ParSD

Software Documentation

Jens Fruhstorfer

Version 1.3

April 6, 2022

ParSD Software Documentation

Contents

Methodology Documentation: Background and theory **1**

1 Models independent on a minimum particle size . . . 2
- 1.1 Andreasen model . . . 2
- 1.2 Ψ model . . . 3
- 1.3 Kawamura model . . . 4

2 Models including a minimum particle size parameter . . . 5
- 2.1 Dinger and Funk model . . . 5
- 2.2 Modified Ψ model . . . 6
- 2.3 Modified Kawamura model . . . 6

Technical Documentation **7**

1 Main menu and database functions . . . 7
2 Design a batch . . . 10
3 Verify a batch . . . 14
4 Calculate model parameters . . . 18
5 Logfile and Debugging . . . 23

Installation Documentation **25**

1 Software requirements . . . 25
2 Installation notes . . . 25

End User Documentation **26**

1 Main window and settings . . . 26
2 Database functions . . . 29
- 2.1 Create new database . . . 30
  - Add first material . . . 31
- 2.2 Open existent database to remove, edit or add materials . . . 33
  - Remove material or complete database . . . 33
  - Edit material . . . 34
  - Add material . . . 38

3 Main functions . . . 39
- 3.1 Select raw materials from a database or recipe . . . 41
- 3.2 Set up batch properties . . . 41
  - Batch definition ('Verify a recipe' & 'Calculate model parameters' main functions) . . . 43
  - Bounds definition ('Design a batch' main function) . . . 43
- 3.3 Set up model . . . 45
  - Model definition ('Verify a recipe' & 'Design a batch' main functions) 45
  - Parameter bounds definition ('Calculate model parameters' main function) . . . 48
- 3.4 Calculations . . . 48
  - Batch optimization ('Design a batch' main function) . . . 48
  - Model optimization ('Calculate model parameters' main function) . . . 51
- 3.5 Results . . . 53
  - Results of 'Verify a recipe' main function . . . 53

Results and fine-tuning of 'Design a batch' main function 55
a) Results of 'Design a batch' main function 56
b) Fine-tuning of results of 'Design a batch' main function . . . 58
Results of 'Calculate model parameters' main function and their fine-tuning 61

GNU Free Documentation License **64**

Bibliography **71**

Methodology Documentation: Background and theory

For the processing and technological properties of materials, the particle size distribution plays a key role [LS94]. The impact is especially high on coarse grained materials like concrete or refractories. In fact, the particle size distribution has to be tailored according to the application needs [Fru19a; Fru21].

In general, there are two approaches: simulation and curve fitting. In the simulations, the initial properties of the raw materials are input and optimal curves for single properties like packing density, compressibility or viscosity are calculated. Interactions (like friction) between the particles are included in the simulation models. The calculations return a composition and a particle size distribution optimizing the target property. The second approach defines an 'ideal' final particle size distribution and the batch composition is calculated to fit this ideal curve optimally [LS94; Lar99; Fru21].

Both approaches have issues. The simulations lean heavily on preliminary investigations of the raw material properties [Fru19b; Fru21]. Moreover, as till today only friction-interactions were investigated, only target properties correlating with the packing density can be simulated [LS94]. Exemplarily, for a thermal shock resistance of refractories, where the bonding strength as well as the packing density and microstructure play a key role, the interactions between particles on this property will not be the same as for the target property packing density.

For the fitting to an 'ideal' curve, the drawbacks are that dependencies in reality are numerous. Models taking these dependencies into account, soon get complicated. If more simple models are chosen, after fitting to the ideal curve, subsequently an optimization by trial and error has to be done [Lar99; Fru19a]. An advantage is that you can always investigate how the parameters of a specific model influence a target property. From this, you can derive an ideal curve and also possible interactions. In this sense, investigating ideal curves can deliver some data and understanding required for the development of adequate simulation models for the target property [Fru21].

This software aims at supporting the second approach [Fru21]. As the significance of an ideal curve and the reciprocal number of necessary trial and error experiments somehow depends on the complexity of the curve model, a variety of models was developed to describe particle packings which can be generally parted into discrete and continuous model approaches [Ree95; MJ13]. In discrete particle size distributions, single particle fractions each of uniform size are packed. Particles of sizes between these fractions are not present and thus discrete packings are always gap-graded. In contrast, the continuous approach assumes that all possible particle sizes are present in the particle size distribution [MJ13]. As gap-grading increases the demixing (especially segregation) tendency, the continuous models are typically preferred against the discrete approaches regarding common applications [UB99; WIW89; FD92].

The most frequently used continuous particle size distribution models for ceramics are the Andreasen model and the Dinger and Funk model [Fru19a]. The models are defined as cumulative passthrough–functions ($CPFT(d)$ = cumulative percent finer than particle size d). Besides this type, particle size distributions are also frequently expressed by cumulative retention curves, relative passthrough or relative retention frequency curves. The latter are e.g. returned by sieve analyses. The relative frequency curves can be also referred to as increments of the corresponding cumulative distribution within the regarded particle size

Table 1: Comparison of passthrough and retention cumulative and relative frequency distributions for a alumina fraction of 0.5 to 1 mm [FA14]

Sieve/Component	Rel. frequency curves in %		Cum. frequency curves in %	
size in µm	Retention	Passthrough	Retention	Passthrough (CPFT)
90	0	0	100	0
150	0.2	0	100	0
315	15.49	0.2	99.8	0.2
630	79.22	15.49	84.31	15.69
1000	4.89	79.22	5.09	94.91
1250	0.2	4.89	0.2	99.8
2000	0	0.2	0	100
2500	0	0	0	100

interval. Table 1 shows for a fraction of tabular alumina how the four types would be expressed.

Both models, Andreasen's as well as Dinger and Funk's, have however the issue that a single distribution modulus is applied to describe the complete particle size distribution. Fine (< 0.1 mm [Rou97]), medium sized (0.1–1 mm [Rou97]) and coarse grain (> 1 mm [Rou97]) amounts are consequently in a fixed relationship and by changing the distribution modulus, the amounts of all grain sizes are changed. This is an issue because it is well documented [WH30; Ree95] that for a high density of the intended final material or product, distributions giving theoretically highest densities differ slightly from the distributions giving them practically. Generally, higher amounts of the coarse grains on costs of the medium sized grains are beneficial. By qualitatively similar changes of a distribution also lower density gradients in uniaxially pressed compacts [FA17; FHA19], lower water demands for castables [FA14], and increased microcracking [Fru18] improving the thermal shock resistance [LZK04; UB99] of refractories can be obtained or adjusted. For such distributions, Kawamura [KHK73] introduced the term 'continuous gap-graded' particle size distributions. Quantitatively, the extents differ of how the distributions should be manipulated to obtain best results for the specific target property—for every property a unique optimal curve might exist.

In the following sections, models included in the software are introduced together with standard values for different products/applications.

1 Models independent on a minimum particle size

Excluding the minimum particle size from the models means that a minimum particle size of zero length units is assumed which simplifies the models considerably. Often also minimum particle sizes cannot be measured adequately and an inclusion of an estimated minimum particle size might be as good as an estimation of a minimum particle size of zero length units.

Figure 1 shows the three models included in the software for comparison.

1.1 Andreasen model

The cumulative ($CPFT(d)$ = cumulative percent finer than particle size d) Andreasen model [AA30] shown in equation 1 is fully described by the maximum particle size $d_{\max}$ and a distribution modulus n_{And}. Standard values for a maximized density are listed in Table 2.

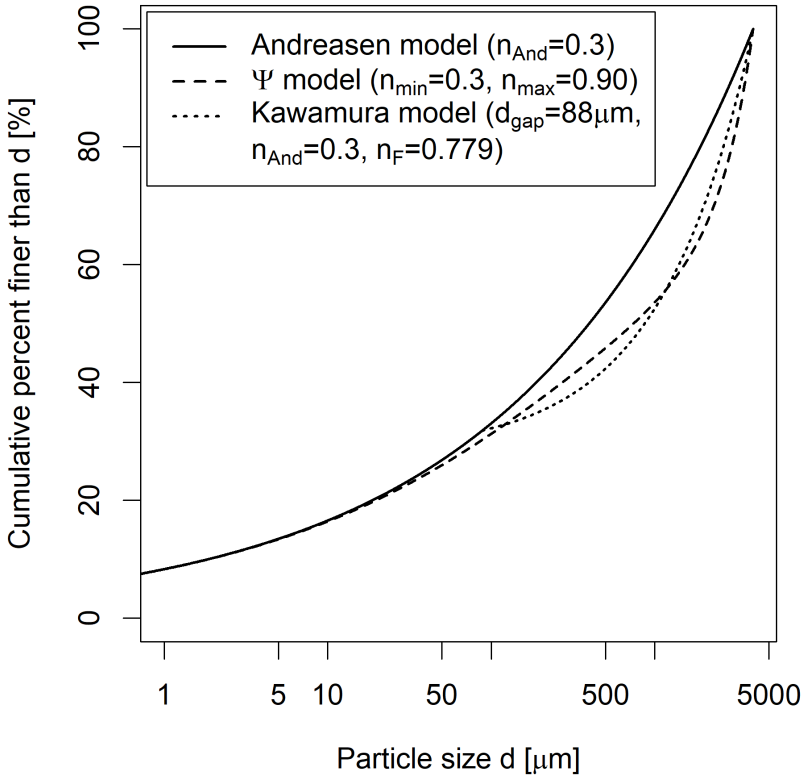


Figure 1: Comparison of the models included in the software which are independent on a minimum particle size (exemplary maximum particle size is 4 mm here)

Table 2: Standard values for the Andreasen model optimizing the packing density

Application	Maximum particle size in mm	Distribution modulus	Reference
Appolonian sphere packing	independent	0.53 *	[FHA19]
Bed of broken solids	3	0.33-0.66, 0.5	[AA30; UB99]
Refractory castables	≈ 2.5	0.22-0.26	[SSD06]
	4	0.2-0.25	[Myh08]
	6	0.28-0.3	[TU98]
Concrete (Aggregate size distributions)	independent	0.5	[Ful05]
	4.7	1.0-1.2	[TR23]
Pressing masses		> 0.5	[Ree95]
	1	0.9	[YWH00]

* Preset standard value in the software

$$CPFT_{\mathrm{And}}(d) = 100\,\% \cdot \left(\frac{d}{d_{\max}}\right)^{n_{\mathrm{And}}} \tag{1}$$

1.2 Ψ model

The Ψ model (equation 2) is a modified Andreasen model where the constant distribution modulus n_{And} is replaced by a modulus function $n_{\Psi}(d)$. Generally, several funtion types are possible, but it was found that a linear distribution modulus function (equation 3) serves the

Table 3: Standard values for the Ψ model optimizing the packing density

Application	Maximum particle size in mm	Min. dist. modulus	Max. dist. modulus	Reference
Refractory castables	3	≤ 0.28	0.8	[FA14]
	4	0.3	0.87	[Fru19b]
Concrete	8–63	0.28	0.28–1.2	[Fru19b]
Pressing masses	1-3	0.2 *	0.6 *,**	[FA17; FHA19]

* Preset standard values in the software

** Optimum maximum distribution modulus for 1 mm max. grain size was 0.6-1.0, for 3 mm it was 0.4-0.6. Intersection for 1-3 mm is 0.6.

purpose of application in concrete and ceramics well [Fru19a].

$$CPFT_{\Psi}(d) = 100\,\% \cdot \left(\frac{d}{d_{\max}}\right)^{n_{\Psi}(d)} \tag{2}$$

$$n_{\Psi}(d) = n_{\min} + d \cdot \frac{n_{\max} - n_{\min}}{d_{\max}} \tag{3}$$

The linear modulus function depends on the particle size d, on $d_{\max}$ and on two parameters, the minimum and maximum distribution modulus $n_{\min}$ and $n_{\max}$, respectively. With the latter two parameters, the amounts of the fine and coarse fractions can be adjusted separately [FA14; Fru19a]. The gap location and its extent are only adjusted indirectly and vary if $n_{\min}$, $n_{\max}$ or $d_{\max}$ are changed [FHA19]. Table 3 presents standard values maximizing the density.

