Definition of a Format for Interchange of Thermodynamic Model Parameters

Bo Sundman

Division of Physical Metallurgy

Royal Institute of Technology

S 100 44 Stockholm, Sweden

January 1990

This report was written as a part of an EG project for developing a thermodynamic database. It defines a format for interchange of thermodynamic model parameters between different software.

Revision history

Date	Revision	Notes
80.09.12	Draft	
81.03.16	Revision	
81.05.10	Revision	
81.11.15	Version 1.0	
82.03.31	Version 1.1	
90.01.08	Version 2.0	Completely revised and without subroutine interface.
90.12.17	Version 2.1	Some additions.
99.08.09	Version 2.2	Small clearifications and section 5 and Appendix III added.

Any quearies, comments or suggestions to this document is welcome and end this by fax to +46 8 100 411 or E-mail to bosse@met.kth.se.

Contents

1 General

A thermodynamic database, in particular the the solution database of SGTE, contains thermodynamic data for elements and phases which make it possible to calculate, on a computer, equilibria of thermodynamic systems. Such a database can be incorporated in various software systems and in order to make it easy to interchange data between different software systems this paper defines a format with a small number of keywords that can be used to describe the content of the database.

A computer program, specific to each software system, can then read (upload) the database from this format and store the data in the internal format used by the software system. There should also be a computer program to write (download) the content of a database from the internal format of the software system into this format in order to transfer it to another system.

The format can easily handle future extensions of the types of data stored in the databank, for example thermophysical data. For more information on this see section 3.15 and 3.16.

1.1 Text Oriented Format

The description of the thermodynamic data for a system is interchanged as text with a given syntax which is (almost) readable by a person. When the interchange format is interpreted by a computer program (upload) all lower case letters are treated as upper case, thus ALPHA and Alpha are the same name.

The text should be divided into lines no longer than 80 characters, the end-of-line may be an LF (line-feed) character or any combination of LF and CR (carriage-return). A single CR which is not preceded or followed by LF should be ignored. Any TAB (tabulation) character or FF (form-feed) within a line can be replaced by a single space. The only control characters used by the interchange format are LF, CR, TAB and FF. Other control characters may be ignored by the upload program. The division into lines is only for ease of manipulation. The end-of-line has no significance in the interchange format as described below.

1.2 Nomenclature

The description of the format used in this paper is simple and possibly not unambiguous. A string enclosed within brackets describes the type of information that shall appear there but other words or symbols used in the description should be used literally. An exception to this is necessary in order to denote that a piece of information may be repeated. Such information is enclosed by slashes and normally the number of repetitions will precede the slash. In this case neither brackets nor slashes appear in the actual data string.

Names, identifiers and keywords defined in this paper may consist of the letters A-Z, the numbers 0-9 and the underscore character "_" in any order but must start with a letter. In a few cases, explicitly described below, names may contain other characters or have a special form. Usually a limited number of the leftmost characters in a name or identifier are significant but the actual length may be longer. Notice that lower case letters are legal but not distinguished from upper case.

Numeric values that should appear in certain places according to this format but which have no value should be indicated by the value NONE. Note also the use of UNASSESSED for a missing parameter in section 3.5.

2 The Model Parameters

The data stored in a thermodynamic database describe the Gibbs energy of each individual phase as a function of the temperature, pressure and the constitution of the phase. The reason for selecting the Gibbs energy rather than any other quantity is discussed in ref database one must store data for elements, species, phases with their constitution and model parameters for each phase.

2.1 Phase or Composition Oriented Data Storage

In thermodynamic databanks for substances it is common to store data according to composition. This means that the data for iron with bcc structure (ferrite), with fcc structure (austenite) and liquid iron are all stored as one dataset for pure iron. For the given composition a single expression is given for temperatures from 298.15 K and up. This method is not suitable for solution databases for two reasons.

The first reason is that for solutions it is necessary to extrapolate the thermodynamic data outside the stable range of the constituents. (In substance databases this is made for gaseous constituents.) Thus there must be thermodynamic data for austenite for the whole temperature range. This also means that it is impossible to know the stable phase for pure iron at a given temperature without making a calculation in order to compare the Gibbs energy of the various possible phases.

The second reason is that interaction parameters are needed to describe the non-ideality of the solution phases. Such parameters are associated with two or more constituents but only one phase. Thus duplication and ambiguity is avoided if one stores these referred primarily to the phase.

In solution databases one must in addition decide on a model to describe the interaction between the constituents of a phase. This is discussed in detail in ref each phase can have a separate model. However, a specific phase must be described with the same model independent of its constituents because otherwise it would not be possible to combine components in an arbitrary way. This makes the decision on the model to use for each phase of crucial importance in a solution database.

2.2 Composition dependence

In all solution models of the Gibbs energy the composition dependence is expressed explicitly as a series expansion in the fractions of the constituents of the phase. A few models also have implicit composition dependence because some additional property of the phase is also composition dependent. One example of this is the Inden model for the magnetic contribution to the Gibbs energy due to ferromagnetic ordering. In this case the composition dependence

of the critical temperature for magnetic ordering, and the Bohr magneton number, must be described separately and the value of this critical temperature and the Bohr magneton number for a given composition are needed in order to calculate the magnetic contribution to the Gibbs energy.

The interchange format described here can express the composition dependence of any number of composition dependent quantities, in addition to the Gibbs energy. For each phase the same series expansion in the fraction of the constituents of the phase is used for all quantities. The constituents of a phase can be the components of the phase, molecules in a gas phase or constituents on different sublattice sites in a crystalline phase. The basic expression for this series expansion is

$$Z_m = \sum_i x_i {}^0 Z_i + {}^E Z_m \tag{1}$$

where Z is the quantity which is composition dependent, x_i is the mole fraction of constituent i, ${}^o Z_i$ is the value of this quantity for the pure constituent i (this value may be a function of temperature and pressure). The summation in eq. 1 represents a linear combination of the values for the constituents. ${}^E Z_m$ is the excess part which depends upon the interaction between the constituents. The subscript m means that the data are for one mole of formula unit of the phase.

Note that eq. 1 does not contain a term for the ideal entropy of mixing, $RT \sum_i x_i \ln(x_i)$, because such a term is only present in the Gibbs energy expression. The ideal entropy does not contain any adjustable parameters and it is not included in the interchange format.

If a phase has sublattices there will be one additional summation in the first term of eq. 1 for each sublattice. On the sublattices one may have the same constituents (in order to describe chemical ordering) or different (for example interstitial solutions or intermetallic compounds).

$$Z_m = \sum_i y_i^{(1)} \sum_j y_j^{(2)} ... \sum_s y_n^{(s)} {}^0 Z_{ij...n} + {}^E Z_m \tag{2}$$

where $y_n^{(s)}$ is the site fraction of constituent n on sublattice s. For the sublattice model the entropy expression is given by

$$\sum_s a^{(s)} \sum_i y_s^{(s)} \ln(y_s^{(s)}) \tag{3}$$

where $a^{(s)}$ is the relative number of sites on sublattice s. The values of $a^{(s)}$ are given in the definition of the phase.

The excess part of Z_m must be zero whenever the mole fraction of any constituent is unity. The simplest expression with this property is the regular solution model

$$^E Z_m = \sum_i \sum_{j>i} x_i x_j Z_{ij} \tag{4}$$

The value of Z_{ij} is a binary parameter for the system i-j. One may make Z_{ij} temperature, pressure and composition dependent as defined by the model index. If there are sublattices it should be noted that i and j must belong to the same sublattice and there will be one

additional summation in eq. 4 for each sublattice. The composition dependence of Z_{ij} is often expressed by a Redlich-Kister polynomial:

$$Z_{ij} = \sum_{\nu} (x_i - x_j)^{\nu} Z_{ij}^{\nu} \tag{5}$$

One additional subscript, ν, is needed to specify the place of the parameter Z_{ij}^{ν} in the Redlich-Kister expression. This subscript is called the degree of the parameter. For other models the degree can mean other things. There is no additional complications if one has sublattices also but note that one may have composition dependent excess terms on each sublattice separately.

One may have ternary parameters with three subscripts for constituents on the same sublattice and in some cases even higher order excess terms. However, from eqs. 1 to 5 it should be evident that the interchange format need only contain the parameters $Z_{ij...}$, where the three dots mean the degree or additional constituent indices, for each phase together with the model index and the structure of the phase in order to reconstruct the full expression for Z_m. The place of a parameter in this expression is determined by the phase, the constituents and the degree of the parameter given as subscripts.

