NanoDome

H2020-NMP-2014-two-stage

NMP-20-2014 Widening materials models

Grant Agreement n. 646121

DELIVERABLE 4.1

Specifications of data format and functions

WP number: WP4	**Lead Beneficiary: UNIBO**
Type: Report	**Public**
Start of the project: 15/09/2015	**Due date: month 6 (15/03/2016)**

NanoDome

H2020-NMP-2014-two-stage

NMP-20-2014 Widening materials models

Grant Agreement n. 646121

DELIVERABLE 4.1

Interface and Linking Description Document

Contents

1 Introduction

This document is to be intended as an extension of the Deliverable 3.2. Following the directives described in D.3.2, this can be considered a part of an SDD. Software Design Document (ISO 1016 2009 [1]) and the design views addressed here are the ones related to the file formats for loading and managing the data provided by the atomistic scale Molecular Dynamics simulations, files for linking with other CFD simulation environments and the definition of the API functions for coupling the mesoscale simulation with the continuum regime.

2 Material Data File Formats

In this section we describe the XML and JSON file formats for loading data describing the material. This file will provide data and parameters regarding the nanoparticle nucleation, sintering and growth. In order to maximize the portability of these file, we decided to use XML and JSON standards to format the data. Moreover, for parsing, reading and writing these standards is available a wide spectrum of libraries and softwares (open or not). In NanoDome will be integrated the one that fits best the requirements of reliability, usability and integrability imposed by the project. At this stage of the project, regarding on the formation of nanoparticles, we refer to Classical Nucleation Theory (CNT), described in Section 3.4.2 of D.3.1. Other behaviors and parameters for reactive species (like SiH_4) will be considered following results of WP1 and WP2.

2.1 XML Input Files

In this section is presented the XML file for the materials data. This file will be used to load all the parameters for describing the nucleation, the sintering and the growth of nanoparticles inside the system.

Listing 1: XML Material Data File

```
<?xml version="1.0" encoding="UTF-8"?>
<MATERIALS>
 <N_SPECIES> Param </N_SPECIES>
 <A_SPECIES>
  <NAME> Species name </NAME>
  <SYMB> Species symbol </SYMB>
  <SINTER>
   <E_SINT> Param </E_SINT>
   <A_SINT> Param </A_SINT>
  </SINTER>
  <NUCL>
   <RADIUS> Param </RADIUS>
   <VOL> Param </VOL>
   <MASS> Param </MASS>
   <SURF_A> Param </SURF_A>
   <P_SAT_PAR> M </P_SAT_PAR>
    <T_SAT> List </T_SAT>
    <P_SAT> List </P_SAT>
   <SIGMA_PAR> N </SIGMA_PAR>
    <T_SIGMA> List </T_SIGMA>
    <SIGMA> List  </SIGMA>
  </NUCL>
  <GROW>
   <DS> Param </DS>
  </GROW>
 </A_SPECIES>
</MATERIALS>
```

The root of the tree is the <MATERIALS> tag that can have n (<N_SPECIES>) sons called <A_SPECIES>. Each <A_SPECIES> tag describes the molecular species with its name (<NAME>), symbol (<SYMB>), sintering parameters (<SINTER>), growth parameters (<GROW>) and nucleation (<NUCL>) parameters. The <SINTER> tag encapsulate all the sintering parameters: E_{sint} (<E_SINT>) and A_{sint} (<A_SINT>) described in equation 3.50 of D.3.1. The <GROW> encapsulates the diffusion parameter, <DS> for the growth of the particle, referred to equation 3.21 of D.3.1. The <NUCL> tag is referred to the nucleation parameters for equations 3.16 and 3.17 of D.3.1. <RADIUS>, <VOL>, <MASS>, <SURF_A> are the particle radius r_0, the volume v_s, the mass m_s and the surface area s_s respectively. <P_SAT_PAR> indicates the cardinality of samples taken