In comparison to the Kawamura model, cf. Figure 1, the gap-grading character of the Ψ model is less pronounced [Fru19a].

1.3 Kawamura model

The approach by Kawamura *et al.* [KHK73] was to combine the models by Andreasen [AA30; Kaw73] and Furnas [Fur31; KAO71] in one model to achieve a higher freedom in designing a particle size distribution [Fru19a].

Kawamura's continuous gap–grading model applies for the fine fractions the Andreasen model $CPFT_{\mathrm{And}}(d)$ (cf. equation 1) up to a specified gap particle size d_{gap}. The associated $CPFT_{\mathrm{And}}(d_{\mathrm{gap}})/100\,\%$ value from which the gap-grading starts can be termed A. From there on up to $d_{\max}$, the Furnas-based model part is applied whose general form is described by $A + (1 - A) \cdot CPFT_{\mathrm{Fur}}(x)$. The Furnas part is described by equation 4 with the distribution modulus n_{F}. For $d < d_{\mathrm{gap}}$, the Andreasen part applies and for $d \geq d_{\mathrm{gap}}$, equation 4 applies [Fru19a].

$$\frac{CPFT_{\mathrm{Kaw,Fur}}(d)}{100\,\%} = \left(\frac{d_{\mathrm{gap}}}{d_{\max}}\right)^{n_{\mathrm{And}}} + \left[1 - \left(\frac{d_{\mathrm{gap}}}{d_{\max}}\right)^{n_{\mathrm{And}}}\right] \times \frac{d^{n_{\mathrm{F}}} - d_{\mathrm{gap}}^{n_{\mathrm{F}}}}{d_{\max}^{n_{\mathrm{F}}} - d_{\mathrm{gap}}^{n_{\mathrm{F}}}} \tag{4}$$

Standard values preset in the software: Kawamura *et al.* [Kaw77] investigated refractory castables with a maximum grain size $d_{\max} = 4\,\mathrm{mm}$. Highest densities were obtained for $n_{\mathrm{And}} = 0.3$, $d_{\mathrm{gap}} = 0.088\,\mathrm{mm}$ and $n_{\mathrm{F}} = 0.779 \approx 0.78$ [Fru19a].

In comparison to the Ψ model, cf. Figure 1, the gap-grading character of the Kawamura model is more pronounced [Fru19a].

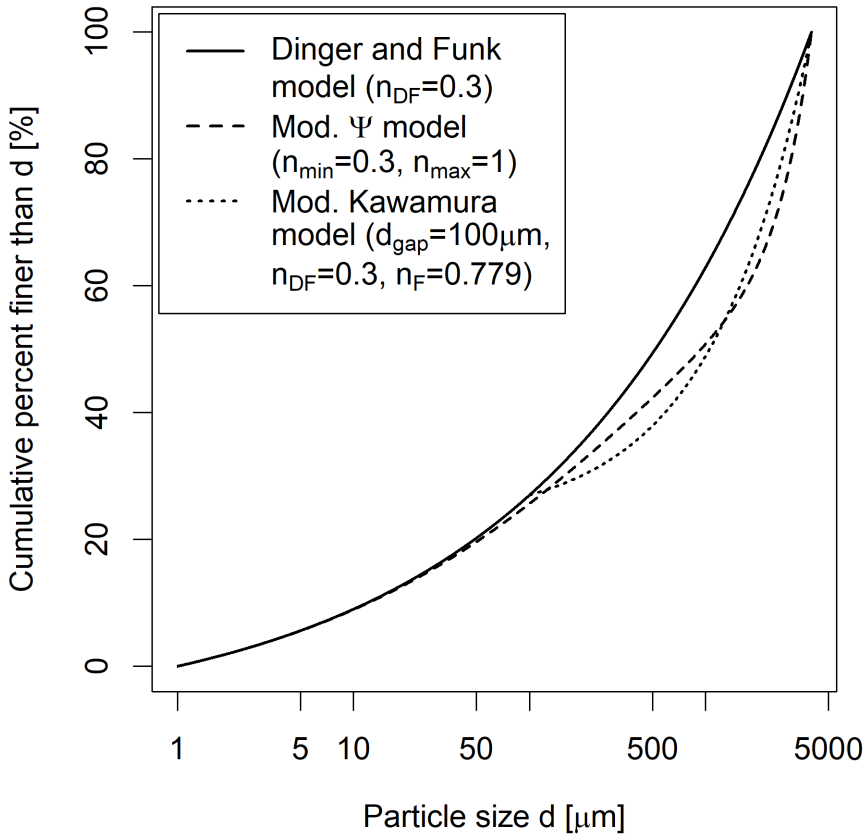


Figure 2: Comparison of the models included in the software which depend on a minimum particle size. The minimum particle size in the examples here is 1 µm and the maximum particle size 4 mm.

2 Models including a minimum particle size parameter

The minimum particle size gives more freedom for the design. Exemplarily, it is essential for the flowability of a concrete or castable because the viscosity increases strongly if nanometer sized particles are present [Fru16].

Figure 2 shows the three models included in the software for comparison.

2.1 Dinger and Funk model

In the model by Dinger and Funk [DF92] (equation 5), in addition to the maximum grain size and a distribution modulus n_{DF}, a minimum particle size d_{min} was introduced.

$$CPFT_{\mathrm{DF}}(d) = 100\,\% \cdot \frac{d^{n_{\mathrm{DF}}} - d_{\mathrm{min}}^{n_{\mathrm{DF}}}}{d_{\mathrm{max}}^{n_{\mathrm{DF}}} - d_{\mathrm{min}}^{n_{\mathrm{DF}}}} \tag{5}$$

Standard values: Dinger and Funk simulated numerically, that an optimal packing is achieved for $n_{\mathrm{DF}} = 0.37$ independent on the particle size range [DF93], which was chosen as preset standard value in the software. For slip cast as well as spray dried products with raw material particle sizes $< 100\,\mu\mathrm{m}$ they recommend a distribution modulus of 0.19-0.21 [FD94]. Although the Dinger and Funk model is frequently applied in industries like ceramics, refractories and concrete, the publications often lack a description of d_{min}—but d_{min} influences considerably the properties.

2.2 Modified Ψ model

The Ψ model can be modified to comprise a minimum particle size $d_{\min}$. The modification can be done by applying the modulus function $n_\Psi(d)$ (equation 3) to the model by Dinger and Funk replacing its constant modulus n_{DF}, cf. equation 6.

$$CPFT_{\Psi,\mathrm{mod}}(d) = 100\,\% \cdot \frac{d^{n_\Psi(d)} - d_{\min}^{n_\Psi(d)}}{d_{\max}^{n_\Psi(d)} - d_{\min}^{n_\Psi(d)}} \tag{6}$$

Regarding the freedom of design, this proposed model is on the same level like the Kawamura model with four parameters (including $d_{\max}$) to adjust. The adjustment of the gap-location is still only done indirectly by adjusting $n_{\min}$, $n_{\max}$, $d_{\min}$ and $d_{\max}$. However, it is known that e.g. by using finer particles in castables, the gap location moves towards smaller particle sizes [Hof92]. By reducing $d_{\min}$, the gap location will move towards smaller sizes in this model, but it cannot be adjusted to a defined value without changing also the other parameters [Fru19a].

The model was only recently applied. In one study [Fru19a], parameter values of $n_{\min} = 0.1$, $n_{\max} = 0.5$ and a correlated $d_{\min} = 0.32\,\mu\mathrm{m}$ were deduced from another study [Myh94] on refractory castables with a $d_{\max} = 3\,\mathrm{mm}$. However, in the base study [Myh94], especially the aggregate size distribution seems not optimal. Zienert *et al.* [Zie21] used the modified Ψ model to design a coarse-grained refractory ceramic-metal composite castable. The adjusted parameter set was $d_{\min} = 0.01\,\mu\mathrm{m}$, $d_{\max} = 3.15\,\mathrm{mm}$, $n_{\min} = 0.2716$ and $n_{\max} = 0.295$. These values are set as standard values in the software. However, the parameter values were influenced by the availability of the synthesized aggregate material and do not represent a parameter set giving special properties of the castables (like a high density).

In comparison to the Modified Kawamura model, cf. Figure 2, the gap-grading character of the Modified Ψ model is less pronounced [Fru19a].

2.3 Modified Kawamura model

To further increase the design freedom, Kawamura's model can be extended for a minimum particle size. This is possible by exchanging the Andreasen part of the model by the Dinger and Funk approach. For $d < d_{\mathrm{gap}}$, then, $CPFT_{\mathrm{DF}}(d)$, cf. equation 5, is applied whereas for $d \geq d_{\mathrm{gap}}$, equation 7 is used.

$$\frac{CPFT_{\mathrm{Kaw,mod,Fur}}(d)}{100\,\%} = \frac{d_{\mathrm{gap}}^{n_{\mathrm{DF}}} - d_{\min}^{n_{\mathrm{DF}}}}{d_{\max}^{n_{\mathrm{DF}}} - d_{\min}^{n_{\mathrm{DF}}}} + \left[1 - \frac{d_{\mathrm{gap}}^{n_{\mathrm{DF}}} - d_{\min}^{n_{\mathrm{DF}}}}{d_{\max}^{n_{\mathrm{DF}}} - d_{\min}^{n_{\mathrm{DF}}}}\right] \times \frac{d^{n_{\mathrm{F}}} - d_{\mathrm{gap}}^{n_{\mathrm{F}}}}{d_{\max}^{n_{\mathrm{F}}} - d_{\mathrm{gap}}^{n_{\mathrm{F}}}} \tag{7}$$

On the one hand, with this modified Kawamura model important points of particle size distributions known to date, are adjustable. On the other hand, the number of parameters which have to be adjusted is quite high. Extensive testing will be required to define ranges for the parameters—also in interaction with other parameters. For example, the optimal d_{gap} will decrease with decreasing $d_{\min}$ in accordance with Hoffman [Hof92].

