# CpFit program

Alexey L. Voskov

Laboratory of Chemical Thermodynamics,
Department of Chemistry,
Lomonosov Moscow State University, Moscow, Russia

December 31, 2018

# Contents

# Chapter 1

# Thermodynamic Model

CpFit program implements thermodynamic model of heat capacity and other thermodynamic functions based on usage of Einstein-Plank functions sum and empirical parameters. The used approach has been developed by Voronin et al. [1].

All thermodynamic functions (e.g. heat capacity, entropy and enthalpy) are represented as sums of Einstein-Plank terms:

$$C_p(T) = \sum_{i=1}^{m} \alpha_i C_{\mathrm{E}} \left( \frac{\theta_i}{T} \right); \ \frac{C_{\mathrm{E}}(x)}{R} = \frac{3x^2 e^x}{(e^x - 1)^2} \tag{1.1}$$

$$S(T) - S(0) = \sum_{i=1}^{m} \alpha_i S_{\mathrm{E}} \left( \frac{\theta_i}{T} \right); \ \frac{S_{\mathrm{E}}(x)}{R} = 3 \left[ \frac{x}{e^x - 1} - \ln\left(1 - e^{-x}\right) \right] \tag{1.2}$$

$$H(T) - H(0) = \sum_{i=1}^{m} \alpha_i \left[ U_{\mathrm{E}} \left( \frac{\theta_i}{T} \right) - U_0 \right]; \ \frac{U_{\mathrm{E}}(x) - U_0}{RT} = \frac{3x}{e^x - 1} \tag{1.3}$$

where $\alpha_i$ and $\theta_i$ are adjustable (usually by means of the least squares method) model parameters. They can be estimated from experimental data using least squares method. A universal gas constant value $R = 8.3144598\ \mathrm{J \cdot mol^{-1}K^{-1}}$ from CODATA 2014 is used in the CpFit program. In eq 1.1, 1.2 and 1.3 $\alpha_i$ values are dimensionless and $\theta_i$ values are expressed in K.

The thermodynamic model of heat capacity based on Einstein-Plank functions allows to approximate $C_p(T)$, $S(T) - S(0)$ and $H(T) - H(0)$ in a wide temperature range using a unified set of parameters. Also unlike polynomial models it can give a physically correct extrapolation to a wider temperature range.

Implementation of this model in CpFit hides all analytical expressions and technical details and allow to use it as an ordinary statistical package for a nonlinear regression.

# Chapter 2

# Optimization Procedure

## 2.1 Theory

In CpFit program $\alpha_i$ and $\theta_i$ parameters are found by the least squares method using the next sum of squares $\chi^2$:

$$\chi^2(\beta) = \sum_{i-1}^{n} W_i^2 \left[ Y_i^{\text{calc}}(\beta) - Y_i^{\text{exp}} \right]^2 \tag{2.1}$$

where $Y_i = c_p(T_i)$ or $Y_i = H(T_{2,i}) - H(T_{1,i})$, $W_i = \omega_i$ for the case of absolute deviation or $W_i = \omega_i \left(Y_i^{\text{exp}}\right)^{-1}$ for the case of relative deviation where $\omega_i$ are user-defined statistical weights (default values are $\omega_i = 1$) and $\beta$ is generalized notation for the $\alpha_i$ and $\theta_i$ model parameters. To minimize $\chi^2$ value Levenberg-Marquardt algorithm implementation from `levmar` library [2] are used.

Standard deviations of the $\alpha_i$ and $\theta_i$ parameters are calculated using the next formula:

$$s_\beta^2 = \hat{\sigma}^2 \text{diag}\left[ \left( J^\top J \right)^{-1} \right];\ J_{ij} = \frac{\partial Y_i^{\text{calc}}}{\partial \beta_j};\ \hat{\sigma}^2 = \frac{\chi^2}{n - 2m} \tag{2.2}$$

where $J$ is Jacobian, $n$ is number of experimental points, $m$ is number of terms (and $2m$ is number of parameters respectively), $\hat{\sigma}^2$ is the model standard error. Parameters confidence intervals can be estimated as:

$$\Delta\beta = s_\beta \cdot t_{p,f};\ f = n - 2m \tag{2.3}$$

where $t_{p,f}$ is $t$-distribution two-sided quantile with probability $p$ (CpFit uses $p = 0.95$ for 95 % confidence intervals), $f = n - 2m$ is number of freedom degrees. Usually $t_{\alpha,f} \approx 2 \div 3$.

CpFit also can estimate standard deviations of thermodynamic functions that can be used to estimate their confidence intervals (don't mix with prediction intervals!) using the next formula:

$$s_Y(T) = \sqrt{J_x^\top C J_x} = \hat{\sigma} \sqrt{J_x^\top \left( J^\top J \right)^{-1} J_x} \tag{2.4}$$

where $C$ is covariance matrix, $J_x = J_x(T)$ is column vector that has the next layout:

$$J_x(T) = \begin{pmatrix} \frac{\partial Y(T)}{\partial \beta_1} \\ \vdots \\ \frac{\partial Y(T)}{\partial \beta_{2m}} \end{pmatrix} \tag{2.5}$$

Confidence intervals of $Y(T)$ thermodynamic functions (i.e. $c_p$ and $H(T) - H(T_0)$ at the point $(T)$ can be calculated using the next relationship similar to eq 2.3:

$$\Delta Y = s_Y \cdot t_{p,f};\ f = n - 2m \tag{2.6}$$

If heat capacity dependence has anomalies that cannot be described by Einstein-Planck functions sum it is possible to add extra terms, i.e. use the next formula for $C_p(T)$:

$$C_p(T, \vec{\alpha}, \vec{\theta}, \vec{b}_1, \ldots, \vec{b}_k) = C_p^{\mathrm{BL}}(T, \vec{\alpha}, \vec{\theta}) + \sum_{i=1}^{k} C_{p,i}^{\mathrm{ex}}(T, \vec{b}_i) \tag{2.7}$$

where $C_p^{\mathrm{BL}}$ is Einstein-Planck function sum ("baseline" term), $C_p^{\mathrm{ex}}$ is arbitrary terms ("extra" terms) and $\vec{b}_i$ is vector with optimized model parameters for $i$-th extra term. A list of extra terms supported by CpFit program is given in Chapter 4.2. The next algorithm is used to optimize model parameters:

1. Optimize $\vec{\alpha}$ and $\vec{\theta}$ parameters by minimizing eq. 2.1 using only data outside $C_p$ anomalies.

   *Note: in the current version of CpFit program experimental enthalpies are not excluded at all.*