2.3   Temperature and pressure dependence

Most parameters $Z_{ij...}$ in eqs. 1 to 5 are functions of temperature and pressure and this function is given explicitly in the interchange format.

3   Syntax of the data interchange format

The interchange format consists of sequences of the following form:

`<keyword> <information> END.`

The keyword must be the first non-blank part of a line. It may be preceded by spaces and tab characters only. The period after END is a part of the sequence and must be present in order to distinguish this terminating END from other occurrences of the letters END as a part of a longer word.

A keyword must be unique with its first 6 characters and its first character must be alphabetic. Notice that the keywords must be followed by at least one space and the terminator "END." must also be preceded by a space. The keywords currently defined in this interchange format will be described below.

The order of the keywords is free, except for some cases specified below, and they must not be nested, i.e. after a keyword there must not be any other keywords until after the terminating "END.".

Note that only one keyword can appear on each line!

3.1 Element

Elements are those included in the periodic chart. They are referenced by their chemical symbols, i.e. their names are one or two letters long. It is possible to have hypothetical element names e.g. A or Z. A few extensions to the periodic chart are necessary. Thus vacant sites on a sublattice are treated as a component and the name VA has been adopted. Electrons are denoted by the combination /- and a positive charge (electron hole) can be denoted by /+ or /- -.

Upper and lower case letters are not distinguished and thus CO, Co and co can all be used to denote cobalt. Note that this does not mean that one cannot use lower case for the second letter in a particular implementation of the database. The upload program can change this when it uploads data from the interchange format.

All elements have a defined state at 298.15 K which will be called Stable Element Reference state (SER). The name of this state (e.g. FCC or H2_GAS etc.), the atomic mass, the entropy at 298.15 K and the difference in enthalpy between 298.15 and 0 K for the element in that same state will be given at the element keyword.

```
ELEMENT <element name> <SER state> <atomic mass>
    <enthalpy at 298.15 K - enthalpy at 0 K>
    <entropy  at 298.15 K> END.
```

Example:

```
element fe bcc_a2 55.847 4489 27.28 end.
element am double_hcp 243.06 NONE NONE end.
```

3.2 Species

Species are aggregates of elements with fixed stoichiometry. The elements are the simplest types of species and they do not have to be given as species also. The name of a species is usually its chemical formula but must not exceed 24 characters. The stoichiometric factor 1 can be excluded if there is no possibility of misinterpreting, e.g. CO1 or CO is cobalt while carbonmonoxide must be written C1O or C1O1.

Note that in chemistry it is common to use a stoichiometric formula to denote both the phase and the species. This is natural when one deals mainly with phases with fixed composition. However, it is not suitable when dealing with solutions. As a species can be a constituent of many phases it is convenient to define the stoichiometry of the species separate from the phase. A species which is present in a particular phase is called a constituent of the phase.

```
SPECIES <species name> /<element name> <stoichiometric factor>/ END.
```

Examples

```
species h2o h2o1 end.
species c1o c1o1 end.
species mg2sio4 mg1o2si0.5 end.       This is allowed but not recommended
species s1o2 s1o2 end.
species s/-2 s1/-1 end.
species fe/+2 fe1/--2 end.
species s1o4/-2 s1o4/-2 end.
species c2cl2h4_cis c2cl2h4 end. These two species -
species c2cl2h4_trans c2cl2h4 end.          are isomers
species this_is_a_very_long_name fe1 end.    Allowed but not recommeneded.
species feo fe0.987O end,
```

Note that the elements do not have to be in alphabetical order. The stoichiometric factors can be non-integer numbers. The electric charge is treated as an element here.

3.3   Phase

Phases are homogeneous parts of the thermodynamic system with uniform structure and composition. In some cases one may use the word phase also to denote parts which have uniform structure but non-uniform composition although such a system cannot be in equilibrium. For solutions the phase is the central concept as data are stored primarily according to the phase it is associated with.

```
PHASE <phase name> <type code> <number of sublattices>
    /<number of sites in the sublattice>/ END.
```

Examples

```
phase fcc_a1 zm 1 1.0 end. Substitutional solution
phase fcc_l12 zm 4 0.25 0.25 0.25 0.25 end. Sublattices for long range order
phase fcc_int zm 2 1.0 1.0 end. Sublattices for interstitials
phase wustite z 2 1.0 1.0 end.
phase m23c6 z 3 2.0 21.0 6.0 end. Two types of sites for metallic atoms
```

Note that the model used for a phase (except the sublattices) is defined using the type code. A type code is a single character. A phase may have several type codes and they must be written together as one word, note that zm in the examples means both type codes z and m.

The constituents of a phase are a subset of the species. This subset is given either by a "constituent" keyword or must be deduced from the PARAMETER keywords.

3.4   Constituents

Constituents is an optional keyword to give the constituents of a phase. The syntax is

```
CONSTITUENTS <phase name> : <species on first sublattice>
/ : <species on other sublattices in sublattice order>/ : end.
```

The number of sublattices must correspond to the number given at the phase keyword. Note that colon ":" are used between the sublattices and comma "," or a space between constituents in the same sublattice

Examples

```
constituent gas :c1o2 c1o h2o h2 o2 c c2: end.
constituent fcc :fe al si: end.
constituent wustite :fe/+2 fe/+3 va: o/-2 : end.
```

The reason this keyword is optional is that the information about the constituents of a phase can be deduced from the parameters (see the PARAMETER keyword). In practice it has turned out to be useful to collect such information using this keyword. However, the upload program should not depend on the use of a constituent keyword and it can be ignored.

In a general thermodynamic database one will find that most phases will dissolve most elements. Thus when retrieving data from the database for a system a large number of phases may have to be included in order to calculate those which are stable. This is due to the fact that a phase has no indication for which constituents it is stable and it is indeed impossible to find that out without a calculation. In the Cu-Zn system for example neither Cu nor Zn exist as pure in the bcc phase but nevertheless there are four bcc related phases stable in the middle of this system.

In order to improve this situation somewhat the CONSTITUENT keyword can be used in order to specify MAJOR constituents on each sublattice of a phase. The major constituents should have a suffix consisting of a per-cent sign "

```
CONSTITUENTS FCC_A1 :FE% CR NI% MN% S: VA% C: END.
```

Where Fe, Cr and Mn are major in the first sublattice and Va in the second (interstitial solution).

3.5 Parameters

All temperature and pressure dependent thermodynamic data is given using the PARAMETER keyword. With this keyword a phase and a composition or composition range must be specified followed by an expression. In the simplest case the parameter gives the Gibbs energy of formation of a phase with a single constituent. In more sophisticated models the parameter may be part of a complex Gibbs energy expression. Note that the Gibbs energy expression of a solution phase may have many coefficients in its composition dependence. Each of these coefficients will appear in a PARAMETER keyword, giving the temperature and pressure dependence of one coefficient. See section 2.2 for more information. The syntax is

```
PARAMETER <identifier> ( <phase name>, <component array> ; <degree> )
    <expression> <reference identifier> END.
```

The first part of the information after the keyword PARAMETER but before the equality sign is the "name" of the parameter. The <identifier> after the PARAMETER denote the type of composition dependent quantity and it must be standardized. The following identifiers are used currently:

Symbol	Unit	Notes
G	J/mol formula unit	Parameters that are part of the explicit composition dependence of the Gibbs energy
TC	Kelvin	Curie temperature parameters
BMAGN	dimensionless	Bohr magneton parameters

The syntax of the <component array> in the PARAMETER information is composed of names of species separated by comma "," for species in the same sublattice and using colon ":" to separate groups of species that go into different sublattices. A recursive definition is:

```
<component array> is
            <species name> or
            <species name>,<component array> when there are more species in
                 the same sublattice,
            <species name>:<component array> when there are more sublattices.
```

There must not be any spaces in a component array. Note that the parameter in the Gibbs energy expression should be multiplied with the fraction of the constituents given by the component array.

The <degree> in a parameter can be void or a value from 0 (zero) to 9. If the parameter is a binary interaction parameter in a Redlich-Kister excess model it means the degree as a coefficient in a Redlich-Kister polynomial. For other parameters and models the degree has different meanings.

The expression will contain coefficients which should be multiplied with temperature and pressure. The actual form of the expression is specified by a EXPTYPE keyword which is described in the next section.