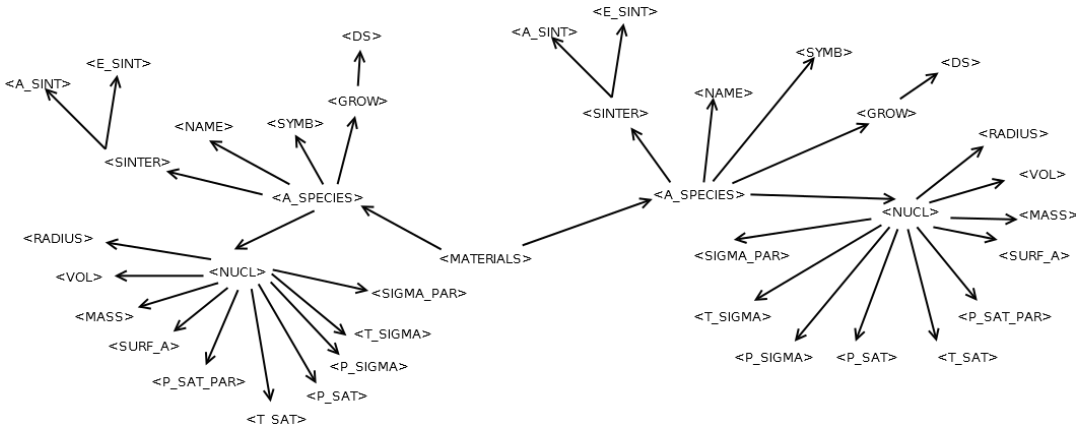

Figure 1: Material Data XML Tree

to describe (using a spline or a linear interpolation) the partial saturation pressure <P_SAT>, p_s^{sat} in function of temperature (<T_SAT>), T for each species. <SIGMA_PAR> indicates the cardinality of sample taken to describe (using a spline or a linear interpolation) the surface tension <SIGMA> σ_s in function of temperature (<T_SIGMA>) T for each species. The tree of the XML file is depicted in Figure 1.

2.2 JSON Input Files

In this section is presented the JSON file for storing the material characteristics.

Listing 2: JSON Material Data File

```
{"Materials" :
{
 "N_Species":"param",
 "Species":[
  {
  "Name":"param",
  "Symb":"param",
  "Sinter":
  {
   "A_sint":"Param",
   "E_sint":"Param",
  },
  "Nucl":
  {
   "Radius":"Param",
   "Vol":"Param",
   "Mass":"Param",
   "Surf_a":"Param",
   "P_sat_par":"Param",
   "T_sat":[],
   "P_sat":[],
   "Sigma_par":"Param",
   "T_sigma":[],
   "Sigma":[]
  },
 "Grow":{"Ds":"Param"}
 }
 ]
}
}
```

The names inside the object *Material* are the same identified by the tags of XML file reported in section 2.2. The name *Species* is an array containing the objects describing the properties of the materials. For each material, we have the same parameters and lists exposed in the XML file.

3 Potential Input Files

In this section is described the file format to manage the data related to the inter-particle potential [1] evaluated using the Force Matching Approach, as formalized in section 3.5.1 of D. 3.1. Following the approach chosen for the file describing the material data, XML and JSON are the languages used to accomplish this task.

3.1 XML Input Files

The XML file is structured as follows:

Listing 3: XML Potential Input File

```
<POTENTIAL>
 <N_SPECIES> Param </N_SPECIES>
 <SPECIES>
  <NAME> Param  </NAME>
  <SYMB> Species symbol </SYMB>
  <N_COEFF> N </N_COEFF>
  <A_N> N Coefficients </A_N>
 </SPECIES>
 <SPECIES>
  <NAME> Param  </NAME>
  <SYMB> Species symbol </SYMB>
  <N_COEFF> N </N_COEFF>
  <A_N> N Coefficients </A_N>
 </SPECIES>
 <SPECIES>
  <NAME> Param  </NAME>
  <N_COEFF> N </N_COEFF>
  <A_N> N Coefficients </A_N>
 </SPECIES>
</POTENTIAL>
```

The root tag is <POTENTIAL> encapsulating an arbitrary number (<N_SPECIES>) of species (<SPECIES>). Each species contains a number (<N_COEFF>) of A_n coefficients (<A_N>) used in FMA to describe the interparticle potentials.