2. Optimize $\vec{b}_i$ values for each extra term separately using the next objective function:

$$\chi^{\mathrm{ex}}\left(\vec{b}_i\right) = \sum_j \left(W_j^{(i)}\right)^2 \left[Y_j^{(i)} - Y_j^{(i),\mathrm{BL}} - Y_j^{(i),\mathrm{ex}}\left(\vec{b}_i\right)\right]^2 \tag{2.8}$$

## 2.2 Practical Tips

The goal of the optimization process is to minimize the model standard error $\hat{\sigma}^2$. However during increasing the number of parameters there is a risk to obtain an ill-conditioned task and not reliable values set. To avoid it you can follow the next tips:

1. Begin the optimization for 2–3 terms (or 4–6 parameters) and gradually increase it using previous results as initial approximations

2. Avoid statistically not significant coefficients (i.e. the conditions $\alpha_i \geq \Delta\alpha_i \approx 2s_{\alpha_i}$ and $\theta_i \geq \Delta\theta_i \approx 2s_{\theta_i}$ should be satisfied). CpFit program will automatically highlight in red statistically not significant coefficients (see Figure 4.2).

3. Both $\alpha_i$ and $\theta_i$ must be positive numbers (automatically controlled by the CpFit optimization engine).

You can also use evaluations procedures with automatic selection of terms number (see Chapter 4.2). They use iterative addition of new terms and control of statistical significance of them using $t$-criterion. However they sometimes can give number of terms less than can be obtained by manual selection of initial approximations.

If you optimize heat capacity and enthalpy data together always remember that their orders of magnitude are different, i.e. $\frac{\Delta H}{c_p} \approx 10^3$. In this case either relative deviations or adequate statistical weights must be used. In the case of $\omega_i = 1$ for all data and absolute deviations $c_p$ data will not be optimized correctly.

# Chapter 3

# File Formats

## 3.1 Experimental data file format

Input data can be supplied in human readable text files with `.dat` or `.txt` extension. A description of the format is present in Table 3.1.

The main features of the format are:

- .dat file is an ordinary ASCII text file
- Dot (and only dot) is used as a decimal separator
- Both DOS/Windows and UNIX line endings are supported
- Numbers are separated by either spaces or tabs (one or multiple)
- Empty lines and comments in the data set are ignored without warning

It currently supports two kind of data: heat capacities $C_p(T)$ and enthalpies/heat content $H(T_2) - H(T_1)$.

## 3.2 Model parameters file format

Files with model parameters can be either exported from the Results Window or imported to the Main Window. That files have `.csv` format, semicolon `;` is used as a separator and dot `.` is used as a decimal point sign. An example of the `.csv` file with model parameters is given below:

```
alpha;dalpha;salpha;theta;dtheta;stheta
14.0168725905787;0.914895;0.457827;3583.8945135367;151.031;75.5782
6.26905917915019;0.401292;0.200813;1393.30718925711;88.5056;44.2895
7.3042223855265;0.350348;0.175319;688.948063566627;41.8236;20.9292
5.69813172425687;0.238324;0.119261;306.421049769356;20.2346;10.1257
2.86883301382133;0.329537;0.164905;145.550553624711;12.1593;6.0847
0.289410000831521;0.171896;0.0860195;66.9793881821218;12.0881;6.04905
```

The first row contains columns description and can contain the next values:

- `alpha` — $\alpha_i$ model parameter, dimensionless.

Table 3.1: .dat file format used by CpFit

| **Line number** | **Example** | **String content** |
|---|---|---|
| 1–3 | `Heat capacity (ScF3)`<br>`Temperature`<br>`Cp` | Arbitrary text information |
| 4 | `539` | Number of dots |
| 5 | `0` | Reserved and should be equal to zero (intended to be a number of dots to be ignored at the end of file but it is not used now) |
| 6–$k$ | `303.35 82.806`<br>`302.98 82.421  0  1`<br>`303.01 82.417  1  0.85`<br>`303.01 82.417  1  0.85 Rom15` | Heat capacity experimental data. Either $T$–$C_\mathrm{p}$ lines or $T$-$C_\mathrm{p}$–$rf$–$\omega$–Data series lines where $\omega$ are statistical weights from eq 2.1 and $rf$ is "use relative deviation" flag (can be either 0 or 1). If $\omega$ or $rf$ are omitted their default values will be used for this line ($\omega = 1$ and $rf = 0$). Data series name can be omitted. |
| 6–$k$ | `H 298.15 940 46534 1 1 Aga03` | Enthalpy experimental data. Kept as $T_1$–$T_2$–$\Delta H$–$rf$–$\omega$–Data series lines where $\Delta H = H(T_2) - H(T_1)$. $rf$, $\omega$ and Data series names can be omitted. Format is similar to $C_p$ data format. |
| 6–$k$ | `# Some comment` | Comment (will be ignored by CpFit). Note: comments won't be saved by CpFit if "Save as" option is used. |
| 6–$k$ | `X 265 310 DEFAULT 293.91 \|`<br>`1 1e-005 0.5` | Extra (excess) heat capacity term. Kept as `X`–$T_\mathrm{min}$–$T_\mathrm{max}$–Term name–Constant parameters—`\|`—Optimized parameters (initial values). |
| 6–$k$ | `AL 3.012 1.022` | Initial approximation for $\alpha$ parameters. Note: it won't be saved by CpFit if "Save as" option is used! |
| 6–$k$ | `TH 1023.2 3.02e2` | Initial approximation for $\theta$ parameters. Note: it won't be saved by CpFit if "Save as" option is used! |

- `dalpha` — $\Delta\alpha_i$, 95 % confidence interval (using two-sided $t$-distribution quantile) for the model parameter $\alpha_i$.
- `salpha` — $s(\alpha_i)$ standard deviation for the model parameter $\alpha_i$.
- `theta` — $\theta_i$ model parameter, K.
- `dtheta` — $\Delta\theta_i$, 95 % confidence interval (using two-sided $t$-distribution quantile) for the model parameter $\theta_i$.
- `stheta` — $s(\theta_i)$, standard deviation for the model parameter $\theta_i$.

Export from the Results Window (see Figure 4.2) gives all six columns. Import to the Main Window uses only two columns `alpha` and `theta`.

