! オブジェクト情報

戻り値 :

なし

機能 :

直前の MD ステップでの物理量を保存する。

44. Thermostat_Method

ファイル名 : Thermostat.f90

44. 1. Scale_Velocity : 温度制御の主制御

名称 : Scale_Velocity

呼び出し形式 : subroutine Scale_Velocity(vel, preVel, mass, freeNum, targetTemp, obj, dyn)

引数 :

real*8, intent(inout), dimension(3, MAX_ATOM)::vel	自由原子の速度
real*8, intent(in), dimension(3, MAX_ATOM)::preVel	直前のステップでの自由原子速度
real*8, intent(in), dimension(MAX_ATOM)::mass	原子の質量
integer, intent(in)::freeNum	自由原子数
real*8, intent(in)::targetTemp	シミュレーション温度
type(Object_t), intent(inout)::obj	オブジェクト情報
type(Dynamics_t), intent(inout)::dyn	MD 情報

戻り値 :

なし

機能 :

各温度制御種別に従い、温度制御処理を呼び出す。

44. 2. Exec_BerendsenThermostat : Berendsen の温度制御

名称 : Exec_BerendsenThermostat

呼び出し形式 : subroutine Exec_BerendsenThermostat(vel, settingTemp, currentTemp, & dt, taut)

引数 :

real*8, intent(inout), dimension(3, MAX_ATOM)::vel	自由原子の速度
real*8, intent(in)::settingTemp	シミュレートの設定温度
real*8, intent(in)::currentTemp	現在の系の温度
real*8, intent(in)::dt	MD 一回での時間増加
real*8, intent(in)::taut	緩衝時間

戻り値 :

なし

機能 :

Berendsen の温度制御を使用して、速度をスケーリングする。

44. 3. Exec_NoseHooverThermostat : 能勢-Hoover の温度制御

名称 : Exec_NoseHooverThermostat

引数 :

real*8, dimension(3, MAX_ATOM), intent(inout)::vel	自由原子速度
integer, intent(in)::freeNum	自由原子数
real*8, intent(in)::temp	スケーリング温度
integer, intent(in)::freedom	系全体の自由度
type(NoseHoover_t), intent(inout)::nose	能瀬-Hoover 情報
real*8, intent(in)::dt	MD の 1 ステップ時間

戻り値 :

なし

機能 :

能瀬-Hoover の温度制御を行い、自由原子の速度をスケーリングする。

処理を以下に示す。

- (1) 仮想粒子のモーメントと質量を求める。
- (2) 仮想粒子のパラメータで自由原子の速度をスケーリングする。
- (3) 速度スケーリング後の仮想粒子のモーメントと仮想粒子のハミルトニアンを計算する。

44. 4. Exec_EachMoleculeNoseHoover : 分子単位の能勢-Hoover の温度制御

名称 : Exec_EachMoleculeNoseHoover

引数 :

real*8, intent(inout), dimension(3, MAX_ATOM)::vel	自由原子速度
integer, intent(in)::freeNum	自由原子数
real*8, intent(in)::temp	スケーリング温度
integer, intent(in)::freedom	系全体の自由度
type(NoseHoover_t), intent(inout)::nose	能瀬-Hoover 情報
real*8, intent(in)::dt	MD の 1 ステップ時間

戻り値 :

なし

機能 :

各分子に対し、能瀬-Hoover の温度制御を行い、自由原子の速度をスケーリングする。

Exec_NoseHooverThermostat では系全体の温度スケーリングであるため、仮想粒子が 1 つであるのに対し、本モジュールでは各分子に熱浴を用意する。

処理を以下に示す。

- (1) 各分子の仮想粒子のモーメントと質量を求める。
- (2) 仮想粒子のパラメータで各分子の自由原子の速度をスケーリングする。
- (3) 速度スケーリング後の仮想粒子のモーメントを計算する。

44. 5. Exec_ShakeConvergeLoop : SHAKE ループ実行

名称 : Exec_ShakeConvergeLoop

呼び出し形式 : subroutine Exec_ShakeConvergeLoop(cord, preCord, vel, preVel, &
mass, object, dynamics)

引数 :

real*8, intent(inout), dimension(3, MAX_ATOM)::cord	現時刻の原子位置
real*8, intent(in), dimension(3, MAX_ATOM)::preCord	直前の時刻の原子位置
real*8, intent(inout), dimension(3, MAX_ATOM)::vel	自由原子の速度
real*8, intent(in), dimension(3, MAX_ATOM)::preVel	直前の時刻の自由原子の速度
real*8, intent(in), dimension(MAX_ATOM)::mass	原子の質量
type(Object_t), intent(in)::object	オブジェクト情報
type(Dynamics_t), intent(in)::dynamics	MD 情報

戻り値 :

なし

機能 :

SHAKE の実行において、拘束後の温度が収束までループを繰り返す。

44. 6. Exec_HooverEvansThermostat : Hoover-Evans の温度スケーリング

名称 : Exec_HooverEvansThermostat

呼び出し形式 : Exec_HooverEvansThermostat (vel, refVel, velDiff, &
freeNum, object, dynamics)

引数 :

real*8, intent(inout), dimension(3, MAX_ATOM)::vel	自由原子の速度
real*8, intent(in), dimension(3, MAX_ATOM)::refVel	当該時刻の自由原子の速度
real*8, intent(in), dimension(3, MAX_ATOM)::velDiff	速度の差分
integer, intent(in)::freeNum	自由原子数
type(Object_t), intent(in)::object	オブジェクト情報
type(Dynamics_t), intent(in)::dynamics	MD 情報

戻り値 :

なし

機能 :

シミュレート温度に合うように原子の速度をスケーリングする。

当該時刻の速度で温度を求め、この温度と目標の温度で当該時刻の速度をスケーリングする。

44. 7. Scale_VelocityWithShakeGradient : SHAKE 向け温度スケーリング

名称 : Scale_VelocityWithShakeGradient

呼び出し形式 : subroutine Scale_VelocityWithShakeGradient(vel, refVel, velDiff, &
freeNum, object, dynamics)

引数 :

real*8, intent(inout), dimension(3, MAX_ATOM)::vel	自由原子の速度
real*8, intent(in), dimension(3, MAX_ATOM)::refVel	当該時刻の自由原子の速度
real*8, intent(in), dimension(3, MAX_ATOM)::grad	原子に加わる力
integer, intent(in)::freeNum	自由原子数
type(Object_t), intent(in)::object	オブジェクト情報
type(Dynamics_t), intent(in)::dynamics	MD 情報

戻り値 :

なし

機能 :

シミュレート温度に合うように原子の速度をスケーリングする。

当該時刻の速度で温度を求め、この温度と目標の温度で当該時刻の速度をスケーリングする。

44. 8. Is_ConvergeTemperature : 収束温度判定

名称 : Is_ConvergeTemperature

呼び出し形式 : function Is_ConvergeTemperature(vel, freeNum, dynamics) result (isConverge)

引数 :

real*8, intent(inout), dimension(3, MAX_ATOM)::vel	自由原子の速度
integer, intent(in)::freeNum	自由原子数
type(Dynamics_t), intent(in)::dynamics	MD 情報

戻り値 :

logical::isConverge

収束フラグ

機能 :

系の当該温度が収束条件内にあるかを判定する。

45. Rigid_Method

ファイル名 : Rigid_Method.f90

45.1. Calc_RigidForce : 剛体の力の計算(並進+回転)

名称 : Calc_RigidForce

呼び出し形式 : subroutine Calc_RigidForce(atom, rigid, totalNum)

引数 :

type(Atom_t), intent(in), dimension(MAX_ATOM)::atom	原子情報
type(Rigid_t), intent(inout), dimension(MAX_RIGID)::rigid	剛体情報
integer, intent(in)::totalNum	剛体数

戻り値 :

なし

機能 :

全ての剛体に対し、構成原子の力を剛体の力に分離する処理 (Calc_TorqueAndForce) を呼び出す。

45.2. Calc_RigidVelocity : 剛体の速度計算

名称 : Calc_RigidVelocity

呼び出し形式 : subroutine Calc_RigidVelocity(vel, dt, freeID, rigid, totalNum)

引数 :

real*8, intent(in)::dt	タイムステップ
type(Rigid_t), intent(inout), dimension(MAX_RIGID)::rigid	剛体情報
integer, intent(in)::totalNum	剛体数

戻り値 :

なし

機能 :

全ての剛体に対し、並進速度と角運動量の更新を行う。

45.3. Calc_RigidAngularMomentum : 剛体の角運動量補正

名称 : Calc_RigidAngularMomentum

呼び出し形式 : subroutine Calc_RigidAngularMomentum(dt, rigid, totalNum, isReverse)

引数 :

real*8, intent(in)::dt	タイムステップ
type(Rigid_t), intent(inout), dimension(MAX_RIGID)::rigid	剛体情報
integer, intent(in)::totalNum	剛体数
logical, intent(in)::isReverse	角運動量計算前に速度を計算

戻り値 :

なし

機能 :

全剛体に対し、角運動量補正処理を呼び出す。

45.4. Calc_RigidCoordinate : 剛体の座標計算

名称 : Calc_RigidCoordinate

呼び出し形式 : subroutine Calc_RigidCoordinate(cord, dt, rigid, totalNum)

引数 :

real*8, intent(out), dimension(3, MAX_ATOM)::cord	原子座標
real*8, intent(in)::dt	タイムステップ
type(Rigid_t), intent(inout), dimension(MAX_RIGID)::rigid	剛体情報
integer, intent(in)::totalNum	剛体数

戻り値 :

なし

機能 :

全剛体に対し、剛体角の更新、剛体の位置更新、構成原子の座標更新を行う。

45.5. Calc_TorqueAndForce : 剛体への力の分離(並進+回転)

名称 : Calc_TorqueAndForce

呼び出し形式 : subroutine Calc_TorqueAndForce(rigid, atom)

引数 :

type(Rigid_t), intent(inout)::rigid	剛体情報
type(Atom_t), intent(in), dimension(MAX_ATOM)::atom	原子情報

戻り値 :

なし

機能 :

当該剛体に対し、構成原子の力の総和を並進と回転への力に分離する。

45.6. Convert_Torque : トルクの補正

名称 : Convert_Torque

呼び出し形式 : function Convert_Torque(quarter, torque) result (result)

引数 :

real*8, intent(in), dimension(0:3)::quarter	剛体角(4元数表現)
real*8, intent(in), dimension(3)::torque	直交座標系でのトルク

戻り値 :

なし

機能 :

直交座標系でのトルクを、当該剛体角の向きに補正する。

45.7. Calc_AngularPhase : 剛体角の計算

名称 : Calc_AngularPhase

呼び出し形式 : subroutine Calc_AngularPhase(quarter, moment, inertia, dt)

引数 :

real*8, intent(inout), dimension(0:3)::quarter	剛体角(4元数表現)
real*8, intent(in), dimension(3)::moment	角運動量
real*8, dimension(3)::inertia	慣性モーメント
real*8, intent(in)::dt	タイムステップ

戻り値 :

なし

機能 :

タイムステップ後の剛体角を計算する。

45.8. Calc_AngularMomentum : 剛体の角運動量補正

名称 : Calc_AngularMomentum

呼び出し形式 : Calc_AngularMomentum(rigid, dt, isExact, isReverse)

引数 :

type(Rigid_t), intent(inout)::rigid	剛体情報
real*8, intent(in)::dt	タイムステップ
logical, intent(in)::isExact	厳密解=true
logical, intent(in)::isReverse	角運動量計算前に速度を計算

戻り値 :

なし

機能 :

剛体の形状(線形、対称、非対称)にあわせて、角運動量の補正を行う。

45.9. Calc_EllipticFunction : ヤコビの楕円関数計算

名称 : Calc_EllipticFunction

呼び出し形式 : subroutine Calc_EllipticFunction(uu, emmc, sn, cn, dn)

引数 :

real*8, intent(in)::uu	楕円関数項
real*8, intent(in)::emmc	楕円関数項
real*8, intent(out)::sn	楕円関数の結果(sn)
real*8, intent(out)::cn	楕円関数の結果(cn)
real*8, intent(out)::dn	楕円関数の結果(dn)

戻り値 :

なし

機能 :

以下の式を満たす sn, cn, dn を求める。

$$u = F(\phi, k)$$

$$\text{SIN}(\phi) = \text{sn}(u, k)$$

$$\text{sn}^2 + \text{cn}^2 = 1$$

$$k^2 \text{sn}^2 + \text{dn}^2 = 1$$

※Press, W.H. et al (1998) Numerical Recipes in Fortran Second Edition 261 page
(Jacobian Elliptic Functions) を引用

45.10. Calc_RigidVirialTensor : 剛体のビリアル計算

名称 : Calc_RigidVirialTensor

呼び出し形式 : function Calc_RigidVirialTensor(atomInfo, rigidInfo, nonBonded, &
optn, cutlen, vdwReplica, hydReplica)

引数 :

type(AtomInfo_t), intent(in)::atomInfo	原子情報
type(RigidInfo_t), intent(in)::rigidInfo	剛体情報
type(Option_t), intent(in)::optn	オプション情報
integer, intent(in)::residNum	残基数
real*8, intent(in)::cutlen	カットオフ長
type(NonBondedInfo_t), intent(in)::nonBonded	1-5 相互作用情報

戻り値 :

なし

機能 :

剛体に掛かる力に関するビリアルを計算する。

46. FMM_Method

ファイル名 : FMM_Method.f90

46.1. System_Size : 系の大きさの計算

名称 : System_Size

呼び出し形式 : subroutine System_Size(totalNum, cord, length, center)

引数 :

integer, intent(in)::totalNum	原子数
real*8, dimension(3, totalNum)::cord	原子座標
real*8, intent(out)::length	系を囲む立方体の 1 辺の長さ
real*8, intent(out), dimension(3)::center	系の中心座標

戻り値 :

なし

機能 :

当該の系の原子位置から、系を囲む立方体の 1 辺の長さと中心位置を計算する。

46.2. Create_CellGeometry : セルの中心座標計算

名称 : Create_CellGeometry

呼び出し形式 : subroutine Create_CellGeometry(length, center)

引数 :

real*8, intent(in)::length	系を囲む立方体の 1 辺の長さ
real*8, intent(in)::center(3)	系の中心座標

戻り値 :

なし

機能 :

FMM の各ツリーレベルに対して、分割したセルの中心座標を計算する。

46.3. Downward_Local : 局所展開係数計算の制御

名称 : Downward_Local

呼び出し形式 : Downward_Local

引数 :

integer, intent(in)::totalNum	原子数
real*8, intent(in), dimension(3, totalNum)::cord	原子座標
real*8, intent(in)::length	系を囲む立方体の 1 辺の長さ
real*8, intent(in), dimension(3)::center	center of the system

戻り値 :

なし

機能 :

セルの局所展開の係数(localReal, localImag)計算の制御を行う。

- 1) セルの局所展開係数をクリアする。
- 2) 各ツリーレベルの局所展開係数と多重極展開係数を計算する。
 - 2-1) 0 でない多重極展開に対する局所展開の係数を計算する。
 - 2-2) 当該セルの child セルに対し、局所展開の係数を計算する。

46.4. Calc_LocalToLocal : 局所展開係数計算

名称 : Calc_LocalToLocal

呼び出し形式 : subroutine Calc_LocalToLocal (parent, child)

引数 :

integer, intent(in)::parent	parent セルの ID
integer, intent(in)::child	child セルの ID

戻り値 :

なし

機能 :

parent-child セル間の局所展開の係数 (localReal, localImag) を計算する。

- 1) parent-child セル間の中心座標距離を計算する。
- 2) xy 平面に対するセル間の角度での Legendre 陪多項式を求める。
- 3) 多重極展開数分の SIN, COS 係数を計算する。
- 4) 局所展開の係数を計算する。

46.5. Upward_Multipole : 多重極展開計算の制御

名称 : Upward_Multipole

呼び出し形式 : subroutine Upward_Multipole (totalNum, cord, charge)

引数 :

integer, intent(in)::totalNum	原子数
real*8, intent(in), dimension(3, totalNum)::cord	原子座標
real*8, intent(in), dimension(totalNum)::charge	原子電荷

戻り値 :

なし

機能 :

多重極モーメント計算の制御を行う。

- 1) 各セルの多重極モーメント (multiReal, multiImag) をクリアする。
- 2) 原子が所属するセルに寄与する多重極展開の係数を加算する。
- 3) child セルからの多重極展開の係数を加算する。

46.6. Calc_PolynomialToMultipole : 多重極モーメント計算

名称 : Calc_PolynomialToMultipole

呼び出し形式 : subroutine Calc_PolynomialToMultipole (cord, charge, cellID)

引数 :

real*8, intent(in), dimension(3)::cord	原子座標
real*8, intent(in)::charge	原子電荷
integer, intent(in)::cellID	セルの ID

戻り値 :

なし

機能 :

当該原子がセルに寄与する多重極モーメントを計算する。

- 1) セルの中心と原子の距離を計算する。
- 2) xy 平面に対するセルと原子の角度での Legendre 陪多項式を求める。
- 3) 多重極モーメントを計算し、当該セルの多重極展開係数に加算する。

46.7. Calc_MultipoleToMultipole : 多重極モーメントの展開

名称 : Calc_MultipoleToMultipole

呼び出し形式 : subroutine Calc_MultipoleToMultipole(child, parent)

引数 :

integer, intent(in)::child	child セルの ID
integer, intent(in)::parent	parent セルの ID

戻り値 :

なし

機能 :

child セル、parent セル間の多重極モーメントを計算し、parent セルの多重極係数に加算する。

- 1) child セル、parent セル間の距離を計算する。
- 2) xy 平面に対するセルと原子の角度での Legendre 陪多項式を求める
- 3) 多重極展開数分の SIN, COS 係数を計算する。
- 4) 多重極モーメントを計算し、parent セルの多重極展開係数に加算する。

46.8. Update_FMMCell : FMM のセル情報の更新

呼び出し形式 : subroutine Update_FMMCell(totalNum)

引数 :

integer, intent(in)::totalNum	原子数
-------------------------------	-----

戻り値 :

なし

機能 :

原子の座標で変化する項目を計算する処理を呼び出す。

- 1) 立方体での系の長さを中心座標を計算する。
- 2) 分割したセルの座標を計算する。
- 3) 各セルに原子を分配する。
- 4) 多重極展開係数を計算する。
- 5) 局所展開係数を計算する。

46.9. Calc_LegendrePolynomial : Legendre 陪多項式計算

名称 : Calc_LegendrePolynomial

呼び出し形式 : subroutine Calc_LegendrePolynomial(legend, x)

引数 :

real*8 legend(0:MP, 0:MP), x	P, x (Legendre 陪多項式 : $P[mp][mp](x)$)
------------------------------	--

戻り値 :

なし

機能 :

Legendre 陪多項式の結果を計算する。

46.10. Calc_MultipoleToLocal : 多重極局所展開

名称 : Calc_MultipoleToLocal

呼び出し形式 : subroutine Calc_MultipoleToLocal (multi, local)

引数 :

integer, intent(in)::multi	多重極展開側のセルの ID
integer, intent(in)::local	局所展開側のセルの ID

戻り値 :

なし

機能 :

多重極展開側の係数を局所展開側の係数に反映させる。

- 1) 多重極展開側と局所展開側のセルの距離を計算する。
- 2) Legendre 陪多項式を計算する。
- 3) Legendre 陪多項式を更新する。(factor = 1/dist**(n+1))
- 4) 多重極展開数分の SIN, COS 係数を計算する。
- 5) 局所展開の係数を更新する。

46.11. Calc_DirectForceInCell : 同一セル間相互作用直接計算

名称 : Calc_DirectForceInCell

呼び出し形式 : subroutine Calc_DirectForceInCell (cord, charge, pot, grad, cell)

引数 :

real*8, intent(in), dimension(3, MAX_ATOM)::cord	原子の座標
real*8, intent(out), dimension(3, MAX_ATOM)::grad	原子への力
real*8, intent(in), dimension(MAX_ATOM)::charge	原子の電荷
real*8, intent(out), dimension(MAX_ATOM)::pot	各原子のポテンシャル
integer, intent(in)::cell	原子の所属するセル

戻り値 :

なし

機能 :

当該セルに所属する原子間の 1-5 静電相互作用を計算する。

46.12. Calc_DirectForceNeighborCell : 隣接セル間相互作用直接計算

名称 : Calc_DirectForceNeighborCell

呼び出し形式 : subroutine Calc_DirectForceNeighborCell (cord, charge, pot, grad, cellID1)

引数 :

real*8, intent(in), dimension(3, MAX_ATOM)::cord	原子の座標
real*8, intent(out), dimension(3, MAX_ATOM)::grad	原子への力
real*8, intent(in), dimension(MAX_ATOM)::charge	原子の電荷
real*8, intent(out), dimension(MAX_ATOM)::pot	各原子のポテンシャル
integer, intent(in)::cell1	原子の所属するセル

戻り値 :

なし

機能 :

当該セルに所属する原子と隣接するセルに所属する原子との間の 1-5 静電相互作用を計算する。

46. 13. Calc_LocalForce : 非隣接セルからの相互作用計算

名称 : Calc_LocalForce

呼び出し形式 : subroutine Calc_LocalForce(cord, charge, pot, grad, cellID)

引数 :

real*8, intent(in), dimension(3, MAX_ATOM)::cord	原子の座標
real*8, intent(out), dimension(3, MAX_ATOM)::grad	原子への力
real*8, intent(in), dimension(MAX_ATOM)::charge	原子の電荷
real*8, intent(out), dimension(MAX_ATOM)::pot	各原子のポテンシャル
integer, intent(in)::cell	原子の所属するセル

戻り値 :

なし

機能 :