The tree is described in Figure 2.

[1] The file presented is designed to manage data for mono-species particles. More complex interactions among particles composed by reactive species or different species (e.g. $Al_2O_3 - Pt$), are not taken into account at this stage of the project. These aspects will be considered after the completion of tasks T.1.1, T.1.2, T1.3, T1.4

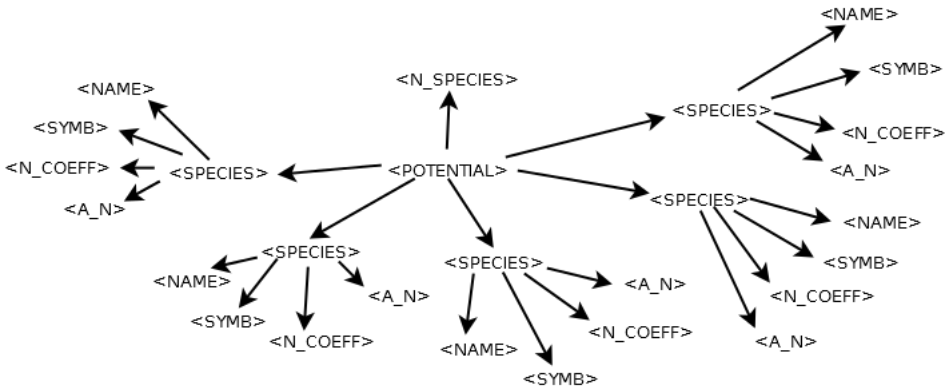

Figure 2: Potential File XML Tree

3.2 JSON Input Files

In this section is presented the JSON file for storing the data regarding the inter-particles potential.

Listing 4: JSON Potential Input File

```
{"Potential":
{
 "N_species":"Param",
 "Species":[
 {
  "Name":"Param",
  "Symb":"Param",
  "N_coeff":"Param",
  "An":[]
 }
 ]
}
}
```

The main object contains the name *Potential* that encapsulates the names *N_species* and *Species*. The name *Species* is an array of objects with the same data listed in the `<SPECIES>` tag of the XML file.

4 CFD Linking

In this section is reported the file format for linking a mesoscopic simulation to a continuum simulation. The concept of linking is to post-process the data produced by a CFD simulation, divided into streamlines, with the mesoscopic simulation ad send back to the CFD updated gas-phase parameters, using the same file format. In this case too, we present the file formats in XML and JSON.

4.1 XML Input Files

The XML file is structured as follows:

Listing 5: XML CFD Linking File

```
<GP>
 <N_STREAM> Param </N_STREAM>
 <T_START> Param </T_START>
 <T_END> Param </T_END>
 <STREAM>
  <ID> Param </ID>
  <N_TIME_SAMPLES> N </N_TIME_SAMPLES>
  <TIME_SAMPLES> list of N Time samples </TIME_SAMPLES>
  <T> list of N Temperature samples </T>
  <P> List of N  Pressure samples </P>
  <N_SPIECES> Ns </N_SPIECES>
  <SPECIES> Ns Values </SPECIES>
  <MOLAR_C> Ns Values </MOLAR_C>
 </STREAM>
</GP>
```

The tree of the XML file is described in Figure 3.

The root tag for the file is <GP> encapsulating <N_STREAM> streamlines (<STREAM>) with a starting time <TIME_START> and an ending time <TIME_END>. Each streamline is composed by an univocal index, <ID>, the number of samples, <N_TIME_SAMPLES> for interpolating in time <TIME_SAMPLES>, the temperature, <T> and pressure <P>values of the streamline. Each streamline contains a list of molecular species <SPECIES> and for each species, its molar concentration value <MOLAR_C>.