## 3.3 Examples

Several examples of `.dat` files with heat capacity and heat content are supplied with CpFit program. They contain real experimental data:

- `natrolite.dat` — natrolite $Na_2Al_2Si_3O_{10} \cdot 2H_2O$ [3]. Contains no extra terms.
- `K_natrolite.dat` — potassium-substituted natrolite $K_{1.86}Na_{0.01}Ca_{0.04}Mg_{0.01}\,[Al_{1.96}Si_{3.04}O_{10}] \cdot 2.33H_2O$ [4]. Contains one extra term that describes one lambda-transition.
- `Tl_natrolte.dat` — thallium-substituted natrolite $Tl_{1.87}Na_{0.05}Mg_{0.03}\,[Al_{1.98}Si_{3.04}O_{10}] \cdot 2.72H_2O$ [5]. Contains one extra term that describes two partially overlapping phase transitions.
- `ThO2.dat` — thorium dioxide $ThO_2$ [6, 7, 8, 9, 10, 11, 12]. Contains one extra term that describes one (pre-melting) lambda-transition.
- `UO2.dat` — uranium dioxide $UO_2$. Contains one extra term that describes two lambda-transitions. [13, 14, 15, 16, 17, 18, 19, 20].

`K_natrolite.dat` and `Tl_natrolite.dat` contain only heat capacity data, other `.dat` files — both heat capacity and heat content data.

## 3.4 Custom extra terms file format

Custom extra terms are written in Lua 5.3 language in text files with `.lua` extension. Recompilation of CpFit is not required for their usage. Each file is Lua script that can contain only one extra term. The term is returned in the form of table (array) by `return` statement. An example of the script:

```
-- Custom excess heat capacity term for CpFit
-- Cp(T)/R = CpE(T)/R + a*T + b*T^2 model implementation
-- Chase M.W. et al. // Calphad Vol. 19, NO. 4, pp. 437-447, 1995
return {
```

```
  type = 'EPPOLY',
  description = 'Einstein + poly',
  formula = 'C_p/R=3x^2e^x/(e^x-1)^2+aT+bT^2; x=TH/T',
  cpfunc = function(T, lsqparam, fixparam, Tmin, Tmax)
    local theta = lsqparam[1]
    local a, b = lsqparam[2], lsqparam[3]
    local x = theta / T
    local ex = math.exp(x)
    local cp = 0
    if x < 300 then
      cp = R * (3*x^2 * ex / (ex - 1) ^ 2 + a*T + b*T^2);
    else
      cp = R * (3*x^2 / ex + a*T + b*T^2);
    end
    return cp
  end,
  lsqparam = {100, 0, 0},
  lsqparam_names = {'theta', 'a', 'b'},
  lsqparam_lb = {0,   -1e5, -1e5},
  lsqparam_ub = {1e5,  1e5,  1e5},
  fixparam = {},
  fixparam_names = {}
  Tmin = 0.0,
  Tmax = 3000.0
}
```

The next fields in the table are required:

- `type` — term type name. Must be a string. Will be used in `.dat` files as term type identifier.
- `description` — term short description (will be shown in CpFit GUI). Must be a string.
- `formula` — formula that will be shown in CpFit GUI. Must be a string. Simplified subset of LaTeX formatting is supported (`_` for subscripts and `^` for superscripts; no brackets support). Greek letters and other mathematical symbols must be written in UTF-8, LaTeX formatting is not supported for them.
- `cpfunc` — function that calculates heat capacity value. It has five inputs and one output, i.e. `function(T, lsqparam, fixparam, Tmin, Tmax)`. The function must return $C_p$ numeric value in $\mathrm{J} \cdot (\mathrm{mol} \cdot \mathrm{K})^{-1}$.
  - `T` — temperature, K (numeric value)
  - `lsqparam`, `fixparam` — tables with current values of model parameters that are filled with numeric values. Their size is consistent to the default values given in the script (see below).
  - `Tmin`, `Tmax` — current values of $T_{\min}$ and $T_{\max}$ parameters. They are numeric values.
- `lsqparam` — default initial approximation for LSQ optimized parameters.

- `lsqparam_lb`, `lsqparam_ub` — lower and upper boundaries for LSQ optimized parameters.
- `lsqparam_names` — LSQ optimized parameters names.
- `fixparam` — default values for fixed parameters.
- `fixparam_names` — fixed parameters names.
- `Tmin`, `Tmax` — default value for minimal and maximal temperatures, K.

There are several predefined constants and functions:

- `R` — universal gas constant
- `CAL` — number of joules in calorie (4.184)
- `CpDebye` — `CpDebye(T, theta)` function, implementation of Debye heat capacity model:

$$\frac{C_p^{\mathrm{D}}\left(T_D, T\right)}{R} = 9\left(\frac{T}{T_{\mathrm{D}}}\right)^3 \int_0^{\frac{T_{\mathrm{D}}}{T}} \frac{x^4 e^x}{\left(e^x - 1\right)^2}\, dx \tag{3.1}$$

# Chapter 4

# User interface

## 4.1 Installation

CpFit program has the next system requirements:

- Microsoft Windows XP/Vista/7/8/8.1/10 (32-bit or 64-bit)
- 256 Mb of RAM
- 5 Mb of free disk space
- SVGA with 1024x768 resolution or higher

The program doesn't require any installation. You can just unpack an archive with the program and begin the work.

If you want to make the program from the source code (that is not required) you will need:

- GCC C++ compiler 4.5.2 or higher and MinGW environment
- wxWidgets 3.0.2 or higher
- levmar 2.6 or higher
- Lua 5.3
- Doxygen and MathJax are not required but highly recommended (to generate documentation about the source code)

More detailed information about compilation and linking is given in the source code.

## 4.2 Main window

Main window (see Figure 4.1) allows to load, save and edit experimental data and manage the parameters optimization process.

A menu of the main window contains the next items:

- **File** — Contains operations for loading and saving experimental data from and to files.