Fruhstorfer *et al.* [Fru22] applied the modified Kawamura model to investigate the effect of the fines for uniaxially compacted ordinary refractory ceramics. The parameters d_{gap} and $d_{\max}$ were kept constant on $200\,\mu\mathrm{m}$ and $3.15\,\mathrm{mm}$, respectively. The remaining parameters were varied and their effect investigated. Highest bulk densities after sintering were achieved for the parameter set $d_{\min} = 0.09\,\mu\mathrm{m}$, $n_{\mathrm{DF}} = 0.27$ and $n_{\mathrm{F}} = 0.75$. These values are also the preset standard values in the software ParSD.

In comparison to the Modified Ψ model, cf. Figure 2, the gap-grading character of the Modified Kawamura model is more pronounced [Fru19a].

Technical Documentation

The application is scripted in R. It requires the non-base package 'tcltk' which is automatically downloaded and installed (if required) and then loaded.

The linear script flow is made user-friendly by 'Back'-menu-buttons which are based on turn-back (while) loops and nested functions. Therefore, the hub or linchpin of the software is the main window/main menu to which has to be returned before another main function can be started. In the main menu, furthermore, the settings are defined.

The three main functions of the application are the 'Design a batch' main function, the 'Verify a recipe' main function and the 'Calculate model parameters' main function. The functions are described by their intent, basic flow chart and description of the core subroutine more detailed in the following sections.

Additionally, for a running session a logfile is created which will be overwritten by a subsequent session. The logfile captures all the output (messages, errors) of the software and can be used for debugging. For a power-user, the logfile can also support to solve optimization errors.

1 Main menu and database functions

The main menu is the hub of the software as is also visualized in Figure 1. From the main menu, the three main functions are started, but also the database subroutines.

The database subroutines were implemented to complete the software. However, the databases (CSV files) can be handled also manually by a spread sheet software like LibreOffice Calc or Microsoft Excel. The database-layout (cf. Figure 2), hence, is the core of the database functions. In the database, the materials are listed in columns. The first column specifies the property or for the particle sizes the diameter of specified length units (specified in global DB settings). The particle size distributions of the materials are saved as relative retention frequency distributions in wt% as returned e.g. by a sieve analysis. Mandatory properties of the materials which have to be given are the identifier and the retention density distribution. Moreover, the identifier of the material has to be unique! Recommended is furthermore to give the density because otherwise only volume-calculations are possible. In fact, the grain bulk denity would be required, but for most applications a true density will suffice. When creating a new database with a spread sheet software, it has to be noted that the software ParSD cannot handle empty databases, but that always at least one raw material has to be contained in the database.

The database functions comprise conversion tools for different particle size distribution types (relative/cumulative retention/passthrough frequency curves), the possibility to define a raw material as a model, and importing functions. To be able to import distributions defined by a different set of particle sizes, the database functions apply linearization and averaging between specified particle sizes. Furthermore, the minimum and maximum particle size of the database can be filled out automatically if not specified by the user (this step is a prerequisite for the linearization). Therefore, it is mandatory that the raw materials to import are (1) defined completely meaning that the distribution runs from 0 % to 0 % for relative frequency curves or from 0 % to 100 % or vice-versa for cumulative frequency curves and that (2) the corresponding extremum particle sizes are within the span of the particle sizes in the database. Otherwise, obviously, the raw material cannot be included in a completely defined way. The

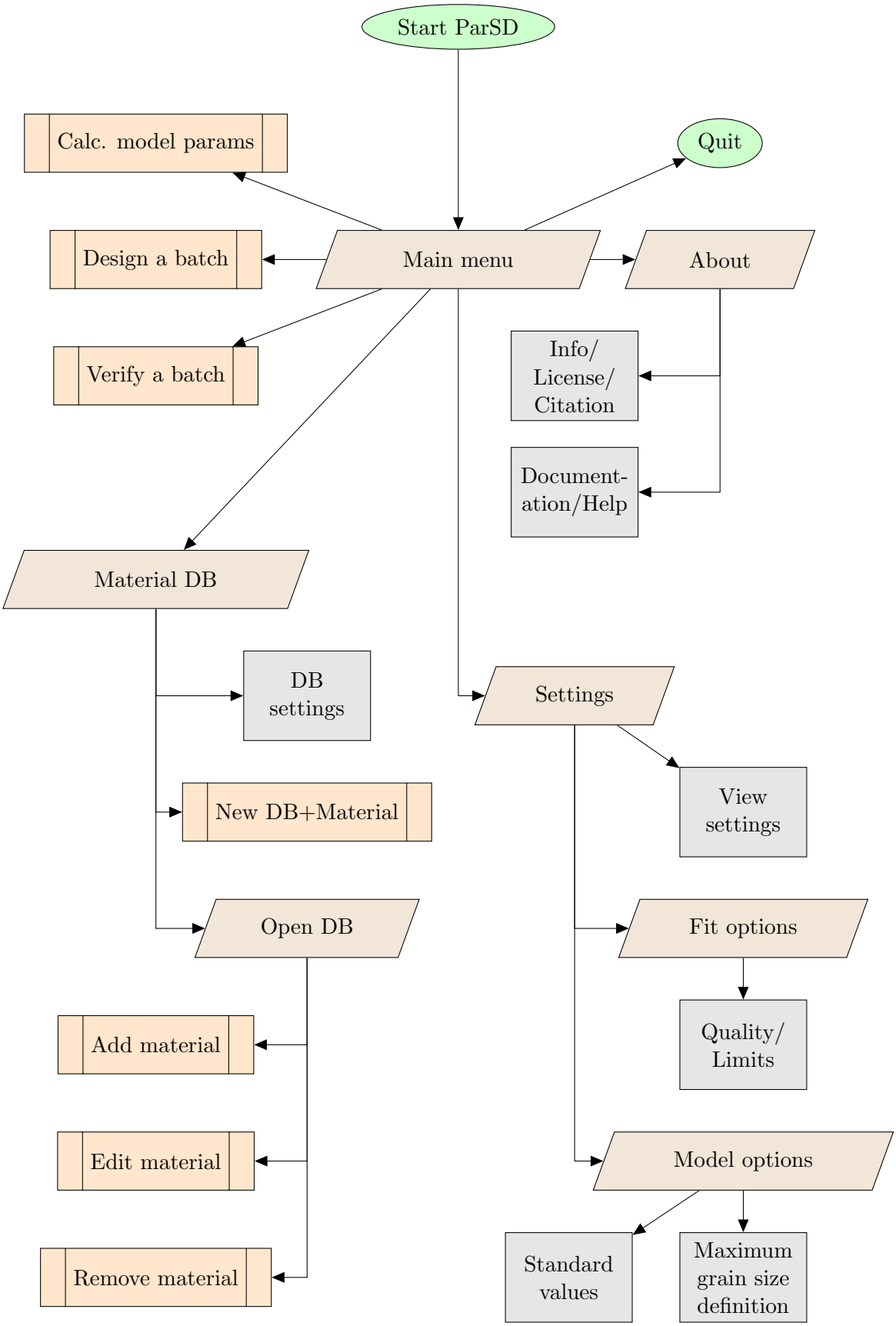


Figure 1: Flowchart comprising the main menu—the hub of the software (DB – database)

Identifier	T60_3000-1000	T60_1000-500	T60_500-0	T60_200-0	T60_20-0
Name					
Last modified					
Origin/Supplier					
Date supplied					
Price per MT					
True density	3,645	3,678	3,8	3,884	3,919
Date measured	2013	2013	2013	2013	2013
Measurement method	He-Pycnometry	He-Pycnometry	He-Pycnometry	He-Pycnometry	He-Pycnometry
Specific surface area					
Date measured					
Measurement method					
d10					
d25					
d50					
d75					
d90					
Date measured	2007	2007	2007	2007	2007
Measurement method					
0,001	0	0	0	0,49	2,42
0,1	0	0	0	3,47	16,98
0,4	0	0	0	3,01	14,7
1	0	0	0	5,53	22,4
4	0	0	0	7,4	16,4
10	0	0	0	22,9	19
40	0	0	0	21,9	1,3
90	0	0	22,82	19,5	1,3
150	0	0,2	24,49	14,8	5,45
315	0,07	15,49	48,78	1	0,05
630	2,76	79,22	3,91	0	0
1000	6,6	4,89	0	0	0
1250	32,89	0,2	0	0	0
2000	25,53	0	0	0	0
2500	28,84	0	0	0	0
3150	3,31	0	0	0	0
4000	0	0	0	0	0

Figure 2: Layout of a database

same applies if the user inputs the distribution data manually. In the example of Table 1 in the Methodology Documentation on page 2, a material given by a sieve analysis (relative retention frequency curve) must be given for the particle sizes from 90 µm to 2 mm to run from 0 % to 0 % and the database also requires particle sizes from $\leq$ 90 µm to $\geq$ 2 mm. If the material, however, would be given as relative passthrough frequency curve, then the extremum particle sizes would be 150 µm and 2.5 mm to allow for a complete definition.

For the import of a material, a CSV raw material file has to be prepared which contains two columns with the exact labels 'Diameter' and 'Percent' with these labels situated in line one. The subsequent lines have to contain the relevant data without missing values and not more. The file can have more columns and also the column numbers for the mandatory 'Diameter' and 'Percent' columns are allowed to vary. Figure 3 shows a fictive example for a test material to be imported as a cumulative pass-through curve into a database spanning particle sizes from ≤ 1 to ≥ 10 length units.

	A	B	C
1	Diameter	Percent	
2	1	0	
3	4	20	
4	7	70	
5	10	100	
6			

Figure 3: Example for a prepared raw-material file for importing as cumulative pass-through curve

2 Design a batch

To design a batch is probably the most important function of the software. As seen in the flowchart in Figure 4, firstly, the materials are put together from recipes and/or a database and bounds are allowed to set. The bounds can be set in vol% or wt% if all selected raw materials have a density saved in the database. Secondly, the model is chosen and defined. If it was returned from the optimization results to the model selection or before, the user is asked if he wants to keep the model from the last run (not shown in flowchart). Thirdly, the optimization results are presented and can be plotted and saved as well. A special feature is the possibility to fine-tune the data as e.g. one might want to round the material amounts to whole numbers/integers.