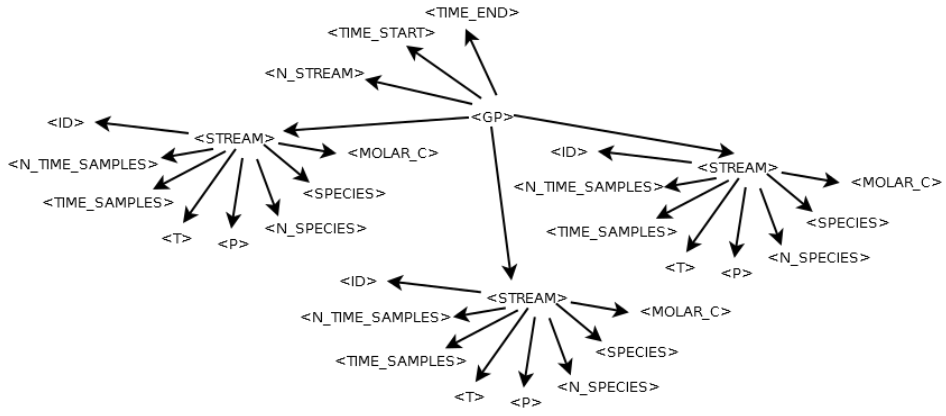

Figure 3: CFD Link XML File tree

4.2 JSON Input Files

In this section is described the JSON file designed to link the mesoscopic simulation to the continuum one.

Listing 6: JSON CFD Linking File

```
{"Gf":
 {
  "N_stream":"Param",
  "T_start":"Param",
  "T_end":"Param",
  "Streams":
    [
     {
      "Id":"Param",
      "N_time_samples":"Param",
      "Time_samples":[],
      "T":[],
      "P":[],
      "N_species":"Param",
      "Species":[],
      "Molar_c":[]
     }
    ]
 }
}
```

The main object is called *Gf* and has the same data contained in the XML version. The *Stream* name is an array containing all the streamlines exported from the CFD or with the updated parameters for updating the CFD simulation. Each object in the *Streams* array has the same parameters described in the `<STREAM>` tag.

5 CFD Coupling

For coupling the CFD with the mesoscopic simulation we mean the use of custom APIs that connect directly the two environments running at the same time on the same machine (cluster of computers or workstation). The fundamental difference with the linking process is that the computations and the data communication among the mesoscopic simulations and the CFD's streamlines is performed without using any files and with a time step smaller than the one used in linking.

NDStreamline
+id: integer **+n_time_samples: integer** **+time_samples: list<double>** **+T_samples: list<double>** **+P_samples: list<double>** **+n_species: integer** **+stream_species: list<string>** **+molar_c: list<double>**
+NDStreamline() **+_NDStreamline()**

Table 1: NDStreamline class

The interface methods can be described using two simple functions:

- `MESOtoCFD(NDStreamline s[]).`
- `CFDtoMESO(NDStreamline s[]).`

Each of these functions take as a parameter an array of objects of the `streamline` class, that incorporates all the parameters described in section 4 of this document. In task T.4.2 we will develop the library to properly couple the mesoscopic simulation with ANSYS FLUENT and OPEN FOAM. In task T.4.3 we will define the methods and algorithms implementing the linking/coupling process.

The `streamline` class is described in the Table 1.

6 Chemical Kinetics

The XML based Cantera CTML (Cantera Markup Language) format is currently used by CMCL's proprietary software kinetics™ for data storage and representation. As part of the NanoDome project's tasks, the Chemkin format will be included as a new data storage format for kinetics™.

Available Chemkin mechanisms will be used for process modelling. It will be possible to automatically convert Chemkin into CTML files and viceversa.

The Chemkin format facilitates the formation, solution and interpretation of problems involving elementary gas-phase chemical kinetics, and allows to incorporate complex chemical kinetic mechanisms into reactor modelling and multidimensional CFD toolkits. Thermodynamic and mechanism data can be incorporated in Chemkin format using:

- A mechanism file (often with the suffix *.inp) containing elements, species, reactions and rate data.
- A thermodynamic data file (suffix *.therm) containing the polynomials employed to compute thermodynamic data.