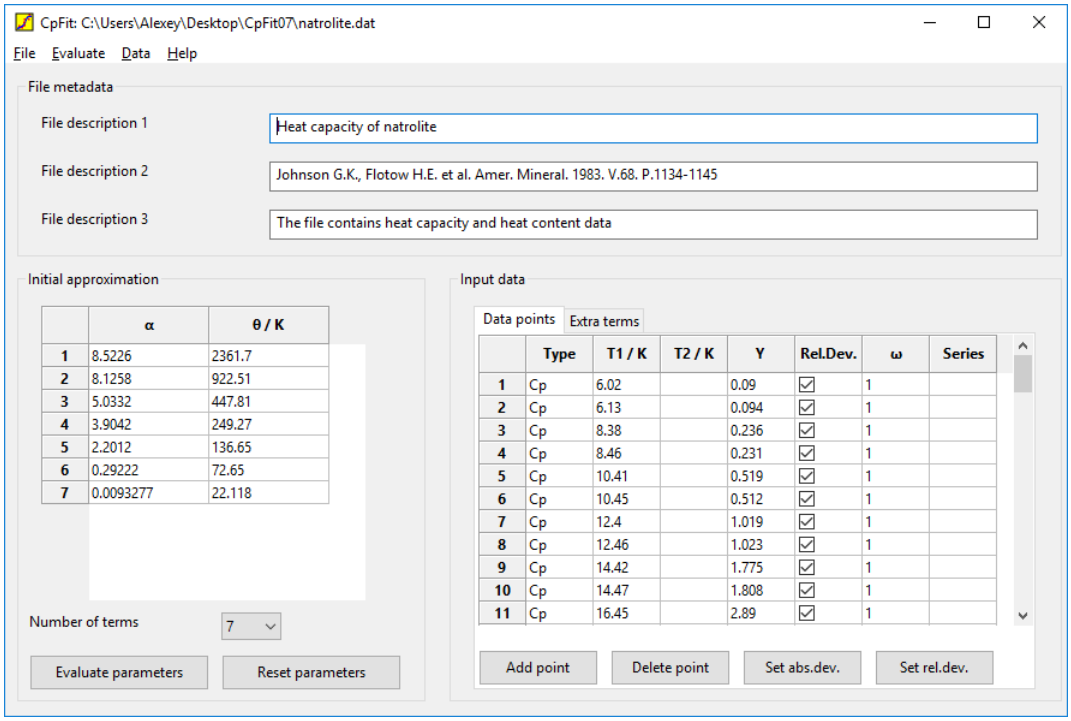


Figure 4.1: CpFit program main window

- **New** — Clear all data in the window.
- **Open** — Open `.dat` file with $C_p$ data.
- **Save as** — Save current data set into `.dat` file.
- **Load parameters** — Loads parameters (i.e. initial approximation) from `.csv` file and puts them into an initial approximation table.
- **Load custom extra term** — Loads custom user-defined extra terms from `.lua` file (must be written in Lua 5.3 language)
- **Exit** — Close the main window and exit from the program.

- **Evaluate** — Different variants of the model parameters evaluation.
  - **Parameters values** — Evaluate only values, number of terms is fixed (similar to "evaluate parameters" button).
  - **Parameters set (don't use init.approx.)** — Evaluation of both number of terms and parameters values without using of an initial approximation given by a user.
  - **Parameters set: extra terms only** — Evaluate only extra terms parameters , use base line parameters $\vec{\alpha}$ and $\vec{\theta}$ from the initial approximation as constants.
  - **Parameters set (use init.approx.)** — Evaluation of both number of terms and parameters values with using of an initial approximation given by a user.
  - **Evaluate without optimization** — Creates results window with parameters given as initial approximation by a user but doesn't run any optimization process. Can be used without any experimental data. If experimental data are given the program will evaluate deviations and parameters confidence intervals.

- **Data** — Operations with experimental data
  - **Sort data** — Sort data points by temperature and by data series identifier.
- **Help** — Contains information about the program.
  - **About** — Show the window with a brief information about the program.

All evaluation methods that evaluate number of terms use the next algorithm:

1. Set initial approximation. CpFit can use either user-defined initial approximation (see Figure 4.1) or set it automatically. In the case of automatic selection one term with $\alpha_1 = 0.1$, $\theta_1 = 1$ will be used.
2. Run optimization for a given initial approximation and test an obtained set of parameters for statistical significance (using $t$-test with two-sided 95 % confidence intervals, see eq. 2.3).
3. If all parameters are statistically significant — add new term with $\alpha_i = 10^{-3}$ and $\theta = 1$ and go to the Step 2. If some parameters are statistically not significant — take the previous result of optimization and return it.

Such algorithm allows to obtain reasonable results in most cases. However in some cases it can give less model terms than possible and manual setting of an initial approximation consisting from several terms may be beneficial.

The main window contains three areas: file metadata area, initial approximation area and input data area.

*File metadata area* contains three arbitrary text strings that are used to describe a content of the data set. It is recommended to use the first line for data series description and leave default values in the second and in the third line.

*Initial approximation area* contains tools for setting initial values of model parameters and number of terms in eq 1.1, 1.2 and 1.3.

- *Table* — This table contains values for the model parameters. Each row correspond to the model term, $\alpha$ and $\theta$ / `K` columns correspond to $\alpha_i$ and $\theta_i$ parameters respectively. Note that the content of this table can be modified by results window (by means of transferring results into initial approximation area).
- *Number of terms* combo box — Use this combo box to select a desired number of terms.
- *Evaluate parameters* button — Runs a proces of the parameters optimization. Results window will be opened.
- *Reset parameters* button — Sets each $\alpha_i$ and each $\theta_i$ value in the table to $10^{-6}$ and 1 respectively.

*Input data area* contains the data set and extra terms that will be used for model parameters optimization. It consists of two tabs: *Data points* tab and *Extra terms* tab.

*Data points* tab allows to view and edit all experimental points. It contains the next elements:

- *Table* — Contains the used data set and contains seven columns:

  - `Type` — data type (either heat capacity `Cp` or enthalpy `H`). Selected by combo box.
  - `T1 / K` — temperature in K. It is either $T$ for $C_p(T)$ (heat capacity) or $T_1$ for $H(T_2) - H(T_1)$ (enthalpy) data.
  - `T2 / K` — temperature in K. It is $T_2$ for $H(T_2) - H(T_1)$ (enthalpy) data. In the case of heat capacity data it is ignored and uneditable.
  - `Y` is for $C_p$ in $\mathrm{J} \cdot \mathrm{mol} \cdot \mathrm{K}^{-1}$.
  - `Rel.dev.` — absolute/relative deviation selection
  - $\omega$ — statistical weights $\omega_i$ from eq 2.1.
  - `Series` — data series ID that is used for markers and legends at 2D plots.