Note that it may be important in some cases to distinguish between parameters that are zero or have not been assessed. A parameter that is zero could of course be omitted completely but it is recommended that unary and binary parameters should be included even if they are zero. The reason for this is that all parameters possible according to the model that are not included in the interchange text by default must be considered unassessed.

If one particularlys want to stress that a parameter is not assessed the function can have the value UNASSESSED (independent of expression type).

The <reference identifier> is an index to the REFERENCE keyword and should give a reference to the assessment when the parameter was determined. See further the REFERENCE keyword below.

3.6   3.6 Expression type

It is necessary to allow the expression of T and P to be formulated in different ways due to different software requirements. The expression type is a keyword which gives information how the expression part of parameters should be interpreted. Note that the expression type keyword may appear several times in an interchange text and the last value should always be used.

```
EXPTYPE <integer code> END.
```

The value 1 is used for this simple and flexible way to giving a temperature function:

```
<low temperature limit>
   <number of coefficients in the temperature polynomial>
   /<coefficient value> <power of temperature>/
   <high temperature limit>
   <number of coefficients in the next temperature range, (zero if no more)>
   etc.
```

An example of expression type 1 is:

```
298.15 3 2345 0 -3.353 1 1.123 100 2100.00 0
```

Where 298.15 is the low limit and 3 is the number of coefficient-power pairs in the polynomial that will follow. The expression should be interpreted as

```
2345 - 3.353*T + 1.123*T*LN(T)
```

The value 2100.00 is the upper temperature limit and 0 indicates that there are no more ranges. The value 100 is used here to denote the T*LN(T) factor. Notice that all powers must be integral values.

The value 2 is used for the following method of giving the data for a species using Cp data:

```
<enthalpy of formation and the standard entropy at the low temperature limit>
   <low temperature limit>
   <number of coefficients in Cp expression>
   /<coefficient value> <power of temperature>/
   <high temperature limit> <enthalpy of transformation at this temperature>
   <number of coefficients in next temperature range, (zero if no more)>
   etc.
```

An example of expression type 2 is:

```
-3453 3.543 298.15 2 2.674 0 1.566E-3 1 400.00 320.0 1 7.669 0
1200.00 0.0 0
```

Where -3453 is the enthalpy of formation and 3.543 is the standard entropy, both at 298.15 K. In the temperature range up to 400.00 K the Cp-expression is 2.674+0.001566*T. At 400 there is an enthalpy of transformation equal to 320 J. Between 400 and 1200 K Cp is constant and equal to 7.669. Above 1200.00 K the enthalpy is zero and there are no data.

The standard entropy is the entropy of the species in the given phase, at 298.15 K. An alternative method of giving the entropy is EXPTYPE 3:

```
<enthalpy and entropy of formation from the SER states of the elements
    at the low temperature limit> <low temperature limit>
    <number of coefficients in Cp expression>
    /<coefficient value> <power of temperature>/
    <high temperature limit> <enthalpy of transformation at this temperature>
    <number of coefficients in next temperature range, (zero if no more)>
    etc.
```

This expression type has never been used and Exptype 4 is no longer used.

The most important exptype is 5. The syntax of EXPTYPE 5 is

```
<low temperature limit> / <function> ; <high temperature limit> <Y or N> /
```

The syntax of the function part is given below. A semicolon must be use to terminate the function and it is followed by the high temperature limit. Finally a "Y" or an "N" must be given. N means that there are no more temperature ranges and Y that there will be another function above the current high temperature limit. The information between the slashes is repeated until it is terminated by an N.

The function is written similar to a Fortran like statement but without the use of parenthesis for grouping terms. The basic entity of the function is called a "simple term". A simple term is:

```
<real number> * <symbol name> **<power> *T** <power> *P** <power>
```

The text between < and > describe the item which should appear there. The other items must be given literally and have their usual meaning i.e. * is multiplication and ** is exponentiation, T is the temperature and P is the pressure. A symbol can mean a numeric value or another expression, see the FUNCTION keyword below.

If the simple term does not depend on any symbol or T or P that part can be omitted. The power can only be an integer but non-integer powers can be handled with a LOG and EXP pair as described below. Negative powers must be surrounded by parenthesis. Redundant parts of a simple term can be omitted and if the power is unity the exponentiation can be omitted altogether. Note that the real number must be the first item and any symbol must precede the T and P. Examples of simple terms are:

```
1.15*T      -V1     1E-12*P**2    -456754.65*T**(-1)*P        10*R*T
```

In order to include the logarithm and exponential in these functions it is allowed to multiply a simple term with the logarithm or exponential of another simple term. This more generalized entity is called a term and it is defined

```
<term> = <simple term> or
 <simple term> * <symbol> or
 <simple term> * LOG( <simple term> ) or
 <simple term> * EXP( <simple term> )
```

A term can be equal to one of the lines above. It is illegal to have both exponentials and logarithms in the same term.

Examples of terms are:

```
+1.15*T*LOG(T)        +1E-6*LOG(-32000*T**(-1))    -5*V3*T*EXP(V4*P)
```

A function of this type is thus a number of terms written after each other in order to form an expression. All terms except the first must be preceded by a sign. Examples of expressions are:

```
-10000 +10*T+1.15*T*LOG(T) +7.5E-5*P +134567*T**(-1) -1.13E-12*T*P
```

```
F1+2.5*R*T*LOG(T)+R*T*LOG(P)
```

Note that there must be a semicolon to separate the function from the following high temperature limit. Note also that the symbols can denote a numeric value or another function. See the FUNCTION keyword.

Some of the restrictions on a function are:

- The order of the factors in a simple term must be followed,

- No parenthesis allowed except for exponentials, logarithms or negative powers

- No division allowed,

- No spaces allowed between a sign and a numeric value,

- Only one symbol in each simple term,

- Only integers as powers.

Some of these restrictions are due to simplifications in the parsing of the functions and some are due to the requirement that it must be possible to calculate quickly the value of the function.

3.7   Function

This keyword can be used only in connection with EXPTYPE 5. It is useful because many thermodynamic parameters are related. For example the Gibbs energy of formation of metastable phases is often based on the Gibbs energy of the stable phase at 298.15 K.

```
FUNCTION <name> <expression> END.
```

The name of a function can be used as symbol name in EXPTYPE 5 expressions. The name must not be longer than 8 characters and must start with a letter. Note that the keyword defining the function must appear before its name is used in an expression. The expression must be given in the way defined by EXPTYPE 5.

It is illegal for a function to refer to itself in the expression part. Implicit references are not allowed either.

In order to represent sqrt(T), i.e the square root of T, one must use two functions

```
FUNCTION F1 0 0.5*LOG(T) 6000 N; END.
FUNCTION SQRTT 0 EXP(F1); 6000 N END.
```

Then the function SQRTT can be used whereever SQRT(T) is needed.

3.8 Models

This keyword should be elaborated to give more information in the future. At present it must be manually decoded. See ref. information.

```
MODEL <model name> <model code> '<description>' END.
```

The following models are used

```
model redlich-kister_muggianu 1 'binary and ternary excess model' end.
model redlich-kister_kohler   2 'binary and ternary excess model' end.
model redlich-kister_hillert 3  'binary and ternary excess model' end.
model ionic_liquid_sublattice 4 'two-sublattice ionic liquid model' end.
model associate 5 'hypothetical species and Redlich_Kister interactions' end.
model inden-magnetic 10 'magnetic ordering' end.
model kaphor-frohberg-gaye 20 'irsid slag model' end.
model dilute 90 'Dilute entropy of mixing term' end.
model pitzer 100 'aqueous solutions' end.
model debue_huckel 101 'Debue-Huckel aqueous term' end.
```

3.9 Type Code

The type information after the phase name is necessary for specifying the model for a phase. A type code is a single character. A phase may have several type codes and they must be written together as one word.

```
TYPE_CODE <type code> <value> '<comment>' END.
```

Examples

```
type_code z 10 'Inden magnetic model' end.
```

The type code thus gives the coupling between the phase and the model. But in the future the type code may also be used for other phase specific information.

3.10   Messages

The keyword MESSAGE is useful for giving information that cannot be incorporated in the other keywords.

```
MESSAGE <integer code> '<text>' END.
```

The integer code should eventually be standardized to identify the message when there has been some experience with the format. The actual message must be enclosed with quotes. Double quotes can be used instead of single quotes.