For example, the general structure of a Chemkin file for mechanism data includes the elements, species and reactions, as shown in listing 7.

Listing 7: Chemkin File

```
ELEMENTS  A  B/2.014  C  D  ...  END

SPECIES  AB  A2  AC  CD3  ...  ...  END

REACTIONS

 A + B = AB                    x1  y1  z1

 2C + D = C2D                  x2  y2  z2

 AB + M = A + D + M            x3  y3  z3  !Third-body reaction

 D/1.8/  AC/2/  AD/3.2/

 A + B(+M) = D(+M)             x3  y3  z3  !Fall-off reaction

 LOW                           /x3  y3  z3/

 ...

 ...

END
```

Elements and species identifier must be followed by at least one space. For an isotope, the atomic weight must follow the identifying symbol and be delimited by slashes (/). Each chemical reaction entry is divided into two fields.

The first contains the symbolic description of the reaction, while the second contains the Arrhenius rate coefficients (x is the pre-exponential factor, y the temperature exponent and z the activation energy).

For each reaction one or more auxiliary information lines can be provided. In this a character-string keyword that defines the reaction type is followed by a slash-delimited (/) field containing an appropriate number of parameters (either integer, floating point, or E format).

Chemkin format can be used to represent a variety of reactions, including reversible, third-body, fall-off and Landau-Teller reactions. In the specific case of a third body reaction, for example, M is added as a reactant and/or product. For third-body reactions the keyword is the name of the third body species. Pressure dependencies can also be specified.

This are indicated by a (+M) or by a species contained in parentheses in the

reaction. For a fall-off reactions, the keyword `LOW` must appear on the auxiliary information line, and the three low-pressure limit rate parameters have to be specified between dashes (/).

Any chemical species that appears in a simulation must have thermodynamic data associated with it. The Chemkin package is designed to work with thermodynamic data (molar heat capacities at constant pressure, standard-state molar enthalpies and standard-state molar entropies) in the form used in the NASA chemical equilibrium code. In this case, seven coefficients are needed for each of two temperature range.

The $*$.therm file includes the species name, the elemental composition of species, the temperature ranges over which the polynomial fits to thermodynamic data are valid and the fits to C_p^0/R, H^0/RT, and S^0/R, consisting of seven coefficients for each of two temperature ranges.

A wide range of other rules for mechanism and thermodynamic data representation are standardised by Chemkin and can be sourced from its user manual.

A Files Examples

In this section of the Appendix we report some examples for the files described in this document.

A.1 Material Files Examples

We want to describe two materials, Species1 (Sp1) and Species2 (Sp2). Each of these materials has the following dummy parameters:

- Species1:
  - **Sintering**: $A_{sint} = 5 \cdot 10^{-14}$, $E_{sint} = 25000$, $Dp = 20 \cdot 10^{-9}$.
  - **Nucleation**: $r_{0,Sp1} = 0.000002\ m$,
    $v_{Sp1} = 0.0000023\ m^3$, $m_{Sp1} = 0.00012\ Kg$, $s_{Sp1} = 0.00000002\ m_2$,
    $Tsamples = \{3000.0, 2978.5, 2845.5, 2598.3\}$,
    $p^{sat}_{Sp1}samples = \{5 \cdot 10^3, 3 \cdot 10^3, 2 \cdot 10^3, 1 \cdot 10^3\}$,
    $\sigma_{Sp1} = \{650.56, 645.67, 634.53, 612.35\}$.
  - **Growth**: $D_{Sp1} = 1.5$.
- Species2:
  - **Sintering**: $A_{sint} = 1.3 \cdot 10^{-13}$, $E_{sint} = 6000$, $Dp = 54 \cdot 10^{-9}$.
  - **Nucleation**: $r_{0,Sp2} = 0.0000013\ m$, $v_{Sp2} = 0.0000057\ m^3$, $m_{Sp2} = 0.00021\ Kg$, $s_{Sp2} = 0.00000012\ m_2$,
    $Tsamples = \{3000.0, 2978.5, 2845.5, 2598.3\}$,
    $p^{sat}_{Sp2}samples = \{34 \cdot 10^3, 35 \cdot 10^3, 38 \cdot 10^3, 39 \cdot 10^3\}$,
    $\sigma_{Sp2} = \{350.56, 345.67, 334.53, 312.35\}$.
  - **Growth**: $D_{Sp2} = 2.4$.