当該セルに所属する原子への非隣接セルからの 1-5 静電相互作用を計算する。

46. 14. Calc_FMMForce : FMM での静電相互作用計算の主制御

名称 : Calc_FMMForce

呼び出し形式 : subroutine Calc_FMMForce(totalNum, cord, charge, grad, energy)

引数 :

integer, intent(in)::totalNum	原子の個数
real*8, intent(in), dimension(3, totalNum)::cord	原子座標
real*8, dimension(totalNum)::charge	原子の電荷
real*8, intent(out), dimension(3, totalNum)::grad	原子への力
real*8, intent(out)::energy	1-5 静電ポテンシャル

戻り値 :

なし

機能 :

FMM の各手続きを呼び出し、1-5 静電の原子への力とポテンシャルを計算する。

- ・ Calc_DirectForceInCell
- ・ Calc_DirectForceNeighborCell
- ・ Calc_MultipoleToLocal
- ・ Calc_LocalForce

46. 15. Calc_InteractCellList : 隣接セルリストの作成主制御

名称 : Calc_InteractCellList

呼び出し形式 : Calc_InteractCellList

引数 :

なし

戻り値 :

なし

機能 :

child-parent セル間、同一レベルのセル間で、隣接したセルのリストを作成する。

46. 16. Set_InteractIndex : 隣接セルリストへのセルの登録

名称 : Set_InteractIndex

呼び出し形式 : subroutine Set_InteractIndex(index, cellID, level,
initX, lastX, initY, lastY, initZ, lastZ)

引数 :

integer, intent(inout)::index	隣接セルテーブルのインデックス
integer, intent(in)::cellID	当該セルの ID
integer, intent(in)::level	対象ベル
integer, intent(in)::initX	対象レベルのセルの x インデックス初期値
integer, intent(in)::lastX	対象レベルのセルの x インデックス終値
integer, intent(in)::initY	対象レベルのセルの y インデックス初期値
integer, intent(in)::lastY	対象レベルのセルの y インデックス終値
integer, intent(in)::initZ	対象レベルのセルの z インデックス初期値
integer, intent(in)::lastZ	対象レベルのセルの z インデックス終値

戻り値 :

なし

機能 :

対象レベルの (initX:lastX, initY:lastY, initZ:lastZ) 内にあるセルを、当該セルの隣接セルとして interactIndex に登録する。

46. 17. Calc_Coefficients : 多重極展開係数の計算

名称 : Calc_Coefficients

呼び出し形式 : subroutine Calc_Coefficients(multipole)

引数 :

integer, intent(in)::multipole	多重極展開数
--------------------------------	--------

戻り値 :

なし

機能 :

以下の係数を計算する。

- ・ sphere : 楕円
- ・ transMM : 多重極展開-多重極展開
- ・ transML : 多重極展開-局所展開
- ・ transLL : 局所展開-局所展開

46. 18. Init_FMM : FMM の初期処理

名称 : Init_FMM

呼び出し形式 : subroutine Init_FMM(level, multiPole, processID)

引数 :

integer, intent(in)::level	ツリーレベル
integer, intent(in)::multiPole	多重極展開数
integer, intent(in)::processID	CPU 数

戻り値 :

なし

機能 :

FMM の初期化を行う。

- ・ FMM で使用する変数の領域確保
- ・ 各レベルの先頭セルのインデックス (TopCells) 作成
- ・ 隣接セルの計算呼び出し
- ・ 係数の作成呼び出し

46. 19. Final_FMM : FMM の終了処理

呼び出し形式 : subroutine Final_FMM

引数 :

なし

戻り値 :

なし

機能 :

FMM 用に確保した領域を解放する。

46. 20. Calc_FMMExcessEnergy : 結合原子間の相互作用計算

名称 : Calc_FMMExcessEnergy

呼び出し形式 : subroutine Calc_FMMExcessEnergy(cord, grad, charge, energy, init, last,
step, eleCoeff, atom)

引数 :

real*8, intent(in), dimension(3, MAX_ATOM)::cord	原子座標
real*8, intent(out), dimension(3, MAX_ATOM)::grad	原子への力
real*8, intent(in), dimension(MAX_ATOM)::charge	原子の電荷
real*8, intent(in)::eleCoeff	静電係数
real*8, intent(out)::energy	ポテンシャルエネルギー
integer, intent(in)::init	ループ初期値(原子先頭 ID)
integer, intent(in)::last	ループ終値(原子最後 ID)
integer, intent(in)::step	ループ増分値
type(Atom_t), intent(in), dimension(MAX_ATOM)::atom	原子情報

戻り値 :

なし

機能 :

全原子に対し、1-4 まで共有結合している原子に対する静電の力とポテンシャルを計算する。

46. 21. Get_ParentCell : parent セル取得

名称 : Get_ParentCell

呼び出し形式 : function Get_ParentCell(cellID, level) result (parentID)

引数 :

integer cellID	当該セル ID
integer level	当該セルのレベル

戻り値 :

integer::parentID	parent セル ID
-------------------	--------------

機能 :

当該セルの parent セルの ID を取得する。

46. 22. Get_ChildCells : child セルの取得

名称 : Get_ChildCells

呼び出し形式 : function Get_ChildCells(id, level) result (childs)

引数 :

integer, intent(in)::id	当該セル ID
integer, intent(in)::level	当該セルのレベル

戻り値 :

integer, dimension(8)::childs	child セルの ID(8 個)
-------------------------------	-------------------

機能 :

当該セルの child セルの ID を取得する。

46. 23. ID_OfIndex : セル ID 取得

名称 : ID_OfIndex

呼び出し形式 : function ID_OfIndex(x, y, z, cellLevel) result (cellID)

引数 :

integer, intent(in)::x	セルの x インデックス
integer, intent(in)::y	セルの y インデックス
integer, intent(in)::z	セルの z インデックス
integer, intent(in)::cellLevel	セルのツリーレベル

戻り値 :

integer::cellID current cell ID

機能 :

入力したセルのインデックスとツリーレベルから当該セルの ID を取得する。

46. 24. Index_OfID : セルのインデックス取得

呼び出し形式 : function Index_OfID(cellID, cellLevel) result(locate)

引数 :

integer, intent(in)::cellID	当該セルの ID
integer, intent(in)::cellLevel	当該セルのツリーレベル

戻り値 :

integer, dimension(3)::locate セルのインデックス

機能 :

当該セルの ID とツリーレベルから、インデックスを取得する。

47. Umbrella_Method

ファイル名 : Umbrella_Method.f90

47.1. Calc_UmbrellaEnergyH1 : 参照座標数 1 調和型 Umbrella ポテンシャル計算

名称 : Calc_UmbrellaEnergyH1

呼び出し形式 : function Calc_UmbrellaEnergyH1(atom, umb, init, last, step)
result (energy)

引数 :

type(Atom_t), intent(inout), dimension(MAX_ATOM)::atom	! 原子情報
type(Umbrella_t), intent(in)::umb	! Umbrella-Potential 情報
integer, intent(in)::init	ループ初期値(対象原子先頭)
integer, intent(in)::last	ループ終値(対象原子最後)
integer, intent(in)::step	ステップ数

戻り値 :

real*8::energy	ポテンシャルエネルギー
----------------	-------------

機能 :

参照座標が 1 点の調和型 umbrella ポテンシャルを計算する。

47.2. Calc_UmbrellaEnergyH2 : 参照座標数 2 調和型 Umbrella ポテンシャル計算

名称 : Calc_UmbrellaEnergyH2

呼び出し形式 : function Calc_UmbrellaEnergyH2(atom, umb, init, last, step)
result (energy)

引数 :

type(Atom_t), intent(inout), dimension(MAX_ATOM)::atom	! 原子情報
type(Umbrella_t), intent(in)::umb	! Umbrella-Potential 情報
integer, intent(in)::init	ループ初期値(対象原子先頭)
integer, intent(in)::last	ループ終値(対象原子最後)
integer, intent(in)::step	ステップ数

戻り値 :

real*8::energy	ポテンシャルエネルギー
----------------	-------------

機能 :

対象座標が 2 点の調和型 umbrella ポテンシャルを計算する。

47.3. Calc_UmbrellaEnergyL1 : 参照座標数 1 線形 Umbrella ポテンシャル計算

名称 : Calc_UmbrellaEnergyL1

呼び出し形式 : function Calc_UmbrellaEnergyL1(atom, umb, init, last, step)
result (energy)

引数 :

type(Atom_t), intent(inout), dimension(MAX_ATOM)::atom	! 原子情報
type(Umbrella_t), intent(in)::umb	! Umbrella-Potential 情報
integer, intent(in)::init	ループ初期値(対象原子先頭)
integer, intent(in)::last	ループ終値(対象原子最後)
integer, intent(in)::step	ステップ数

戻り値 :

real*8::energy	ポテンシャルエネルギー
----------------	-------------

機能 :

参照座標が 1 点の線形 umbrella ポテンシャルを計算する。

47.4. Calc_UmbrellaEnergyL2 : 参照座標数 2 線形 Umbrella ポテンシャル計算

名称 : Calc_UmbrellaEnergyL2

呼び出し形式 : function Calc_UmbrellaEnergyL2(atom, umb, init, last, step)
result (energy)

引数 :

type (Atom_t), intent (inout), dimension (MAX_ATOM)::atom	! 原子情報
type (Umbrella_t), intent (in)::umb	! Umbrella-Potential 情報
integer, intent (in)::init	ループ初期値 (対象原子先頭)
integer, intent (in)::last	ループ終値 (対象原子最後)
integer, intent (in)::step	ステップ数

戻り値 :

real*8::energy ポテンシャルエネルギー

機能 :

対象座標が 2 点の線形 umbrella ポテンシャルを計算する。

47.5. Calc_FilledPotential : 全体の FilledPotential 計算

名称 : Calc_FilledPotential

呼び出し形式 : function Calc_FilledPotential(atom, umb, init, last, step)
result (energy)

引数 :

type (Atom_t), intent (inout), dimension (MAX_ATOM)::atom	! 原子情報
type (Umbrella_t), intent (in)::umb	! Umbrella-Potential 情報
integer, intent (in)::init	ループ初期値 (対象原子先頭)
integer, intent (in)::last	ループ終値 (対象原子最後)
integer, intent (in)::step	ステップ数

戻り値 :

real*8::energy ポテンシャルエネルギー

機能 :

Filled ポテンシャルの全体のポテンシャルを計算する。

47.6. Calc_UmbrellaPotential : 対象次元の FilledPotential 計算

名称 : Calc_UmbrellaPotential

呼び出し形式 : function Calc_UmbrellaPotential (atom, umb, init, last, step)
result (energy)

引数 :

type (Atom_t), intent (inout), dimension (MAX_ATOM)::atom	! 原子情報
type (Umbrella_t), intent (in)::umb	! Umbrella-Potential 情報
integer, intent (in)::init	ループ初期値 (対象原子先頭)
integer, intent (in)::last	ループ終値 (対象原子最後)
integer, intent (in)::step	ステップ数

戻り値 :

real*8::energy ポテンシャルエネルギー

機能 :

Filled ポテンシャルの対象次元でのポテンシャルを計算する。

47.7. Calc_FlowEnergy : FlowForce 計算主制御

名称 : Calc_FlowForce

呼び出し形式 : function Calc_FlowEnergy(atom, flow) result (energy)

引数 :

type(Atom_t), intent(inout), dimension(MAX_ATOM)::atom ! 原子情報
type(FlowForce_t), intent(in)::flow flow-force 情報

戻り値 :

real*8::energy ポテンシャルエネルギー

機能 :

FlowForce の計算の制御を行う。FlowForce は「蛋白質と溶媒」、「蛋白質と薬物」、「薬物と溶媒」の間の 1-5 相互作用の関係にある原子間に加わる力である。

47.8. Calc_FlowForce : FlowForce 計算

呼び出し形式 : function Calc_FlowForce(atom, atomList, topAtomIndex, lastAtomIndex, init, last, step, flow) result (energy)

引数 :

type(Atom_t), intent(inout), dimension(MAX_ATOM)::atom ! 原子情報
integer, intent(in), dimension(:)::atomList ! 相互作用リスト
integer, intent(in), dimension(MAX_ATOM)::topAtomIndex ! 相互作用リスト先頭
integer, intent(in), dimension(MAX_ATOM)::lastAtomIndex ! 相互作用リスト最後
integer, intent(in)::init ループ初期値(先頭原子)
integer, intent(in)::last ループ終値(最後原子)
integer, intent(in)::step ループステップ
type(FlowForce_t), intent(in)::flow FlowForce 情報

戻り値 :

real*8::energy ポテンシャルエネルギー

機能 :

FlowForce 計算を行う。

47.9. UpdateFPAverage : FP 対象原子の力および座標の平均値の更新

呼び出し形式 : subroutine Update_FPAverage(energy, temp, atomInfo, rest)

引数 :

real*8, intent(in)::energy アンブレラポテンシャルエネルギー
real*8, intent(in)::temp 温度
type(AtomInfo_t), intent(in)::atomInfo 原子情報
type(Restrain_t), intent(inout)::rest 拘束情報

戻り値 :

なし

機能 :

FP 対象原子の力（アンブレラポテンシャルは除く）と座標の平均値を更新する。

48. Tsallis_Method

ファイル名 : Tsallis_Method.f90

48.1. Calc_Kinetic : 系の運動量計算

名称 : Calc_Kinetic

呼び出し形式 : function Calc_Momentum(vel, freeNum, freeID) result (kinetic)

引数 :

real*8, intent(in), dimension(3, MAX_ATOM)::vel
integer, intent(in)::freeNum
integer, intent(in), dimension(MAX_ATOM)::freeID

戻り値 :

real*8::kinetic

機能 :

系の運動量計算を行う。

48.2. Calc_TsallisEquation : 速度と座標の微分値計算

名称 : Calc_TsallisEquation

呼び出し形式 : subroutine Calc_TsallisEquation(grad, vel, potential, freeNum,
freeID, zeta, diffCord, diffVel, zetaDash, vkd)

引数 :

real*8, intent(in), dimension(3, MAX_ATOM)::grad
real*8, intent(in), dimension(3, MAX_ATOM)::vel
real*8, intent(in)::potential
integer, intent(in)::freeNum
real*8, intent(in)::zeta
integer, intent(in), dimension(MAX_ATOM)::freeID
real*8, intent(out), dimension(3, MAX_ATOM)::diffCord
real*8, intent(out), dimension(3, MAX_ATOM)::diffVel
real*8, intent(out)::zetaDash
real*8, intent(out)::vkd

戻り値 :

なし

機能 :

Runge-Kutta 法での速度と座標の n 次微分の値を計算する。

48.3. Exec_RungeKuttaTsallis : 4 次の Runge-Kutta 法での速度、座標の増分計算

名称 : Exec_RungeKuttaTsallis

呼び出し形式 : subroutine Exec_RungeKuttaTsallis(obj, intr, rest, cond,
dyn, option, bound, cluster, addCord, addVel)

引数 :

```

type(Object_t), intent(inout)::obj      ! aggregate of object
type(Interact_t), intent(inout)::intr    ! aggregate of interaction
type(Restrain_t), intent(in)::rest       ! aggregate of restrain
type(Condition_t), intent(in)::cond      ! aggregate of condition
type(Dynamics_t), intent(inout)::dyn     ! aggregate of dynamics
type(Option_t), intent(in)::option       ! aggregate of option
type(Boundary_t), intent(in)::bound      ! aggregate of boundary
type(MultiCluster_t), intent(inout)::cluster ! aggregate of cluster
real*8, intent(out), dimension(3, MAX_ATOM)::addCord
real*8, intent(out), dimension(3, MAX_ATOM)::addVel

```

戻り値 :

なし

機能 :

4 次の Runge-Kutta 法で、当該ステップでの原子の速度、座標の増分を計算する。

48.4. Set_DataAfterIntegration : 積分器後の物理量計算

名称 : Init_Tsallis

呼び出し形式 : subroutine Init_Tsallis(obj, intr, rest, cond,
dyn, option, bound, cluster)

引数 :

```

type(Object_t), intent(inout)::obj      ! aggregate of object
type(Interact_t), intent(inout)::intr    ! aggregate of interaction
type(Restrain_t), intent(in)::rest       ! aggregate of restrain
type(Condition_t), intent(in)::cond      ! aggregate of condition
type(Dynamics_t), intent(inout)::dyn     ! aggregate of dynamics
type(Option_t), intent(in)::option       ! aggregate of option
type(Boundary_t), intent(in)::bound      ! aggregate of boundary
type(MultiCluster_t), intent(inout)::cluster ! aggregate of cluster

```

戻り値 :

なし

機能 :

積分器後の以下の物理量を計算する。

- ・ジュール換算のポテンシャル+シフトエネルギー
- ・Tsallis 積分のチェック値
- ・ ρ のモード
- ・ $\ln(\rho)$ の値
- ・スケーリングファクター

48.5. Exec_TsallisIntegration : Tsallis での積分器主制御

名称 : Exec_TsallisIntegration

呼び出し形式 : subroutine Exec_TsallisIntegration(obj, intr, rest, cond,
dyn, option, bound, cluster)

引数 :

```
type(Object_t), intent(inout)::obj      ! aggregate of object
type(Interact_t), intent(inout)::intr    ! aggregate of interaction
type(Restrain_t), intent(in)::rest       ! aggregate of restrain
type(Condition_t), intent(in)::cond      ! aggregate of condition
type(Dynamics_t), intent(inout)::dyn     ! aggregate of dynamics
type(Option_t), intent(in)::option       ! aggregate of option
type(Boundary_t), intent(in)::bound      ! aggregate of boundary
type(MultiCluster_t), intent(inout)::cluster ! aggregate of cluster
```

戻り値 :

なし

機能 :

Tsallis の積分器を実行する。

- (1) 速度・座標の増分の計算
- (2) 速度・座標の更新
- (3) 積分器後の物理量計算

48.6. Init_Tsallis : Tsallis 法初期処理

名称 : Init_Tsallis

呼び出し形式 : subroutine Init_Tsallis(dyn)

引数 :

```
type(Dynamics_t), intent(inout)::dyn     MD 情報
```

戻り値 :

なし

機能 :

Tsallis 法の初期設定を行う。

49. ReplicaMD_Control

ファイル名 : Control_ReplicaMD.f90

49.1. Init_ReplicaMD : レプリカ法の MD 初期処理

名称 : Init_ReplicaMD

呼び出し形式 : subroutine Init_ReplicaMD

引数 :

なし

戻り値 :

なし

機能 :

レプリカ交換時の MD 初期処理を行う。レプリカ交換でない MD との違いは以下の 2 点。

(1-1) レプリカ交換リスタートファイルの読み込み

(1-2) 初期速度設定で、各レプリカのシミュレーション温度に対し、速度をスケールリング

49.2. Exec_ReplicaMD : レプリカ交換ループの制御

名称 : Exec_ReplicaMD

呼び出し形式 : subroutine Exec_ReplicaMD

引数 :

なし

戻り値 :

なし

機能 :

レプリカ交換回数とレプリカ個数に対しループして、1 レプリカに対する MD 処理を実行する。レプリカの個数ループが終了した時点でレプリカ交換を行う。

```
do レプリカ交換回数
  do レプリカ個数
    ・ 1 レプリカの MD 実行 (Exec_ReplicaMDLoop)
  enddo
  ・ レプリカ交換判定
enddo
```

49.3. Exec_ReplicaMDLoop : レプリカ法での MD の実行

名称 : Exec_ReplicaMDLoop

呼び出し形式 : subroutine Exec_ReplicaMDLoop(replica)

引数 :

integer, intent(in) :: replica レプリカ番号

戻り値 :

なし

機能 :

レプリカ交換の MD を実行する。レプリカ交換でない MD との違いは、出力するトラジェクトリ情報に、レプリカ番号を付加する点である。

49.4. Final_ReplicaMD : レプリカ法での MD の終了処理

名称 : Final_ReplicaMD

呼び出し形式 : Final_ReplicaMD

引数 :

なし

戻り値 :

なし

機能 :

レプリカ交換時の MD の終了処理をレプリカ交換でない MD との違いは以下の 2 点。

(4-1) レプリカ交換リスタートファイルの出力

(4-2) MD 終了時のログをレプリカ数分出力

49.5. ReplicaMD_Main : レプリカ交換 MD の主制御

名称 : ReplicaMD_Main

呼び出し形式 : ReplicaMD_Main

引数 :

なし

戻り値 :

なし

機能 :

レプリカ交換 MD の主制御を行い、MD の初期化、実行、終了処理を呼び出す。

50. Replica_Method

ファイル名 : Replica_Method.f90

50.1. Exchange_Replica : レプリカ交換判定

名称 : Exchange_Replica

呼び出し形式 : Exchange_Replica(replicaLoop)

引数 :

integer, intent(in)::replicaLoop レプリカ交換の回数

戻り値 :

なし

機能 :

レプリカ交換の判定を行い、条件を満たす場合に原子速度と原子座標および、Nose, Rigidの物理量を交換する。

50.2. Init_Replica : レプリカ交換法初期処理

名称 : Init_Replica

呼び出し形式 : subroutine Init_Replica(obj, dyn)

引数 :

type(Object_t), intent(in)::obj オブジェクト情報

type(Dynamics_t), intent(in)::dyn MD 情報

戻り値 :

なし

機能 :

レプリカ交換で使用する動的な領域の確保を行う。

50.3. Set_ReplicaTemperature : レプリカのシミュレーション温度設定

名称 : Set_ReplicaTemperature

呼び出し形式 : Set_ReplicaTemperature(temp, replica)

引数 :

real*8, intent(in)::temp シミュレーション温度

integer, intent(in)::replica レプリカ ID

戻り値 :

なし

機能 :

指定したレプリカのシミュレーション温度を設定する。

50.4. Final_Replica : レプリカ法終了処理

名称 : Final_Replica

呼び出し形式 : Final_Replica(dyn)