- *Add point* button — Adds a new point into the table.
- *Delete point* button — Removes selected rows from the table.
- *Set abs.dev.* button — Sets all deviations for all points as absolute (i.e. unselects all checkboxes in the third row)
- *Set rel.dev.* button — Sets all deviations for all points as relative (i.e. selects all checkboxes in the third row)

*Extra terms* tab allows to modify extra terms in the heat capacity model that cannot be described by Einstein-Planck functions sum. It contains the next controls:

- *Terms* combo box — allows to select the current term.
- *Add* button — adds a new term.
- *Delete* button — removes the current term.
- *Fix.param.* table — contains fixed parameters $c_i$ that are not changed during extra terms parameters optimization.
- *LSQ param.* table — contains varied parameters $b_i$ that are optimized by the least squares method.
- *Term type* combo box — select the type of the current term. The possible options are:
  - `Default` (named as `DEFAULT` in `.dat` files) — default term for description of lambda-transition, i.e.:

$$\frac{C_p^{\mathrm{ex}}}{R} = b_1 \exp\left[b_2 \left(b_3 \Delta T - |\Delta T|\right)\right];\ \Delta T = T - T_{\mathrm{tr}};\ T_{\mathrm{tr}} = c_1 \tag{4.1}$$

  - `Two terms` (named as `TWOTERMS` in `.dat` files) — sum of two terms for description of two partially overlapping lambda-transitions.

$$\frac{C_p^{\mathrm{ex}}}{R} = \sum_{i=1}^{2} d_1^{(i)} \exp\left[d_2^{(i)} \left(d_3^{(i)} (\Delta T)_i - |(\Delta T)_i|\right)\right];\ (\Delta T)_i = T - T_{\mathrm{tr},i};\ T_{\mathrm{tr},i} = c_i \tag{4.2}$$

$$d_i^{(j)} = b_{i+3(j-1)} \tag{4.3}$$

– `Left exp.` (named as `LEFTEXP` in `.dat` files) — exponential function for description of the left side of a peak.

$$\frac{C_p^{\text{ex}}}{R} = b_1 \exp\left[b_2 \left(T - T_{\max}\right)\right] \tag{4.4}$$

– `Right exp.` (named as `RIGHTEXP` in `.dat` files) — exponential function for description of the left side of a peak.

$$\frac{C_p^{\text{ex}}}{R} = b_1 \exp\left[-b_2 \left(T - T_{\min}\right)\right] \tag{4.5}$$

– `Left exp.2` (named as `LEFTEXP2` in `.dat` files) — exponential function for description of the left side of a peak (sum of the two terms).

$$\frac{C_p^{\text{ex}}}{R} = b_1 \exp\left[b_2 \left(T - T_{\max}\right)\right] + b_3 \exp\left[b_4 \left(T - T_{\max}\right)\right] \tag{4.6}$$

– `Right exp.2` (named as `RIGHTEXP2` in `.dat` files) — exponential function for description of the left side of a peak (sum of the two terms).

$$\frac{C_p^{\text{ex}}}{R} = b_1 \exp\left[-b_2 \left(T - T_{\min}\right)\right] + b_3 \exp\left[-b_4 \left(T - T_{\min}\right)\right] \tag{4.7}$$

– `Skewed` (named as `SKEWED` in `.dat` files) — function based on Gauss bell curve but with possibility of asymmetry.

$$\frac{C_p^{\text{ex}}}{R} = \frac{b_1 e^{-x^2}}{1 + e^{-b_2 x}};\ x = \frac{T - b_3}{b_4} \tag{4.8}$$

– `Empty` (named as `EMPTY` in `.dat` files) — function that is always equal to 0. Useful for just excluding heat capacity anomalies without their approximation.

– Custom terms written in Lua. See Chapter 3.4 for details.

## 4.3 Results window

Results window (see Figure 4.2) contains an optimized set of parameters. It allows to create reports, draw 2D plots and different tables using the optimized parameters.

A menu of the main window contains the next items:

- **File** — File input-output (exporting parameters to files)
  - **Save as CSV** — Exports optimized baseline parameters into `.csv` file readable by MS Excel and another programs.
  - **Save as HTML** — Exports optimized parameters and model errors into `.html` file that can be considered as a report suitable for printing.
- **Plots** — Creation of plots (thermodynamic functions, absolute deviations, relative deviations). Every new plot will be opened in a separate window.
  - **Cp function** — Plots for the $C_p(T)$ function.