Listing 8: XML Material Data File Example

```
<?xml version="1.0" encoding="UTF-8"?>
<MATERIALS>
 <N_SPECIES> 2 </N_SPECIES>
 <A_SPECIES>
  <NAME> Species1 </NAME>
  <SYMB> Sp1 </SYMB>
  <SINTER>
   <E_SINT> 25000 </E_SINT>
   <A_SINT> 5.0e-14 </A_SINT>
   <DP> 20.0e-9 </DP>
  </SINTER>
  <NUCL>
   <RADIUS> 0.000002 </RADIUS>
   <VOL> 0.0000023 </VOL>
   <MASS> 0.00012 </MASS>
   <SURF_A> 0.00000002 </SURF_A>
   <P_SAT_PAR> 4 </P_SAT_PAR>
    <T_SAT>3000.0 2978.5 2845.5 2598.3</T_SAT>
    <P_SAT>5.0e3 3.0e3 2.0e3 1.0e3</P_SAT>
   <SIGMA_PAR> 4 </SIGMA_PAR>
    <T_SIGMA>3000.0 2978.5 2845.5 2598.3</T_SIGMA>
    <SIGMA>650.56 645.67 634.53 612.35</SIGMA>
  </NUCL>
  <GROW>
   <DS> 1.5 </DS>
  </GROW>
 </A_SPECIES>
 <A_SPECIES>
  <NAME> Species2 </NAME>
  <SYMB> Sp2 </SYMB>
  <SINTER>
   <E_SINT> 6000 </E_SINT>
   <A_SINT> 1.3e-13 </A_SINT>
   <DP> 54.0e-9 </DP>
  </SINTER>
  <NUCL>
   <RADIUS> 0.0000013 </RADIUS>
   <VOL> 0.0000057 </VOL>
   <MASS> 0.00021 </MASS>
```

```
   <SURF_A> 0.00000012 </SURF_A>
   <P_SAT_PAR> 4 </P_SAT_PAR>
    <T_SAT>3000.0 2978.5 2845.5 2598.3</T_SAT>
    <P_SAT>34.0e3, 35.0e3, 38.0e3, 39.0e3</P_SAT>
   <SIGMA_PAR> 4 </SIGMA_PAR>
    <T_SIGMA>3000.0 2978.5 2845.5 2598.3</T_SIGMA>
    <SIGMA>350.56 345.67 334.53 312.35</SIGMA>
  </NUCL>
  <GROW>
   <DS> 2.4 </DS>
  </GROW>
 </A_SPECIES>
</MATERIALS>
```

Listing 9: JSON Material Data File Example

```
{"Materials" :
{
 "N_Species":"2",
 "Species":[
  {
  "Name":"Species1",
  "Symb":"Sp1",
  "Sinter":
  {
   "A_sint":"5.0e-14",
   "E_sint":"25000",
   "Dp":"20.0e-9"
  },
  "Nucl":
  {
   "Radius":"0.000002",
   "Vol":"0.0000023",
   "Mass":"0.00012",
   "Surf_a":"0.00000002",
   "P_sat_par":"4",
   "T_sat":[3000.0, 2978.5, 2845.5, 2598.3],
   "P_sat":[5.0e3 3.0e3 2.0e3 1.0e3],
   "Sigma_par":"4",
   "T_sigma":[3000.0 2978.5 2845.5 2598.3],
   "Sigma":[650.56 645.67 634.53 612.35]
  },
 "Grow":{"Ds":"1.5"}
 },
 {
  "Name":"Species2",
  "Symb":"Sp2",
  "Sinter":
  {
   "A_sint":"1.3e-13",
   "E_sint":"6000",
   "Dp":"54.0e-9"
  },
  "Nucl":
  {
```

```
    "Radius":"0.0000013",
    "Vol":"0.0000057",
    "Mass":"0.00021␣",
    "Surf_a":"0.00000012",
    "P_sat_par":"4",
    "T_sat":[3000.0, 2978.5, 2845.5, 2598.3],
    "P_sat":[34.0e3, 35.0e3, 38.0e3, 39.0e3],
    "Sigma_par":"4",
    "T_sigma":[3000.0 2978.5 2845.5 2598.3],
    "Sigma":[350.56 345.67 334.53 312.35]
   },
  "Grow":{"Ds":"2.4"}
  }
]
}
}
```