引数 :

type(Dynamics_t), intent(in)::dyn MD 情報

戻り値 :

なし

機能 :

レプリカ交換法で使った動的な領域を解放する。

50.5. Set_ReplicaData : 系の状態の保存

名称 : Set_ReplicaData

呼び出し形式 : subroutine Set_ReplicaData(obj, dyn, replica)

引数 :

type(Object_t), intent(in)::obj	オブジェクト情報
type(Dynamics_t), intent(in)::dyn	MD 情報
integer, intent(in)::replica	レプリカ ID

戻り値 :

なし

機能 :

当該レプリカの MD 情報とオブジェクト情報でレプリカごとに変わるものを保存する。

50.6. Set_ReplicaPotential : エネルギーの保存

名称 : Set_ReplicaPotential

呼び出し形式 : Set_ReplicaPotential(obj, dyn, replica)

引数 :

type(Object_t), intent(in)::obj	オブジェクト情報
type(Dynamics_t), intent(in)::dyn	MD 情報
integer, intent(in)::replica	レプリカ ID

戻り値 :

なし

機能 :

当該レプリカのポテンシャルエネルギーと運動エネルギーを保存する。

50.7. Get_ReplicaData : 形の状態の取得

名称 : Get_ReplicaData

呼び出し形式 : subroutine Get_ReplicaData(obj, dyn, replica)

引数 :

type(Object_t), intent(out)::obj	オブジェクト情報
type(Dynamics_t), intent(out)::dyn	MD 情報
integer, intent(in)::replica	レプリカ ID

戻り値 :

なし

機能 :

当該レプリカの情報を MD 情報とオブジェクト情報に設定する。

50.8. Get_ReplicaPotential : エネルギーの取得

名称 : Get_ReplicaPotential

呼び出し形式 : Get_ReplicaPotential(obj, dyn, replica)

引数 :

type(Object_t), intent(out)::obj	オブジェクト情報
type(Dynamics_t), intent(out)::dyn	MD 情報
integer, intent(in)::replica	レプリカ ID

戻り値 :

なし

機能 :

当該レプリカのポテンシャルエネルギーと運動エネルギーを MD 情報に設定する。

50.9. Output_ReplicaLog : 各レプリカのログ出力

名称 : Output_ReplicaLog

呼び出し形式 : Output_ReplicaLog

引数 :

なし

戻り値 :

なし

機能 :

全レプリカのログ情報を標準出力に出力する。

50.10. Input_ReplicaRestartFile : レプリカ法のリスタートファイル入力

名称 : Input_ReplicaRestartFile

呼び出し形式 : subroutine Input_ReplicaRestartFile(inp)

引数 :

type(Input_t), intent(in)::inp

戻り値 :

なし

機能 :

レプリカ交換法のリスタートファイルを読み込む。レプリカ交換法のリスタートファイルはバイナリ形式であり、1つのMDのリスタートファイルが昇順に並んだ形である。

この処理がレプリカ数分実行され、レプリカ交換法のリスタート入力が行われる。

50.11. Output_ReplicaRestartFile : レプリカ法のリスタートファイル出力

名称 : Output_ReplicaRestartFile

呼び出し形式 : subroutine Output_ReplicaRestartFile(simulateTime, unit, file)

引数 :

real*8, intent(in)::simulateTime シミュレート時間

integer, intent(in)::unit 論理装置番号

character (NAME_SIZE), intent(in)::file ファイル名

戻り値 :

なし

機能 :

レプリカ交換法のリスタートファイルを出力する。レプリカ交換法のリスタートファイルはバイナリ形式であり、1つのMDのリスタートファイルが昇順に並んだ形である。

当手続きでレプリカ数分のループを実行し、リスタートファイルを生成する。

50.12. Get_ReplicaTargetTemperature : レプリカのシミュレーション温度取得

名称 : Get_ReplicaTargetTemperature

呼び出し形式 : function Get_ReplicaTargetTemperature(replica) result (temp)

引数 :

integer, intent(in)::replica レプリカ ID

戻り値 :

real*8::temp 当該レプリカのシミュレート設定温度

機能 :

当該レプリカのシミュレート設定温度を取得する。

50. 13. Add_ReplicaInfoToTrajectory : トラジェクトリへのレプリカ情報追加

名称 : Add_ReplicaInfoToTrajectory

呼び出し形式 : Add_ReplicaInfoToTrajectory (monitorMD, replica, loopNum, freeNum)

引数 :

type (MD_t), intent (in) :: monitorMD	MD モニター情報
integer, intent (in) :: replica	レプリカ ID
integer, intent (in) :: loopNum	当該 MD ループ回数
integer, intent (in) :: freeNum	自由原子数

戻り値 :

なし

機能 :

現在まで出力したトラジェクトリ情報の最後に、以下の情報を出力する。

[データバイト数] データ識別子 レプリカ数 レプリカ ID freeNum

- ・データバイト数はバイナリ形式の時の浮動小数点のバイト数を示す 4 バイト整数
- ・データ識別子は、'CORD' 'VEL' 'ENER' 'TEMP' 'TRAJ' のいずれかの文字列。
- ・他の項目は 4 バイト整数

51. EffectiveTemp_Method

ファイル名 : EffectiveTemp_Method.f90

51.1. Init_EffectiveTempMCMD : 有効温度マルチカノニカル法初期化

名称 : Init_EffectiveTempMCMD

呼び出し形式 : subroutine Init_EffectiveTempMCMD(dyn, cond, mon)

引数 :

type(Dynamics_t), intent(in)::dyn	MD 情報
type(Condition_t), intent(in)::cond	MD 条件情報
type(Monitor_t), intent(in)::mon	モニター情報

戻り値 :

なし

機能 :

有効温度マルチカノニカル法の初期設定と動的な領域の確保を行う。

51.2. Scale_GradientForEffectiveTemp : 有効温度での力のスケーリング

名称 : Scale_GradientForEffectiveTemp

呼び出し形式 : subroutine Scale_GradientForEffectiveTemp(obj, dyn)

引数 :

type(Object_t), intent(inout)::obj	オブジェクトの情報
type(Dynamics_t), intent(inout)::dyn	MD の情報

戻り値

なし

機能 :

有効温度マルチカノニカル法での原子に加わる力へのスケーリングと統計データ（エネルギー）のサンプリングを行う。

51.3. Calc_DistributionFromLog : 統計データからのスケーリングファクター計算

名称 : Calc_DistributionFromLog

呼び出し形式 : subroutine Calc_DistributionFromLog

引数 :

なし

戻り値 :

なし

機能 :

エネルギー分布のヒストグラムからスケーリングファクターを計算する。

51.4. Output_Historgam : ヒストグラム出力

名称 : Output_Historgam

呼び出し形式 : subroutine Output_Historgam(loopNum)

引数 :

integer, intent(in)::loopNum	当該 MD ループ回数
------------------------------	-------------

戻り値 :

なし

機能 :

Effective-Temperature のデータをファイルに出力する。

51.5. Read_EffectiveTempRestart : リスタートデータの入力

名称 : Read_EffectiveTempRestart

呼び出し形式 subroutine Read_EffectiveTempRestart(unit)

引数 :

integer, intent(in)::unit 入力装置番号

戻り値 :

なし

機能 :

Effective-Temperature のデータをリスタートファイルから入力する。

51.6. Write_EffectiveTempRestart : リスタートデータの出力

名称 : Write_EffectiveTempRestart

呼び出し形式 subroutine Write_EffectiveTempRestart(unit)

引数 :

integer, intent(in)::unit 出力装置番号

戻り値 :

なし

機能 :

Effective-Temperature のリスタートデータをリスタートファイルに出力する。

51.7. Rollback_EffectiveTempRestart : データのロールバック

名称 : Write Rollback_EffectiveTempRestart

呼び出し形式 subroutine Rollback_EffectiveTempRestart

引数 :

なし

戻り値 :

なし

機能 :

Effective-Temperature のデータをリスタート時点に戻す。

52. Periodic_Nonbonded

ファイル名 : Periodic_NonBonded.f90

52.1. Calc_Periodic15InteractEnergy : 周期系での 1-5 相互作用計算

名称 : Calc_Periodic15InteractEnergy

呼び出し形式 : subroutine Calc_Periodic15InteractEnergy(nonBonded, Atom, loopInit, loopLast, dyn, option)

引数 :

type(NonBondedInfo_t), intent(in)::nonBonded	1-5 相互作用情報
type(Atom_t), intent(inout), dimension(MAX_ATOM)::Atom	原子情報
integer, intent(in)::loopInit	ループ初期値
integer, intent(in)::loopLast	ループ終値
type(Dynamics_t), intent(inout)::dyn	MD 情報
type(Option_t), intent(in)::option	オプション情報

戻り値 :

なし

機能 :

周期系での 1-5 相互作用計算を計算する。

注 52.2~52.12 に記述する相互作用計算処理の引数は共通であるため、ここで記述する。

real*8, dimension(3, MAX_ATOM)::cord	原子座標
real*8, dimension(3, MAX_ATOM)::grad	原子への力
real*8, dimension(MAX_ATOM)::ele	原子電荷
real*8, dimension(3, -62:62)::replicaCord	レプリカセル座標
real*8::eleCoeff	静電係数
real*8, dimension(MAX_TYPE, MAX_TYPE)::hyd10Coeff	水素結合 10 乗項係数
real*8, dimension(MAX_TYPE, MAX_TYPE)::hyd12Coeff	水素結合 12 乗項係数
real*8, dimension(MAX_TYPE, MAX_TYPE)::vdw6Coeff	van der Waals 6 乗項係数
real*8, dimension(MAX_TYPE, MAX_TYPE)::vdw12Coeff	van der Waals 12 乗項係数
integer, intent(in)::init	ループ初期値(処理対象原子先頭)
integer, intent(in)::last	ループ終値(処理対象原子の最後)
integer, intent(in)::step	ループ増分値
real*8, dimension(MAX_POT)::energy	ポテンシャルエネルギー
integer, dimension(MAX_ATOM)::vdwTopAtom	van der Waals の先頭インデックス配列
integer, dimension(MAX_ATOM)::vdwLastAtom	van der Waals の最後インデックス配列
integer, dimension(MAX_VDW)::vdwAtomID	van der Waals の相手の原子 ID
integer, dimension(MAX_HYD)::vdwReplica	van der Waals の相手原子のレプリカセル ID
integer, dimension(MAX_ATOM)::hydTopAtom	水素結合の先頭インデックス配列
integer, dimension(MAX_ATOM)::hydLastAtom	水素結合の最後インデックス配列
integer, dimension(MAX_HYD)::hydAtomID	水素結合の相手原子の ID
integer, dimension(MAX_HYD)::hydReplica	水素結合の相手原子のレプリカセル ID
integer, dimension(MAX_ATOM)::type	原子タイプ
type(Virial_t), dimension(MAX_POT)::virial	ビリアル

52.2. Calc_VdwPeriodic : 周期系での van der Waals 計算

名称 : Calc_VdwPeriodic

呼び出し形式 : subroutine Calc_VdwPeriodic(cord, grad, ele, type, energy, eleCoeff,
 replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
 init, last, step,
 vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
 hydTopAtom, hydLastAtom, hydAtomID, hydReplica, virial)

引数 :

52.1. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 原子に周期境界条件を適用し、van der Waals の力とポテンシャルの計算を行う。

52.3. Calc_HydPeriodic : 周期系での水素結合計算

名称 : Calc_HydPeriodic

呼び出し形式 : subroutine Calc_HydPeriodic(cord, grad, ele, type, energy, eleCoeff,
 replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
 init, last, step,
 vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
 hydTopAtom, hydLastAtom, hydAtomID, hydReplica, virial)

引数 :

52.1. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 原子に周期境界条件を適用し、水素結合の力とポテンシャルの計算を行う。

52.4. Calc_ConstElePeriodic : 周期系での距離非依存静電計算

名称 : Calc_ConstElePeriodic

呼び出し形式 : subroutine Calc_ConstElePeriodic(cord, grad, ele, type, energy, eleCoeff,
 replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
 init, last, step,
 vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
 hydTopAtom, hydLastAtom, hydAtomID, hydReplica, virial)

引数 :

52.1. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 原子に周期境界条件を適用し、距離非依存静電の力とポテンシャルの計算を行う。

52.5. Calc_DependElePeriodic : 周期系での距離依存静電計算

名称 : Calc_DependElePeriodic

呼び出し形式 : subroutine Calc_DependElePeriodic (cord, grad, ele, type, energy, eleCoeff, replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff, init, last, step, vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica, hydTopAtom, hydLastAtom, hydAtomID, hydReplica, virial)

引数 :

52.1. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 原子に周期境界条件を適用し、距離依存静電の力とポテンシャルの計算を行う。

52.6. Calc_VdwConstElePeriodic : 周期系での van der Waals と距離非依存静電計算

名称 : Calc_VdwConstElePeriodic

呼び出し形式 : subroutine Calc_VdwConstElePeriodic(cord, grad, ele, type, energy, eleCoeff, replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff, init, last, step, vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica, hydTopAtom, hydLastAtom, hydAtomID, hydReplica, virial)

引数 :

52.1. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 原子に周期境界条件を適用し、van der Waals と距離非依存静電の力とポテンシャルの計算を行う。

52.7. Calc_VdwDependElePeriodic : 周期系での van der Waals と距離依存静電計算

名称 : Calc_VdwDependElePeriodic

呼び出し形式 : subroutine Calc_VdwDependElePeriodic(cord, grad, ele, type, energy, eleCoeff, replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff, init, last, step, vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica, hydTopAtom, hydLastAtom, hydAtomID, hydReplica, virial)

引数 :

52.1. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 原子に周期境界条件を適用し、van der Waals と距離依存静電の力とポテンシャルの計算を行う。

52.8. Calc_VdwHydPeriodic : 周期系での van der Waals と水素結合計算

名称 : Calc_VdwHydPeriodic

呼び出し形式 : subroutine Calc_VdwHydPeriodic(cord, grad, ele, type, energy, eleCoeff,
 replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
 init, last, step,
 vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
 hydTopAtom, hydLastAtom, hydAtomID, hydReplica, virial)

引数 :

52.1. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 原子に周期境界条件を適用し、van der Waals と水素結合の力とポテンシャルの計算を行う。

52.9. Calc_HydConstElePeriodic : 周期系での水素結合と距離非依存静電計算

名称 : Calc_HydConstElePeriodic

呼び出し形式 : subroutine Calc_HydConstElePeriodic(cord, grad, ele, type, energy, eleCoeff,
 replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
 init, last, step,
 vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
 hydTopAtom, hydLastAtom, hydAtomID, hydReplica, virial)

引数 :

52.1. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 原子に周期境界条件を適用し、水素結合と距離非依存静電の力とポテンシャルの計算を行う。

注意 :

スプライン補間は水素結合をサポートしていないため、本手続きはエラーを出力してプログラムを終了する。

52.10. Calc_HydDependElePeriodic : 周期系での水素結合と距離依存静電計算

名称 : Calc_HydDependElePeriodic

呼び出し形式 : subroutine Calc_HydDependElePeriodic(cord, grad, ele, type, energy, eleCoeff,
 replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
 init, last, step,
 vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
 hydTopAtom, hydLastAtom, hydAtomID, hydReplica, virial)

引数 :

52.1. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 原子に周期境界条件を適用し、水素結合と距離依存静電の力とポテンシャルの計算を行う。

52. 11. Calc_VdwHydConstElePeriodic : 周期系での van der Waals, 水素結合, 距離非依存静電計算

名称 : Calc_VdwHydConstElePeriodic

呼び出し形式 : subroutine Calc_VdwHydConstElePeriodic(cord, grad, ele, type, energy, eleCoeff,
replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
hydTopAtom, hydLastAtom, hydAtomID, hydReplica, virial)

引数 :

52. 1. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 原子に周期境界条件を適用し、van der Waals, 水素結合, 距離非依存静電の力とポテンシャルの計算を行う。

52. 12. Calc_VdwHydDependElePeriodic : 周期系での van der Waals, 水素結合, 距離依存静電計算

名称 : Calc_VdwHydDependElePeriodic

呼び出し形式 : subroutine Calc_VdwHydDependElePeriodic(cord, grad, ele, type, energy,
eleCoeff,
replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
hydTopAtom, hydLastAtom, hydAtomID, hydReplica, virial)

引数 :

26. 1. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の van der Waals, 水素結合, 距離依存静電の力とポテンシャルの計算を行う。

52. 13. Calc_PmeEwald15InteractEnergy : PME/Ewald での 1-5 相互作用計算

名称 : Calc_PmeEwald15InteractEnergy

呼び出し形式 : subroutine Calc_PmeEwald 15InteractEnergy (nonBonded, Atom, loopInit, loopLast, dyn, option)

引数 :

type (NonBondedInfo_t), intent (in) :: nonBonded	1-5 相互作用情報
type (Atom_t), intent (inout), dimension (MAX_ATOM) :: Atom	原子情報
integer, intent (in) :: loopInit	ループ初期値
integer, intent (in) :: loopLast	ループ終値
type (RigidInfo_t), intent (in) :: rigidInfo	原子情報
type (Dynamics_t), intent (inout) :: dyn	MD 情報
type (Option_t), intent (in) :: option	オプション情報
integer, intent (in) :: updatePeriod	相互作用テーブル更新間隔

戻り値 :

なし

機能 :

周期系で PME/Ewald 法を使用した場合の 1-5 相互作用計算を計算する。

注 52. 14~52. 17 に記述する相互作用計算処理の引数は共通であるため、ここで記述する。

real*8, dimension (3, MAX_ATOM) :: cord	原子座標
real*8, dimension (3, MAX_ATOM) :: grad	原子への力
real*8, dimension (MAX_ATOM) :: ele	原子電荷
real*8, dimension (3, -62:62) :: replicaCord	レプリカセル座標
real*8 :: eleCoeff	静電係数
real*8, dimension (MAX_TYPE, MAX_TYPE) :: hyd10Coeff	水素結合 10 乗項係数
real*8, dimension (MAX_TYPE, MAX_TYPE) :: hyd12Coeff	水素結合 12 乗項係数
real*8, dimension (MAX_TYPE, MAX_TYPE) :: vdw6Coeff	van der Waals 6 乗項係数
real*8, dimension (MAX_TYPE, MAX_TYPE) :: vdw12Coeff	van der Waals 12 乗項係数
integer, intent (in) :: init	ループ初期値 (処理対象原子先頭)
integer, intent (in) :: last	ループ終値 (処理対象原子の最後)
integer, intent (in) :: step	ループ増分値
real*8, dimension (MAX_POT) :: energy	ポテンシャルエネルギー
integer, dimension (MAX_ATOM) :: vdwTopAtom	van der Waals の先頭インデックス配列
integer, dimension (MAX_ATOM) :: vdwLastAtom	van der Waals の最後インデックス配列
integer, dimension (MAX_VDW) :: vdwAtomID	van der Waals の相手の原子 ID
integer, dimension (MAX_HYD) :: vdwReplica	van der Waals の相手原子のレプリカセル ID
integer, dimension (MAX_ATOM) :: hydTopAtom	水素結合の先頭インデックス配列
integer, dimension (MAX_ATOM) :: hydLastAtom	水素結合の最後インデックス配列
integer, dimension (MAX_HYD) :: hydAtomID	水素結合の相手原子の ID
integer, dimension (MAX_HYD) :: hydReplica	水素結合の相手原子のレプリカセル ID
integer, dimension (MAX_ATOM) :: type	原子タイプ
real*8, dimension (3, 3) :: virial	系全体のビリアル
real*8 :: alphaEwald	逆空間計算比率

52.14. Calc_EleDirect : 実空間の静電計算

名称 : Calc_EleDirect

呼び出し形式 : subroutine Calc_EleDirect(cord, grad, ele, type, energy,
eleCoeff,
replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
hydTopAtom, hydLastAtom, hydAtomID, hydReplica,
virial, alphaEwald)

引数 :

26.13. を参照

戻り値 :

なし

機能 :

PME/Ewald 法での実空間の静電の力とポテンシャルの計算を行う。

52.15. Calc_VdwEleDirect : 実空間の van der Waals, 静電計算

名称 : Calc_VdwHydDependEleDirect

呼び出し形式 : subroutine Calc_VdwHydDependEleDirect(cord, grad, ele, type, energy,
eleCoeff,
replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
hydTopAtom, hydLastAtom, hydAtomID, hydReplica,
virial, alphaEwald)

引数 :

26.13. を参照

戻り値 :

なし

機能 :

PME/Ewald 法での実空間の van der Waals, 静電の力とポテンシャルの計算を行う。

52.16. Calc_HydEleDirect : 実空間の水素結合, 静電計算

名称 : Calc_HydEleDirect

呼び出し形式 : subroutine Calc_HydEleDirect(cord, grad, ele, type, energy,
eleCoeff,
replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
hydTopAtom, hydLastAtom, hydAtomID, hydReplica,
virial, alphaEwald)

引数 :

26.13. を参照

戻り値 :

なし

機能 :

PME/Ewald 法での実空間の水素結合, 距離依存静電の力とポテンシャルの計算を行う。

52.17. Calc_VdwHydDirect : 実空間の van der Waals, 水素結合, 静電計算

名称 : Calc_VdwHydDirect

呼び出し形式 : subroutine Calc_VdwHydDirect(cord, grad, ele, type, energy,
eleCoeff,
replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
hydTopAtom, hydLastAtom, hydAtomID, hydReplica,
virial, alphaEwald)

引数 :

26.13. を参照

戻り値 :

なし

機能 :