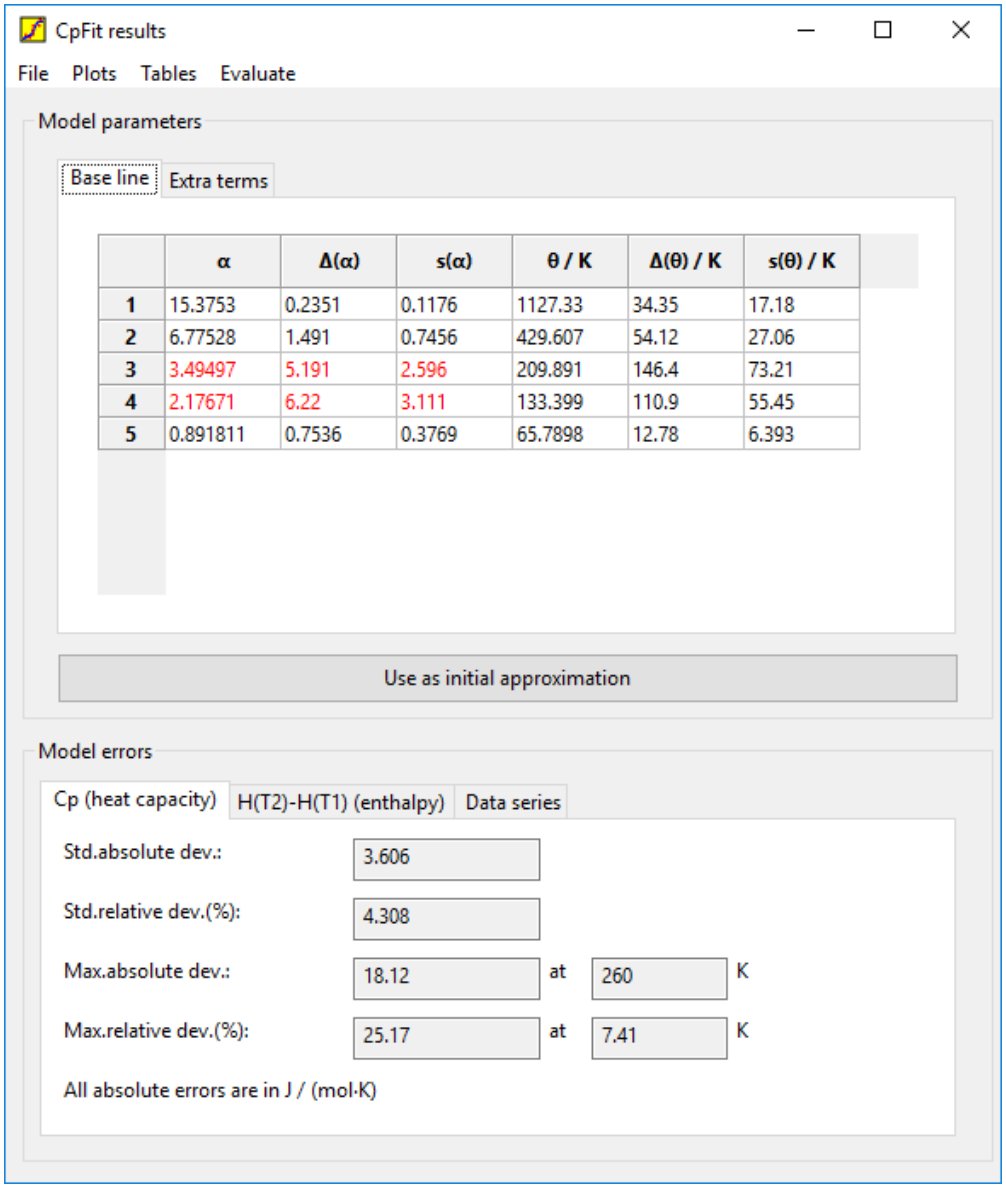


Figure 4.2: CpFit results window

  - **Cp/T function** — Plots for the $C_p(T)/T$ function.
  - **H function** — Plots for the $H(T) - H_0$ function. The next graphs are supported:
    * **H plot** — Plots $H_T - H_0$ function without experimental points.
    * **Absolute residuals** — Plots absolute residuals for all experimental points.
    * **Relative residuals** — Plots relative residuals for all experimental points.
    * **H(T)-H(T0) plot** — Plots a figure with both theoretical $H_T - H_{T_0}$ function and experimental data. Two forms are possible: $H_T - H_{T_0}$ function and $(H_T - H_{T_0}) / (T - T_0)$ function.
  - **S function** — Plots for the $S(T)$ function.

- **Tables** — Creation of tables with thermodynamic data in separate windows. All tables can be exported into `.csv` files.
  - **Thermodynamic functions** — Customizable table with tabulated thermodynamic functions values (and their confidence intervals). $C_p(T)$, $H(T) - H(0)$ and $S(T)$ functions are supported.
  - **Residuals** — Table with absolute and relative residuals for experimental points.
    * **Cp residuals** — residuals for $C_p(T)$ experimental points.
    * **H residuals** — residuals for $H(T_2) - H(T_1)$ experimental points.

- **Evaluate** — Evaluates some extra properties and values useful for management of optimization procedure.
  - **Low-T S correction** — Allows to evaluate deviation of isobaric heat capacity $C_p(T)$ from Debye function at low temperatures. Such estimation can be used for correction of entropy values.

Results window contains two areas: model parameters area and Cp model errors area.

*Model parameters area* contains the optimized parameters of the model. It contains two tabs: base line tab and extra terms tab:

- **Base line** tab contains $\alpha_i$ and $\theta_i$ parameters with their standard errors ($s(\alpha)$ and $s(\theta)$) and 95 % confidence intervals ($\Delta\alpha$ and $\Delta\theta$) calculated using eq 2.2 and eq 2.3 respectively. Statistically not significant terms are automatically highlighted in red colour. Each row correspond to one model term.

- **Extra terms** tab contains results of extra terms optimization. They are presented in the form of text report.

*Model errors area* contains information about accuracy of the optimized parameters data set present in the window. There are three tabs:

- **Cp (heat capacity)** — heat capacity data approximation uncertainties for all data series including points inside anomalies.

- **H(T2)-H(T1) (enthalpy)** — heat content data approximation uncertainties for all data series including points inside anomalies.

- **Data series** — data series approximation uncertainties. It is possible to see these uncertainties for each data series and for each data type separately. Separate consideration of data inside and outside anomalies is also supported.

for heat capacities and for enthalpies respectively. Note that all deviations are calculated without taking statistical weights $\omega_i$ from 2.1 into account (however they were used during the optimization).

### 4.3.1 Debye function

**Evaluate — Low-T S correction** menu item of *Results window* uses Debye model. In Debye model heat capacity is approximated by means of the next formula:

$$\frac{C_p^{\mathrm{D}}(T)}{3NRT} = 4D\left(\frac{T_{\mathrm{D}}}{T}\right) - \frac{3\frac{T_{\mathrm{D}}}{T}}{\exp\left(\frac{T_{\mathrm{D}}}{T}\right) - 1} \tag{4.9}$$

where $N$ is the number of atoms in the compound, $T_{\mathrm{D}}$ is Debye temperature, $D(x)$ is Debye function:

$$D(x) = \frac{3}{x^3}\int_0^x \frac{y^3}{e^y - 1}\,dy \tag{4.10}$$

The next algorithm of entropy correction calculation is used:

- Get the lowest trusted point $T_0$ and $C_p(T_0)$ at the curve (i.e. Einstein-Plank model). $T_0$ is manually set by user and usually is 5–30 K. The user also must give $N$ (number of atoms) value.
- Calculate Debye model parameters. Two options are available. The first one is evaluate only $T_{\mathrm{D}}$ value by solving $C_p^{\mathrm{D}}(T_0) = C_p(T_0)$ equation and use $N$ value given by a user. The second one is evaluate both $T_{\mathrm{D}}$ and $N$ (i.e. treat it as "effective" number of atoms) by solving the next system of equations:
$$\begin{cases} C_p^{\mathrm{D}}\left(T_0, T_{\mathrm{D}}, N\right) & = C_p(T_0) \\ \left.\frac{\partial C_p^{\mathrm{D}}(T,T_{\mathrm{D}},N)}{\partial T}\right|_{T_0} & = \left.\frac{\partial C_p(T)}{\partial T}\right|_{T_0} \end{cases} \tag{4.11}$$
- Calculate $S^{\mathrm{D}}(T_0)$ value using eq. 4.9. It is also possible to add virtual points to the input data to improve Einstein-Planck function sum behaviour at low temperatures.

## 4.4 Plots

CpFit 2-dimensional plot is a resizable window that can show experimental data points, analytical functions obtained by a nonlinear regression or both of them. They can also contain plots with absolute or relative deviations. Use results window to build new plots.

A menu of any 2D plot window in the CpFit program contains the next items:

- **File** — Export of the plot into graphic files.
  - **Save as** — Export into `.png` file with screen resolution.

Table 4.1: Terms colors in the $C_p(T)/T$ function plots.

| Term number | Color |
|---|---|
| 1 | Black |
| 2 | Blue |
| 3 | Green |
| 4 | Cyan |
| 5 | Red |
| 6 | Magenta |
| 7 | Brown |
| 8 | Gray |

- **Edit** — Figure customization and clipboard operations.
  - **Copy** — Copy the plot into Clipboard (in a vector format similar to Windows Metafile)
  - **Legend** — Enable/disable legend.
  - **Grid** — Enable/disable grid.