A.2 Potential File Examples

We want to describe the potentials, using the Force Matching Approach, for two species, Species1 (Sp1) and Species2 (Sp2) with the following dummy values:

- Sp1:
  - $A_n = \{0.12356, -12583.45, 1240854.1234, 124095.2345, -5679.341\}$.
- Sp2:
  - $A_n = \{125067.01254, 340653.56, 395403.6745, 34565.9832, 435234.56\}$.

Listing 10: XML Potential Data Example

```
<POTENTIAL>
 <N_SPECIES> 2 </N_SPECIES>
 <SPECIES>
  <NAME> Species1  </NAME>
  <SYMB> Sp1 </SYMB>
  <N_COEFF> 5 </N_COEFF>
  <A_N>0.12356 -12583.45 1240854.1234
        124095.2345 -5679.341</A_N>
 </SPECIES>
 <SPECIES>
  <NAME> Species2  </NAME>
  <SYMB> Sp2 </SYMB>
  <N_COEFF> 5 </N_COEFF>
  <A_N> 125067.01254 340653.56 395403.6745
        34565.9832 435234.56</A_N>
 </SPECIES>
</POTENTIAL>
```

Listing 11: JSON Potential Data Example

```
{"Potential":
{
 "N_species":"2",
 "Species":[
 {
  "Name":"Species1",
  "Symb":"Sp1",
  "N_coeff":"5",
  "An":[0.12356, -12583.45, 1240854.1234,
        124095.2345, -5679.341]
 },
 {
  "Name":"Species2",
  "Symb":"Sp2",
  "N_coeff":"5",
  "An":[125067.01254, 340653.56, 395403.6745,
        34565.9832, 435234.56]
 }
 ]
}}
```

A.3 CFD Link File Examples

In this section we provide an example of files for linking a CFD simulation to the mesoscopic one with dummy values.

We have these streamlines in the time interval $\tau\,[1.5, 2.3]$

- Stream 1:
  - $\tau_{samples} = \{1.5, 1.6, 1.7, 1.8, 1.9\}$.
  - $T_{samples} = \{540.8, 550.0, 560.8, 580.6, 600.5\}$.
  - $P_{samples} = \{2.2, 2.5, 2.7, 2.3, 2.4\}$.
  - Species: Si ($M_c = 3.2$), Al($M_c = 5.4$), H($M_c = 2.5$), N($M_c = 8.6$).
- Stream 2:
  - $\tau_{samples} = \{1.5, 1.6, 1.7, 1.8, 1.9\}$.
  - $T_{samples} = \{1200.0, 1250.0, 1270.8, 1234.6, 1278.5\}$.
  - $P_{samples} = \{1.2, 1.5, 1.7, 1.3, 1.4\}$.
  - Species: Si ($M_c = 1.2$), Al($M_c = 3.4$), H($M_c = 5.5$), N($M_c = 7.6$).
- Stream 3:
  - $\tau_{samples} = \{1.5, 1.6, 1.7, 1.8, 1.9\}$.
  - $T_{samples} = \{700.0, 723.0, 780.8, 789.6, 800.5\}$.
  - $P_{samples} = \{5.2, 5.5, 5.7, 5.3, 5.4\}$.
  - Species: Si ($M_c = 4.5$), Al($M_c = 7.4$), H($M_c = 1.4$), N($M_c = 4.7$).