3.11   Reference

As all parameters in the interchange text must originate from some assessment it is important to give a reference to the paper or report where the parameter is determined. The syntax of this keyword is

```
REFERENCE /<reference identifier> '<text>'/ END.
```

The <reference identifier> is the same as used in the PARAMETER keyword. The text should give the publication or where to find the report. The text must be enclosed within single or double quotes because more than one reference identifier can be defined in one keyword.

3.12   Redefine

This keyword can be used to redefine any keyword or other part of this description in a controlled way. Excessive use of this keyword may make it very difficult to interpret an interchange text but it is included to allow small modifications for convenience.

```
REDEFINE <item> <value> <new value> END.
```

Examples

```
redefine terminator end. ! end.
redefine keyword EXPTYPE expession_type !
```

The first redefine changes the terminator for the keywords from " END." to an exclamation mark "!". This means that exclamation marks cannot be used within the information part of a keyword, not even in texts. But the exclamation mark can be used within comments between the keywords as described in the next section.

3.13   Comments

Before the first keyword or after an "END." and up to the next keyword, or the end of the text, any characters can appear, for example comments which can be useful for a person trying to understand how to deal with this.

3.14 End of transfer

The interchange text should finish with a keyword

```
END_OF_TRANSFER <checksum> END.
```

The checksum should be calculated in the following way:

All printable characters (excluding also space) from the beginning of the text should be added together using the ASCII values of the characters. The value should be given after the keyword. Upon reading the text one may thus check that the same checksum is obtained. If the values are not the same there might be a corruption in the data. The line with END_OF_TRANSFER should not be included in the checksum.

If the checksum becomes larger than 2000000000 (a 32 bit integer can become 2147483647) then all digits in checksum should be added together to form a new number before summation continues. Thus 2000000012 should become 5.

3.15 Material

In some cases data is characterized for a material which may consist of several phases. In order to accomodate such data, particularly for thermophysical use, one may use the keyword MATERIAL. The data for this keyword is

```
MATERIAL <name> <composition> '<comments>' END.
```

The name is 1 to 24 alphanumerical characters or underscore, it must start with a letter. The composition is specified in mass percent of the components, normally the elements but species may be used. The chemical symbol of the component is followed by the mass percent, the major component is specified by giving the value as *. The comments may be any text.

Examples

```
MATERIAL PIG_IRON C 4 SI 2.5 MN 0.4 FE * END.
MATERIAL LIMESTONE CACO3 3 CAC1O3 * 'Almost pure' END.
```

3.16 New_Identifier

Only three identifiers for use in parameters are defined in this text. In order to allow more free addition of such identifiers this keyword has been added. The syntax is

```
NEW_IDENTIFIER <identifier> <unit> '<meaning>' END.
```

The identifier must be 8 characters or less and consist of letters only. The units must be specified in SI. If it is dimensionless that must be given as "1". The meaning should be as explicit as possible so no missunderstanding is possible. Example

```
NEW_IDENTIFIER DC m**2/s 'Diffusion coefficient' END.
```

4   Order of appearance of the keywords

The recommended order of the keywords are

- all MODEL keywords

- all ELEMENT keywords

- all SPECIES keywords

- all PHASE keywords

- all CONSTITUENT keywords

- all TYPE_CODE keywords

- all FUNCTION keywords

- all PARAMETER keywords

- all REFERENCE keywords

However, the actual order may be much more free but the following rules must be obeyed

- All elements used in a species keyword must have been defined by ELEMENT keywords.

- All constituents of a phase used in CONSTITUENT or PARAMETER keywords must have been defined by the SPECIES keyword.

- All function names used in PARAMETER or FUNCTION keywords must have been defined by the FUNCTION keyword. As a function may refer to other functions this means that there must be an order even within the FUNCTION keywords.

Notice that all parameters for a phase may not come together. Except when using the CONSTITUENT keyword it is not possible to identify all constituents of a phase until all data have been fetched. The constituents of the various phases are defined by the component array after the PARAMETER keyword.

A number of error conditions may also occur when reading the interchange format. The same species name must not be used twice (but note that several species may have the same stoichiometry). The same phase name must not be used twice.

The keywords EXPTYPE and MESSAGE may appear anywhere.

5 Example of an interchange

On the following pages an example of the use of the interchange format is given in order to describe a database for C-Cr-Fe. This database has been extracted from the SGTE solution database and in the example there are a number of phases that are metastable in this ternary system.