The core routine (cf. algorithm in Listing 1) after setting the model parameters uses for an optimization without bounds or with bounds in vol% the Nonlinear Least Squares nls()-function with the port-algorithm. The auxiliary condition, that the calculated optimal amounts of all raw materials add up to 100 %, is forced by a volume iteration. If bounds in wt% were given, besides the volume iteration, it is iterated optimizing the theoretical density of the batches. During every iteration step, the nls()-function with the port algorithm is applied (for the vol%), returning a batch for which a theoretical density can be calculated. The approach cannot fit directly the masses because the particle size distribution models are volumetric ones. In case wt% bounds are given, the vol% bounds are preset to 0 for the lower bounds and to 1 for the upper bounds. Then during the iteration, the volume bounds are adjusted the way, that the calculable mass% move into the range of the given mass%-bounds.

Listing 1: Algorithm of the design-function

```
design <- function (Materials, preparation, VolumeModel, densities,
LowerBoundsVolume, UpperBoundsVolume, LowerBoundsMass,
UpperBoundsMass, limdesign, accuracy) {

#Materials: List of selected Materials
#preparation: data frame contains column for diameters and columns
#             for Materials as cumulative passthrough curves (CPFT)
#VolumeModel: vector of defined CPFT-model in vol% with values for
#             every diameter
#densities: vector contains densities of Materials
#limdesign, accuracy: limit of iterations and accuracy (decimal
#                     places) for calculations defined by the user
#                     in the settings
#LowerBoundsVolume, UpperBoundsVolume: Specified vol%/100% bounds,
#                                      if not specified 0 for lower
#                                      bounds and 1 for upper ones
#LowerBoundsMass, UpperBoundsMass: Specified mass%/100% bounds, if
```

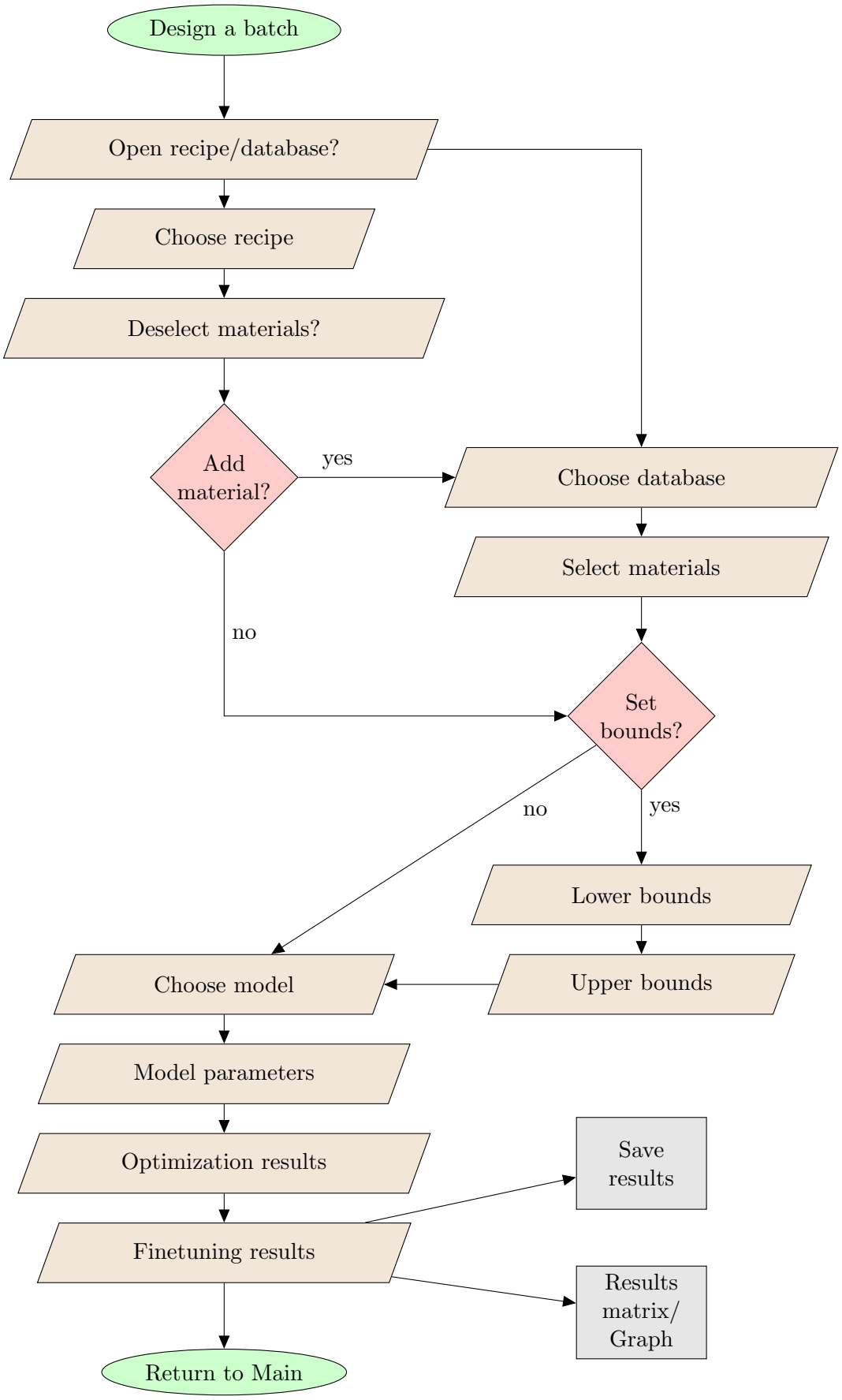


Figure 4: Flowchart for the 'Design a batch' main function

```
#                                    not specified 0 for lower bounds
#                                    and 1 for upper bounds

 preparation <- cbind(preparation, VolumeModel)#add vol% model to
                                                #dataframe as column

 run <- 0 #iteration step of normalization to 100 vol%
 runx <- 0 #iteration step of mass% calculation

 mcheck <- 0 #evaluation variable for mass% calculation

 while (mcheck!=1) { #mass% calculation-iteration

  if (run+runx > limdesign) { #limit reached
   #return to: model, bounds, material selection or main menu
  }

  check <- 0 #evaluation variable for normalization to 100 vol%

  while (check!=1) { #normalization to 100vol%-iteration

   if (run+runx > limdesign) { #limit reached
    #return to: model, bounds, material selection or main menu
   }

   fit <- nls(VolumeModel~Coefficients*Materials, data=preparation,
   start=WithinBounds, algorithm="port", lower=LowerBoundsVolume,
   upper=UpperBoundsVolume)#returns coefficients equivalent to vol%
                           #of the Materials in the batch

   if (is.null(fit)) { #nls()-error
    #return to: model, bounds, material selection or main menu
   }

   check = sum(coef(fit)) #Auxiliary condition, that coefficients
                          #(vol%) add up to 1 (100%) -> check if
                          #it does is done in the condition of the
                          #corresponding while-loop

   preparation <- rbind(preparation, c(100,...))#adds a row of 100s
                                                #to preparation,
                                                #giving more focus
                                                #in the next step
                                                #of the iteration
                                                #on the 100vol%

   run=run+1

  } #end while loop of normalization to 100vol%-iteration

  Den <- 0 #Calculate Batch density
```

```
  for (n in seq(from=1, to=length(Materials), by=1)) {
  Den <- Den+coef(fit)[n]*densities[n] }

  masses <- vector(length=length(Materials)) #calculate mass% of
                                             #Materials in batch
  for (o in seq(from=1, to=length(Materials), by=1)) {
   masses[o] <- coef(fit)[o]*densities[o]/Den
  }

  mcheck <- 1
  for (p in seq(from=1, to=length(Materials), by=1)) {#check if
                                                      #mass% are
                                                      #within set
                                                      #mass%-
                                                      #bounds. If
                                                      #too low,
                                                      #increase
                                                      #by arbitrary
                                                      #amount
                                                      #1/10^();
                                                      #if too high,
                                                      #decrease by
                                                      #this amount
   if (masses[p] < LowerBoundsMass[p]) {
    LowerBoundsVolume[p] = LowerBoundsVolume[p] + 1/10^(accuracy+2)
    mcheck=mcheck+1
   } else if (masses[p] > UpperBoundsMass) {
    UpperBoundsVolume[p] = UpperBoundsVolume[p] - 1/10^(accuracy+2)
    mcheck=mcheck+1
   }
  }

  for (r in seq(from=1, to=length(Materials), by=1)) { #move bounds
                                                       #if they
                                                       #catch up
                                                       #with each
                                                       #other
   if (LowerBoundsVolume[r] > UpperBoundsVolume[r]) {
    if (masses[r] < LowerBoundsMass[r]) {
     UpperBoundsVolume[r] <- LowerBoundsVolume[r]
    } else if (masses[r] > UpperBoundsMass[r]) {
     LowerBoundsVolume[r] <- UpperBoundsVolume[r]
    }
   }
  }

  runx=runx+1

 } #end while-loop of mass% calculation-iteration

#show optimization results:
```

```
#coef(fit) contains vol% for Materials
#masses contains mass% for Materials

} #end design-function
```

The applied 'port' algorithm of the nls()-function is the 'NL2SOL' algorithm based on the FORTRAN PORT library [Por]. The NL2SOL uses a model/trust-region method along with an adaptive choice of the model Hessian [DGW81a]. For problems with large number of residuals (as can happen for a high number of component sizes), NL2SOL is more reliable than Gauss–Newton ot Levenberg–Marquardt method and more efficient than the secant or variable metric algorithms (used for general function minimization) [DGW81b].

Regarding troubleshooting, the software returns an error message and suggestions what to do if the optimization is stopped due to singularities or non-convergence. These suggestions comprise giving different or no bounds or selecting different raw materials. On the other hand it is possible that there are rounding issues and that an optimization would run forever. In the settings (Main window), therefore, the user can set a limit for the number of iterations comprising both volume and wt% iterations. It could happen, that a result is faster received if in a first run, it is optimized without bounds and in a second run, bounds are set or instead of the second run directly the Main function 'Verify a batch' is used to manually check for the optimum amounts. It should be also mentioned that the software has no possibility to check if a found solution is a local or a global optimization minimum. For this, besides or together with an educated evaluation of the solution by the End User, runs with different starting values (adjusted by the parameter bounds) are the only possibility to check if there is another solution with a lower sum of squared deviations.