PME/Ewald 法での実空間の van der Waals, 水素結合, 静電の力とポテンシャルの計算を行う。

53. Rattle_Method

ファイル名 : Rattle.f90

53.1. Exec_Rattle : Rattle 法主制御

名称 : Exec_Rattle

呼び出し形式 : subroutine Exec_Rattle(cord)

引数 :

real*8, dimension(3, MAX_ATOM), intent(in)::cord ! 原子座標

戻り値 :

なし

機能 :

Rattle 法による速度補正計算の主制御を行う。

53.2. Calc_TwoAtomsRattle : 速度補正計算 (2 原子間 Rattle 拘束)

名称 : Calc_TwoAtomsRattle

呼び出し形式 : subroutine Calc_TwoAtomsRattle(cord)

引数 :

real*8, dimension(3, MAX_ATOM), intent(in)::cord ! 原子座標

戻り値 :

なし

機能 :

Rattle 拘束対象原子 (2 原子間) の速度を補正する。

53.3. Calc_ThreeAtomsRattle : 速度補正計算 (3 原子間 Rattle 拘束)

名称 : Calc_ThreeAtomsRattle

呼び出し形式 : subroutine Calc_ThreeAtomsRattle(cord)

引数 :

real*8, dimension(3, MAX_ATOM), intent(in)::cord ! 原子座標

戻り値 :

なし

機能 :

Rattle 拘束対象原子 (3 原子間) の速度を補正する。

53.4. Calc_FourAtomsRattle : 速度補正計算 (4 原子間 Rattle 拘束)

名称 : Calc_FourAtomsRattle

呼び出し形式 : subroutine Calc_FourAtomsRattle(cord)

引数 :

real*8, dimension(3, MAX_ATOM), intent(in)::cord ! 原子座標

戻り値 :

なし

機能 :

Rattle 拘束対象原子 (4 原子間) の速度を補正する。

54. Calc_ExtCap

ファイル名 : ExtCap.f90

54.1. Allocate_ExtCapData : 計算パラメータの設定および計算領域の確保

名称 : Allocate_ExtCapData

呼び出し形式 : subroutine Allocate_ExtCapData(sphereNumber, &
 sphereAtomNumber, &
 sphereResidNumber, &
 sphereChainNumber, &
 algEllipNumber, &
 algEllipAtomNumber, &
 algEllipResidNumber, &
 algEllipChainNumber, &
 geomEllipNumber, &
 geomEllipAtomNumber, &
 geomEllipResidNumber, &
 geomEllipChainNumber)

引数 :

integer, intent(in)::sphereNumber	!	球体 CAP 拘束数
integer, intent(in)::sphereAtomNumber	!	球体 CAP 拘束対象原子数
integer, intent(in)::sphereResidNumber	!	球体 CAP 拘束対象残基数
integer, intent(in)::sphereChainNumber	!	球体 CAP 拘束対象チェーン数
integer, intent(in)::algEllipNumber	!	代数的楕円体 CAP 拘束数
integer, intent(in)::algEllipAtomNumber	!	代数的楕円体 CAP 拘束対象原子数
integer, intent(in)::algEllipResidNumber	!	代数的楕円体 CAP 拘束対象残基数
integer, intent(in)::algEllipChainNumber	!	代数的楕円体 CAP 拘束対象チェーン数
integer, intent(in)::geomEllipNumber	!	幾何的楕円体 CAP 拘束数
integer, intent(in)::geomEllipAtomNumber	!	幾何的楕円体 CAP 拘束対象原子数
integer, intent(in)::geomEllipResidNumber	!	幾何的楕円体 CAP 拘束対象残基数
integer, intent(in)::geomEllipChainNumber	!	幾何的楕円体 CAP 拘束対象チェーン数

戻り値 :

なし

機能 :

拡張 CAP 拘束数、拡張 CAP 拘束対象数を設定し、必要な計算領域を確保する。

54.2. Allocate_DynamicArray : 計算領域の確保

名称 : Allocate_DynamicArray

呼び出し形式 : subroutine Allocate_DynamicArray

引数 :

なし

戻り値 :

なし

機能 :

拡張 CAP 拘束計算用の領域を動的に確保する。

54.3. Set_ExtCapParameter : 計算パラメータの設定

名称 : Set_ExtCapParameter

呼び出し形式 : subroutine Set_ExtCapParameter (tempSphere, &
tempAlgEllip, &
tempGeomEllip, &
tempAtomIDForSphere, &
tempResidIDForSphere, &
tempChainIDForSphere, &
tempAtomIDForAlgEllip, &
tempResidIDForAlgEllip, &
tempChainIDForAlgEllip, &
tempAtomIDForGeomEllip, &
tempResidIDForGeomEllip, &
tempChainIDForGeomEllip)

引数 :

```

type(Sphere_t), dimension(:), intent(in)::tempSphere      ! 球体 CAP 拘束情報
type(AlgEllip_t), dimension(:), intent(in)::tempAlgEllip    ! 代数的楕円体 CAP 拘束情報
type(GeomEllip_t), dimension(:), intent(in)::tempGeomEllip  ! 幾何的楕円体 CAP 拘束情報
integer, dimension(:, :), intent(in)::tempAtomIDForSphere
! 球体 CAP 拘束対象原子番号
integer, dimension(:, :), intent(in)::tempResidIDForSphere
! 球体 CAP 拘束対象残基番号
integer, dimension(:, :), intent(in)::tempChainIDForSphere
! 球体 CAP 拘束対象チェーン番号
integer, dimension(:, :), intent(in)::tempAtomIDForAlgEllip
! 代数的楕円体 CAP 拘束対象原子番号
integer, dimension(:, :), intent(in)::tempResidIDForAlgEllip
! 代数的楕円体 CAP 拘束対象残基番号
integer, dimension(:, :), intent(in)::tempChainIDForAlgEllip
! 代数的楕円体 CAP 拘束対象チェーン番号
integer, dimension(:, :), intent(in)::tempAtomIDForGeomEllip
! 幾何的楕円体 CAP 拘束対象原子番号
integer, dimension(:, :), intent(in)::tempResidIDForGeomEllip
! 幾何的楕円体 CAP 拘束対象残基番号
integer, dimension(:, :), intent(in)::tempChainIDForGeomEllip
! 幾何的楕円体 CAP 拘束対象チェーン番号

```

戻り値 :

なし

機能 :

拡張 CAP 拘束計算パラメータ、拘束対象原子番号、拘束対象残基番号
および拘束対象チェーン番号を設定する。

54.4. Transfer_ExtCapParameter : 計算パラメータの送信

名称 : Transfer_ExtCapParameter

呼び出し形式 : subroutine Transfer_ExtCapParameter (optn)

引数 :

```

type(Option_t), intent(in)::optn      ! オプション情報

```

戻り値 :

なし

機能 :

拡張 CAP 拘束計算パラメータを子プロセスに送信する。

54.5. Calc_ExtCapEnergy : 拡張 CAP 拘束計算

名称 : Calc_ExtCapEnergy

呼び出し形式 : function Calc_ExtCapEnergy(atom, init, last, step) result (totalEnergy)

引数 :

type(Atom_t), dimension(MAX_ATOM), intent(inout)::atom	!	原子情報
integer, intent(inout)::init	!	ループ初期値
integer, intent(in)::last	!	ループ最終値
integer, intent(inout)::step	!	ループ増分値

戻り値 :

real*8::totalEnergy	!	ポテンシャルエネルギー
---------------------	---	-------------

機能 :

拡張 CAP 拘束のポテンシャルエネルギーと、原子に加わる拘束力を計算する。

55.3. Set_RangeParameter : 計算パラメータの設定

名称 : Set_RangeParameter

呼び出し形式 : subroutine Set_RangeParameter (tempRange, &
tempAtomIDForRange, &
tempResidIDForRange, &
tempChainIDForRange, &
tempIsMassAtom, &
tempIsMassResid, &
tempIsMassChain)

引数 :

type(Range_t), dimension(:), intent(in)::tempSphere ! 範囲付き位置拘束情報
integer, dimension(:), intent(in)::tempAtomIDForSphere
! 範囲付き位置拘束対象原子番号
integer, dimension(:), intent(in)::tempResidIDForSphere
! 範囲付き位置拘束対象残基番号
integer, dimension(:), intent(in)::tempChainIDForSphere
! 範囲付き位置拘束対象チェーン番号
logical, dimension(:), intent(in)::tempIsMassAtom ! MASS 指定フラグ
logical, dimension(:), intent(in)::tempIsMassResid ! MASS 指定フラグ
logical, dimension(:), intent(in)::tempIsMassChain ! MASS 指定フラグ

戻り値 :

なし

機能 :

範囲付き位置拘束計算パラメータ、拘束対象原子番号、拘束対象残基番号、
拘束対象チェーン番号および MASS 指定フラグを設定する。

55.4. Transfer_RangeParameter : 計算パラメータの送信

名称 : Transfer_RangeParameter

呼び出し形式 : subroutine Transfer_RangeParameter (optn)

引数 :

type(Option_t), intent(in)::optn ! オプション情報

戻り値 :

なし

機能 :

範囲付き位置拘束計算パラメータを子プロセスに送信する。

55.5. Calc_RangeEnergy : 範囲付き位置拘束計算

名称 : Calc_RangeEnergy

呼び出し形式 : function Calc_RangeEnergy(atom, init, last, step) result (totalEnergy)

引数 :

type(Atom_t), dimension(MAX_ATOM), intent(inout)::atom ! 原子情報
integer, intent(inout)::init ! ループ初期値
integer, intent(in)::last ! ループ最終値
integer, intent(inout)::step ! ループ増分値

戻り値 :

real*8::totalEnergy ! ポテンシャルエネルギー

機能 :

範囲付き位置拘束のポテンシャルエネルギーと、原子に加わる拘束力を計算する。

56. GeneralizedBorn_Method

ファイル名 : GeneralizedBorn_Method.f90

56.1. Allocate_GeneralizedBornData : GB 法の動的メモリ確保

名称 : Allocate_GeneralizedBornData

呼び出し形式 : subroutine Allocate_GeneralizedBornData (atomInfo)

引数 :

type (AtomInfo_t), intent (in) :: atomInfo 原子情報

戻り値 :

なし

機能 :

GB 法で使用する動的な配列を確保する。

すでに確保されている場合には、この領域を破棄して再確保する。

- | | |
|-----------------------|--------------|
| 1) 各原子の電荷 | : radius |
| 2) 各原子の vdW 半径 (計算用) | : vdW |
| 3) 各原子の vdW 半径 (元データ) | : vdWRadius |
| 4) スケールファクター | : scale |
| 5) Born 半径 | : bornRadius |
| 6) 原子への力 | : grad |
| 7) Born 半径での力 | : bornGrad |
| 8) 電荷を持つ原子のリスト | : atomList |
| 9) 各原子のポテンシャル | : energy |

56.2. Set_GeneralizedBornData : GB 法の原子データ設定

名称 : Set_GeneralizedBornData

呼び出し形式 : subroutine Set_GeneralizedBornData (vd, sc, atomInfo)

引数 :

real*8, intent (in), dimension (*) :: vd	vdW 半径
real*8, intent (in), dimension (*) :: sc	スケールファクター
type (AtomInfo_t), intent (in) :: atomInfo	原子情報

戻り値 :

なし

機能 :

次の GB の原子データ情報を設定する。

- | | |
|-----------------------|-------------|
| 1) 各原子の vdW 半径 (元データ) | : vdWRadius |
| 2) スケールファクター | : scale |

56.3. Init_GeneralizedBorn : GB 法の初期設定

名称 : Init_GeneralizedBorn

呼び出し形式 : subroutine Init_GeneralizedBorn (option, offset, atomInfo)

引数 :

type(Option_t), intent(in)::option	オプション情報
real*8, intent(in)::offset	vdw 半径のオフセット
type(AtomInfo_t), intent(in)::atomInfo	原子情報

戻り値 :

なし

機能 :

次の GB の計算パラメータを設定する。

並列実行時には 0 番プロセスから他プロセスへのデータ転送を行う。

- | | |
|----------------------|--------------|
| 1) 各原子の vdw 半径 (計算用) | : vdW |
| 2) 電荷を持つ原子のリスト | : atomList |
| 3) 電荷を持つ原子の個数 | : chargedNum |
| 4) 電荷を持つ原子の電荷 | : charge |

56.4. Create_AtomList : 原子ペアリストの作成

名称 : Create_AtomList

呼び出し形式 : subroutine Create_AtomList (topAtom, lastAtom, atomID, atom, list, num)

引数 :

integer, intent(in)::topAtom	当該原子の 1-5 相互作用テーブル先頭インデックス
integer, intent(in)::lastAtom	当該原子の 1-5 相互作用テーブル最後インデックス
integer, intent(in), dimension(*)::atomID	1-5 相互作用テーブル
type(Atom_t), intent(in)::atom	当該原子情報
integer, intent(out), dimension(*)::list	作成した原子ペアリスト
integer, intent(out)::num	原子ペアリスト数

戻り値 :

なし

機能 :

当該原子に対する、非結合原子と結合原子のペアリストを作成する。

56.5. Calc_GeneralizedBornEnergy : GB 法計算主制御

名称 : Calc_GeneralizedBornEnergy

呼び出し形式 : subroutine Calc_GeneralizedBornEnergy (obj, intr, dyn, updateGB)

引数 :

type(Object_t), intent(inout)::obj	オブジェクト情報
type(Interact_t), intent(in)::intr	相互作用情報
type(Dynamics_t), intent(out)::dyn	ダイナミクス情報
logical, intent(in)::updateGB	GB 計算更新フラグ

戻り値 :

なし

機能 :

Born 半径計算処理 (Calc_BornRadius)、GB 法のエネルギー・力の計算処理 (Calc_GPolarization) を呼び出す。

Born 半径計算は updateGB が true のときのみ呼び出しを行う。

56.6. Calc_BornRadius : Born 半径計算

名称 : Calc_BornRadius

呼び出し形式 : subroutine Calc_BornRadius (cord, lambda, delta, atom,
topAtom, lastAtom, atomID)

引数 :

real*8, intent(in), dimension(3,*)::cord	原子座標
real*8, intent(in)::lambda	体積の Born 半径補正值
real*8, intent(in)::delta	蛋白質の Born 半径補正值
type(Atom_t), intent(in), dimension(*)::atom	原子情報
integer, intent(in), dimension(*)::topAtom	原子リストの先頭インデックス
integer, intent(in), dimension(*)::lastAtom	原子リストの最後インデックス
integer, intent(in), dimension(*)::atomID	相互作用原子 ID リスト

戻り値 :

なし

機能 :

各原子の Born 半径 (bornRadius) を計算する。

56.7. Calc_GPolarization : Born 半径計算

名称 : Calc_GPolarization

呼び出し形式 : function Calc_GPolarization(cord, atom,
saltCoeff, proteinEpsilon, waterEpsilon,
topAtom, lastAtom, atomID, updateGB) result(totalEnergy)

引数 :

real*8, intent(in), dimension(3,*)::cord	原子座標
real*8, intent(in)::proteinEpsilon	蛋白質のエネルギー係数
real*8, intent(in)::waterEpsilon	水のエネルギー係数
real*8, intent(in)::saltCpeff	塩こうか係数
type(Atom_t), intent(in), dimension(*)::atom	原子情報
integer, intent(in), dimension(*)::topAtom	原子リストの先頭インデックス
integer, intent(in), dimension(*)::lastAtom	原子リストの最後インデックス
integer, intent(in), dimension(*)::atomID	相互作用原子 ID リスト
logical, intent(in)::updateGB	GB 計算更新フラグ

戻り値 :

real*8::totalEnergy GeneralizedBorn の総エネルギー

機能 :

各原子の GeneralizedBorn による力とエネルギーを計算する。

エネルギー微分項計算は遠距離の原子ペアと近距離の原子ペアに分離し、updateGB が true の場合のみ遠距離の原子ペアの微分項を計算する。

56.8. Create_AtomList : Born 半径計算

名称 : Create_AtomList

呼び出し形式 : subroutine Create_FarAtomList (topAtom, lastAtom, atomID, atom, current
list, num)

引数 :

integer, intent(in)::topAtom	当該原子の 1-5 相互作用テーブル先頭インデックス
integer, intent(in)::topAtom	当該原子の 1-5 相互作用テーブル最後インデックス
integer, intent(in), dimension(*)::atomID	1-5 相互作用テーブル
type(Atom_t), intent(in), dimension(*)::atom	原子情報
integer, intent(in)::current	当該原子の ID
integer, intent(out), dimension(*)::list	作成した原子ペアリスト
integer, intent(out)::num	原子ペアリスト数

戻り値 :

なし

機能 :

当該原子に対する、カットオフ半径外の原子のペアリストを作成する。

当該原子、相手の原子共に固定原子の場合はペアリストにこの組は登録しない。

57. AccessibleSurface_Method

ファイル名 : AccessibleSurface_Method.f90

57.1. Allocate_AccessibleSurfaceData : ASA 法の動的メモリ確保

名称 : Allocate_AccessibleSurfaceData

呼び出し形式 : subroutine Allocate_AccessibleSurfaceData (atomInfo)

引数 :

type (AtomInfo_t), intent (in) :: atomInfo 原子情報

戻り値 :

なし

機能 :

GB 法で使用する動的な配列を確保する。

すでに確保されている場合には、この領域を破棄して再確保する。

- | | |
|-------------------|----------------|
| 1) 各原子の ASA 半径 | : radius |
| 2) 各原子の係数 | : coeff |
| 3) ASA 向け係数 (保存用) | : coeffSA |
| 4) GB 向け係数 (保存用) | : coeffGB |
| 5) 接触原子リスト | : atomIDTable |
| 6) 接触原子数 | : surfaceNum |
| 7) 非水素原子リスト | : notHydAtomID |

57.2. Set_AccessibleSurfaceData : ASA 法の原子データ設定

名称 : Set_AccessibleSurfaceData

呼び出し形式 : subroutine Set_AccessibleSurfaceData

(asaRadius, asaCoeff, gbCoeff, atomInfo, isNotHyd, asa)

引数 :

real*8, intent (in), dimension (*) :: asaRadius	ASA 半径
real*8, intent (in), dimension (*) :: asaCoeff	ASA 向け係数
real*8, intent (in), dimension (*) :: gbCoeff	GB 向け係数
type (AtomInfo_t), intent (in) :: atomInfo	原子情報
logical, intent (in), dimension (*) :: isNotHyd	非水素原子フラグ
type (AccessibleSurface_t), intent (inout) :: asa	ASA 情報

戻り値 :

なし

機能 :

次の ASA の原子データ情報を設定する。また、非水素原子の個数をカウントし、ASA 情報に設定する。

- | | |
|-------------------|----------------|
| 1) 各原子の ASA 半径 | : radius |
| 2) ASA 向け係数 (保存用) | : coeffSA |
| 3) GB 向け係数 (保存用) | : coeffGB |
| 4) 非水素原子リスト | : notHydAtomID |

57.3. Set_AccessibleSurfaceParameter : ASA 法のパラメータ設定

名称 : Set_AccessibleSurfaceParameter

呼び出し形式 : subroutine Set_AccessibleSurfaceParameter (option, asa)

引数 :

type (Option_t), intent (in) :: option	オプション情報
type (AccessibleSurface_t), intent (in) :: asa	ASA 情報

戻り値 :

なし

機能 :

次の ASA の計算パラメータを設定する。

並列実行時には 0 番プロセスから他プロセスへのデータ転送を行う。

- | | |
|----------------|-------------|
| 1) 各原子の ASA 半径 | : radius |
| 2) 各原子の係数 | : coeff |
| 3) プローブ半径 | : asaProbe |
| 4) ASA のカットオフ長 | : asaCutoff |
| 5) ASA の係数 | : asaWeight |

57.4. Update_AccessibleTable : ASA 法の相互作用テーブル作成

名称 : Update_AccessibleTable

呼び出し形式 : subroutine Update_AccessibleTable (atom)

引数 :

type (Atom_t), intent (in), dimension (*) :: atom	原子情報
---	------

戻り値 :

なし

機能 :

全原子に対し、次の条件を満たす原子を相互作用テーブル (atomIDTable) に登録し、個数を surfaceNum に登録する。

原子間距離 $\leq$ 原子 1 のプローブ半径 + 原子 2 のプローブ半径 + カットオフ距離

57.5. Calc_AccessibleSurfaceEnergy : ASA 法のエネルギー・力計算

名称 : Calc_AccessibleSurfaceEnergy

呼び出し形式 : function Calc_AccessibleSurfaceEnergy (atom) result (energy)

引数 :

type (Atom_t), intent (inout), dimension (*) :: atom	原子情報
--	------

戻り値 :

real*8 :: energy	ポテンシャルエネルギー
------------------	-------------

機能 :

ASA 法のエネルギー、力の計算を行う。

58. SimulatedTemp_Method

ファイル名 : SimulatedTemp_Method.f90

58.1. Init_SimulatedTempMCMD : Simulated-Tempering 法初期化

名称 : Init_SimulatedTempMCMD

呼び出し形式 : subroutine Init_SimulatedTempMCMD (dyn, cond, mon)

引数 :

type (Dynamics_t), intent(in)::dyn	MD 情報
type (Condition_t), intent(in)::cond	MD 条件情報
type (Monitor_t), intent(in)::mon	モニター情報

戻り値 :

なし

機能 :

Simulated-Tempering 法の初期設定と動的な領域の確保を行う。

58.2. Scale_GradientForSimulatedTemp : Simulated-Tempering での力のスケーリング

名称 : Scale_GradientForSimulatedTemp

呼び出し形式 : subroutine Scale_GradientForSimulatedTemp (obj, dyn)

引数 :

type (Object_t), intent(inout)::obj	オブジェクトの情報
type (Dynamics_t), intent(inout)::dyn	MD の情報

戻り値

なし

機能 :