Important note: this example is taken from the LIST_DATA output from Thermo-Calc and does not exactly follow the standard.

```
REDEFINE END. ! END.
MESSAGE 77 'Nonstandard references' !
EXPTYPE 5 !

 ELEMENT /-   ELECTRON_GAS                   0.0000E+00  0.0000E+00  0.0000E+00!
 ELEMENT VA   VACUUM                         0.0000E+00  0.0000E+00  0.0000E+00!
 ELEMENT C    GRAPHITE                       1.2011E+01  1.0540E+03  5.7400E+00!
 ELEMENT CR   BCC_A2                         5.1996E+01  4.0500E+03  2.3560E+01!
 ELEMENT FE   BCC_A2                         5.5847E+01  4.4890E+03  2.7280E+01!

 SPECIES C1                            C!
 SPECIES C2                            C2!
 SPECIES C3                            C3!
 SPECIES C4                            C4!
 SPECIES C5                            C5!
 SPECIES C6                            C6!
 SPECIES C7                            C7!

 FUNCTION GHSERCC    2.98150E+02  -17368.441+170.73*T-24.3*T*LN(T)
     -4.723E-04*T**2+2562600*T**(-1)-2.643E+08*T**(-2)+1.2E+10*T**(-3);
     6.00000E+03   N !
 FUNCTION GPCLIQ     2.98150E+02  +YCLIQ#*EXP(ZCLIQ#);   6.00000E+03   N !
 FUNCTION GHSERCR    2.98150E+02  -8856.94+157.48*T-26.908*T*LN(T)
     +.00189435*T**2-1.47721E-06*T**3+139250*T**(-1);  2.18000E+03   Y
      -34869.344+344.18*T-50*T*LN(T)-2.88526E+32*T**(-9);  6.00000E+03   N !
 FUNCTION GPCRLIQ    2.98150E+02  +YCRLIQ#*EXP(ZCRLIQ#);   6.00000E+03   N !
 FUNCTION GFELIQ     2.98150E+02  +12040.17-6.55843*T-3.6751551E-21*T**7
     +GHSERFE#;  1.81100E+03   Y
      -10839.7+291.302*T-46*T*LN(T);  6.00000E+03   N !
 FUNCTION GPFELIQ    2.98150E+02  +YFELIQ#*EXP(ZFELIQ#);   6.00000E+03   N !
 FUNCTION GPCRBCC    2.98150E+02  +YCRBCC#*EXP(ZCRBCC#);   6.00000E+03   N !
 FUNCTION GPCGRA     2.98150E+02  +YCGRA#*EXP(ZCGRA#);   6.00000E+03   N !
 FUNCTION GHSERFE    2.98150E+02  +1225.7+124.134*T-23.5143*T*LN(T)
     -.00439752*T**2-5.8927E-08*T**3+77359*T**(-1);  1.81100E+03   Y
      -25383.581+299.31255*T-46*T*LN(T)+2.29603E+31*T**(-9);  6.00000E+03   N !
 FUNCTION GPFEBCC    2.98150E+02  +YFEBCC#*EXP(ZFEBCC#);   6.00000E+03   N !
```

```
FUNCTION GFECEM      2.98150E+02  -10745+706.04*T-120.6*T*LN(T)+GPCEM1#;
    6.00000E+03   N !
FUNCTION GFEFCC      2.98150E+02  -1462.4+8.282*T-1.15*T*LN(T)+6.4E-04*T**2
    +GHSERFE#;  1.81100E+03   Y
     -27098.266+300.25256*T-46*T*LN(T)+2.78854E+31*T**(-9);  6.00000E+03  N !
FUNCTION GPCFCC      2.98150E+02  +YCFCC#*EXP(ZFEFCC#);   6.00000E+03   N !
FUNCTION GCRFCC      2.98150E+02  +7284+.163*T+GHSERCR#;   6.00000E+03   N !
FUNCTION GPFEFCC     2.98150E+02  +YFEFCC#*EXP(ZFEFCC#);   6.00000E+03   N !
FUNCTION GCRM23C6    2.98150E+02  -521983+3622.24*T-620.965*T*LN(T)
    -.126431*T**2;  6.00000E+03   N !
FUNCTION GFEM23C6    2.98150E+02  +7.666667*GFECEM#-1.666667*GHSERCC#+66920
    -40*T;  6.00000E+03   N !
FUNCTION GCRM7C3     2.98150E+02  -201690+1103.128*T-190.177*T*LN(T)
    -.0578207*T**2;  6.00000E+03   N !
FUNCTION GPSIG1      2.98150E+02  +1.09E-04*P;  6.00000E+03   N !
FUNCTION GPSIG2      2.98150E+02  +1.117E-04*P;  6.00000E+03   N !
FUNCTION YCLIQ       2.98150E+02  +VCLIQ#*EXP(-ECLIQ#);  6.00000E+03   N !
FUNCTION ZCLIQ       2.98150E+02  +1*LN(XCLIQ#);  6.00000E+03   N !
FUNCTION YCRLIQ      2.98150E+02  +VCRLIQ#*EXP(-ECRLIQ#);   6.00000E+03   N !
FUNCTION ZCRLIQ      2.98150E+02  +1*LN(XCRLIQ#);  6.00000E+03   N !
FUNCTION YFELIQ      2.98150E+02  +VFELIQ#*EXP(-EFELIQ#);   6.00000E+03   N !
FUNCTION ZFELIQ      2.98150E+02  +1*LN(XFELIQ#);  6.00000E+03   N !
FUNCTION YCRBCC      2.98150E+02  +VCRBCC#*EXP(-ECRBCC#);   6.00000E+03   N !
FUNCTION ZCRBCC      2.98150E+02  +1*LN(XCRBCC#);  6.00000E+03   N !
FUNCTION YCGRA       2.98150E+02  +VCGRA#*EXP(-ECGRA#);   6.00000E+03   N !
FUNCTION ZCGRA       2.98150E+02  +1*LN(XCGRA#);  6.00000E+03   N !
FUNCTION YFEBCC      2.98150E+02  +VFEBCC#*EXP(-EFEBCC#);   6.00000E+03   N !
FUNCTION ZFEBCC      2.98150E+02  +1*LN(XFEBCC#);  6.00000E+03   N !
FUNCTION GPCEM1      2.98150E+02  +VCEM1#*P;  6.00000E+03   N !
FUNCTION YCFCC       2.98150E+02  +VCFCC#*EXP(-EFEFCC#);   6.00000E+03   N !
FUNCTION ZFEFCC      2.98150E+02  +1*LN(XFEFCC#);  6.00000E+03   N !
FUNCTION YFEFCC      2.98150E+02  +VFEFCC#*EXP(-EFEFCC#);   6.00000E+03   N !
FUNCTION VCLIQ       2.98150E+02  +7.626E-06*EXP(ACLIQ#);  6.00000E+03   N !
FUNCTION ECLIQ       2.98150E+02  +1*LN(CCLIQ#);  6.00000E+03   N !
FUNCTION XCLIQ       2.98150E+02  +1*EXP(.5*DCLIQ#)-1;  6.00000E+03   N !
FUNCTION VCRLIQ      2.98150E+02  +7.653E-06*EXP(ACRLIQ#);   6.00000E+03   N!
FUNCTION ECRLIQ      2.98150E+02  +1*LN(CCRLIQ#);  6.00000E+03   N !
FUNCTION XCRLIQ      2.98150E+02  +1*EXP(.8*DCRLIQ#)-1;  6.00000E+03   N !
FUNCTION VFELIQ      2.98150E+02  +6.46677E-06*EXP(AFELIQ#);   6.00000E+03 N !
FUNCTION EFELIQ      2.98150E+02  +1*LN(CFELIQ#);  6.00000E+03   N !
FUNCTION XFELIQ      2.98150E+02  +1*EXP(.8484467*DFELIQ#)-1;   6.00000E+03 N !
FUNCTION VCRBCC      2.98150E+02  +7.188E-06*EXP(ACRBCC#);   6.00000E+03   N!
FUNCTION ECRBCC      2.98150E+02  +1*LN(CCRBCC#);  6.00000E+03   N !
FUNCTION XCRBCC      2.98150E+02  +1*EXP(.8*DCRBCC#)-1;  6.00000E+03   N !
```

```
FUNCTION VCGRA      2.98150E+02  +5.259E-06*EXP(ACGRA#);   6.00000E+03    N !
FUNCTION ECGRA      2.98150E+02  +1*LN(CCGRA#);   6.00000E+03    N !
FUNCTION XCGRA      2.98150E+02  +1*EXP(.9166667*DCGRA#)-1;   6.00000E+03 N!
FUNCTION VFEBCC     2.98150E+02  +7.042095E-06*EXP(AFEBCC#);   6.00000E+03 N!
FUNCTION EFEBCC     2.98150E+02  +1*LN(CFEBCC#);   6.00000E+03    N !
FUNCTION XFEBCC     2.98150E+02  +1*EXP(.7874195*DFEBCC#)-1;   6.00000E+03 N!
FUNCTION VCEM1      2.98150E+02  +2.339E-05*EXP(ACEM1#);   6.00000E+03    N !
FUNCTION VCFCC      2.98150E+02  +1.031E-05*EXP(ACFCC#);   6.00000E+03    N !
FUNCTION EFEFCC     2.98150E+02  +1*LN(CFEFCC#);   6.00000E+03    N !
FUNCTION XFEFCC     2.98150E+02  +1*EXP(.8064454*DFEFCC#)-1;   6.00000E+03 N!
FUNCTION VFEFCC     2.98150E+02  +6.688726E-06*EXP(AFEFCC#);   6.00000E+03 N!
FUNCTION ACLIQ      2.98150E+02  +2.32E-05*T+2.85E-09*T**2;   6.00000E+03  N!
FUNCTION CCLIQ      2.98150E+02  1.6E-10;   6.00000E+03    N !
FUNCTION DCLIQ      2.98150E+02  +1*LN(BCLIQ#);   6.00000E+03    N !
FUNCTION ACRLIQ     2.98150E+02  +1.7E-05*T+9.2E-09*T**2;   6.00000E+03    N !
FUNCTION CCRLIQ     2.98150E+02  3.72E-11;   6.00000E+03    N !
FUNCTION DCRLIQ     2.98150E+02  +1*LN(BCRLIQ#);   6.00000E+03    N !
FUNCTION AFELIQ     2.98150E+02  +1.135E-04*T;   6.00000E+03    N !
FUNCTION CFELIQ     2.98150E+02  +4.22534787E-12+2.71569924E-14*T; 6000   N !
FUNCTION DFELIQ     2.98150E+02  +1*LN(BFELIQ#);   6.00000E+03    N !
FUNCTION ACRBCC     2.98150E+02  +1.7E-05*T+9.2E-09*T**2;   6.00000E+03    N !
FUNCTION CCRBCC     2.98150E+02  2.08E-11;   6.00000E+03    N !
FUNCTION DCRBCC     2.98150E+02  +1*LN(BCRBCC#);   6.00000E+03    N !
FUNCTION ACGRA      2.98150E+02  +2.32E-05*T+2.85E-09*T**2;   6.00000E+03 N!
FUNCTION CCGRA      2.98150E+02  3.3E-10;   6.00000E+03    N !
FUNCTION DCGRA      2.98150E+02  +1*LN(BCGRA#);   6.00000E+03    N !