Listing 12: XML CFD Linking File Example

```
<GP>
 <N_STREAM> 3 </N_STREAM>
 <T_START> 1.5 </T_START>
 <T_END> 2.3 </T_END>
 <STREAM>
  <ID> 1 </ID>
  <N_TIME_SAMPLES> 5 </N_TIME_SAMPLES>
  <TIME_SAMPLES>1.5 1.6 1.7 1.8 1.9</TIME_SAMPLES>
  <T> 540.8 550.0 560.8 580.6 600.5 </T>
  <P> 2.2 2.5 2.7 2.3 2.4 </P>
  <N_SPIECES> Ns </N_SPIECES>
  <SPECIES> Si Al H N Values </SPECIES>
  <MOLAR_C> 3.2 5.4 2.5 8.6 </MOLAR_C>
 </STREAM>
 <STREAM>
  <ID> 2 </ID>
  <N_TIME_SAMPLES> 5 </N_TIME_SAMPLES>
  <TIME_SAMPLES>1.5 1.6 1.7 1.8 1.9</TIME_SAMPLES>
  <T> 1200.0 1250.0 1270.8 1234.6 1278.5 </T>
  <P> 1.2 1.5 1.7 1.3 1.4 </P>
  <N_SPIECES> 4 </N_SPIECES>
  <SPECIES> Si Al H N Values </SPECIES>
  <MOLAR_C> 1.2 3.4 5.5 7.6 </MOLAR_C>
 </STREAM>
 <STREAM>
  <ID> 3 </ID>
  <N_TIME_SAMPLES> 5 </N_TIME_SAMPLES>
  <TIME_SAMPLES>1.5 1.6 1.7 1.8 1.9</TIME_SAMPLES>
  <T> 700.0 723.0 780.8 789.6 800.5 </T>
  <P> 5.2 5.5 5.7 5.3 5.4 </P>
  <N_SPIECES> 4 </N_SPIECES>
  <SPECIES> Si Al H N Values </SPECIES>
  <MOLAR_C> 4.5 7.4 1.4 4.7 </MOLAR_C>
 </STREAM>
</GP>
```

Listing 13: JSON CFD Linking File Example

```
{"Gf":
 {
  "N_stream":"3",
  "T_start":"1.5",
  "T_end":"2.3",
  "Streams":
   [
    {
     "Id":"1",
     "N_time_samples":"5",
     "Time_samples":[1.5, 1.6, 1.7, 1.8, 1.9],
     "T":[540.8, 550.0, 560.8, 580.6, 600.5],
     "P":[2.2, 2.5, 2.7, 2.3, 2.4],
     "N_species":"4",
     "Species":[Si, Al, H, N],
     "Molar_c":[3.2, 5.4, 2.5, 8.6]
    },
                {
     "Id":"2",
     "N_time_samples":"5",
     "Time_samples":[1.5, 1.6, 1.7, 1.8, 1.9],
     "T":[1200.0, 1250.0, 1270.8, 1234.6, 1278.5],
     "P":[1.2, 1.5, 1.7, 1.3, 1.4],
     "N_species":"4",
     "Species":[Si, Al, H, N],
     "Molar_c":[1.2, 3.4, 5.5, 7.6]
    },
                {
     "Id":"3",
     "N_time_samples":"5",
     "Time_samples":[1.5, 1.6, 1.7, 1.8, 1.9],
     "T":[700.0, 723.0, 780.8, 789.6, 800.5],
     "P":[5.2, 5.5, 5.7, 5.3, 5.4],
     "N_species":"4",
     "Species":[Si, Al, H, N],
     "Molar_c":[4.5, 7.4, 1.4, 4.7]
    }
   ]
 }}
```

References

[1] IEEE. Ieee standard for information technology–systems design–software design descriptions - redline. *IEEE Std 1016-2009 (Revision of IEEE Std 1016-1998) - Redline*, pages 1–58, July 2009.