Simulated-Tempering 法での原子に加わる力へのスケーリングと統計データ (温度) のサンプリングを行う。

58.3. Calc_DistributionFromLog : 統計データからのスケーリングファクター計算

名称 : Calc_DistributionFromLog

呼び出し形式 : subroutine Calc_DistributionFromLog

引数 :

なし

戻り値 :

なし

機能 :

温度分布のヒストグラムからスケーリングファクターを計算する。

58.4. Output_Historgam : ヒストグラム出力

名称 : Output_Historgam

呼び出し形式 : subroutine Output_Historgam(loopNum)

引数 :

integer, intent(in)::loopNum	当該 MD ループ回数
------------------------------	-------------

戻り値 :

なし

機能 :

Simulated-Tempering のデータをファイルに出力する。

58.5. Get_Integration : ヒストグラムからの確率計算

名称 : Get_Integration

呼び出し形式 : function Get_Integration(i, binMin, binMax, binSize, histData) result (value)

引数 :

integer, intent(in)::i	確率計算対象のインデックス
integer, intent(in)::binMin	ヒストグラムの最小インデックス
integer, intent(in)::binMax	ヒストグラムの最大インデックス
real*8, intent(in)::binSize	ビンのサイズ
real*8, dimension(binMin:binMax)::histData	ヒストグラムデータ

戻り値 :

real*8::value	当該インデックスでのデータ確率
---------------	-----------------

機能 :

当該インデックスまでの値をとる確率をヒストグラムから求める。

58.6. Read_SimulatedTempRestart : リスタートデータの入力

名称 : Read_SimulatedTempRestart

呼び出し形式 subroutine Read_SimulatedTempRestart(unit)

引数 :

integer, intent(in)::unit	入力装置番号
---------------------------	--------

戻り値 :

なし

機能 :

Simulated-Tempering のデータをリスタートファイルから入力する。

58.7. Write_SimulatedTempRestart : リスタートデータの出力

名称 : Write_SimulatedTempRestart

呼び出し形式 subroutine Write_SimulatedTempRestart(unit)

引数 :

integer, intent(in)::unit	出力装置番号
---------------------------	--------

戻り値 :

なし

機能 :

Simulated-Tempering のリスタートデータをリスタートファイルに出力する。

58.8 Rollback_SimulatedTempRestart : データのロールバック

名称 : Write Rollback_SimulatedTempRestart

呼び出し形式 subroutine Rollback_SimulatedTempRestart

引数 :

なし

戻り値 :

なし

機能 :

Simulated-Tempering のデータをリスタート時点に戻す。

59. ForceBias_Method

ファイル名 : ForceBias_Method.f90

59.1. Init_ForceBiasMCMD : Force-Bias 法初期化

名称 : Init_ForceBiasMCMD

呼び出し形式 : subroutine Init_ForceBiasMCMD (dyn, cond, mon)

引数 :

type(Dynamics_t), intent(in)::dyn	MD 情報
type(Condition_t), intent(in)::cond	MD 条件情報
type(Monitor_t), intent(in)::mon	モニター情報

戻り値 :

なし

機能 :

Force-Bias 法の初期設定と動的な領域の確保を行う。

59.2. Scale_GradientForForceBias : Force-Bias での力のスケーリング

名称 : Scale_GradientForForceBias

呼び出し形式 : subroutine Scale_GradientForForceBias (obj, dyn)

引数 :

type(Object_t), intent(inout)::obj	オブジェクトの情報
type(Dynamics_t), intent(inout)::dyn	MD の情報

戻り値

なし

機能 :

Force-Bias 法での原子に加わる力へのスケーリングと統計データ（温度）のサンプリングを行う。

59.3. Calc_DistributionFromLog : 統計データからのスケーリングファクター計算

名称 : Calc_DistributionFromLog

呼び出し形式 : subroutine Calc_DistributionFromLog

引数 :

なし

戻り値 :

なし

機能 :

温度分布のヒストグラムからスケーリングファクターを計算する。

59.4. Output_Historgam : ヒストグラム出力

名称 : Output_Historgam

呼び出し形式 : subroutine Output_Historgam(loopNum)

引数 :

integer, intent(in)::loopNum	当該 MD ループ回数
------------------------------	-------------

戻り値 :

なし

機能 :

Force-Bias のデータをファイルに出力する。

59.5. Get_Integration : ヒストグラムからの確率計算

名称 : Get_Integration

呼び出し形式 : function Get_Integration(i, binMin, binMax, binSize, histData) result (value)

引数 :

integer, intent(in)::i	確率計算対象のインデックス
integer, intent(in)::binMin	ヒストグラムの最小インデックス
integer, intent(in)::binMax	ヒストグラムの最大インデックス
real*8, intent(in)::binSize	ビンのサイズ
real*8, dimension(binMin:binMax)::histData	ヒストグラムデータ

戻り値 :

real*8::value 当該インデックスでのデータ確率

機能 :

当該インデックスまでの値をとる確率をヒストグラムから求める。

59.6. Read_ForceBiasRestart : リスタートデータの入力

名称 : Read_ForceBiasRestart

呼び出し形式 subroutine Read_ForceBiasRestart(unit)

引数 :

integer, intent(in)::unit 入力装置番号

戻り値 :

なし

機能 :

Force-Bias のデータをリスタートファイルから入力する。

59.7. Write_ForceBiasRestart : リスタートデータの出力

名称 : Write_ForceBiasRestart

呼び出し形式 subroutine Write_ForceBiasRestart(unit)

引数 :

integer, intent(in)::unit 出力装置番号

戻り値 :

なし

機能 :

Force-Bias のリスタートデータをリスタートファイルに出力する。

59.8. Rollback_ForceBiasRestart : データのロールバック

名称 : Rollback_ForceBiasRestart

呼び出し形式 subroutine Rollback_ForceBiasRestart

引数 :

なし

戻り値 :

なし

機能 :

Force-Bias のデータをリスタート時点に戻す。

60. Charmm_Method

ファイル名 : Charmm_Method.f90

60.1. Calc_UreyBradlyEnergy : Urey-Bradly 項の計算

名称 : Calc_UreyBradlyEnergy

呼び出し形式 : function Calc_UreyBradlyEnergy (angle, atom, rigid, virial, init, last, step)
result (energy)

引数 :

type (Angle_t), intent (in), dimension (MAX_Angle) :: angle	angle 情報
type (Atom_t), intent (inout), dimension (MAX_ATOM) :: atom	原子情報
type (Rigid_t), intent (in), dimension (MAX_RIGID) :: rigid	剛体情報
real*8, dimension (3, 3), intent (inout) :: virial	ビリアル
integer, intent (in) :: init	ループ初期値
integer, intent (in) :: last	ループ終値 (angle 数)
integer, intent (in) :: step	ループ増分値

戻り値 :

real*8 :: energy Urey Bradly 項のポテンシャルエネルギー

機能 :

系の全 angle の Urey-Bradly 項 (angle の 1-3 原子間の相互作用) のポテンシャルと力を計算する。

60.2. Calc_CharmmImproperEnergy : CHARMM 用 Improper-Torsion の計算

名称 : Calc_CharmmImproperEnergy

呼び出し形式 : function Calc_CharmmImproperEnergy (improper, atom, virial, init, last, step)
result (energy)

引数 :

type (Torsion_t), intent (in), dimension (MAX_TORSION) :: torsion	improper-torsion 情報
type (Atom_t), intent (inout), dimension (MAX_ATOM) :: atom	原子情報
type (Rigid_t), intent (in), dimension (MAX_RIGID) :: rigid	剛体情報
real*8, dimension (3, 3), intent (inout) :: virial	ビリアル
integer, intent (in) :: init	ループ初期値
integer, intent (in) :: last	ループ終値 (improper-torsion 数)
integer, intent (in) :: step	ループ増分値

戻り値 :

real*8 :: energy Improper-Torsion のポテンシャルエネルギー

機能 :

系の全 Improper-Torsion のポテンシャルと力を計算する。

60.3. Calc_CharmmTorsionalEnergy : CHARMM 用 1-4 相互作用の計算

名称 : Calc_CharmmTorsionalEnergy

呼び出し形式 : subroutine Calc_CharmmTorsionalEnergy (torsion, atom, init, last, step,
potential, vdw14Vir, ele14Vir, isCalcVdw, isCalcEle)

引数 :

type (Torsion_t), intent (in), dimension (MAX_TORSION) :: torsion	torsion 情報
type (Atom_t), intent (inout), dimension (MAX_ATOM) :: atom	原子情報
type (Rigid_t), intent (in), dimension (MAX_RIGID) :: rigid	剛体情報
integer, intent (in) :: init	ループ初期値
integer, intent (in) :: last	ループ終値 (torsion 数)
integer, intent (in) :: step	ループ増分値
real*8, dimension (MAX_POT), intent (out) :: potential	ポテンシャルエネルギー
real*8, dimension (3, 3), intent (inout) :: vdw14Vir	ビリアル (1-4 ele)
real*8, dimension (3, 3), intent (inout) :: ele14Vir	ビリアル (1-4 vdW)
logical, intent (in) :: isCalcVdw	1-4 vdW エネルギー計算フラグ
logical, intent (in) :: isCalcEle	1-4 ele エネルギー計算フラグ

戻り値 :

なし

機能 :

系の全 1-4 相互作用 (静電、van der Waals) のポテンシャルエネルギーと力を求める。

60.4. Calc_Charmm15InteractEnergy : 孤立系の CHARMm 用 1-5 相互作用の計算の主制御

名称 : Calc_Charmm15InteractEnergy

呼び出し形式 : subroutine Calc_Charmm15InteractEnergy (nonBonded, Atom, loopInit, loopLast, dyn, option, updatePeriod)

引数 :

type (NonBondedInfo_t), intent (in) :: nonBonded	非結合相互作用情報
type (Atom_t), intent (inout), dimension (MAX_ATOM) :: Atom	原子情報
integer, intent (in) :: loopInit	ループ初期値
integer, intent (in) :: loopLast	ループ終値 (原子数)
type (Dynamics_t), intent (inout) :: dyn	MD 情報
type (Option_t), intent (in) :: option	オプション情報
integer, intent (in) :: updatePeriod	更新間隔

戻り値 :

なし

機能 :

孤立系に対する CHAMMm 指定時の 1-5 相互作用計算の制御を行う。

60.5. Calc_CharmmVdwCutoff : 孤立系の CHARMm 用 1-5 van der Waals 相互作用の計算

名称 : Calc_Charmm15VdwCutoff

呼び出し形式 : subroutine Calc_Charmm15VdwCutoff (cord, grad, ele, type, energy, eleCoeff, init, last, step, vdwTopAtom, vdwLastAtom, vdwAtomID)

引数 :

real*8, dimension (3, MAX_ATOM) :: cord	各原子座標
real*8, dimension (3, MAX_ATOM) :: grad	各原子への力
real*8, dimension (MAX_ATOM) :: ele	各原子の静電荷
real*8 :: eleCoeff	静電係数
real*8, dimension (MAX_POT) :: energy	ポテンシャルエネルギー
integer, dimension (MAX_ATOM) :: vdwTopAtom	相互作用相手のインデックス先頭
integer, dimension (MAX_ATOM) :: vdwLastAtom	相互作用相手のインデックス最後
integer, dimension (MAX_VDW) :: vdwAtomID	相互作用相手の原子リスト
integer, dimension (MAX_ATOM) :: type	原子タイプ
integer :: init	ループ初期値
integer :: last	ループ終値 (原子数)
integer :: step	ループ増分値

戻り値 :

なし

機能 :

孤立系に対する CHAMMm 指定時の 1-5 van der Waals の力とポテンシャルを計算する。

60.6. Calc_CharmmVdwEleCutoff : 孤立系の CHARMM 用 1-5 静電、van der Waals 相互作用の計算

名称 : Calc_Charmm15VdwEleCutoff

呼び出し形式 : subroutine Calc_Charmm15VdwEleCutoff (cord, grad, ele, type,
energy, eleCoeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID)

引数 :

real*8, dimension(3, MAX_ATOM) :: cord	各原子座標
real*8, dimension(3, MAX_ATOM) :: grad	各原子への力
real*8, dimension(MAX_ATOM) :: ele	各原子の静電荷
real*8 :: eleCoeff	静電係数
real*8, dimension(MAX_POT) :: energy	ポテンシャルエネルギー
integer, dimension(MAX_ATOM) :: vdwTopAtom	相互作用相手のインデックス先頭
integer, dimension(MAX_ATOM) :: vdwLastAtom	相互作用相手のインデックス最後
integer, dimension(MAX_VDW) :: vdwAtomID	相互作用相手の原子リスト
integer, dimension(MAX_ATOM) :: type	原子タイプ
integer :: init	ループ初期値
integer :: last	ループ終値(原子数)
integer :: step	ループ増分値

戻り値 :

なし

機能 :

孤立系に対する CHAMMM 指定時の 1-5 静電と van der Waals の力とポテンシャルを計算する。

60.7. Calc_Charmm15PeriodicEnergy : 周期系の CHARMM 用 1-5 相互作用の計算の主制御

名称 : Calc_Charmm15InteractEnergy

呼び出し形式 : subroutine Calc_Charmm15InteractEnergy (nonBonded, Atom, loopInit, loopLast,
dyn, option)

引数 :

type(NonBondedInfo_t), intent(in) :: nonBonded	非結合相互作用情報
type(Atom_t), intent(inout), dimension(MAX_ATOM) :: Atom	原子情報
integer, intent(in) :: loopInit	ループ初期値
integer, intent(in) :: loopLast	ループ終値(原子数)
type(Dynamics_t), intent(inout) :: dyn	MD 情報
type(Option_t), intent(in) :: option	オプション情報

戻り値 :

なし

機能 :

周期系に対する CHAMMM 指定時の 1-5 相互作用計算の制御を行う。

60.8. Calc_CharmmVdwPeriodic : 周期系の CHARMM 用 1-5 van der Waals 相互作用の計算

名称 : Calc_Charmm15VdwPeriodic

呼び出し形式 : subroutine Calc_Charmm15VdwPeriodic(cord, grad, ele, type,
energy, eleCoeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID, virial)

引数 :

real*8, dimension(3, MAX_ATOM)::cord	各原子座標
real*8, dimension(3, MAX_ATOM)::grad	各原子への力
real*8, dimension(MAX_ATOM)::ele	各原子の静電荷
real*8::eleCoeff	静電係数
real*8, dimension(MAX_POT)::energy	ポテンシャルエネルギー
integer, dimension(MAX_ATOM)::vdwTopAtom	相互作用相手のインデックス先頭
integer, dimension(MAX_ATOM)::vdwLastAtom	相互作用相手のインデックス最後
integer, dimension(MAX_VDW)::vdwAtomID	相互作用相手の原子リスト
integer, dimension(MAX_ATOM)::type	原子タイプ
integer::init	ループ初期値
integer::last	ループ終値(原子数)
integer::step	ループ増分値
type(Virial_t), dimension(MAX_POT)::virial	ビリアル情報
integer, intent(in)::ensemble	アンサンブル
logical, intent(in)::isRigid	剛体モデル使用フラグ
type(Atom_t), dimension(MAX_ATOM), intent(inout)::atom	原子情報

戻り値 :

なし

機能 :

周期系に対する CHAMM 指定時の 1-5 van der Waals の力とポテンシャルを計算する。

60.9. Calc_CharmmVdwElePeriodic : 周期系の CHARMM 用 1-5 静電、van der Waals 相互作用の計算

名称 : Calc_Charmm15VdwElePeriodic

呼び出し形式 : subroutine Calc_Charmm15VdwElePeriodic(cord, grad, ele, type,
energy, eleCoeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID, virial)

引数 :

real*8, dimension(3, MAX_ATOM)::cord	各原子座標
real*8, dimension(3, MAX_ATOM)::grad	各原子への力
real*8, dimension(MAX_ATOM)::ele	各原子の静電荷
real*8::eleCoeff	静電係数
real*8, dimension(MAX_POT)::energy	ポテンシャルエネルギー
integer, dimension(MAX_ATOM)::vdwTopAtom	相互作用相手のインデックス先頭
integer, dimension(MAX_ATOM)::vdwLastAtom	相互作用相手のインデックス最後
integer, dimension(MAX_VDW)::vdwAtomID	相互作用相手の原子リスト
integer, dimension(MAX_ATOM)::type	原子タイプ
integer::init	ループ初期値
integer::last	ループ終値(原子数)
integer::step	ループ増分値
type(Virial_t), dimension(MAX_POT)::virial	ビリアル情報
integer, intent(in)::ensemble	アンサンブル
logical, intent(in)::isRigid	剛体モデル使用フラグ
type(Atom_t), dimension(MAX_ATOM), intent(inout)::atom	原子情報

戻り値 :

なし

機能 :

周期系に対する CHAMMm 指定時の 1-5 静電と van der Waals の力とポテンシャルを計算する。

60.10. Calc_CharmmPotentialParameter : CHARMM 用ポテンシャルパラメータの設定

名称 : Calc_CharmmPotentialParameter

呼び出し形式 : subroutine Calc_CharmmPotentialParameter(nonBond)

引数 :

type(NonBondedInfo_t), intent(in)::nonBond 非結合相互作用情報

戻り値 :

なし

機能 :

CHARMM 用の 1-5, 1-4 van der Waals のパラメータを設定する。

60.11. Set_CharmmCutoff : CHARMM 用カットオフ関連パラメータの算出

名称 : Set_CharmmCutoff

呼び出し形式 : subroutine Set_CharmmCutoff(intr)

引数 :

type(NonBondedInfo_t), intent(in)::nonBond 非結合相互作用情報

戻り値 :

なし

機能 :

CHARMM 用のカットオフに関連する普遍的なパラメータを計算する。

60.12. Init_Charmm : CHARMM ポテンシャル計算用初期設定

名称 : Init_Charmm

呼び出し形式 : subroutine Init_Charmm(charmm)

引数 :

type(Charmm_t), intent(in)::charmm CHARMMm 情報

戻り値 :

なし

機能 :

CHARMM 用の動的な領域を取得する。

60.13. Calc_CharmmRigidVirialTensor : 剛体のビリアル計算 (Charmm ポテンシャル用)

名称 : Calc_CharmmRigidVirialTensor

呼び出し形式 : function Calc_CharmmRigidVirialTensor (atomInfo, rigidInfo, nonBonded, &
optn, residNum, cutlen, vdwReplica) &
result (rigidVir)

引数 :

type(AtomInfo_t), intent(in)::atomInfo 原子情報

type(RigidInfo_t), intent(in)::rigidInfo 剛体情報

type(NonBondedInfo_t), intent(in)::nonBonded 1-5 相互作用情報

type(Option_t), intent(in)::optn オプション情報

integer, intent(in)::residNum 残基数

real*8, intent(in)::cutlen カットオフ長

integer, dimension(MAX_HYD)::vdwReplica vdw 結合の相手原子のレプリカセル ID

戻り値 :

real*8, dimension(3,3):: rigidVir ビリアル行列

機能 :

剛体に掛かる力に関するビリアルを計算する。

計算には、Charmm ポテンシャルによる力の値を使用する。

61. GeneralST_Method

ファイル名 : GeneralST_Method.f90

61.1. Init_GeneralSTMCMD

名称 : Init_GeneralSTMCMD

呼び出し形式 : subroutine Init_GeneralSTMCMD(dyn, cond, mon)

引数 :

type(Dynamics_t), intent(in)::dyn	dynamics 情報
type(Condition_t), intent(in)::cond	計算条件情報
type(Monitor_t), intent(in)::mon	モニター情報

戻り値 :

なし

機能 :

General Simulated Tempering 法の初期化、オプション値の取り込み、およびヒストグラム用の配列の確保を行う。

61.2. Allocate_GuideFunctionData

名称 : Allocate_GuideFunctionData

呼び出し形式 : subroutine Allocate_GuideFunctionData

引数 :

なし

戻り値 :

なし

機能 :

ガイド関数用の 2 次元配列を確保する。

61.3. Scale_GradientForGeneralST

名称 : Scale_GradientForGeneralST

呼び出し形式 : subroutine Scale_GradientForGeneralST(obj, dyn)

引数 :

type(Object_t), intent(inout)::obj	オブジェクト情報
type(Dynamics_t), intent(inout)::dyn	dynamics 情報

戻り値 :

なし

機能 :

General Simulated Tempering 法でのヒストグラム更新、力のスケールファクタ計算、および力のスケーリングを行う。

61.4. Calc_DistributionFromLog

名称 : Calc_DistributionFromLog

呼び出し形式 : subroutine Calc_DistributionFromLog(loopNum, totalEnergy)

引数 :

integer, intent(in)::loopNum	MD ループ数
real*8, intent(in)::totalEnergy	総エネルギー

戻り値 :

なし

機能 :

現在までの MD で蓄積したヒストグラムからスケーリングファクターを再計算する。

61.5. Output_Histogram

名称 : Output_Histogram

呼び出し形式 : subroutine Output_Histogram(loopNum, totalEnergy)

引数 :

integer, intent(in)::loopNum	当該 MD 回数
real*8, intent(in)::totalEnergy	総エネルギー

戻り値 :

なし

機能 :

ヒストグラムの内容をファイルに出力する。

61.6. Calc_ApproxWeight

名称 : Calc_Weight

呼び出し形式 : function Calc_ApproxWeight(energy, lambda) result (weight)

引数 :

real*8, intent(in)::energy	総エネルギー
real*8, intent(in)::lambda	当該アンサンブル

戻り値 :

real*8::weight 当該アンサンブルの重み

機能 :

当該アンサンブルの重みを近似計算する。

61.7. Calc_EnsembleWeight

名称 : Calc_EnsembleWeight

呼び出し形式 : function Calc_EnsembleWeight(energy, lambda) result (lambdaWeight)