FUNCTION AFEBCC     2.98150E+02  +2.3987E-05*T+1.2845E-08*T**2; 6.00000E+03 N!
FUNCTION CFEBCC     2.98150E+02  +2.20949565E-11+2.41329523E-16*T; 6000 N!
FUNCTION DFEBCC     2.98150E+02  +1*LN(BFEBCC#);   6.00000E+03    N !
FUNCTION ACEM1      2.98150E+02  -1.36E-05*T+4E-08*T**2;   6.00000E+03    N !
FUNCTION ACFCC      2.98150E+02  +1.44E-04*T;   6.00000E+03    N !
FUNCTION CFEFCC     2.98150E+02  +2.62285341E-11+2.71455808E-16*T; 6000   N !
FUNCTION DFEFCC     2.98150E+02  +1*LN(BFEFCC#);   6.00000E+03    N !
FUNCTION AFEFCC     2.98150E+02  +7.3097E-05*T;   6.00000E+03    N !
FUNCTION BCLIQ      2.98150E+02  +1+3.2E-10*P;   6.00000E+03    N !
FUNCTION BCRLIQ     2.98150E+02  +1+4.65E-11*P;   6.00000E+03    N !
FUNCTION BFELIQ     2.98150E+02  +1+4.98009787E-12*P+3.20078924E-14*T*P;
     6.00000E+03    N !
FUNCTION BCRBCC     2.98150E+02  +1+2.6E-11*P;   6.00000E+03    N !
FUNCTION BCGRA      2.98150E+02  +1+3.6E-10*P;   6.00000E+03    N !
FUNCTION BFEBCC     2.98150E+02  +1+2.80599565E-11*P+3.06481523E-16*T*P;
     6.00000E+03    N !
FUNCTION BFEFCC     2.98150E+02  +1+3.25236341E-11*P+3.36607808E-16*T*P;
     6.00000E+03    N !
```

```
TYPE_DEFINITION % SEQ *!

PHASE GAS:G %  1  1.0  !
   CONSTITUENT GAS:G :C1,C2,C3,C4,C5,C6,C7,CR,FE :  !

 PARAMETER G(GAS,C1;0)  2.98150E+02  +710400-112.911*T-21.054*T*LN(T)
 +2.5309E-04*T**2-3.86667E-08*T**3+3707.6*T**(-1)+RTLNP#;  3.00000E+03  Y
  +710400+2729.045-112.911*T-21.54137*T-18.17*T*LN(T)-7.385E-04*T**2
 +1.77967E-08*T**3+75203*T**(-1)+RTLNP#;  6.00000E+03  N REF189 !
 PARAMETER G(GAS,C2;0)  2.98150E+02  +809200.492+476557.6*T**(-1)
 +378.302884*T+RTLNP#-92.200716*T*LN(T)+.0861553799*T**2
 -2.31015411E-05*T**3;  4.00000E+02  Y
  +RTLNP#+828935.548-657802.204*T**(-1)-12.8384069*T-28.5984768*T*LN(T)
 -.00362018508*T**2+2.30113027E-07*T**3;  1.60000E+03  Y
  +RTLNP##+820236.64+1372216.02*T**(-1)+41.6450504*T-35.9012304*T*LN(T)
 -7.296896E-04*T**2+1.02996133E-09*T**3;  6.50000E+03  Y
  +RTLNP##+8037.62873+7.13408905E+08*T**(-1)+1579.97331*T-209.6184*T*LN(T)
 +.0157926754*T**2-2.93952499E-07*T**3;  9.00000E+03  Y
  +RTLNP#-1077409.5+1.86823941E+09*T**(-1)+3305.57979*T-400.194579*T*LN(T)
 +.0306129473*T**2-5.09367133E-07*T**3;  9.99999E+03  N REF189 !
 PARAMETER G(GAS,C3;0)  2.98150E+02  +833687.446-59322.844*T**(-1)
 +1.3684176*T+RTLNP#-35.0527152*T*LN(T)-.0121251483*T**2
 +1.29642356E-06*T**3;  1.20000E+03  Y
  +RTLNP##+807138.569+4179196.77*T**(-1)+226.569994*T-66.6226688*T*LN(T)
 +.0047850316*T**2-4.25622979E-07*T**3;  3.80000E+03  Y
  +RTLNP##+1044778.63-82501747.9*T**(-1)-703.89726*T+48.8896216*T*LN(T)
 -.01927596*T**2+4.90258805E-07*T**3;  6.00000E+03  Y
  +RTLNP#-1375294.3+1.83904714E+09*T**(-1)+4307.29326*T-523.565258*T*LN(T)
 +.0410331156*T**2-7.04079336E-07*T**3;  9.99999E+03  N REF189 !
 PARAMETER G(GAS,C4;0)  2.98150E+02  +1003706.34+199317.392*T**(-1)+RTLNP#
 +108.871898*T-52.057328*T*LN(T)-.0189560304*T**2+1.85112363E-06*T**3;
 1.20000E+03  Y
  +RTLNP##+973125.871+5399339.03*T**(-1)+356.692844*T-86.5853696*T*LN(T)
 -.00108131296*T**2+5.1440188E-08*T**3;  4.40000E+03  Y
  +RTLNP##+761565.209+1.35864787E+08*T**(-1)+911.433567*T-152.038192*T*LN(T)
 +.00813926072*T**2-1.99640257E-07*T**3;  7.50000E+03  Y
  +RTLNP#-3267589.93+4.00345375E+09*T**(-1)+7962.54222*T-940.431404*T*LN(T)
 +.0768258549*T**2-1.32375275E-06*T**3;  9.99999E+03  N REF189 !
 PARAMETER G(GAS,C5;0)  2.98150E+02  +1020031.18+299459.34*T**(-1)+RTLNP#
 +182.707296*T-64.5034728*T*LN(T)-.0266960664*T**2+2.92755805E-06*T**3;
 1.00000E+03  Y
  +RTLNP##+993156.233+3417001.67*T**(-1)+474.880979*T-107.079857*T*LN(T)
```

```
+.00274934824*T**2-8.26890196E-07*T**3;  2.20000E+03   Y
 +RTLNP#+1142512.8-32394322.9*T**(-1)-389.924723*T+6.7835192*T*LN(T)
-.0350827563*T**2+1.49320614E-06*T**3;  3.80000E+03   Y
 +RTLNP#+445025.439+2.78698861E+08*T**(-1)+2006.34751*T-286.104849*T*LN(T)
+.0195105568*T**2-4.12189549E-07*T**3;  7.00000E+03   Y
 +RTLNP#-4691131.49+5.08294597E+09*T**(-1)+11187.9121*T-1315.33621*T*LN(T)
+.111168608*T**2-1.94300567E-06*T**3;  9.99999E+03   N REF189 !
 PARAMETER G(GAS,C6;0)  2.98150E+02  +1246184.88+241241.072*T**(-1)+RTLNP#
+216.150737*T-74.8358608*T*LN(T)-.037038337*T**2+4.08827705E-06*T**3;
1.00000E+03   Y
 +RTLNP#+1200861.09+6717468.48*T**(-1)+635.721059*T-134.462463*T*LN(T)
-.00196717664*T**2+1.80331795E-07*T**3;  2.80000E+03   Y
 +RTLNP#+1214236.89-3611507.46*T**(-1)+640.871166*T-136.25991*T*LN(T)
+6.6676224E-04*T**2-3.87661547E-08*T**3;  6.50000E+03   Y
 +RTLNP#-2092852.91+2.85686451E+09*T**(-1)+6983.1698*T-853.633069*T*LN(T)
+.0698978622*T**2-1.29326185E-06*T**3;  8.50000E+03   Y
 +RTLNP#-9029732.89+1.02753605E+10*T**(-1)+17993.2677*T-2069.59301*T*LN(T)
+.164605987*T**2-2.67600481E-06*T**3;  9.99999E+03   N REF189 !
 PARAMETER G(GAS,C7;0)  2.98150E+02  +1317471.14+404461.004*T**(-1)+RTLNP#
+349.562891*T-98.7160408*T*LN(T)-.0314259612*T**2+2.66429519E-06*T**3;
1.00000E+03   Y
 +RTLNP#+1351184.27-6064281.23*T**(-1)+141.244152*T-71.7125048*T*LN(T)
-.0361169993*T**2+1.89586803E-06*T**3;  2.00000E+03   Y
 +RTLNP#+1036356.26+91042911.2*T**(-1)+1596.3469*T-256.701789*T*LN(T)
+.00993852716*T**2-1.01904109E-07*T**3;  3.60000E+03   Y
 +RTLNP#+1158858.06+13357756.8*T**(-1)+1335.6657*T-227.611274*T*LN(T)
+.0088048096*T**2-2.16460635E-07*T**3;  7.00000E+03   Y
 +RTLNP#-5584816.37+6.25252621E+09*T**(-1)+13524.3285*T-1595.97299*T*LN(T)
+.132400902*T**2-2.31373666E-06*T**3;  9.99999E+03   N REF189 !
 PARAMETER G(GAS,CR;0)  2.98150E+02  +391905.7-30.81*T-20.7861*T*LN(T)
+RTLNP#;  8.00000E+02   Y
 +391703.8-26.821*T-21.4288*T*LN(T)+8.23558E-04*T**2-1.82785E-07*T**3
+11071.9*T**(-1)+RTLNP#;  1.20000E+03   Y
 +401183.5-84.883*T-13.8336*T*LN(T)-.00108035*T**2-2.20692E-07*T**3
-1931730*T**(-1)+RTLNP#;  2.20000E+03   Y
 +393019-98.327*T-10.996*T*LN(T)-.00448127*T**2+1.1658E-07*T**3
+3710580*T**(-1)+RTLNP#;  3.60000E+03   Y
 +304527.7+160.8627*T-55.8648*T*LN(T)+.0053938*T**2-2.83633E-07*T**3
+35390400*T**(-1)+RTLNP#;  4.80000E+03   Y
 +853670.9-1290.85*T+115.378*T*LN(T)-.0183853*T**2+3.3669E-07*T**3
-2.95432E+08*T**(-1)+RTLNP#;  6.00000E+03   N REF105 !
 PARAMETER G(GAS,FE;0)  2.98150E+02  +406417-59.0479*T-33.0079*T*LN(T)
+.0092758*T**2-1.37996E-06*T**3+112882*T**(-1)+R#*T*LN(P);  9.50000E+02   Y
 +415082.7-149.23*T-19.9029*T*LN(T)+1.35653E-04*T**2-1.38831E-07*T**3
```

```
  -996892*T**(-1)+R#*T*LN(P);  2.35000E+03  Y
   +407130.1-128.688*T-22.1594*T*LN(T)-1.41918E-04*T**2-6.89897E-08*T**3
  +2445910*T**(-1)+R#*T*LN(P);  6.00000E+03  N REF7 !



PHASE LIQUID:L %  1  1.0  !
   CONSTITUENT LIQUID:L :C,CR,FE :  !

 PARAMETER G(LIQUID,C;0)  2.98150E+02  +117369-24.63*T+GHSERCC#+GPCLIQ#;
 6.00000E+03   N REF283 !
 PARAMETER G(LIQUID,CR;0)  2.98150E+02  +24339.955-11.420225*T
 +2.37615E-21*T**7+GHSERCR#+GPCRLIQ#;  2.18000E+03  Y
  +18409.36-8.563683*T+2.88526E+32*T**(-9)+GHSERCR#+GPCRLIQ#;  6.00000E+03
 N REF283 !
 PARAMETER G(LIQUID,FE;0)  2.98150E+02  +GFELIQ#+GPFELIQ#;   6.00000E+03
 N REF283 !
 PARAMETER G(LIQUID,C,CR;0)  2.98150E+02  -90526-25.9116*T;   6.00000E+03
  N REF101 !