3 Verify a batch

The main function 'Verify a batch' comprises the functionality of the software EMMA by Elkem [Mat]. Figure 5 shows the corresponding flowchart. In difference to the 'Design a batch' and 'Calculate model parameters' main functions, to verify a batch, no curve fittings and iterations are required but the user defines firstly the materials and their amounts and secondly the model with its parameters. In the results dialog, batch and model can be compared. By making use of the 'Back to...' menu bar entries, different parameter sets and/or amterial combinations can be manually tested and compared.

Except for the 'Back to...' menu bar entries which are implemented by while()-loops and nested functions shown exemplarily in Listing 2, there is no core function requiring more detailed description.

Listing 2: Algorithm for the back-buttons

```
openmat <- function(fun, rec) {
 #fun = 3 verify a batch
 #rec = open database or recipe

 #choose and read db/recipe depending on rec-value
 #select materials
 #save information in vectors, like densities
 #prepare data frame 'preparation' with first column diameters
 #  and then the selected material-CPFT's

 if (fun==3) { #Verify recipe
  error <- batch(Materials,preparation,densities,fun)
 }
```

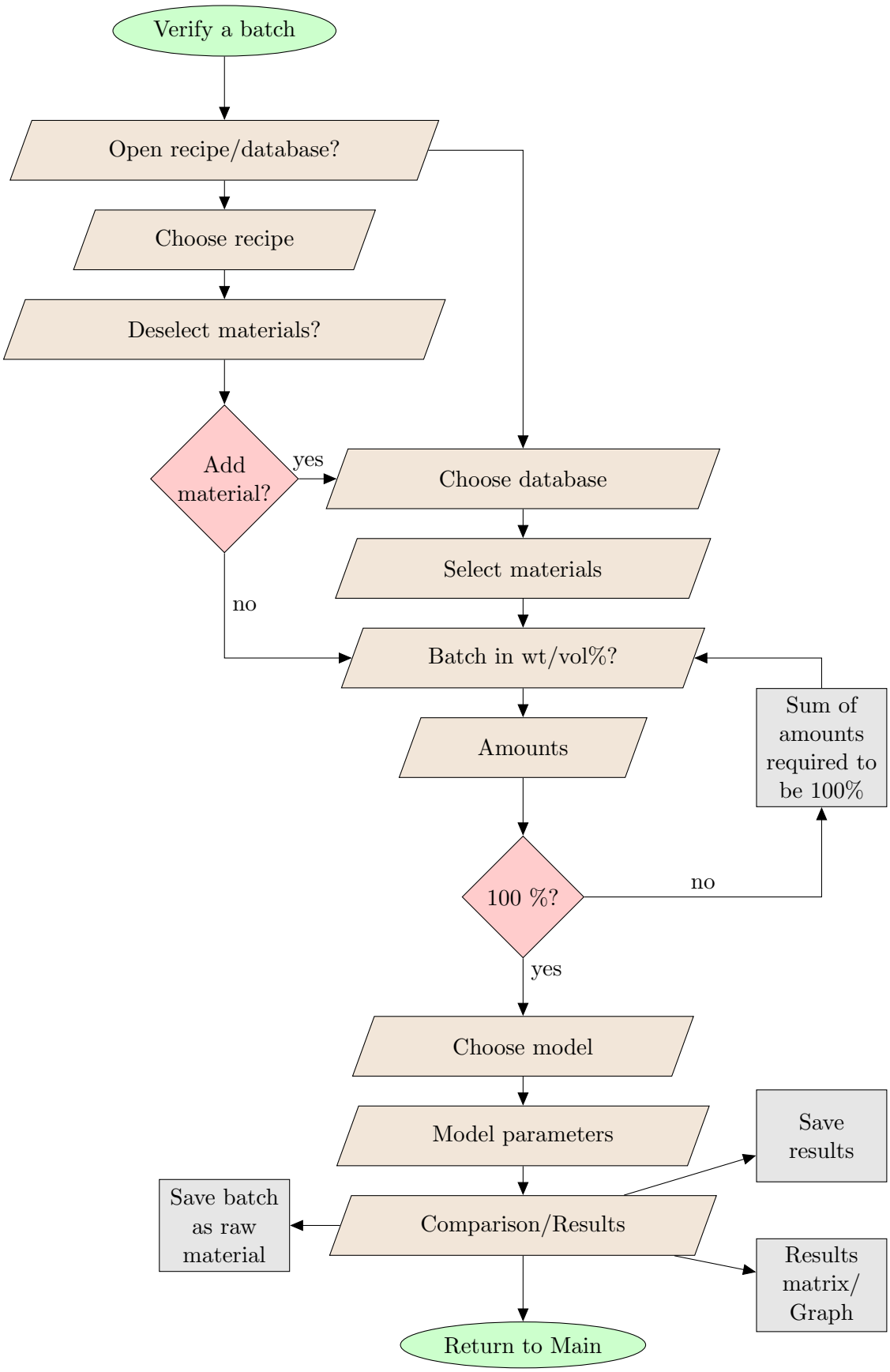


Figure 5: Flowchart of the 'Verify a batch' main function

```
 return(error) #values: 0 quit, 1 main menu, 2 open recipe/database
} #end openmat-function


batch <- function (Materials,preparation,densities,fun) {
 while (TRUE) { #batch definition-loop

  #choose batch type-window (vol% or mass%)
  #tcl-variable cm, initialized with 0
  #cm <- 1 back to main menu, cm <- 2 back to material selection,
  #cm <- 3 ma%, cm <- 4 vol%
  #wait till cm non-zero and assign to var:
  var <- tclvalue(cm)

  if (var < 3) {
   return(var) #values: 0 quit, 1 main menu, 2 material selection
  } else if (var==3) {
   #set ma% amounts: boundsMass
   #and calculate vol% amounts boundsVol with densities
  } else if (var==4) {
   #set vol% amounts boundsVol
   #and calculate ma% amounts boundsMass with densities
  }

  if (100*sum(boundsVol)!=100 || 100*sum(boundsMass)!=100) {
   #message that batch has to add up to 100%
   next #starting over with batch definition-loop
  }

  if (fun==3) {
   value <- modelver(Materials,preparation,densities,boundsMass,
   boundsVol)
  }

  if (value < 3) { return(value) }
  #return values: 0 quit, 1 main menu, 2 material selection

 } #while loop bounds
} #end batch-function


modelver <- function(Materials,preparation,densities,boundsMass,
boundsVol) {
 while (TRUE) {

  #choose model-window; tcl-variable cm (init 0)
  #cm <- 1 main menu, cm <- 2 material selection, cm <- 3 batch
  #  definition, cm <- 4-10 the different models
  #wait till cm non-zero, then assign to var:
  var <- tclvalue(cm)
```

```
  if (var < 4) { return(var) }
  #return values: 1 back to main menu, 2 back to open recipe/
  #  database, 3 back to batch definition

  if (var == ...) {
   #define and calculate VolumeModel (as global variable)
  }

  valmod <- verify(Materials, preparation, densities, boundsMass,
  boundsVol)

  if (valmod < 4) { return(valmod) }
  #return values: 0 quit, 1 back to main menu, 2 back to open
  #               recipe/database, 3 back to batch definition,
  #               4 back to model selection

 } #while loop model selection
} #end modelver-function


verify <- function (Materials, preparation, densities, boundsMass,
boundsVol) {
 #add Volume Model to preparation data frame
 #save preparation in tuning
 #calculate batch density

 while (TRUE) {
  #show results-window; tcl-variable done <- 0
  #menu: save-operations
  #done <- 1 main menu, done <- 2 material selection,
  #done <- 3 batch definition, 4 model selection
  #wait till done non-zero, assign to don:
  don <- tclvalue(done)

  if (don < 5) { return(don) }
  #return values: 0 quit, 1 back to main menu, 2 back to open
  #               recipe/database, 3 back to batch definition,
  #               4 back to model selection, 5 show results again
 } #while loop results-window
} #end verify-function


#---MAIN WINDOW (no function)---
while (TRUE) { #main window while loop

 #init main window with tcl-variable app with start value 0
 #destroy event and 'Quit' menu bar button asign value 1 to app
 #functions asign other values, verify-function asigns value 3
 #the window is kept alive till app is non-zero, meaning, till
 #  something to do was chosen by user
```

```
#then the value of the tcl variable app is asigned to a common
#  variable var1:
var1 <- tclvalue(app)

if (var1 == 1) {
 #save options and settings
 break #get out of main window-loop == exit software
} else if (...) { #other functions including database functions,
                  #except of main functions 'design a batch',
                  #'verify a batch' and 'calc. model parameters'
} else { #main functions
 while (TRUE) { #open recipe/database window
  #open dtabase or recipe?
  #tcl-variable db initialized with 0
  #back to main db <- 1
  #open database or recipe db <- 2 or 3
  #wait for decision until db non-zero
  #then the value of db is asigned to a common variable var2:
  var2 <- tclvalue(db)

  if (var2 == 1) {
   appdb <- 1
   break #get out of open recipe/database-loop, meaning:
         #return to main menu
  }

  appdb <- openmat(fun=var1, rec=var2)

  if (appdb == 0) { break } #return to main menu and then exit
  if (appdb == 1) { break } #return to main menu

 } #while db vs recipe
} #end else - main functions

if (appdb == 0) {
 #save options and settings
 break #get out of main window-loop == exit software
}
} #end while main window loop
```

4 Calculate model parameters

The calculate model parameters main function is the opposite of the design a batch function. The user defines a batch, chooses a model and possible model parameter bounds and then it is simulated for which parameters the model fits best to the defined batch, cf. flowchart in Figure 6.