引数 :

real*8, intent(in)::energy	total energy
real*8, intent(in)::lambda	当該アンサンブル

戻り値 :

real*8::lambdaWeight 当該アンサンブルの重み

機能 :

当該アンサンブルの重みを厳密に計算する。

61.8. Get_Integration

名称 : Get_Integration

呼び出し形式 : function Get_Integration(i, binMin, binMax, binSize, histData) result (value)

引数 :

integer, intent(in)::i	当該 bin の値
integer, intent(in)::binMin	bin の最小値
integer, intent(in)::binMax	bin の最大値
real*8, intent(in)::binSize	bin の幅
real*8, dimension(binMin:binMax)::histData	ヒストグラムデータ

戻り値 :

real*8::value ヒストグラムの総和量

機能 :

ヒストグラムデータから当該 bin の値までの総和量を計算する。

61.9. Read_GeneralSTRestart

名称 : Read_GeneralSTRestart

呼び出し形式 : subroutine Read_GeneralSTRestart(unit)

引数 :

integer, intent(in)::unit 論理装置番号

戻り値 :

なし

機能 :

Generalized Simulated Tempering のリスタートファイルを読み込む。

61.10. Write_GeneralSTRestart

名称 : Write_GeneralSTRestart

呼び出し形式 : subroutine Write_GeneralSTRestart(unit)

引数 :

integer, intent(in)::unit 論理装置番号

戻り値 :

なし

機能 :

Generalized Simulated Tempering のリスタートファイルを出力する。

61.11. Rollback_GeneralSTRestart

名称 : Rollback_GeneralSTRestart

呼び出し形式 : subroutine Rollback_GeneralSTRestart

引数 :

なし

戻り値 :

なし

機能 :

Generalized Simulated Tempering の物理量をリスタートファイル読み込み時の値に戻す。

62. Calc_Saturation

ファイル名 : Saturation.f90

62.1. Allocate_SatData : SAT ポテンシャルエネルギー計算の動的メモリの確保

名称 : Allocate_SatData

呼び出し形式 : subroutine Allocate_SatData(satValue, maxAtomInChain, maxElements, &
chainInfo, atom)

引数 :

```
real*8, intent(in), dimension(maxAtomInChain, chainInfo%TotalNum)::satValue
integer, intent(in)::maxAtomInChain
integer, intent(in)::maxElements
type(ChainInfo_t), intent(in)::chainInfo
type(Atom_t), intent(in), dimension(MAX_ATOM)::atom
```

戻り値 :

なし。

機能 :

SAT ポテンシャルエネルギー計算で使用する変数の動的メモリの確保を行う。

62.2. Allocate_SatTemporaryArray : SAT ポテンシャルエネルギー計算の動的メモリの確保 (配列)

名称 : Allocate_SatTemporaryArray

呼び出し形式 : subroutine Allocate_SatTemporaryArray

引数 :

なし。

戻り値 :

なし。

機能 :

SAT ポテンシャルエネルギー計算で使用する配列の動的メモリの確保を行う。

62.3. Init_Saturation : SAT パラメータの初期化

名称 : Init_Saturation

呼び出し形式 : Init_Saturation(opt, sat)

引数 :

```
type(Option_t), intent(in)::opt
type(Sat_t), intent(in)::sat
```

戻り値 :

なし。

機能 :

SAT パラメータの初期化および子プロセスへの送信を行う。

62.4. Calc_SatParameter : SAT パラメータの算出

名称 : Calc_SatParameter

呼び出し形式 : subroutine Calc_SatParameter(atom, chain, target, another)

引数 :

```
type(Atom_t), intent(in), dimension(MAX_ATOM)::atom
type(Chain_t), intent(in), dimension(MAX_CHAIN)::chain
integer, intent(in)::target
integer, intent(in)::another
```

戻り値 :

なし。

機能 :

SAT 計算パラメータの算出を行う。

62.5. Calc_SaturationEnergy : SAT ポテンシャルエネルギー計算主制御

名称 : Calc_SaturationEnergy

呼び出し形式 : function Calc_SaturationEnergy(atom, chain) result (energy)

引数 :

type (Atom_t), intent (inout), dimension (MAX_ATOM)::atom
type (Chain_t), intent (in), dimension (MAX_CHAIN)::chain

戻り値 :

real*8::energy

機能 :

SAT ポテンシャルエネルギー計算の主制御を行う。

62.6. eig : 固有値および固有ベクトルの計算

名称 : eig

呼び出し形式 : subroutine eig(matrixSize, eval, evec, ep, rev, rho)

引数 :

integer, intent (in)::matrixSize
real*8, intent (out), dimension (matrixSize)::eval
real*8, intent (out), dimension (maxProton, maxProton)::evec
real*8, intent (out), dimension (maxProton)::ep
real*8, intent (out), dimension (maxProton)::rev
real*8, intent (in), dimension (maxProton, maxProton)::rho

戻り値 :

なし。

機能 :

次数が matrixSize の正方行列について、固有値および固有ベクトルを求める。

62.7. txt : 行列と行列の積の計算

名称 : txt

呼び出し形式 : subroutine txt(matrixSize, evec, xvec, tevect)

引数 :

integer, intent (in)::matrixSize
real*8, dimension (maxProton, maxProton)::evec
real*8, dimension (*)::xvec
real*8, dimension (maxProton, maxProton)::tevect

戻り値 :

なし。

機能 :

2つの行列の積を求める。

62.8. emi : (単位行列一行列) の計算

名称 : emi

呼び出し形式 : subroutine emi(matrixSize, tept)

引数 :

integer, intent (in)::matrixSize
real*8, intent (out), dimension (:, ::) ::tept

戻り値 :

なし。

機能 :

入力された行列に対して、(単位行列一行列) を求める。

62.9. new : 行列とベクトルの積の計算

名称 : new

呼び出し形式 : subroutine new(tept, trevt, si, matrixSize, ne)

引数 :

```

real*8, intent(in), dimension(:, :)::tept
real*8, intent(in), dimension(:, :)::trevt
real*8, dimension(SAT_SIZE)::si
integer::matrixSize
real*8, intent(out), dimension(*)::ne

```

戻り値 :

なし。

機能 :

行列とベクトルの積を求める。

62.10. delexp : SAT ポテンシャルエネルギー計算

名称 : delexp

呼び出し形式 : subroutine delexp(protonNum, evec, eval, ep, trevt, tept, &
sigma, cnh, sath, il, ne, difne, nsanum, sqdne, targetID, &
anotherID)

引数 :

```

integer, intent(in), dimension(2)::protonNum
real*8, intent(in), dimension(maxProton, maxProton)::evec
real*8, intent(in), dimension(maxProton)::eval
real*8, intent(in), dimension(maxProton)::ep
real*8, intent(in), dimension(maxProton, maxProton)::trevt
real*8, intent(in), dimension(maxProton, maxProton)::tept
real*8, intent(in), dimension(maxProton)::sigma
real*8, intent(in), dimension(3, maxProton)::cnh
real*8, intent(in), dimension(3, maxHydNum)::sath
integer, intent(in)::il
real*8, intent(in), dimension(maxProton)::ne
real*8, intent(out), dimension(3, maxProton)::difne
integer, intent(in), dimension(maxProton)::nsanum
real*8, intent(inout)::sqdne
integer, intent(in)::targetID
integer, intent(in)::anotherID

```

戻り値 :

なし。

機能 :

SAT ポテンシャルエネルギーおよび各原子に加わる力を求める。

62.11. Output_SatFile : SAT データの出力

名称 : Output_SatFile

呼び出し形式 : subroutine Output_SatFile(file, atom, chain)

引数 :

```

type(File_t), intent(in)::file
type(Atom_t), intent(in), dimension(MAX_ATOM)::atom
type(Chain_t), intent(in), dimension(MAX_CHAIN)::chain

```

戻り値 :

なし。

機能 :

SAT データをファイルに出力する。

63. Calc_ResidueDipolar

ファイル名 : ResidueDipolar.f90

63.1. Allocate_RDCData : RDC ポテンシャルエネルギー計算の動的メモリの確保

名称 : Allocate_RDCData

呼び出し形式 : subroutine Allocate_RDCData(totalNum)

引数 :

integer, intent(in) :: totalNum

戻り値 :

なし。

機能 :

RDC ポテンシャルエネルギー計算で使用する変数の動的メモリの確保を行う。

63.2. Set_RDCParameter : RDC ポテンシャルエネルギー計算のパラメータ設定

名称 : Set_RDCParameter

呼び出し形式 : subroutine Set_RDCParameter(residID, split, sub, add, name, &
chainID, atom, atomNum)

引数 :

integer, intent(in), dimension(*) :: residID

real*8, intent(in), dimension(*) :: split

real*8, intent(in), dimension(*) :: sub

real*8, intent(in), dimension(*) :: add

character*4, intent(in), dimension(*) :: name

integer, intent(in), dimension(*) :: chainID

type(Atom_t), intent(in), dimension(*) :: atom

integer, intent(in) :: atomNum

戻り値 :

なし。

機能 :

RDC ポテンシャルエネルギー計算で使用するパラメータの設定を行う。

63.3. Calc_ResidueDipolarEnergy : RDC ポテンシャルエネルギー計算

名称 : Calc_ResidueDipolarEnergy

呼び出し形式 : function Calc_ResidueDipolarEnergy(atom, rdc) result (energy)

引数 :

type(Atom_t), intent(inout), dimension(*) :: atom

type(RDC_t), intent(in) :: rdc

戻り値 :

real*8 :: energy

機能 :

RDC ポテンシャルエネルギーおよび各原子に加わる力を求める。

63.3. Output_RDCFile : RDC データの出力

名称 : Output_RDCFile

呼び出し形式 : subroutine Output_RDCFile(file, atom)

引数 :

type(File_t), intent(in) :: file

type(Atom_t), intent(in), dimension(*) :: atom

戻り値 :

なし。

機能 :

RDC のデータをファイルに出力する。

64. Calc_NuclearEffect

ファイル名 : NuclearEffect.f90

64.1. Allocate_NOEData : NOE ポテンシャルエネルギー計算の動的メモリの確保

名称 : Allocate_NOEData

呼び出し形式 : subroutine Allocate_NOEData (totalNum)

引数 :

integer, intent(in) :: totalNum

戻り値 :

なし。

機能 :

NOE ポテンシャルエネルギー計算で使用する変数の動的メモリの確保を行う。

64.2. Set_NOEParameter : NOE ポテンシャルエネルギー計算のパラメータ設定

名称 : Set_NOEParameter

呼び出し形式 : subroutine Set_NOEParameter (chainID1, residID1, residName1, name1, &
chainID2, residID2, residName2, name2, &
dLow, dUpper, rLow, rUpper, atom, chain)

引数 :

integer, intent(in), dimension(*) :: chainID1

integer, intent(in), dimension(*) :: residID1

character(*), intent(in), dimension(*) :: residName1

character(*), intent(in), dimension(*) :: name1

integer, intent(in), dimension(*) :: chainID2

integer, intent(in), dimension(*) :: residID2

character(*), intent(in), dimension(*) :: residName2

character(*), intent(in), dimension(*) :: name2

real*8, intent(in), dimension(*) :: dLow

real*8, intent(in), dimension(*) :: dUpper

real*8, intent(in), dimension(*) :: rLow

real*8, intent(in), dimension(*) :: rUpper

type(Atom_t), intent(in), dimension(*) :: atom

type(Chain_t), intent(in), dimension(*) :: chain

戻り値 :

なし。

機能 :

NOE ポテンシャルエネルギー計算で使用するパラメータの設定を行う。

64.3. Calc_NuclearEffectEnergy : NOE ポテンシャルエネルギー計算

名称 : Calc_NuclearEffectEnergy

呼び出し形式 : function Calc_NuclearEffectEnergy (atom, noe) result (energy)

引数 :

type(Atom_t), intent(inout), dimension(*) :: atom

type(Noe_t), intent(in) :: noe

戻り値 :

real*8 :: energy

機能 :

NOE ポテンシャルエネルギーおよび各原子に加わる力を求める。

48.6. Calc_Histogram : エネルギーのヒストグラム計算

名称 : Calc_Histogram

呼び出し形式 : subroutine Calc_Histogram(dyn, momentum, potential)

引数 :

type(Dynamics_t), intent(in)::dyn

real*8, intent(in)::momentum

real*8, intent(in)::potential

戻り値 :

なし

機能 :

エネルギーのヒストグラムを計算する。

64. Tsallis_Function

ファイル名 : Tsallis_Function.f90

64.1. Get_KTValue : $k \cdot T$ の取得

名称 : Get_KTValue

呼び出し形式 : function Get_KTValue() result (value)

引数 :

なし

戻り値 :

real*8::value

機能 :

$k \cdot T$ (ボルツマン定数*シミュレーション温度)の値を取得する

64.2. Calc_LowerParameter : 低エネルギー側パラメータ計算

名称 : Calc_LowerParameter

呼び出し形式 : function Calc_LowerParameter(totalE) result (f1prime)

引数 :

real*8, intent(in)::totalE

戻り値 :

real*8::f1prime

機能 :

入力ポテンシャルエネルギーに対する低エネルギー側パラメータ f1prime を計算する。

64.3. Calc_ScaleFactor : 速度、座標増分のスケールファクター計算

名称 : Calc_ScaleFactor

呼び出し形式 : function Calc_ScaleFactor(totalJoule) result(scale)

引数 :

real*8, intent(in)::totalJoule

戻り値 :

real*8::scale

機能 :

入力したポテンシャルエネルギーにシフトエネルギーを加算、各エネルギーの値に応じてスケールファクターを計算する。

64.4. Calc_LowerScale : 低エネルギー側スケールファクター計算

名称 : Calc_LowerScale

呼び出し形式 : function Calc_LowerScale(totalE) result(scale)

引数 :

real*8, intent(in)::totalE

戻り値 :

real*8::scale

機能 :

入力したポテンシャルエネルギーに対する低エネルギー側のスケールファクターを計算する。

64. 5. Calc_UpperScale : 高エネルギー側スケールファクター計算

名称 : Calc_LowerScale

呼び出し形式 : function Calc_UpperScale(totalE) result(scale)

引数 :

real*8, intent(in)::totalE

戻り値 :

real*8::scale

機能 :

入力したポテンシャルエネルギーに対する高エネルギー側のスケールファクターを計算する。

64. 6. Calc_ZetaDistributionDensity : ζ 分布計算

名称 : Calc_ZetaDistributionDensity

呼び出し形式 : subroutine Calc_ZetaDistributionDensity(zeta, thetaz, dthetaz)

引数 :

real*8, intent(in)::zeta

real*8, intent(out)::thetaz

real*8, intent(out)::dthetaz

戻り値 :

なし

機能 :

ζ の分布値と ζ の分布の微分値を計算する。

64. 7. Calc_LogOfR0 : $\ln(\rho)$ 計算

名称 : Calc_LogOfR0

呼び出し形式 : function Calc_LogOfR0(totalJoule, RO_MODE) result (dlnR0)

引数 :

real*8, intent(in)::totalJoule ! potential + kinetic energy (JOULE)

character(2), intent(inout)::RO_MODE ! ro mode charactger

戻り値 :

$\ln(\rho)$ の値

機能 :

エネルギーの値に応じた $\ln(\rho)$ の値を計算する。

また、エネルギーの範囲を示す文字列を RO_MODE に設定する。

"Lo" : 低エネルギー側

"Nr" : 通常

"Hi" : 高エネルギー側

64. 8. Calc_NormalLogOfR0 : 通常のエネルギーでの $\ln(\rho)$ 計算

名称 : Calc_NormalLogOfR0

呼び出し形式 : function Calc_NormalLogOfR0(totalJoule) result (dlnR0)

引数 :

real*8, intent(in)::totalJoule ! potential + kinetic energy (JOULE)

戻り値 :

$\ln(\rho)$ の値

機能 :

通常のエネルギーでの $\ln(\rho)$ の値を計算する。

64. 9. Calc_LowerLogOfR0 : 低エネルギーでの $\ln(\rho)$ 計算

名称 : Calc_LowerLogOfR0

呼び出し形式 : function Calc_LowerLogOfR0(totalJoule) result (dlnR0)

引数 :

real*8, intent(in)::totalJoule ! potential + kinetic energy (JOULE)

戻り値 :

 $\ln(\rho)$ の値

機能 :

低エネルギーでの $\ln(\rho)$ の値を計算する。**64. 10. Calc_UpperLogOfR0 : 高エネルギーでの $\ln(\rho)$ 計算**

名称 : Calc_UpperLogOfR0

呼び出し形式 : function Calc_UpperLogOfR0(totalJoule) result (dlnR0)

引数 :

real*8, intent(in)::totalJoule ! potential + kinetic energy (JOULE)

戻り値 :

 $\ln(\rho)$ の値

機能 :

高エネルギーでの $\ln(\rho)$ の値を計算する。**64. 11. Init_TsallisFunction : Tsallis 関数の初期化**

名称 : Init_TsallisFunction

呼び出し形式 : Init_TsallisFunction(dyn, cond)

引数 :

type(Dynamics_t), intent(inout)::dyn

type(Condition_t), intent(in)::cond ! aggregate of condition

戻り値 :

なし

機能 :

Tsallis 関数内の不変な変数群の初期設定を行う。

64. 12. Display_InitTsallis : Tsallis の初期情報表示

名称 : Display_InitTsallis

呼び出し形式 : subroutine Display_InitTsallis

引数 :

なし

戻り値 :

なし

機能 :

Tsallis 用関数内の不変な変数群の値を標準出力に出力する。

65. Smooth_Method

ファイル名 : Smooth_Method.f90

65.1. Init_Smoothing : スプライン補間初期化

名称 : Init_Smoothind

呼び出し形式 : subroutine Init_Smoothing(intr, bound)

引数 :

type(Interact_t), intent(in)::intr 相互作用情報

type(Boundary_t), intent(in)::bound 周期境界情報

戻り値 :

なし

機能 :

van der Waals, 静電のスプライン補間での係数を初期化する。

65.2. Calc_TorsionalSmooth : 1-4 相互作用のスプライン補間

名称 : Calc_TorsionalSmooth

呼び出し形式 : subroutine Calc_TorsionalSmooth(torsion, atom, init, last, step, potential,
vdw14Vir, ele14Vir, isCalcVdw, isCalcEle)

引数 :

type(Torsion_t), intent(in), dimension(MAX_TORSION)::torsion	torsion 情報
type(Atom_t), intent(inout), dimension(MAX_ATOM)::atom	原子情報
integer, intent(in)::init	ループの初期値
integer, intent(in)::last	ループの終値
integer, intent(in)::step	ループのステップ
real*8, dimension(MAX_POT), intent(out)::potential	ポテンシャルエネルギー
real*8, dimension(3, 3), intent(inout)::vdw14Vir	1-4 の vdw のビリアル
real*8, dimension(3, 3), intent(inout)::ele14Vir	1-4 の静電のビリアル
logical, intent(in)::isCalcVdw	vdw 計算フラグ
logical, intent(in)::isCalcEle	静電計算フラグ

戻り値 :

なし

機能 :

torsion の末端の原子に対し、静電、vdw のポテンシャルを計算する。

65.2. Calc_Cutoff15InteractSmooth : 孤立系のスプライン補間付き 1-5 相互作用計算主制御

名称 : Calc_Cutoff15InteractSmooth

呼び出し形式 : subroutine Calc_Cutoff15InteractSmooth (nonBonded, Atom, loopInit, loopLast, dyn, optn, updatePeriod)

引数 :

type (NonBondedInfo_t), intent (in) :: nonBonded	非結合相互作用情報
type (Atom_t), intent (inout), dimension (MAX_ATOM) :: Atom	原子情報
integer, intent (in) :: loopInit	ループの初期値
integer, intent (in) :: loopLast	ループの終値
type (Dynamics_t), intent (inout) :: dyn	分子動力学情報
type (Option_t), intent (in) :: optn	オプション情報
integer, intent (in) :: updatePeriod	相互作用テーブル更新間隔

戻り値 :

なし

機能 :

孤立系の 1-5 相互作用計算の主制御を行う。

注 65.3~65.12 に記述する相互作用計算処理の引数は共通であるため、ここで記述する。

real*8, dimension (3, MAX_ATOM) :: cord	原子座標
real*8, dimension (3, MAX_ATOM) :: grad	原子への力
real*8, dimension (MAX_ATOM) :: ele	原子電荷
real*8 :: eleCoeff	静電係数
real*8, dimension (MAX_TYPE, MAX_TYPE) :: hyd10Coeff	水素結合 10 乗項係数
real*8, dimension (MAX_TYPE, MAX_TYPE) :: hyd12Coeff	水素結合 12 乗項係数
real*8, dimension (MAX_TYPE, MAX_TYPE) :: vdW6Coeff	van der Waals 6 乗項係数
real*8, dimension (MAX_TYPE, MAX_TYPE) :: vdW12Coeff	van der Waals 12 乗項係数
integer, intent (in) :: init	ループ初期値 (処理対象原子先頭)
integer, intent (in) :: last	ループ終値 (処理対象原子の最後)
integer, intent (in) :: step	ループ増分値
real*8, dimension (MAX_POT) :: energy	ポテンシャルエネルギー
integer, dimension (MAX_ATOM) :: vdWTopAtom	van der Waals の先頭インデックス配列
integer, dimension (MAX_ATOM) :: vdWLastAtom	van der Waals の最後インデックス配列
integer, dimension (MAX_VDW) :: vdWAtomID	van der Waals の相手の原子 ID
integer, dimension (MAX_ATOM) :: hydTopAtom	水素結合の先頭インデックス配列
integer, dimension (MAX_ATOM) :: hydLastAtom	水素結合の最後インデックス配列
integer, dimension (MAX_HYD) :: hydAtomID	水素結合の相手の原子 ID
integer, dimension (MAX_ATOM) :: type	原子タイプ