 PARAMETER G(LIQUID,C,CR;1)  2.98150E+02  80000;   6.00000E+03   N REF101 !
 PARAMETER G(LIQUID,C,CR;2)  2.98150E+02  80000;   6.00000E+03   N REF101 !
 PARAMETER G(LIQUID,C,CR,FE;0)  2.98150E+02  -496063;   6.00000E+03 N REF322!
 PARAMETER G(LIQUID,C,CR,FE;1)  2.98150E+02  57990;   6.00000E+03   N REF322!
 PARAMETER G(LIQUID,C,CR,FE;2)  2.98150E+02  61404;   6.00000E+03   N REF322!
 PARAMETER G(LIQUID,C,FE;0)  2.98150E+02  -124320+28.5*T;   6.00000E+03
 N REF190 !
 PARAMETER G(LIQUID,C,FE;1)  2.98150E+02  19300;  6.00000E+03   N REF190 !
 PARAMETER G(LIQUID,C,FE;2)  2.98150E+02  +49260-19*T;   6.00000E+03   N
 REF190 !
 PARAMETER G(LIQUID,CR,FE;0)  2.98150E+02  -14550+6.65*T;   6.00000E+03
 N REF107 !



TYPE_DEFINITION & GES A_P_D BCC_A2 MAGNETIC  -1.0    4.00000E-01 !
PHASE BCC_A2  %&  2 1   3 !
   CONSTITUENT BCC_A2  :CR%,FE% : C,VA% :  !

 PARAMETER G(BCC_A2,CR:C;0)  2.98150E+02  +GHSERCR#+3*GHSERCC#+GPCRBCC#
 +3*GPCGRA#+416000;   6.00000E+03   N REF101 !
 PARAMETER TC(BCC_A2,CR:C;0)  2.98150E+02  -311.5;  6.00000E+03 N REF101 !
 PARAMETER BMAGN(BCC_A2,CR:C;0)  2.98150E+02  -.008;   6.00000E+03 N REF101 !
 PARAMETER G(BCC_A2,FE:C;0)  2.98150E+02  +322050+75.667*T+GHSERFE#
 +GPFEBCC#+3*GHSERCC#+3*GPCGRA#;   6.00000E+03   N REF190 !
 PARAMETER TC(BCC_A2,FE:C;0)  2.98150E+02  1043;  6.00000E+03   N REF190 !
 PARAMETER BMAGN(BCC_A2,FE:C;0)  2.98150E+02  2.22;   6.00000E+03   N REF190!
```

```
 PARAMETER G(BCC_A2,CR:VA;0)  2.98150E+02   +GHSERCR#+GPCRBCC#;
6.00000E+03   N REF283 !
 PARAMETER TC(BCC_A2,CR:VA;0)  2.98150E+02  -311.5;   6.00000E+03   N REF281!
 PARAMETER BMAGN(BCC_A2,CR:VA;0)  2.98150E+02  -.01;   6.00000E+03  N REF281!
 PARAMETER G(BCC_A2,FE:VA;0)  2.98150E+02   +GHSERFE#+GPFEBCC#;
6.00000E+03   N REF283 !
 PARAMETER TC(BCC_A2,FE:VA;0)  2.98150E+02  1043;   6.00000E+03   N REF281 !
 PARAMETER BMAGN(BCC_A2,FE:VA;0)  2.98150E+02  2.22;   6.00000E+03  N REF281!
 PARAMETER G(BCC_A2,CR,FE:C;0)  2.98150E+02  -1250000+667.7*T;
6.00000E+03   N REF322 !
 PARAMETER BMAGN(BCC_A2,CR,FE:C;0)  2.98150E+02  -.85;  6.00000E+03 N REF102!
 PARAMETER TC(BCC_A2,CR,FE:C;0)  2.98150E+02  1650;   6.00000E+03   N REF102!
 PARAMETER TC(BCC_A2,CR,FE:C;1)  2.98150E+02  550;   6.00000E+03   N REF102 !
 PARAMETER G(BCC_A2,CR:C,VA;0)  2.98150E+02  -190*T;   6.00000E+03  N REF101!
 PARAMETER G(BCC_A2,FE:C,VA;0)  2.98150E+02  -190*T;   6.00000E+03 N REF190 !
 PARAMETER G(BCC_A2,CR,FE:VA;0)  2.98150E+02  +20500-9.68*T; 6000 N REF107 !
 PARAMETER BMAGN(BCC_A2,CR,FE:VA;0)  2.98150E+02  -.85; 6.00000E+03 N REF107!
 PARAMETER TC(BCC_A2,CR,FE:VA;0)  2.98150E+02  1650;   6.00000E+03 N REF107 !
 PARAMETER TC(BCC_A2,CR,FE:VA;1)  2.98150E+02  550;   6.00000E+03 N  REF107 !


PHASE CEMENTITE  %  2 3   1 !
   CONSTITUENT CEMENTITE  :CR,FE% : C :  !

 PARAMETER G(CEMENTITE,CR:C;0)  2.98150E+02   +3*GHSERCR#+GHSERCC#-48000
-9.2888*T;   6.00000E+03   N REF322 !
 PARAMETER G(CEMENTITE,FE:C;0)  2.98150E+02   +GFECEM#; 6.00000E+03 N REF190 !
 PARAMETER G(CEMENTITE,CR,FE:C;0)  2.98150E+02   +25278-17.5*T;
6.00000E+03   N REF322 !


TYPE_DEFINITION ' GES A_P_D FCC_A1 MAGNETIC  -3.0    2.80000E-01 !
PHASE FCC_A1  %' 2 1   1 !
   CONSTITUENT FCC_A1  :CR,FE% : C,VA% :  !

 PARAMETER G(FCC_A1,CR:C;0)  2.98150E+02   +GHSERCR#+GHSERCC#+1200-1.94*T;
6.00000E+03   N REF322 !
 PARAMETER G(FCC_A1,FE:C;0)  2.98150E+02   +77207-15.877*T+GFEFCC#+GHSERCC#
+GPCFCC#;   6.00000E+03   N REF190 !
 PARAMETER TC(FCC_A1,FE:C;0)  2.98150E+02  -201;   6.00000E+03   N REF190 !
 PARAMETER BMAGN(FCC_A1,FE:C;0)  2.98150E+02  -2.1;   6.00000E+03 N REF190 !
 PARAMETER G(FCC_A1,CR:VA;0)  2.98150E+02   +GCRFCC#+GPCRBCC#;
6.00000E+03   N REF281 !
 PARAMETER TC(FCC_A1,CR:VA;0)  2.98150E+02  -1109;   6.00000E+03 N   REF281 !
```

```
 PARAMETER BMAGN(FCC_A1,CR:VA;0)  2.98150E+02  -2.46;  6.00000E+03   N
REF281 !
 PARAMETER G(FCC_A1,FE:VA;0)  2.98150E+02  +GFEFCC#+GPFEFCC#;
6.00000E+03   N REF283 !
 PARAMETER TC(FCC_A1,FE:VA;0)  2.98150E+02  -201;  6.00000E+03   N REF281 !
 PARAMETER BMAGN(FCC_A1,FE:VA;0)  2.98150E+02  -2.1;  6.00000E+03 N REF281 !
 PARAMETER G(FCC_A1,CR,FE:C;0)  2.98150E+02  -74319+3.2353*T;
6.00000E+03   N REF322 !
 PARAMETER G(FCC_A1,CR:C,VA;0)  2.98150E+02  -11977+6.8194*T;
6.00000E+03   N REF322 !
 PARAMETER G(FCC_A1,FE:C,VA;0)  2.98150E+02  -34671;  6.00000E+03 N REF190 !
 PARAMETER G(FCC_A1,CR,FE:VA;0)  2.98150E+02  +10833-7.477*T;
6.00000E+03   N REF107 !
 PARAMETER G(FCC_A1,CR,FE:VA;1)  2.98150E+02  1410;  6.00000E+03 N  REF107 !


PHASE GRAPHITE  %  1  1.0  !
   CONSTITUENT GRAPHITE  :C :  !

 PARAMETER G(GRAPHITE,C;0)  2.98150E+02  +GHSERCC#+GPCGRA#;  6.00000E+03
 N REF283 !


PHASE M23C6  %  3 20   3   6 !
   CONSTITUENT M23C6  :CR%,FE% : CR%,FE% : C :  !

 PARAMETER G(M23C6,CR:CR:C;0)  2.98150E+02  +GCRM23C6#;  6.00000E+03   N
REF102 !
 PARAMETER G(M23C6,FE:CR:C;0)  2.98150E+02  +.1304348*GCRM23C6#
+.8695652*GFEM23C6#;  6.00000E+03   N REF102 !
 PARAMETER G(M23C6,CR:FE:C;0)  2.98150E+02  +.8695652*GCRM23C6#
+.1304348*GFEM23C6#;  6.00000E+03   N REF102 !
 PARAMETER G(M23C6,FE:FE:C;0)  2.98150E+02  +GFEM23C6#;  6.00000E+03   N
REF102 !
 PARAMETER G(M23C6,CR,FE:CR:C;0)  2.98150E+02  -205342+141.6667*T;
6.00000E+03   N REF322 !
 PARAMETER G(M23C6,CR,FE:FE:C;0)  2.98150E+02  -205342+141.6667*T;
6.00000E+03   N REF322 !


PHASE M7C3  %  2 7   3 !
   CONSTITUENT M7C3  :CR%,FE : C :  !

 PARAMETER G(M7C3,CR:C;0)  2.98150E+02  +GCRM7C3#;  6.00000E+03   N REF322 !
```

```
   PARAMETER G(M7C3,FE:C;0)  2.98150E+02   +7*GHSERFE#+3*GHSERCC#+75000
 -48.2168*T;   6.00000E+03   N REF322 !
   PARAMETER G(M7C3,CR,FE:C;0)  2.98150E+02   -4520-10*T; 6.00000E+03 N REF322 !


 PHASE SIGMA  %  3 8   4   18 !
    CONSTITUENT SIGMA  :FE : CR : CR,FE :  !

  PARAMETER G(SIGMA,FE:CR:CR;0)  2.98150E+02   +8*GFEFCC#+22*GHSERCR#+92300
 -95.96*T+GPSIG1#;   6.00000E+03   N REF107 !
   PARAMETER G(SIGMA,FE:CR:FE;0)  2.98150E+02   +8*GFEFCC#+4*GHSERCR#
 +18*GHSERFE#+117300-95.96*T+GPSIG2#;   6.00000E+03   N REF107 !

 LIST_OF_REFERENCES
 NUMBER  SOURCE
   REF7     'A. Fernandez Guillermet, P. Gustafson,
         High Temp. High Press. vol 16, (1985) p 591-610,
         TRITA-MAC 229(1984); FE'
   REF189   'P. Gustafson, Carbon vol 24, (1986) p 169-176, TRITA-MAC
        236(1984); C'
   REF105   'J-O Andersson, Int. J. Thermophys. Vol 6, (1985), p 411-419
         TRITA 0230 (1984); CR'
   REF283   'Alan Dinsdale, SGTE Data for Pure Elements,
         Calphad Vol 15(1991) p 317-425,
         also in NPL Report DMA(A)195 Rev. August 1990'
   REF101   'J-O Andersson, Calphad Vol 11 (1987) p 271-276, TRITA 0314; C-CR'
   REF322   'Byeong-Joo Lee, unpublished revision (1991); C-Cr-Fe-Ni'
   REF190   'P. Gustafson, Scan. J. Metall. vol 14, (1985) p 259-267
         TRITA 0237 (1984); C-FE'
   REF107   'J-O Andersson, B. Sundman, CALPHAD Vol 11, (1987), p 83-92
         TRITA 0270 (1986); CR-FE'
   REF281   'Alan Dinsdale, SGTE Data for Pure Elements, NPL Report
        DMA(A)195
         September 1989'
   REF102   'J-O Andersson, Met. Trans A, Vol 19A, (1988) p 627-636
         TRITA 0207 (1986); C-CR-FE'
  !

 END_OF_TRANSFER - !
```