This is without issue, if the particle size parameters are defined (lower bound = upper bound). Then the curve-fitting function nls() can be used to directly simulate the optimal remaining parameters. However, if the particle size parameters also have to be modelled, then, iterated curve fitting has to be done. Moreover, in case of the Kawamura and modified Kawamura model, there are two parts of the formula (cf. subsections 1.3 and 2.2 of the

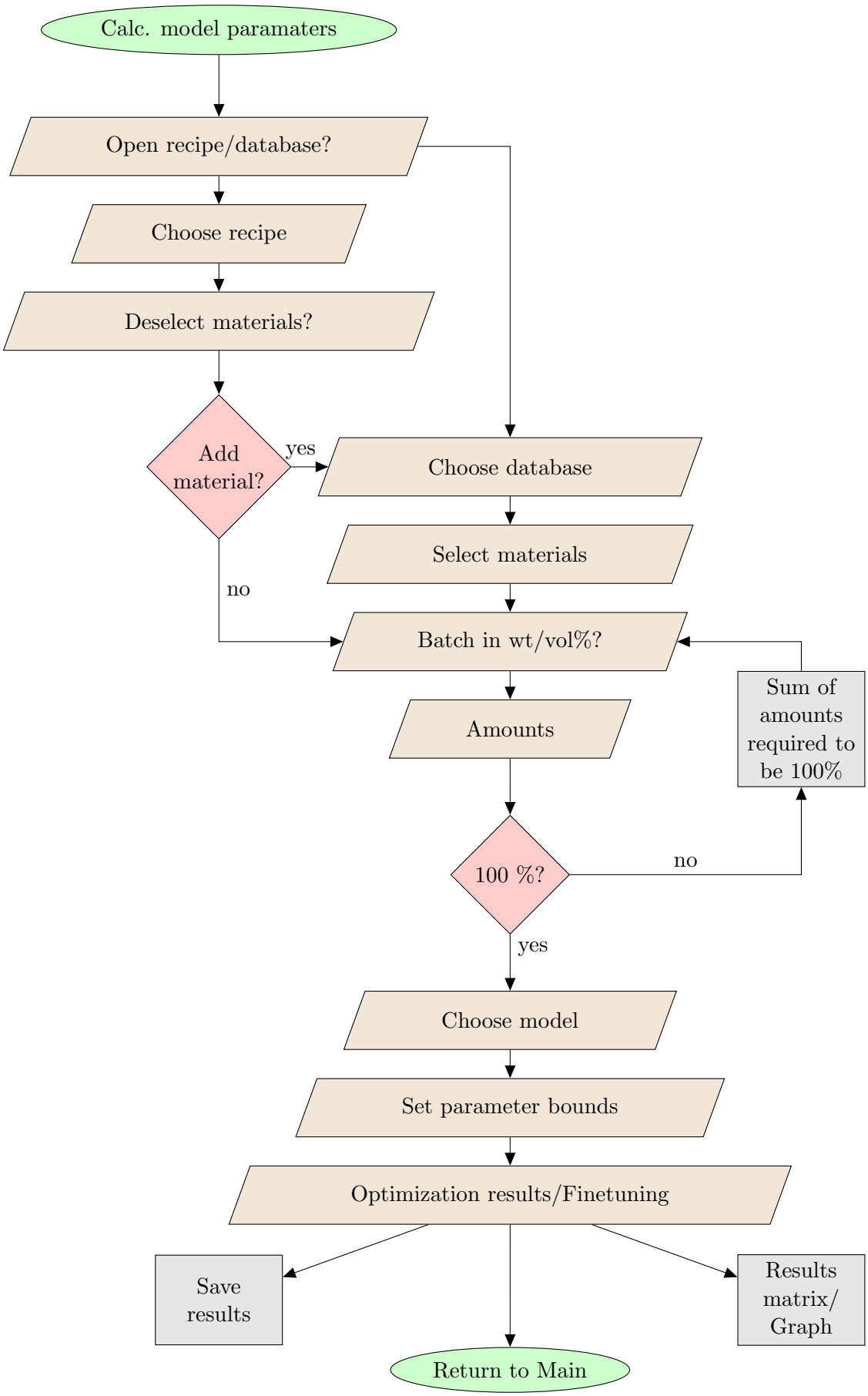


Figure 6: Flowchart of the 'Calculate model parameters' main function

Methodology Documentation) divided by a gap particle size. In such a case, multiple iterated nls()-curve fitting were required to enable finding an optimal parameter set. The listing 3 presents the algorithm on the example of the simple Andreasen model.

The nls()-function is used with the port algorithm (cf. section 2 on the 'Design a batch' main function). However, as especially for the Kawamura model and the modified Kawamura model, iterations are manifold, the calculate model parameters function sometimes aborts. Possibilities are besides normal non-convergences, that for the particle sizes a local minimum was found instead of a global minimum within the given range of bounds for this particle size. To solve this, but also to check in case of a found solution if it is the global minimum, the only possibility is to test different parameter bounds.

Listing 3: Algorithm for the 'Calculate model parameters' main function

```
calcparams <- function (Materials, preparation, densities, Model,
BoundsMass, BoundsVolume, LowerBoundsSizes, UpperBoundsSizes,Batch,
LowerBoundsModuli, UpperBoundsModuli, limparcalc, accuracy, Den) {

#Materials: List of selected Materials
#preparation: data frame contains column for Diameters and columns
#             for Materials as cumulative passthrough curves (CPFT)
#Model: Chosen model
#densities: vector contains densities of Materials
#limparcalc, accuracy: limit of iterations and accuracy (decimal
#                     places) for calculations defined by the user
#                     in the settings
#BoundsVolume, BoundsMass: Defined batch
#LowerBoundsSizes, UpperBoundsSizes: Specified bounds for the
#                                    particle size parameters of
#                                    the chosen Model
#LowerBoundsModuli, UpperBoundsModuli: Specified bounds for the
#                                      distribution moduli of the
#                                      chosen Model
#Den: batch density
#Batch: Cumulative sum curve of batch

 nacount <<- 0 #counter for failed nls-fittings

 preparation <- cbind(preparation, Batch) #add batch to dataframe

 run <- 0 #Iteration step of particle size calculation
 step <- 0 #Refining step of single iteration step

 if (Model == "Andreasen model") {

  dmaxl <<- LowerBoundsSizes   #Initialize running variables
  dmaxu <<- UpperBoundsSizes   #required for fitting particle size
  dmaxmin <<- LowerBoundsSizes #dmax

  while (TRUE) { #while loop to find optimal dmax

   if (run > limparcalc) { #limit reached
    #return to: model, batch def., material selection or main menu
   }
```

```
run <- run + 1
step <- 1

if (dmaxl == dmaxu) { #Calculate final distribution moduli when
                      #fitting of particle size dmax finished

 dmax <- dmaxmin #Initialze dmax with optimum value

 #prepare sub-dataset which ends at dmax for fitting:
 modelsub <- subset(preparation, preparation$Diameter <= dmax)

 fit <- nls(Batch ~ 100*(Diameter/dmax)^Vn , data=modelsub,
 start=WithinBounds, algorithm="port", lower=LowerBoundsModuli,
 upper=UpperBoundsModuli) #returns distribution modulus (Vn)
                          #stored in coef(fit)

 if (is.null(fit)) { #nls()-error
  #error mesage window
  #return to: model, bounds, material selection or main menu
 }

 #Calculate VolumeModel for chosen Model with fitted value
 #saved in coef(fit)
 for (i in seq(from=1,to=length(Materials),by=1)) {
  if (dmax < preparation$Diameter[i]) { VolumeModel[i] <<- 100 }
  else { VolumeModel[i]<<-100*(preparation$Diameter[i]/
                                        dmax)^coef(fit)[1] }
 }

 break #after dmax iterated and distribution modulus fitted,
       #while loop is left

} else { #dmaxl != dmaxu, fitting of dmax not finished

 #Initialize logarithmic steps for dmax within bounds of last
 #run <-> dmaxl and dmaxu
 dmax <- vector(length=4)
 dmax[1] <- dmaxl
 dmax[2] <- 10^(log10(dmaxl)+(log10(dmaxu)-log10(dmaxl))/3)
 dmax[3] <- 10^(log10(dmaxu)-(log10(dmaxu)-log10(dmaxl))/3)
 dmax[4] <- dmaxu

 sumerrsq <- rep(0, length(dmax)) #Evaluation variable for dmax:
                                  #sum of squared deviations

 #For every dmax of the dmax-steps, a distirbution modulus to
 #give optimal results is fitted. Then, the calculable model
 #and given batch are compared and the deviations characterized
 #by the sum of squared deviations per component size are
 #calculated for every dmax of the dmax-steps:
```

```
for(j in seq(from=1, to=length(dmax), by=1)) {
 step <- j

 modelsub<-subset(preparation, preparation$Diameter <= dmax[j])

 fit <- nls(Batch ~ 100*(Diameter/dmax[j])^Vn , data=modelsub,
 start=WithinBounds, algorithm="port", lower=LowerBoundsModuli,
 upper=UpperBoundsModuli) #optimal dist. modulus (Vn) for
                          #specific dmax-step

 if (is.null(fit)) { #nls()-error
  sumerrsq[j] <- NA
  nacount <<- nacount+1
  next #going on with next dmax-step
 }

 #create VolumeModel-column and calculate sum of squared dev.:
 errsq <- vector(length=length(preparation$Diameter))
 for (i in seq(from=1,to=length(preparation$Diameter),by=1)) {
  if (dmax[j]<preparation$Diameter[i]) {VolumeModel[i] <<- 100}
  else { VolumeModel[i] <<- 100*(preparation$Diameter[i]/
                                 dmax[j])^coef(fit)[1] }

  errsq[i] <- ((VolumeModel[i]-Batch[i])* #squared deviation of
           (VolumeModel[i]-Batch[i]))     #batch and model per
                                          #component size
  sumerrsq[j]<- sumerrsq[j]+errsq[i] #sum of squared deviations
 }

} #end for j in dmax loop

if (sum(is.na(sumerrsq)) == 4) { #true if nls()-errors occurred
                                 #for all dmax-steps
 #return to: model, bounds, material selection or main menu
}

#find the dmax with the minimum sum of squared dev. for the
#curve fit and set neighbouring iteration steps as new bounds
if (which.min(sumerrsq) == 1) {
 dmaxl <<- dmax[1]
 dmaxu <<- dmax[2]
} else if (which.min(sumerrsq) == 2) {
 dmaxl <<- dmax[1]
 dmaxu <<- dmax[3]
} else if (which.min(sumerrsq) == 3) {
 dmaxl <<- dmax[2]
 dmaxu <<- dmax[4]
} else {
 dmaxl <<- dmax[3]
 dmaxu <<- dmax[4]
}
```

```
    #save dmax that returned minimum sum of squared deviations
    dmaxmin <<- dmax[which.min(sumerrsq)]

   }#end else (dmaxl != dmaxu)

  }#end while loop Andreasen

  optd <- dmax       #save optimal parameters
  optn <- coef(fit)

 } #end if (Model == "Andreasen model")

 if (nacount > 0) { #did any nls-fittings fail?
  #message, that there were failed nls-fittings although a result
  #was found
 }

 #show optimization results:
 #optd contains optimal values for the particle size parameters
 #optn contains optimal values for distribution moduli

} #end of calcparams (calculate model parameters) function
```

5 Logfile and Debugging

For a running session, a logfile 'parsd.log' is created in the main software folder. It will be overwritten by a subsequent session. The logfile captures all the output (messages, errors) of the software and keeps the standard output, the command line, clear to not concern the End User.