65.3. Calc_VdwSmmoth : スプライン補間付き 1-5 van der Waals 計算

名称 : Calc_VdwSmmoth

呼び出し形式 : subroutine Calc_VdwSmmoth(cord, grad, ele, type, energy, eleCoeff, &
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff, &
init, last, step, &
vdwTopAtom, vdwLastAtom, vdwAtomID, &
hydTopAtom, hydLastAtom, hydAtomID)

引数 :

65.2 を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の van der Waals の力とポテンシャルの計算を行う。

65.4. Calc_HydSmmoth : スプライン補間付き 1-5 水素結合計算

名称 : Calc_HydSmmoth

呼び出し形式 : subroutine Calc_HydSmmoth(cord, grad, ele, type, energy, eleCoeff, &
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff, &
init, last, step, &
vdwTopAtom, vdwLastAtom, vdwAtomID, &
hydTopAtom, hydLastAtom, hydAtomID)

引数 :

65.2 を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の水素結合の力とポテンシャルの計算を行う。

65.5. Calc_ConstEleSmmoth : スプライン補間付き 1-5 距離非依存静電計算

名称 : Calc_ConstEleSmmoth

呼び出し形式 : subroutine Calc_ConstEleSmmoth(cord, grad, ele, type, energy, eleCoeff, &
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff, &
init, last, step, &
vdwTopAtom, vdwLastAtom, vdwAtomID, &
hydTopAtom, hydLastAtom, hydAtomID)

引数 :

65.2 を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の静電の力とポテンシャルの計算を行う。

65. 6. Calc_DependEleSmoth : スプライン補間付き 1-5 距離依存静電計算

名称 : Calc_DependEleSmoth

呼び出し形式 : subroutine Calc_DependEleSmoth(cord, grad, ele, type, energy, eleCoeff, &
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff, &
init, last, step, &
vdwTopAtom, vdwLastAtom, vdwAtomID, &
hydTopAtom, hydLastAtom, hydAtomID)

引数 :

65. 2 を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の静電の力とポテンシャルの計算を行う。

65. 7. Calc_VdwConstEleSmoth : スプライン補間付き 1-5 vdw+距離非依存静電計算

名称 : Calc_VdwConstEleSmoth

呼び出し形式 : subroutine Calc_VdwConstEleSmoth(cord, grad, ele, type, energy, eleCoeff, &
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff, &
init, last, step, &
vdwTopAtom, vdwLastAtom, vdwAtomID, &
hydTopAtom, hydLastAtom, hydAtomID)

引数 :

65. 2 を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の vdw と静電の力とポテンシャルの計算を行う。

65. 8. Calc_VdwDependEleSmoth : スプライン補間付き 1-5 vdw+距離依存静電計算

名称 : Calc_VdwDependEleSmoth

呼び出し形式 : subroutine Calc_VdwDependEleSmoth(cord, grad, ele, type, energy, eleCoeff, &
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff, &
init, last, step, &
vdwTopAtom, vdwLastAtom, vdwAtomID, &
hydTopAtom, hydLastAtom, hydAtomID)

引数 :

65. 2 を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の vdw と静電の力とポテンシャルの計算を行う。

65. 9. Calc_VdwHydSmmoth : スプライン補間付き 1-5 vdw+水素結合計算

名称 : Calc_VdwHydSmmoth

呼び出し形式 : subroutine Calc_VdwHydSmmoth(cord, grad, ele, type, energy, eleCoeff, &
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff, &
init, last, step, &
vdwTopAtom, vdwLastAtom, vdwAtomID, &
hydTopAtom, hydLastAtom, hydAtomID)

引数 :

65. 2 を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の vdw と水素結合の力とポテンシャルの計算を行う。

注意 :

スプライン補間は水素結合をサポートしていないため、本手続きはエラーを出力してプログラムを終了する。

65. 10. Calc_HydConstEleSmmoth : スプライン補間付き 1-5 水素結合+距離非依存静電計算

名称 : Calc_VdwConstEleSmmoth

呼び出し形式 : subroutine Calc_VdwConstEleSmmoth(cord, grad, ele, type, energy, eleCoeff, &
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff, &
init, last, step, &
vdwTopAtom, vdwLastAtom, vdwAtomID, &
hydTopAtom, hydLastAtom, hydAtomID)

引数 :

65. 2 を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の水素結合と静電の力とポテンシャルの計算を行う。

注意 :

スプライン補間は水素結合をサポートしていないため、本手続きはエラーを出力してプログラムを終了する。

65. 11. Calc_HydDependEleSmmoth : スプライン補間付き 1-5 水素結合+距離依存静電計算

名称 : Calc_HydDependEleSmmoth

呼び出し形式 : subroutine Calc_HydDependEleSmmoth(cord, grad, ele, type, energy, eleCoeff, &
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff, &
init, last, step, &
vdwTopAtom, vdwLastAtom, vdwAtomID, &
hydTopAtom, hydLastAtom, hydAtomID)

引数 :

65. 2 を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の水素結合と静電の力とポテンシャルの計算を行う。

注意 :

スプライン補間は水素結合をサポートしていないため、本手続きはエラーを出力してプログラムを終了する。

65. 11. Calc_VdwHvdConstEleSsmoth : スプライン補間付き 1-5vdw+水素結合+距離非依存静電計算

名称 : Calc_VdwHvdConstEleSsmoth

呼び出し形式 : subroutine Calc_VdwHvdConstEleSsmoth(cord, grad, ele, type, energy, eleCoeff, &
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff, &
init, last, step, &
vdwTopAtom, vdwLastAtom, vdwAtomID, &
hydTopAtom, hydLastAtom, hydAtomID)

引数 :

65. 2 を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の水素結合と静電の力とポテンシャルの計算を行う。

注意 :

スプライン補間は水素結合をサポートしていないため、本手続きはエラーを出力してプログラムを終了する。

65. 12. Calc_VdwHvdDependEleSsmoth : スプライン補間付き 1-5vdw+水素結合+距離依存静電計算

名称 : Calc_VdwHvdDependEleSsmoth

呼び出し形式 : subroutine Calc_VdwHvdDependEleSsmoth(cord, grad, ele, type, energy, eleCoeff, &
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff, &
init, last, step, &
vdwTopAtom, vdwLastAtom, vdwAtomID, &
hydTopAtom, hydLastAtom, hydAtomID)

引数 :

65. 2 を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の水素結合と静電の力とポテンシャルの計算を行う。

注意 :

スプライン補間は水素結合をサポートしていないため、本手続きはエラーを出力してプログラムを終了する。

65. 13. Calc_Periodic15InteractSmooth : 周期系のスプライン補間付き 1-5 相互作用計算主制御

名称 : Calc_Periodic15InteractSmooth

呼び出し形式 : subroutine Calc_Periodic15InteractSmooth (nonBonded, Atom, loopInit, loopLast, dyn, optn, updatePeriod)

引数 :

type (NonBondedInfo_t), intent (in) :: nonBonded	非結合相互作用情報
type (Atom_t), intent (inout), dimension (MAX_ATOM) :: Atom	原子情報
integer, intent (in) :: loopInit	ループの初期値
integer, intent (in) :: loopLast	ループの終値
type (Dynamics_t), intent (inout) :: dyn	分子動力学情報
type (Option_t), intent (in) :: optn	オプション情報
integer, intent (in) :: updatePeriod	相互作用テーブル更新間隔

戻り値 :

なし

機能 :

周期系の 1-5 相互作用計算の主制御を行う。

注 65. 14~65. 23 に記述する相互作用計算処理の引数は共通であるため、ここで記述する。

real*8, dimension (3, MAX_ATOM) :: cord	原子座標
real*8, dimension (3, MAX_ATOM) :: grad	原子への力
real*8, dimension (MAX_ATOM) :: ele	原子電荷
real*8, dimension (3, -62:62) :: replicaCord	レプリカセル座標
real*8 :: eleCoeff	静電係数
real*8, dimension (MAX_TYPE, MAX_TYPE) :: hyd10Coeff	水素結合 10 乗項係数
real*8, dimension (MAX_TYPE, MAX_TYPE) :: hyd12Coeff	水素結合 12 乗項係数
real*8, dimension (MAX_TYPE, MAX_TYPE) :: vdW6Coeff	van der Waals 6 乗項係数
real*8, dimension (MAX_TYPE, MAX_TYPE) :: vdW12Coeff	van der Waals 12 乗項係数
integer, intent (in) :: init	ループ初期値 (処理対象原子先頭)
integer, intent (in) :: last	ループ終値 (処理対象原子の最後)
integer, intent (in) :: step	ループ増分値
real*8, dimension (MAX_POT) :: energy	ポテンシャルエネルギー
integer, dimension (MAX_ATOM) :: vdWTopAtom	van der Waals の先頭インデックス配列
integer, dimension (MAX_ATOM) :: vdWLastAtom	van der Waals の最後インデックス配列
integer, dimension (MAX_VDW) :: vdWAtomID	van der Waals の相手の原子 ID
integer, dimension (MAX_HYD) :: vdWReplica	van der Waals の相手原子のレプリカセル ID
integer, dimension (MAX_ATOM) :: hydTopAtom	水素結合の先頭インデックス配列
integer, dimension (MAX_ATOM) :: hydLastAtom	水素結合の最後インデックス配列
integer, dimension (MAX_HYD) :: hydAtomID	水素結合の相手原子の ID
integer, dimension (MAX_HYD) :: hydReplica	水素結合の相手原子のレプリカセル ID
integer, dimension (MAX_ATOM) :: type	原子タイプ
type (Virial_t), dimension (MAX_POT) :: virial	ビリアル

65. 14. Calc_VdwPeriodicSmooth : 周期系のスプライン補間付き van der Waals 計算

名称 : Calc_VdwPeriodic

呼び出し形式 : subroutine Calc_VdwPeriodic(cord, grad, ele, type, energy, eleCoeff,
 replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
 init, last, step,
 vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
 hydTopAtom, hydLastAtom, hydAtomID, hydReplica, virial)

引数 :

65. 13. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 原子に周期境界条件を適用し、van der Waals の力とポテンシャルの計算を行う。

65. 15. Calc_HydPeriodicSmooth : 周期系のスプライン補間付き水素結合計算

名称 : Calc_HydPeriodicSmooth

呼び出し形式 : subroutine Calc_HydPeriodicSmooth(cord, grad, ele, type, energy, eleCoeff,
 replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
 init, last, step,
 vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
 hydTopAtom, hydLastAtom, hydAtomID, hydReplica, virial)

引数 :

65. 13. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 原子に周期境界条件を適用し、水素結合の力とポテンシャルの計算を行う。

注意 :

スプライン補間は水素結合をサポートしていないため、本手続きはエラーを出力してプログラムを終了する。

65. 16. Calc_ConstElePeriodicSmooth : 周期系のスプライン補間付き距離非依存静電計算

名称 : Calc_ConstElePeriodicSmooth

呼び出し形式 : subroutine Calc_ConstElePeriodicSmooth(cord, grad, ele, type, energy, eleCoeff,
 replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
 init, last, step,
 vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
 hydTopAtom, hydLastAtom, hydAtomID, hydReplica, virial)

引数 :

65. 13. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 原子に周期境界条件を適用し、距離非依存静電の力とポテンシャルの計算を行う。

65. 16. Calc_DependElePeriodic : 周期系のスプライン補間付き距離依存静電計算

名称 : Calc_DependElePeriodicSmooth

呼び出し形式 : subroutine Calc_DependElePeriodicSmooth(
 cord, grad, ele, type, energy, eleCoeff,
 replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
 init, last, step,
 vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
 hydTopAtom, hydLastAtom, hydAtomID, hydReplica, virial)

引数 :

65. 13. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 原子に周期境界条件を適用し、距離依存静電の力とポテンシャルの計算を行う。

65. 17. Calc_VdwConstElePeriodicSmooth : 周期系のスプライン補間付き vdw と距離非依存静電計算

名称 : Calc_VdwConstElePeriodicSmooth

呼び出し形式 : subroutine Calc_VdwConstElePeriodicSmooth(
 cord, grad, ele, type, energy, eleCoeff,
 replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
 init, last, step,
 vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
 hydTopAtom, hydLastAtom, hydAtomID, hydReplica, virial)

引数 :

65. 13. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 原子に周期境界条件を適用し、van der Waals と距離非依存静電の力とポテンシャルの計算を行う。

65. 18. Calc_VdwDependElePeriodicSmooth : 周期系のスプライン補間付き vdw と距離依存静電計算

名称 : Calc_VdwDependElePeriodicSmooth

呼び出し形式 : subroutine Calc_VdwDependElePeriodicSmooth(
 cord, grad, ele, type, energy, eleCoeff,
 replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
 init, last, step,
 vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
 hydTopAtom, hydLastAtom, hydAtomID, hydReplica, virial)

引数 :

65. 13. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 原子に周期境界条件を適用し、van der Waals と距離依存静電の力とポテンシャルの計算を行う。

65. 19. Calc_VdwHydPeriodicSmooth : 周期系のスプライン補間付き van der Waals と水素結合計算

名称 : Calc_VdwHydPeriodicSmooth

呼び出し形式 : subroutine Calc_VdwHydPeriodicSmooth(cord, grad, ele, type, energy, eleCoeff,
replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
hydTopAtom, hydLastAtom, hydAtomID, hydReplica, virial)

引数 :

65. 13. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 原子に周期境界条件を適用し、van der Waals と水素結合の力とポテンシャルの計算を行う。

注意 :

スプライン補間は水素結合をサポートしていないため、本手続きはエラーを出力してプログラムを終了する。

65. 20. Calc_HydConstElePeriodicSmooth : 周期系スプライン補間付き水素結合+距離非依存静電計算

名称 : Calc_HydConstElePeriodicSmooth

呼び出し形式 : subroutine Calc_HydConstElePeriodicSmooth(
cord, grad, ele, type, energy, eleCoeff,
replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
hydTopAtom, hydLastAtom, hydAtomID, hydReplica, virial)

引数 :

65. 13. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 原子に周期境界条件を適用し、水素結合と距離非依存静電の力とポテンシャルの計算を行う。

注意 :

スプライン補間は水素結合をサポートしていないため、本手続きはエラーを出力してプログラムを終了する。

65. 21. Calc_HydDependElePeriodicSmooth : 周期系スプライン補間付き水素結合と距離依存静電計算

名称 : Calc_HydDependElePeriodicSmooth

呼び出し形式 : subroutine Calc_HydDependElePeriodicSmooth(
 cord, grad, ele, type, energy, eleCoeff,
 replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
 init, last, step,
 vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
 hydTopAtom, hydLastAtom, hydAtomID, hydReplica, virial)

引数 :

65. 13. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 原子に周期境界条件を適用し、水素結合と距離依存静電の力とポテンシャルの計算を行う。

注意 :

スプライン補間は水素結合をサポートしていないため、本手続きはエラーを出力してプログラムを終了する。

65. 22. Calc_VdwHydConstElePeriodicSmooth : 周期系スプライン補間 vdw, 水素, 距離非依存静電計算

名称 : Calc_VdwHydConstElePeriodicSm

呼び出し形式 : subroutine Calc_VdwHydConstElePeriodicSm(
 cord, grad, ele, type, energy, eleCoeff,
 replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
 init, last, step,
 vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
 hydTopAtom, hydLastAtom, hydAtomID, hydReplica, virial)

引数 :

65. 13. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 原子に周期境界条件を適用し、van der Waals, 水素結合, 距離非依存静電の力とポテンシャルの計算を行う。

65. 23. Calc_VdwHydDependElePeriodicSm : 周期スプライン補間付 vdw, 水素結合, 距離依存静電計算

名称 : Calc_VdwHydDependElePeriodicSm

呼び出し形式 : subroutine Calc_VdwHydDependElePeriodicSm(cord, grad, ele, type, energy,
 eleCoeff,
 replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
 init, last, step,
 vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
 hydTopAtom, hydLastAtom, hydAtomID, hydReplica, virial)

引数 :

65. 13. を参照

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の van der Waals, 水素結合, 距離依存静電の力とポテンシャルの計算を行う。

戻り値 :

なし

機能 :

相互作用テーブルに登録された 1-5 の van der Waals の力とポテンシャルの計算を行う。

65. 24. Calc_PmeEwald15InteractSmooth : PME 向けスプライン補間付き 1-5 相互作用計算主制御

名称 : Calc_PmeEwald15InteractSmooth

呼び出し形式 : subroutine Calc_PmeEwald15InteractSmooth(nonBonded, Atom, loopInit, loopLast, dyn, optn)

引数 :

type (NonBondedInfo_t), intent (in) :: nonBonded	非結合相互作用情報
type (Atom_t), intent (inout), dimension (MAX_ATOM) :: Atom	原子情報
integer, intent (in) :: loopInit	ループの初期値
integer, intent (in) :: loopLast	ループの終値
type (Dynamics_t), intent (inout) :: dyn	分子動力学情報
type (Option_t), intent (in) :: optn	オプション情報

戻り値 :

なし

機能 :

PME/Ewald 法での周期系の 1-5 相互作用計算の主制御を行う。

本モジュールでは、PME/Ewald での静電相互作用の実空間計算を行う。

注 65. 25~65. 23 に記述する相互作用計算処理の引数は共通であるため、ここで記述する。

real*8, dimension (3, MAX_ATOM) :: cord	原子座標
real*8, dimension (3, MAX_ATOM) :: grad	原子への力
real*8, dimension (MAX_ATOM) :: ele	原子電荷
real*8, dimension (3, -62:62) :: replicaCord	レプリカセル座標
real*8 :: eleCoeff	静電係数
real*8, dimension (MAX_TYPE, MAX_TYPE) :: hyd10Coeff	水素結合 10 乗項係数
real*8, dimension (MAX_TYPE, MAX_TYPE) :: hyd12Coeff	水素結合 12 乗項係数
real*8, dimension (MAX_TYPE, MAX_TYPE) :: vdW6Coeff	van der Waals 6 乗項係数
real*8, dimension (MAX_TYPE, MAX_TYPE) :: vdW12Coeff	van der Waals 12 乗項係数
integer, intent (in) :: init	ループ初期値 (処理対象原子先頭)
integer, intent (in) :: last	ループ終値 (処理対象原子の最後)
integer, intent (in) :: step	ループ増分値
real*8, dimension (MAX_POT) :: energy	ポテンシャルエネルギー
integer, dimension (MAX_ATOM) :: vdWTopAtom	van der Waals の先頭インデックス配列
integer, dimension (MAX_ATOM) :: vdWLastAtom	van der Waals の最後インデックス配列
integer, dimension (MAX_VDW) :: vdWAtomID	van der Waals の相手の原子 ID
integer, dimension (MAX_HYD) :: vdWReplica	van der Waals の相手原子のレプリカセル ID
integer, dimension (MAX_ATOM) :: hydTopAtom	水素結合の先頭インデックス配列
integer, dimension (MAX_ATOM) :: hydLastAtom	水素結合の最後インデックス配列
integer, dimension (MAX_HYD) :: hydAtomID	水素結合の相手原子の ID
integer, dimension (MAX_HYD) :: hydReplica	水素結合の相手原子のレプリカセル ID
integer, dimension (MAX_ATOM) :: type	原子タイプ
type (Virial_t), dimension (MAX_POT) :: virial	ビリアル
real*8, intent (in) :: alphaEwald	Ewald 係数

65.25. Calc_EleDirectSmmoth : スプライン補間付き実空間静電計算

名称 : Calc_EleDirectSmmoth

呼び出し形式 : subroutine Calc_EleDirectSmmoth(cord, grad, ele, type, energy, eleCoeff,
replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
hydTopAtom, hydLastAtom, hydAtomID, hydReplica,
virial, alphaEwald)

引数 :

65.24. を参照

戻り値 :

なし

機能 :

PME/Ewald の実空間静電の計算を行う。

65.26. Calc_VdwEleDirectSmmoth : スプライン補間付き van der Waals+実空間静電計算

名称 : Calc_VdwEleDirectSmmoth

呼び出し形式 : subroutine Calc_VdwEleDirectSmmoth(cord, grad, ele, type, energy, eleCoeff,
replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
hydTopAtom, hydLastAtom, hydAtomID, hydReplica,
virial, alphaEwald)

引数 :

65.24. を参照

戻り値 :

なし

機能 :

PME/Ewald の実空間静電および van der Waals の計算を行う。

65.26. Calc_HydEleDirectSmmoth : スプライン補間付き水素結合+実空間静電計算

名称 : Calc_HydEleDirectSmmoth

呼び出し形式 : subroutine Calc_HydEleDirectSmmoth(cord, grad, ele, type, energy, eleCoeff,
replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
hydTopAtom, hydLastAtom, hydAtomID, hydReplica,
virial, alphaEwald)

引数 :

65.24. を参照

戻り値 :

なし

機能 :

PME/Ewald の実空間静電および水素結合の計算を行う。

注意 :

スプライン補間は水素結合をサポートしていないため、本手続きはエラーを出力してプログラムを終了する。

65. 27. Calc_VdwHydEleDirectSmmoth : スプライン補間付き vdw+水素結合+実空間静電計算

名称 : Calc_VdwHydEleDirectSmmoth

呼び出し形式 : subroutine Calc_VdwHydEleDirectSmmoth(cord, grad, ele, type, energy, eleCoeff,
replicaCord, vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff,
init, last, step,
vdwTopAtom, vdwLastAtom, vdwAtomID, vdwReplica,
hydTopAtom, hydLastAtom, hydAtomID, hydReplica,
virial, alphaEwald)

引数 :

65. 24. を参照

戻り値 :

なし

機能 :

PME/Ewald の実空間静電、van der Waals、 および水素結合の計算を行う。

注意 :