In the case of the $C_p(T)/T$ function plot extra dashed lines of different color that correspond to the function terms (see eq 1.1) are shown. A color of the line codes the term number (see Table 4.1).

You can also zoom in/zoom out plots using mouse wheel scrolling and drag-n-drop technique to navigate inside a zoomed plot.

Plots with experimental points can contain legend (it is enabled by default). It is formed by the next steps:

1. Read all data series IDs from the input `.dat` file and find unique values. Sort these values by alphabet.
2. Assign marker (i.e. color and form of dots) for each unique value obtained in the previous step.
3. Draw legend for data series that are present at the plot.

## 4.5 Tables

Two kinds of tables can be constructed by CpFit: tables with tabulated thermodynamic values and tables with comparison of calculated and experimental values. Both of them can be exported into `.csv` file format (readable by Microsoft Excel).

### 4.5.1 Tabulated thermodynamic functions

This kind of table contains tabulated values for $C_p(T)$, $H(T) - H(0)$ and $S(T)$ thermodynamic functions. Every new table contains rows for default $T$ values for $T = 10 \div 1000$ K and always contains a 298.15 K row. However a user can add or delete any rows with any step.

A menu of the tabulated thermodynamic functions table window contains the next items:

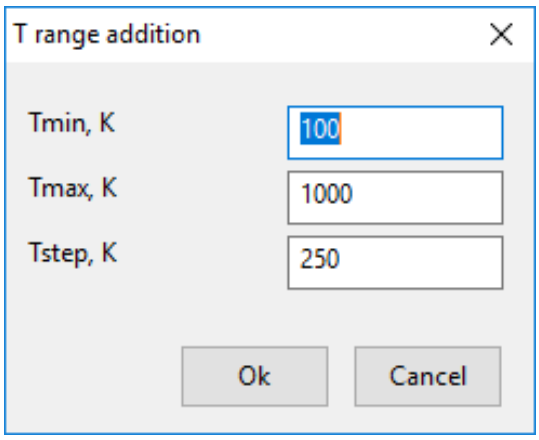


Figure 4.3: T range addition dialog

- **File** — File input-output (exporting table to files).
  - **Save as** — Export into `.csv` file.
- **Edit** — A customization of the table.
  - **Reset the table** — Removes all rows from the table.
  - **Add T range to the table** — Adds new rows for requested temperature range with requested step.

    When a user adds a new T range a special dialog shown in Figure 4.3 is opened. The user can set Tmin (minimal temperature), Tmax (maximal temperature) and Tstep (a step between new points). After the addition a table of tabulated values will be automatically sorted by the temperature.
  - **Errors estimation** — Allows to select the mode used for estimation of errors for tabulated $S$, $H$ and $C_p$ values.
    - **Standard deviation** — calculate standard deviations.
    - **95 % confidence interval** — calculate confidence intervals (95 % confidence intervals using two-sided $t$-distribution quantiles).

### 4.5.2 Calculated vs experimental

A menu of the residuals table window contains the next items:

- **File** — File input-output (exporting table to files).
  - **Save as** — Export into `.csv` file.

# Chapter 5

# Changelog

- 31 December 2018 — CpFit 0.8
  - Support of custom extra terms written in Lua. Examples of such scripts were added (see `eq_debpoly.lua`, `eq_eppoly.lua`, `natrolite_eppoly.dat`).
  - Editing of LSQ parameters lower and upper boundaries was added.
  - New "empty" extra term with zero value (useful for exclusion of anomalies).
  - Sorting of experimental points.
  - Uncertainties for data series inside and outside anomalies.
  - Improved procedure of Debye function based extrapolation. Generation of virtual points.
  - Improved source code documentation.
  - Bugfix: improved calculation of confidence intervals for $C_p$ and $S$ values (around 5%).
  - Bugfix: fixed number of points field in `K_natrolite.dat` (has no influence on the results of optimization)s.
- 25 December 2017 — CpFit 0.7
  - Four new extra terms were added
  - Now data files can have either `.dat` or `.txt` extension.
  - More informative message about exceeding the maximum number of iteration (the program now recommends to use another initial approximation)
  - Support of $\alpha_i$ and $\theta_i$ storage (i.e. initial approximations for them) was added into data files.
  - Some bugfixes
- 03 November 2017 — CpFit 0.6
  - Extra (excess) heat capacity terms support was added. Currently it is designed for description of lambda-transitions.
  - `UO2.dat`, `K_natrolite.dat`, `Tl_natrolite.dat` examples were added. They contain experimental data for $UO_2$, K-substituted natrolite and Tl-substituted natrolite respectively.

  - `ThO2_shifted15.dat` was added. It contains altered high-temperature $C_p$ data to improve description of high-temperature heat content.
  - Heat content data were added to `natrolite.dat` data file.
  - Extra data were addded to data file `ThO2.dat`.
  - Calories support in `.dat` files was added.
  - Heat content plots with experimental points were added. Two kind of plots are supported: $H_T - H_{T_0}$ plots and $(H_T - H_{T_0})/(T - T_0)$ plots where $T_0$ is set by a user.
  - Export of plots to `.wmf` and `.m` (MATLAB scripts) files was added.
  - Some bugfixes.

- 30 August 2017 — CpFit 0.5
  - Enthalpies ($H(T_2) - H(T_1)$ heat content) experimental data approximation support was added.
  - Support multiple data series with different markers was implemented.
  - Support of comments (ignored by CpFit) in `.dat` files was added.
  - Grid and legend management (possibility to enable or disable them) were added in 2D plots.
  - `ThO2.dat` example was added. It contains heat capacity and heat content (enthalpies differences) data for $ThO_2$. It also an example of data series names in the file.
  - Some bugfixes

- 05 July 2017 — CpFit 0.4
  - Errors in columns name (`s(H-H0)` and `s(S-S0)` names mismatch) in `.csv` files containing exported tables of thermodynamic functions was fixed.
  - Some typos in the manual were fixed
  - Bibliographic references were added to examples (`natrolite.dat`), (`ScF3.dat`).
  - Low-temperature entropy correction (by means of Debye function) was added.

- 21 December 2016 — CpFit 0.3
  - Automatic search of model terms number was added.
  - $t$-test for model coefficients (for statistical significance) was added.
  - Negative values of the model cannot be obtained during an optimization now.
  - Confidence intervals for parameters and tabulated values (using two-sided 95 % quantile of $t$-distribution).
  - Improved export of tabulated values.
  - Zooming for 2D plots.
  - Loading of model parameters from `.csv` files.
  - Estimation of confidence intervals for user-supplied parameters.