It logs the session in coarse steps (cf. listing 4), mostly writing only one to three messages per programmed function. The finish of a function is marked by one 'END' message, even if the function was multiply started and multiple 'BEGIN' messages were logged. By default the software also logs failed optimizations of the nls()-function by saving the (last) error message. For a power user, this can support to solve optimization errors.

Listing 4: Algorithm for the 'Calculate model parameters' main function

```
lf <- file("parsd.log", open = "wt") #Create or overwrite logfile
                                     #   parsd.log
sink(lf, type = "output")            #Put output to logfile
sink(lf, type = "message")           #Put messages to logfiles

message("Logfile\n\nParSD\nTool to design and analyze particle size
distributions\n")                    #Starting text in the logfile

if(!require("tcltk")) install.packages("tcltk")
```

Furthermore, it can be used for debugging. The power user can roughly detect the location of the bug by the coarse steps of the message logs. For the message logs, the message() function is used (cf. listing 4). Then in the roughly detected location containing the bug, print() functions can be added in the code to localize the bug in detail. Using print() functions

has the advantages (1) that they are different from the message() functions and can be easily removed after solving the bug and (2) that they format the output according to the variable type. For example, a list c() is displayed without separators by the message() function, but is readable by the print() function output.

Installation Documentation

After downloading the archive, you unpack it to the place where you want to install the application. Following information is also given in the README.txt.

1 Software requirements

The application is independent on the operating system (OS). It was tested on a GNU/Linux OS (LinuxMint 19.3 Xfce 64 bit) and Windows OS (Windows 10 Education 64 bit). Software requirements:

- GNU R (https://cloud.r-project.org/)
- PDF-Reader

2 Installation notes

On windows, an issue with the writing-rights required by ParSD was reported when ParSD was put into the folder 'C:\Program Files'. If you cannot install ParSD on a windows system, try to put it into a non-system folder.

Enter the unpacked archive folder, following referred to as main folder. Go to the subfolder 'Install' and run the starter corresponding to your OS:

- Linux-Starter
- iOS-Starter.command
- Windows-Starter.bat

For GNU/Linux OS and iOS, basically, a shell-script is copied to the main folder. In the shell-script, 'Rscript main.R' is executed to startup the application. For Windows, the batch-script asks for the Rscript.exe-file, which is in your installation folder of R, possibly in the 'bin' subfolder. After selecting Rscript.exe, a batch-script including the path to Rscript.exe and calling main.R is created in the main folder.

To start the application, go back to the main folder and run 'ParSD' on GNU/Linux OS, 'ParSD.command' on iOS and on Windows 'ParSD.vbs'. With these starters, on GNU/Linux OS and Windows OS, the terminal is invisible.

Note: It is possible that you are asked if you really want to execute the file/s or mark them as executale. To run the software, it should be marked as executable.

End User Documentation

As described in the Technical documentation, the application is programmed linearly with turn-back loops and nested functions. It is thus possible to return to previous dialogs and it all meets at the central one, the main dialog.

From the main dialog or window, the main and database functions are started as well as the settings defined. Therefore, the End User Documentation is sectioned accordingly:

1. Main window and settings
2. Database functions
3. Main functions

1 Main window and settings

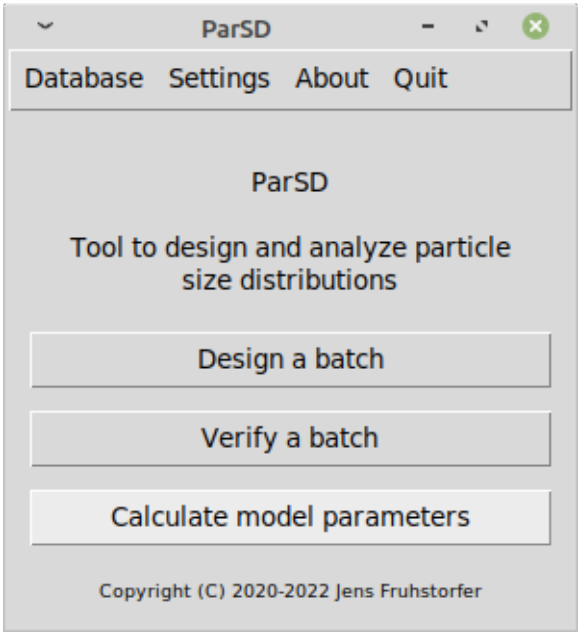


Figure 1: The main window of ParSD app

The main window opens when the application is started. Besides showing the application name, it is constituted of a task bar and a main menu, cf. Figure 1. The main menu leads to the main functions of the application:

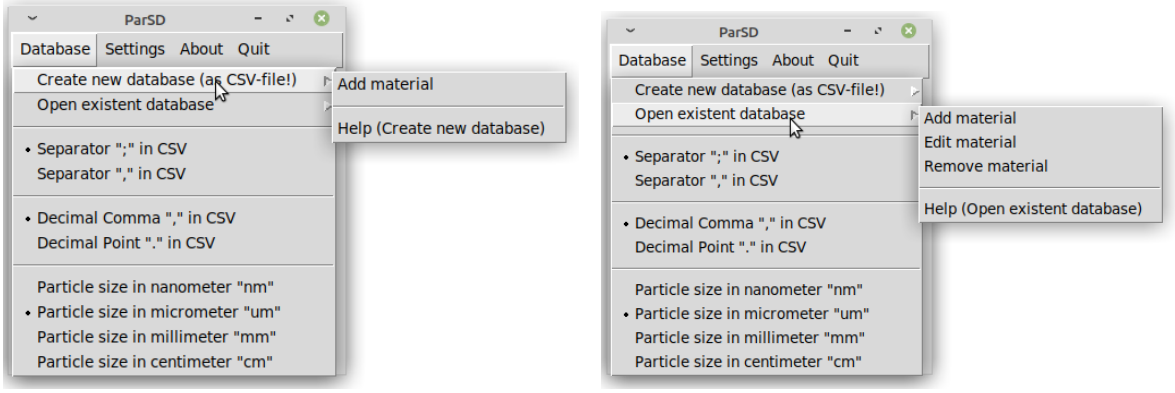
Design a batch In this main function, for a defined particle size distribution, the amounts of selected raw materials (the batch) to approximate optimally this distribution are calculated.

Verify a batch In this main function, a defined particle size distribution and a defined batch (amounts of raw materials) are given and compared.

Calculate model parameters In this main function, for a defined batch (amounts of raw materials), the parameters of a selected particle size distribution model to approximate this batch optimally are calculated. This function can be used for reverse engineering.

The task bar contains the menu items 'Database', 'Settings', 'About' and 'Quit'. The last one closes the application similar to clicking the close-button usually in the upper right corner of the dialog window. Before closing the program, changed settings are saved. The 'About' menu item contains the subitems 'Version/Info', 'Authors', 'Contributors', 'License', 'Citation info', 'Help (Current dialog)' and 'Documentation' of which the last one opens the Documentation, the Help-subitem the part of the documentation about this dialog box and related objects. The others open message-boxes with information on the application, the developers and the license.

The database menu item, cf. Figure 2, contains the settings for the database files of type CSV, abbreviation for 'Comma-separated values'. It is possible to choose a decimal place separator (decimal comma or decimal point) and a separator for the values in the file (comma or semicolon). Standard are either decimal point and data separated by commas or decimal comma and data separated by semicolon. Furthermore, the particle or component size magnification has to be chosen. In the database this is not saved. It is recommended, that the user decides for one style because within a single run of the program, all used files require the same setup to work with each other.



(a) Create new database subitem

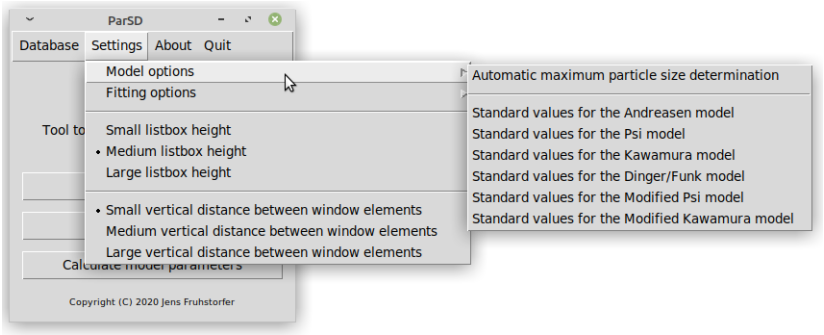
(b) Open existent database subitem

Figure 2: The task bar 'Database' menu item

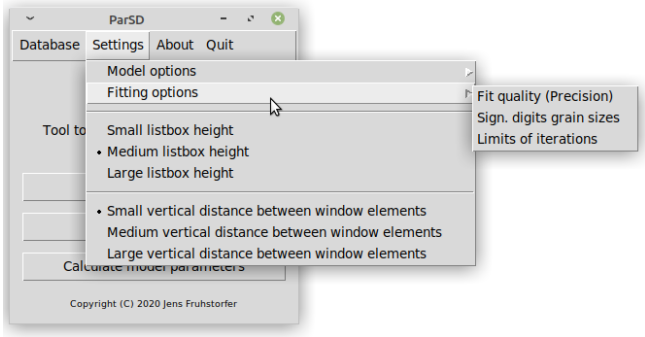
The subitems of the Database menu item link to the database functions. It is possible to create a database and add a first material (cf. Figure 2(a)). The Help for this function is also accessed from the submenu. Furthermore, it is possible to open an existent database and to add, edit or remove materials (cf. Figure 2(a)). Also for these database functions, the corresponding Help is accessed from that submenu.

The settings menu item, cf. Figure 3, contains the view settings for the larger result-windows. If used on a notebook or subnotebook with small display or on a computer with a screen with a low resolution, a small listbox height and a small vertical distance between the window elements might be best. Otherwise it can happen, that the window cannot be shown completely.