スプライン補間は水素結合をサポートしていないため、本手続きはエラーを出力してプログラムを終了する。

65. ECC_Function

ファイル名 : ECC_Function.f90

65.1. Display_ECCSetting : エラー処理付き算術関数の設定情報表示

名称 : Display_ECCSetting

呼び出し形式 : subroutine Display_ECCSetting

引数 :

なし

戻り値 :

なし

機能 :

エラー処理付き算術関数の引数の有効範囲を表示する。

65.2. Set_CommentForECCFunction : エラー時のコメントの設定

名称 : Set_CommentForECCFunction

呼び出し形式 : subroutine Set_CommentForECCFunction(name)

引数 :

character(*), intent(in)::name コメントの内容

戻り値 :

なし

機能 :

エラー付き算術関数でエラーが発生した際に表示するコメントを設定する。
(ここでのコメントは本関数の呼び元の手続き名が望ましい)**65.3. V_EXP : エラー処理付き指数関数**

名称 : V_EXP

呼び出し形式 : function V_EXP(a, comment) result (value)

引数 :

real*8, intent(in)::a EXP の引数

character(*), intent(in)::comment コメント

戻り値

real*8::value

機能 :

引数 a の範囲チェックを行い、EXP(a) の計算値を返す。
引数が範囲外の場合はエラーを出力する。
(ここでのコメントは EXP の計算を示す式が望ましい)**65.3. V_LOG : エラー処理付き対数関数**

名称 : V_LOG

呼び出し形式 : function V_LOG(a, comment) result (value)

引数 :

real*8, intent(in)::a LOG の引数

character(*), intent(in)::comment コメント

戻り値

real*8::value

機能 :

引数 a の範囲チェックを行い、LOG(a) の計算値を返す。
引数が範囲外の場合はエラーを出力する。
(ここでのコメントは LOG の計算を示す式が望ましい)

65. 4. V_DIV : エラー処理付き除算

名称 : V_DIV

呼び出し形式 : function V_DIV(a, b, comment) result (value)

引数 :

real*8, intent(in)::a	分子
real*8, intent(in)::b	分母
character(*), intent(in)::comment	コメント

戻り値

real*8::value

機能 :

0 割の発生チェックを行い、a/b の計算値を返す。

引数が範囲外の場合はエラーを出力する。

(ここでのコメントは除算を示す式が望ましい)

65. 5. V_POW : エラー処理付きべき乗

名称 : V_POW

呼び出し形式 : function V_POW(a, b, comment) result (value)

引数 :

real*8, intent(in)::a	値
integer, intent(in)::b	べき乗の値
character(*), intent(in)::comment	コメント

戻り値

real*8::value

機能 :

a**b の範囲チェックを行い、a**b の計算値を返す。

結果が範囲外の場合はエラーを出力する。

(ここでのコメントは除算を示す式が望ましい)

66. Integrate_Method

ファイル名 : Integrate_Method.f90

66.1. Set_IntegratorFreedom : 自由度の設定

名称 : Set_IntegratorFreedom

呼び出し形式 : subroutine Set_IntegratorFreedom(zeroGrad)

引数 :

logical, intent(in)::zeroGrad 系の力の総和が 0

戻り値 :

なし

機能 :

系の力の総和が 0 の場合、積分器で使用する自由度の値を-3 する。

66.2. Init_GeneralParameter : 共通項目の設定

名称 : Init_GeneralParameter

呼び出し形式 : subroutine Init_GeneralParameter(obj, dyn, cond)

引数 :

type(Object_t), intent(in)::obj オブジェクト情報

type(Dynamics_t), intent(inout)::dyn ダイナミクス情報

type(Condition_t), intent(in)::cond シミュレーション条件情報

戻り値 :

なし

機能 :

各積分器で使用する共通項目を設定する。

66.3. Init_RungeKutta : 4 次 Runge-Kutta 法初期化

名称 : Init_RungeKutta

呼び出し形式 : subroutine Init_RungeKutta(obj, dyn, cond)

引数 :

type(Object_t), intent(in)::obj オブジェクト情報

type(Dynamics_t), intent(inout)::dyn ダイナミクス情報

type(Condition_t), intent(in)::cond シミュレーション条件情報

戻り値 :

なし

機能 :

Runge-Kutta 法の初期化を行う。

66. 4. Exec_RungeKutta : 4 次 Runge-Kutta 法の 1 ステップ実行

名称 : Exec_RungeKutta

呼び出し形式 : subroutine Exec_RungeKutta(obj, intr, rest, cond,
dyn, option, bound, cluster)

引数 :

```

type(Object_t), intent(inout)::obj      ! aggregate of object
type(Interact_t), intent(inout)::intr    ! aggregate of interaction
type(Restrain_t), intent(in)::rest       ! aggregate of restrain
type(Condition_t), intent(in)::cond      ! aggregate of condition
type(Dynamics_t), intent(inout)::dyn     ! aggregate of dynamics
type(Option_t), intent(in)::option       ! aggregate of option
type(Boundary_t), intent(in)::bound      ! aggregate of boundary
type(MultiCluster_t), intent(inout)::cluster ! aggregate of cluster

```

戻り値 :

なし

機能 :

4 次 Runge-Kutta の積分器を実行する。

66. 5. Init_EXVPIntegrator : 拡張相空間体積保存積分法初期化

名称 : Init_EXVPIntegrator

呼び出し形式 : subroutine Init_EXVPIntegrator(obj, dyn, cond)

引数 :

```

type(Object_t), intent(in)::obj      オブジェクト情報
type(Dynamics_t), intent(inout)::dyn  ダイナミクス情報
type(Condition_t), intent(in)::cond   シミュレーション条件情報

```

戻り値 :

なし

機能 :

拡張相空間体積保存積分法の初期化を行う。

66. 6. Exec_EXVPIntegrator : 拡張相空間体積保存積分法の 1 ステップ実行

名称 : Exec_EXVPIntegrator

呼び出し形式 : subroutine Exec_EXVPIntegrator(obj, intr, rest, cond,
dyn, option, bound, cluster)

引数 :

```

type(Object_t), intent(inout)::obj      ! aggregate of object
type(Interact_t), intent(inout)::intr    ! aggregate of interaction
type(Restrain_t), intent(in)::rest       ! aggregate of restrain
type(Condition_t), intent(in)::cond      ! aggregate of condition
type(Dynamics_t), intent(inout)::dyn     ! aggregate of dynamics
type(Option_t), intent(in)::option       ! aggregate of option
type(Boundary_t), intent(in)::bound      ! aggregate of boundary
type(MultiCluster_t), intent(inout)::cluster ! aggregate of cluster

```

戻り値 :

なし

機能 :

拡張相空間体積保存積分法の積分器を実行する。

66.7. Init_PredictorCorrector : Predictor-Corrector 法初期化

名称 : Init_PredictorCorrector

呼び出し形式 : subroutine Init_PredictorCorrector(obj, dyn, cond)

引数 :

type(Object_t), intent(in)::obj	オブジェクト情報
type(Dynamics_t), intent(inout)::dyn	ダイナミクス情報
type(Condition_t), intent(in)::cond	シミュレーション条件情報

戻り値 :

なし

機能 :

Predictor-Corrector 法の初期化を行う。

66.8. Exec_PredictorCorrector : Predictor-Corrector 法の1ステップ実行

名称 : Exec_PredictorCorrector

呼び出し形式 : subroutine Exec_PredictorCorrector(obj, intr, rest, cond,
dyn, option, bound, cluster)

引数 :

type(Object_t), intent(inout)::obj	! aggregate of object
type(Interact_t), intent(inout)::intr	! aggregate of interaction
type(Restrain_t), intent(in)::rest	! aggregate of restrain
type(Condition_t), intent(in)::cond	! aggregate of condition
type(Dynamics_t), intent(inout)::dyn	! aggregate of dynamics
type(Option_t), intent(in)::option	! aggregate of option
type(Boundary_t), intent(in)::bound	! aggregate of boundary
type(MultiCluster_t), intent(inout)::cluster	! aggregate of cluster

戻り値 :

なし

機能 :

Predictor-Corrector 法の積分器を実行する。

66.9. Get_PredictorRestartData : Predictor-Corrector 法のリスタートデータの取得

名称 : Get_PredictorRestartData

呼び出し形式 : subroutine Get_PredictorRestartData

引数 :

なし

戻り値 :

なし

機能 :

Predictor-Corrector 法のリスタートデータを退避領域から本モジュールにコピーする。

66.10. Rollback_PredictorRestartData : Predictor-Corrector 法のリスタートデータの取得

名称 : Rollback_PredictorRestartData

呼び出し形式 : subroutine Rollback_PredictorRestartData

引数 :

なし

戻り値 :

なし

機能 :

Predictor-Corrector 法のリスタートデータを退避領域から本モジュールにコピーし、退避領域を削除する。

66. 11. Put_PredictorRestartData : Predictor-Corrector 法のリスタートデータの退避

名称 : Put_PredictorRestartData

呼び出し形式 : subroutine Put_PredictorRestartData

引数 :

なし

戻り値 :

なし

機能 :

Predictor-Corrector 法のデータを退避領域にコピーする。

66. 12. Init_Gear : GEAR 法初期化

名称 : Init_Gear

呼び出し形式 : subroutine Init_Gear(obj, dyn, cond)

引数 :

type(Object_t), intent(in)::obj	オブジェクト情報
type(Dynamics_t), intent(inout)::dyn	ダイナミクス情報
type(Condition_t), intent(in)::cond	シミュレーション条件情報

戻り値 :

なし

機能 :

Predictor-Corrector 法の初期化を行う。

66. 13. Exec_GearIntegration : GEAR 法の 1 ステップ実行

名称 : Exec_GearIntegration

呼び出し形式 : subroutine Exec_GearIntegration(obj, intr, rest, cond,
dyn, option, bound, cluster)

引数 :

type(Object_t), intent(inout)::obj	! aggregate of object
type(Interact_t), intent(inout)::intr	! aggregate of interaction
type(Restrain_t), intent(in)::rest	! aggregate of restrain
type(Condition_t), intent(in)::cond	! aggregate of condition
type(Dynamics_t), intent(inout)::dyn	! aggregate of dynamics
type(Option_t), intent(in)::option	! aggregate of option
type(Boundary_t), intent(in)::bound	! aggregate of boundary
type(MultiCluster_t), intent(inout)::cluster	! aggregate of cluster

戻り値 :

なし

機能 :

GEAR 法の積分器を実行する。

66. 14. Get_GearRestartData : GEAR 法のリスタートデータの取得

名称 : Get_GearRestartData

呼び出し形式 : subroutine Get_GearRestartData

引数 :

なし

戻り値 :

なし

機能 :

GEAR 法のリスタートデータを退避領域から本モジュールにコピーする。

66. 15. Rollback_GearRestartData : GEAR 法のリスタートデータの取得

名称 : Rollback_GearRestartData

呼び出し形式 : subroutine Rollback_GearRestartData

引数 :

なし

戻り値 :

なし

機能 :

GEAR 法のリスタートデータを退避領域から本モジュールにコピーし、退避領域を削除する。

66. 16. Put_GearRestartData : GEAR 法のリスタートデータの退避

名称 : Put_GearRestartData

呼び出し形式 : subroutine Put_GearRestartData

引数 :

なし

戻り値 :

なし

機能 :

GEAR 法のデータを退避領域にコピーする。

67. Data_Stock

ファイル名 : Data_Stock.f90

67.1. Allocate_StockData : 退避領域の確保

名称 : Allocate_StockData

呼び出し形式 : subroutine Allocate_StockData(intNum, realNum)

引数 :

integer, intent(in)::intNum	整数型データの個数
integer, intent(in)::realNum	実数型データの個数

戻り値 :

なし

機能 :

整数型データと実数型データの退避領域を確保する。

67.2. Deallocate_StockData : 退避領域の解放

名称 : Deallocate_StockData

呼び出し形式 : subroutine Deallocate_StockData

引数 :

なし

戻り値 :

なし

機能 :

退避領域を解放する。

67.3. Set_StockIntScalar : 整数型スカラーのデータ追加

名称 : Set_StockIntScalar

呼び出し形式 : subroutine Set_StockIntScalar(data)

引数 :

integer, intent(in)::data

戻り値 :

なし

機能 :

整数型退避領域にデータを追加する。

67.4. Set_StockInt1Dim : 整数型 1 次元配列のデータ追加

名称 : Set_StockInt1Dim

呼び出し形式 : subroutine Set_StockInt1Dim(data)

引数 :

integer, dimension(1:), intent(in)::data

戻り値 :

なし

機能 :

整数型退避領域にデータを追加する。

67.5. Set_StockInt2Dim : 整数型 2 次元配列のデータ追加

名称 : Set_StockInt2Dim

呼び出し形式 : subroutine Set_StockInt2Dim(data)

引数 :

integer, dimension(1:, 1:), intent(in)::data

戻り値 :

なし

機能 :

整数型退避領域にデータを追加する。

67.6. Set_StockInt3Dim : 整数型 3 次元配列のデータ追加

名称 : Set_StockInt3Dim

呼び出し形式 : subroutine Set_StockInt3Dim(data)

引数 :

integer, dimension(1:, 1:, 1:), intent(in)::data

戻り値 :

なし

機能 :

整数型退避領域にデータを追加する。

67.7. Set_StockRealScalar : 実数型スカラーのデータ追加

名称 : Set_StockRealScalar

呼び出し形式 : subroutine Set_StockRealScalar(data)

引数 :

real*8, intent(in)::data

戻り値 :

なし

機能 :

実数型退避領域にデータを追加する。

67.8. Set_StockReal1Dim : 実数型 1 次元配列のデータ追加

名称 : Set_StockReal1Dim

呼び出し形式 : subroutine Set_StockReal1Dim(data)

引数 :

real*8, dimension(1:), intent(in)::data

戻り値 :

なし

機能 :

実数型退避領域にデータを追加する。

67.9. Set_StockReal2Dim : 実数型 2 次元配列のデータ追加

名称 : Set_StockReal2Dim

呼び出し形式 : subroutine Set_StockReal2Dim(data)

引数 :

real*8, dimension(1:, 1:), intent(in)::data

戻り値 :

なし

機能 :

実数型退避領域にデータを追加する。

67.10. Set_StockReal3Dim : 実数型 3 次元配列のデータ追加

名称 : Set_StockReal3Dim

呼び出し形式 : subroutine Set_StockReal3Dim(data)

引数 :

real*8, dimension(1:, 1:, 1:), intent(in)::data

戻り値 :

なし

機能 :

実数型退避領域にデータを追加する。

67.11. Get_StockIntScalar : 整数型スカラーのデータ取得

名称 : Get_StockIntScalar

呼び出し形式 : subroutine Get_StockIntScalar (data)

引数 :

integer, intent(in)::data

戻り値 :

なし

機能 :

整数型退避領域からデータを取得する。

67.12. Get_StockInt1Dim : 整数型 1 次元配列のデータ取得

名称 : Get_StockInt1Dim

呼び出し形式 : subroutine Get_StockInt1Dim(data)

引数 :

integer, dimension(1:), intent(in)::data

戻り値 :

なし

機能 :

整数型退避領域からデータを取得する。

67.13. Get_StockInt2Dim : 整数型 2 次元配列のデータ取得

名称 : Get_StockInt2Dim

呼び出し形式 : subroutine Get_StockInt2Dim(data)

引数 :

integer, dimension(1:,1:), intent(in)::data

戻り値 :

なし

機能 :

整数型退避領域からデータを取得する。

67.14. Get_StockInt3Dim : 整数型 3 次元配列のデータ取得

名称 : Get_StockInt3Dim

呼び出し形式 : subroutine Get_StockInt3Dim(data)

引数 :

integer, dimension(1:,1:,1:), intent(in)::data

戻り値 :

なし

機能 :

整数型退避領域からデータを取得する。

67.15. Get_StockRealScalar : 実数型スカラーのデータ取得

名称 : Get_StockRealScalar

呼び出し形式 : subroutine Get_StockRealScalar (data)

引数 :

real*8, intent(in)::data

戻り値 :

なし

機能 :

実数型退避領域からデータを取得する。

67.16. Get_StockReal1Dim : 実数型 1 次元配列のデータ取得

名称 : Get_StockReal1Dim

呼び出し形式 : subroutine Get_StockReal1Dim(data)

引数 :

```
real*8,dimension(1:),intent(in)::data
```

戻り値 :

なし

機能 :

実数型退避領域からデータを取得する。

67. 17. Get_StockReal2Dim : 実数型 2 次元配列のデータ取得

名称 : Get_StockReal2Dim

呼び出し形式 : subroutine Get_StockReal2Dim(data)

引数 :

```
real*8,dimension(1:,1:),intent(in)::data
```

戻り値 :

なし

機能 :

実数型退避領域からデータを取得する。

67. 18. Get_StockReal3Dim : 実数型 3 次元配列のデータ取得

名称 : Get_StockReal3Dim

呼び出し形式 : subroutine Get_StockReal3Dim(data)

引数 :

```
real*8,dimension(1:,1:,1:),intent(in)::data
```

戻り値 :

なし

機能 :

実数型退避領域からデータを取得する。

67. 19. Output_StockData : 退避データのファイル出力

名称 : Output_StockData

呼び出し形式 : subroutine Output_StockData(unit)

引数 :

```
integer,intent(in)::unit
```

戻り値 :

なし

機能 :

退避領域を指定された装置番号にバイナリ形式でファイル出力する。

67. 20. Input_StockData : 退避データのファイル入力

名称 : Input_StockData

呼び出し形式 : subroutine Input_StockData(unit)

引数 :

```
integer,intent(in)::unit
```

戻り値 :

なし

機能 :

指定された装置番号からバイナリ形式で退避領域を読み込む。

67. 18. Init_StockIndex : 退避領域のインデックス初期化

名称 : Init_StockIndex

呼び出し形式 : subroutine Init_StockIndex

引数 :

なし

戻り値 :

なし

機能 :

実数型データと整数型データのインデックスを初期化する。

68. Calc_MVO

ファイル名 : MVO.f90

68.1. Calc_MVOEnergy : MVO ポテンシャルエネルギー計算

名称 : Calc_MVOEnergy

呼び出し形式 : function Calc_MVOEnergy(atom, MVO), result(totalEnergy)

引数 :

type(Atom_t), intent(inout), dimension(MAX_ATOM)::atom 原子情報

type(MVO_t), intent(in)::MVO MVO 情報

戻り値 :

real*8::totalEnergy MVO エネルギー

機能 :

MVO ポテンシャルエネルギーおよび力を計算する。

69. Combine_Method

ファイル名 : Combine_Method.f90

69.1. Combine_Main : Combine 法向けポテンシャルエネルギー計算主制御

名称 : Combine_Main

呼び出し形式 : subroutine Combine_Main (molInfo, residInfo, nonBonded, atom, chainInfo, out,
loopInit, loopLast, loopNum)

引数 :

type(MolInfo_t), intent(in)::molInfo	分子情報
type(ResidInfo_t), intent(inout)::residInfo	残基情報
type(NonBondedInfo_t), intent(inout)::nonBonded	1-5 相互作用情報
type Atom_t, intent(inout), dimension(MAX_ATOM)::atom	原子情報
type(FileMonitor_t), intent(in)::out	Combine 法向け出力ファイル情報
integer, intent(in)::loopInit	ループ初期値
integer, intent(in)::loopLast	ループ終値
integer, intent(in)::loopNum	MD ループ回数

戻り値 :

なし。

機能 :

Combine 法ポテンシャルエネルギー計算処理を呼び出す。

最初の実行において、蛋白原子と化合物原子の構成原子と電荷の情報を出力する。

69.2. Calc_CombineVdwConstEle : Combine 法向けポテンシャルエネルギー計算

名称 : Calc_CombineVdwConstEle

呼び出し形式 : subroutine Calc_VdwConstEleELIENocut(cord, ele, type, eleCoeff, &
vdw6Coeff, vdw12Coeff, hyd10Coeff, hyd12Coeff, &
init, last, step, isFreeAtom, funcKind, &
topIndex, lastIndex, &
ensemble, isRigid, residInfo, atom, unit)

引数 :

real*8, dimension(3, MAX_ATOM)::cord	原子座標
real*8, dimension(MAX_ATOM)::ele	原子電荷
real*8::eleCoeff	静電係数
real*8, dimension(MAX_TYPE, MAX_TYPE)::hyd10Coeff	水素結合 10 乗項係数
real*8, dimension(MAX_TYPE, MAX_TYPE)::hyd12Coeff	水素結合 12 乗項係数
real*8, dimension(MAX_TYPE, MAX_TYPE)::vdw6Coeff	van der Waals 6 乗項係数
real*8, dimension(MAX_TYPE, MAX_TYPE)::vdw12Coeff	van der Waals 12 乗項係数
integer, intent(in)::init	ループ初期値(蛋白側先頭残基 ID)
integer, intent(in)::last	ループ終値(蛋白側最後残基 ID)
integer, intent(in)::step	ループ増分値
logical, dimension(MAX_ATOM)::isFreeAtom	自由原子フラグ
integer, dimension(MAX_TYPE, MAX_TYPE)::funcKind	ポテンシャル関数種別マトリックス
integer::topIndex	化合物先頭原子 ID
integer::lastIndex	化合物最後原子 ID
integer::ensemble	アンサンブル
logical::isRigid	剛体モデル使用フラグ
type(ResidInfo_t), intent(in)::residInfo	残基情報
type Atom_t::atom	原子情報
integer::unit	出力ファイルの装置番号

戻り値 :

なし。

機能 :

蛋白側原子(先頭残基から最後残基までに属する原子)と化合物側原子(先頭原子から最後原子)の 1-5 静電(定誘電率)エネルギー計算を no-cut で行う。

— 以上 —