- 31 October 2016 — CpFit 0.2 — First public version

# Bibliography


[1] Gennady F. Voronin and Ilya B. Kutsenok. Universal method for approximating the standard thermodynamic functions of solids. *Journal of Chemical & Engineering Data*, 58(7):2083–2094, 2013.

[2] M.I.A. Lourakis. levmar: Levenberg-Marquardt nonlinear least squares algorithms in C/C++. [web page] `http://www.ics.forth.gr/~lourakis/levmar/`, Jul. 2004. [Accessed on 24 Aug. 2016.].

[3] G. K. Johnson, H. E. Flotow, P. A. G. O'Hare, and W. S. Wise. Thermodynamic studies of zeolites: natrolite, mesolite and scolecite. *American Mineralogist*, 68(11–12):1143–1145, 1983.

[4] I. E. Paukov, Yu. A. Kovalevskaya, Yu. V. Seretkin, and I. A. Belitskii. The thermodynamic properties and structure of potassium-substituted natrolite in the phase transition region. *Russian Journal of Physical Chemistry A*, 76(9):1406–1410, 2002.

[5] I. E. Paukov, Yu. A. Kovalevskaya, V. A. Drebushchak, and Yu. V. Seretkin. The thermodynamic properties of the thallium substituted natrolite form at low temperatures. *Russian Journal of Physical Chemistry A*, 79(12):1926–1930, 2005.

[6] Darrel W. Osborne and Edgar F. Westrum Jr. The heat capacity of thorium dioxide from 10 to 305°K. the heat capacity anomalies in uranium dioxide and neptunium dioxide. *The Journal of Chemical Physics*, 21(10):1884–1887, 1953.

[7] Smruti Dash, S. C. Parida, Ziley Singh, B. K. Sen, and V. Venugopal. Thermodynamic investigations of $ThO_2$–$UO_2$ solid solutions. *Journal of Nuclear Materials*, 393(2):267–281, 2009.

[8] C. Ronchi and J. P. Hiernaut. Experimental measurement of pre-melting and melting of thorium dioxide. *Journal of Alloys and Compounds*, 240(1):179–185, 1996.

[9] R. Agarwal, R. Prasad, and V. Venugopal. Enthalpy increments and heat capacities of $ThO_2$ and $(Th_yU_{(1-y)})O_2$. *Journal of Nuclear Materials*, 322(2):98–110, 2003.

[10] J. C. Southard. A modified calorimeter for high temperatures. the heat content of silica, wollastonite and thorium dioxide above 25°. *Journal of the American Chemical Society*, 63(11):3142–3146, 1941.

[11] Michael Hoch and Herrick L. Johnston. The heat capacity of aluminium oxide from 1000 to 2000° and of thorium oxide from 1000 to 2500°. *The Journal of Physical Chemistry*, 65(7):1184–1185, 1961.

[12] D. F. Fischer, J. K. Fink, and L. Leibowitz. Enthalpy of thorium dioxide to 3400 K. *Journal of Nuclear Materials*, 102(1):220–222, 1981.

[13] James J. Huntzicker and Edgar F. Westrum. The magnetic transition, heat capacity, and thermodynamic properties of uranium dioxide from 5 to 350 K. *The Journal of Chemical Thermodynamics*, 3(1):61–76, 1971.

[14] Fredrik Grønvold, Nils Jørgen Kveseth, Arvid Sveen, and Jiři Tichý. Thermodynamics of the $UO_{2+x}$ phase. I. Heat capacities of $UO_{2.017}$ and $UO_{2.254}$ from 300 to 1000 K and electronic contributions. *The Journal of Chemical Thermodynamics*, 2(5):665–679, 1970.

[15] C. Ronchi, M. Sheindlin, M. Musella, and G. J. Hyland. Thermal conductivity of uranium dioxide up to 2900 K from simultaneous measurement of the heat capacity and thermal diffusivity. *Journal of Applied Physics*, 85(2):776–789, 1999.

[16] Yoichi Takahashi and Masami Asou. High-temperature heat-capacity measurements on $(U, Gd)O_2$ by drop calorimetry and dsc. *Journal of Nuclear Materials*, 201(Supplement C):108–114, 1993.

[17] G. E. Moore and K. K. Kelley. High-temperature heat contents of uranium, uranium dioxide and uranium trioxide. *Journal of the American Chemical Society*, 69(9):2105–2107, 1947.

[18] D. R. Fredrickson and M. G. Chasanov. Enthalpy of uranium dioxide and sapphire to 1500 K by drop calorimetry. *The Journal of Chemical Thermodynamics*, 2(5):623–629, 1970.

[19] R. A. Hein, L. H. Sjodahl, and Szwarc R. Heat content op uranium dioxide from 1200 to 3100°K. *Journal of Nuclear Materials*, 25(1):99–102, 1968.

[20] L. Leibowitz, L. W. Mishler, and M. G. Chasanov. Enthalpy of solid uranium dioxide from 2500°K to its melting point. *Journal of Nuclear Materials*, 29(3):356–358, 1969.

# Chapter 6

# License

CpFit program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

## GNU GENERAL PUBLIC LICENSE

### Version 2, June 1991



51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA



### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## Terms and Conditions For Copying, Distribution and Modification

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

   Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   (a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   (b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

(c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   (a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   (b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   (c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

   The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

   If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

   It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

   This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program

under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

   Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

## No Warranty

11. Because the program is licensed free of charge, there is no warranty for the program, to the extent permitted by applicable law. Except when otherwise stated in writing the copyright holders and/or other parties provide the program "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with you. Should the program prove defective, you assume the cost of all necessary servicing, repair or correction.

12. In no event unless required by applicable law or agreed to in writing will any copyright holder, or any other party who may modify and/or redistribute the program as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use the program (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a failure of the program to operate with any other programs), even if such holder or other party has been advised of the possibility of such damages.

## End of Terms and Conditions

# Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> one line to give the program's name and a brief idea of what it does.
> Copyright (C) yyyy name of author
>
> This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
>
> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
>
> You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

> Gnomovision version 69, Copyright (C) yyyy name of author
> Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
> This is free software, and you are welcome to redistribute it under certain conditions;
> type 'show c' for details.

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w` and `show c`; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

> Yoyodyne, Inc., hereby disclaims all copyright interest in the program
> 'Gnomovision' (which makes passes at compilers) written by James Hacker.
>
> signature of Ty Coon, 1 April 1989
> Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.