6   Appendix I. Modifications

This paper has been revised thoroughly since last time. The subroutine interface which was then included as a guide for writing a database management program has been omitted as it has not been put to any use.

Some general information to make this paper understandable for those who have not participated in the discussions within SGTE has been added. However, the model section is very meagre and will be better documented in a separate paper.

The following changes have been made in the actual interchange format:

Comments allowed between keywords. The maximum length of a line defined and the set of allowed control characters.

Definition of species stoichiometry has been simplified.

Reference identifier added to the PARAMETER keyword.

EXPTYPE 5 added.

keyword FUNCTION added.

keyword CONSTITUENT added.

keyword REFERENCE added.

keyword TYPE_CODE added.

keyword REDEFINE added.

keyword END_OF_TRANSFER added.

6.1   Changes in version 2.1

Major suffix added for CONSTITUENT keyword.

Keyword MATERIAL added.

Keyword NEW_IDENTIFIER added.

7   Appendix II. Summary of keywords

```
ELEMENT <element name> <SER state> <atomic mass>
    <enthalpy at 298.15 K - enthalpy at 0 K>
    <entropy  at 298.15 K> END.


SPECIES <species name> /<element name> <stoichiometric factor>/ END.


PHASE <phase name> <type code> <number of sublattices>
    /<number of sites in the sublattice>/ END.


CONSTITUENT <phase name> : <species on first sublattice>
/ : <species on other sublattices in sublattice order>/ : end.


PARAMETER <identifier> ( <phase name>, <component array> ; <degree> )
    <expression> <reference identifier> END.
```

```
EXPTYPE <integer code> END.

FUNCTION <name> <expression> END.

MODEL <model name> <model code> '<description>' END.

REFERENCE <reference identifier> '<text>' END.

TYPE_CODE <type code> <value> '<comment>' END.

MESSAGE <integer code> '<text>' END.

REDEFINE <item> <value> <new value> END.

END_OF_TRANSFER <checksum> END.

MATERIAL <name> <composition> <comment> END.

NEW_IDENTIFIER <identifier> <unit> <meaning> END.
```

8 Appendix III. Revison of format 2001

9 Appendix IV. Comments for future revisions

When reviewing this document next time the following issues may need to be addressed:
   - information about miscibility gaps in a phase,
   - revision of models and how to identify them,
   - functions not in call order, instead warning of missing functions at end.