The subitems of the Settings menu item are the settings for the models and the fitting. The model settings, cf. Figure 3(a), contain links to the setup of how to determine the maximum particle size automatically and for each model to a dialog where the standard values can be set up. In the dialog 'Automatic maximum particle size determination' (Figure 4(a)), a value for the amount of oversized grains in percent is defined. Every natural crushing or milling process followed by sieving or other separation techniques leads to raw materials spanning from a bit lower to a bit larger particle sizes than the used component sizes of the sieves (cf. Figure 5). For the determination of the maximum particle size only the percentage of oversized grains, the amount of grains larger than the defined upper size, is important. Values between 5 and 15 % are common. Furthermore, it can be noted that the Reset-button



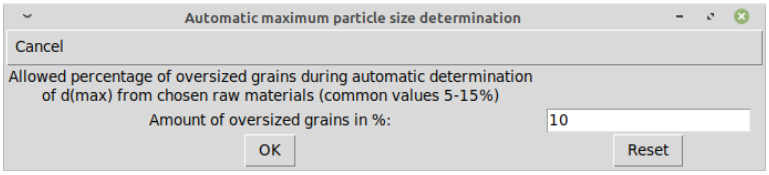
(a) Model settings subitem



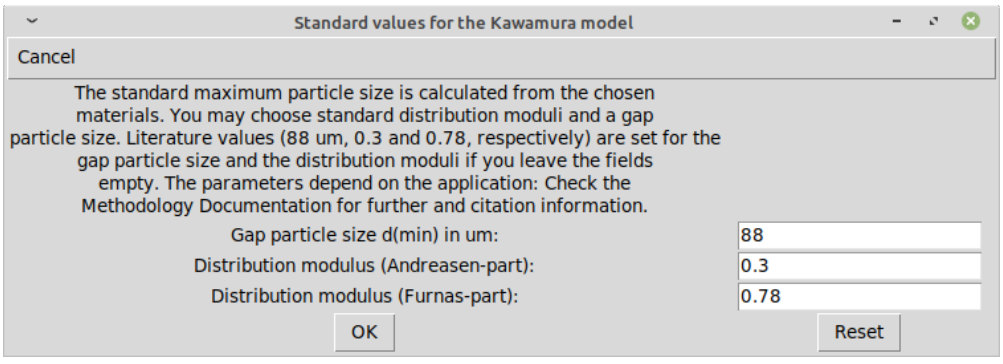
(b) Fitting settings subitem

Figure 3: The task bar 'Settings' menu item

restores not a standard value, but the value from before editing, which is also important to recognize for the dialogs 'Standard values for the ... model' (Example in Figure 4(b)) where chosen literature values can be set by emptying the fields. However, for different applications different standard values might apply. An overview together with citation information is given in the Methodology Documentation.



(a) Overgraining adjustment



(b) Standard values

Figure 4: Dialogs for setting up the models

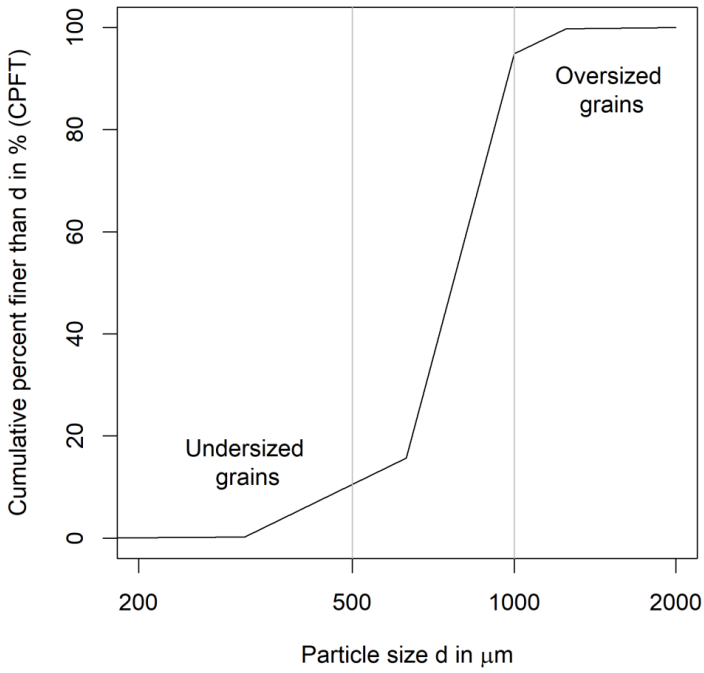


Figure 5: Particle size distribution of an alumina raw material fraction 0.5–1 mm with about 5 % oversized grains (larger than 1 mm) and about 10–15 % undersized grains (smaller than 0.5 mm)

The fitting subitem contains the setup options for the modelling:

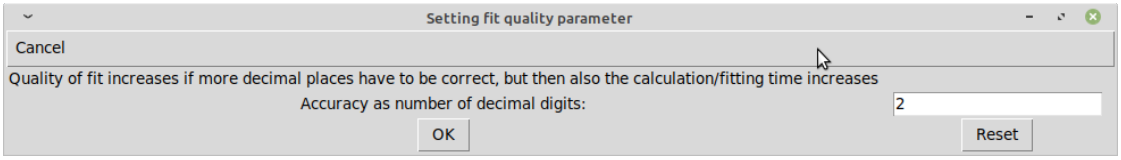
1. The fit quality (Figure 6(a)) is adjusted by defining the number of decimal places of the values to calculate that have to be correct. For example, if a batch has to be calculated that should contain raw materials only as multiples of 5 %, no decimal places are required; but if one raw material is contained with an amount of 0.32 %, then two decimal places are required.

2. The second option dialog defines the number of significant digits for the grain sizes, cf. Figure 6(b). For a standard sieve series, a value of 3 is adequate, but if your component sizes differ, you can adjust it here. For higher numbers, the results for the optimum particle sizes get more detailed—e.g. in a standard sieve series but with a number of signif. digits of 5, it is possible that your optimum maximum particle size for a batch with 3 mm grains is 2983.3 µm which might be not required in this detail, but leads to a much longer calculation time.

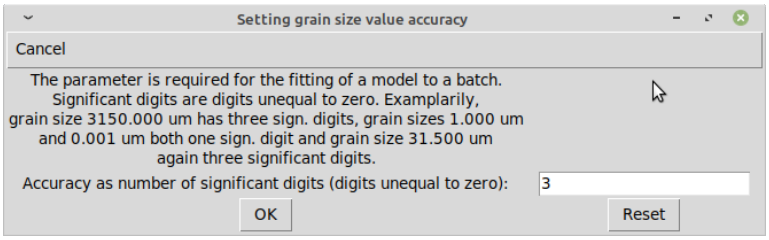
3. The third option lets the user set up limits for the iterations of the different main fitting functions (Figure 6(c)). For the 'Design a batch' function, for the author's experiments up to 7000 iterations were necessary. For the function 'Calculate model parameters', a lower number of iterations (in the author's experiments up to about 30) was required. The limits have to be defined because it is possible that rounding issues occur or that the particle size optimization is not converging and then the optimization process could run infinitely.

2 Database functions

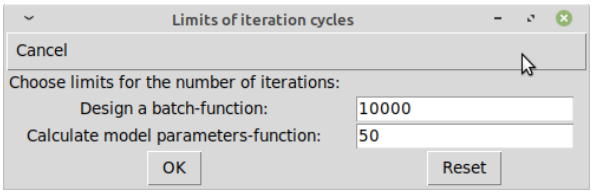
The database functions comprise to create a database and add a first material, to add more materials, edit these materials' data and to remove materials or a database.



(a) Fit quality



(b) Significant digits

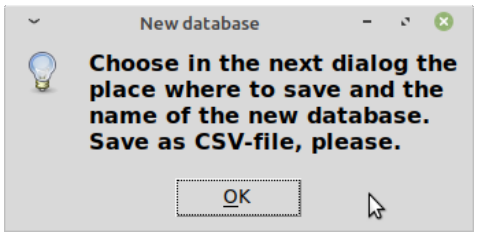


(c) Iteration limits

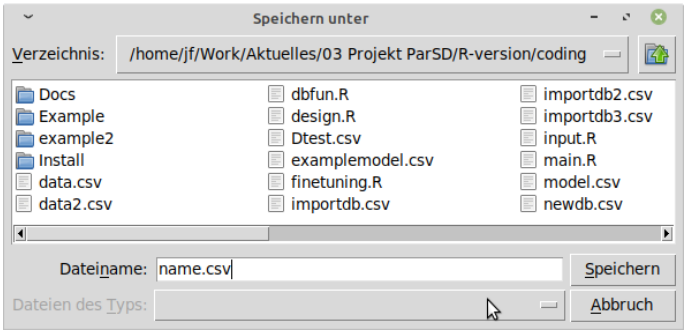
Figure 6: Dialogs for setting up the fitting options

2.1 Create new database

After clicking to create a database and add a first material, the user is asked to enter the path and name for saving the database. The file extension .csv has to be added to the filename of which the user is reminded by a message before choosing path and filename, cf. Figure 7. It can be noted that the database is not created in this step, but only after a first material to include in the database was added because empty databases cannot be handled by the software.



(a) Message CSV-extension

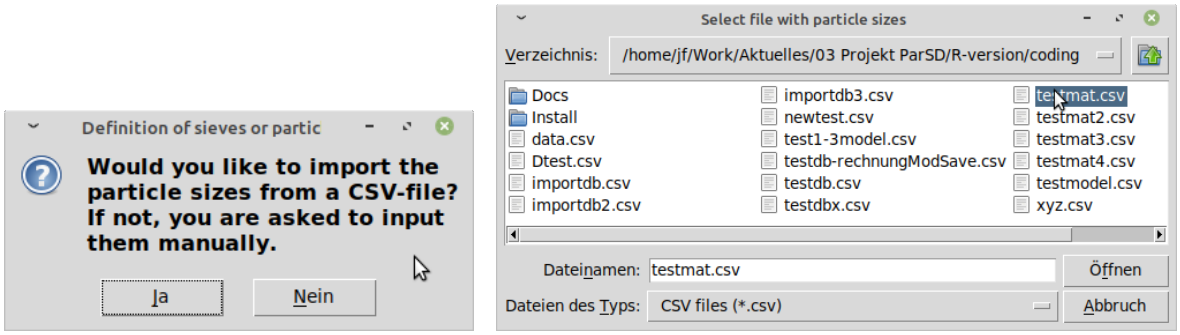


(b) Path and name definition of new database

Figure 7: Choosing path and name for saving the new database

For the later creation of the database, it is essential to define the sieve or component sizes. It is possible to import the component sizes or to input them manually—the user is asked to decide how to define the component sizes, cf. Figure 8(a). If the user chooses to import them, in the next step the CSV file has to be chosen (Figure 8(b)) which contains the component sizes in a separate column with the column head 'Diameter'. Furthermore, the sizes have

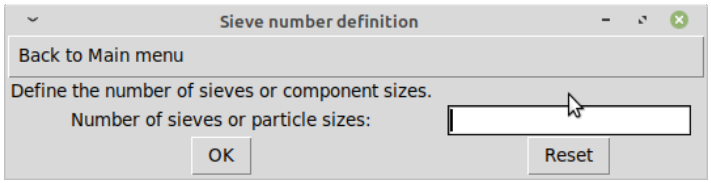
to increase with increasing line number. Thus, it is possible to choose directly a prepared raw material file (an example is shown in Figure 3 in the Technical Documentation). The length units are defined in the database settings as explained in the section 'Main menu and settings' of the End User Documentation.



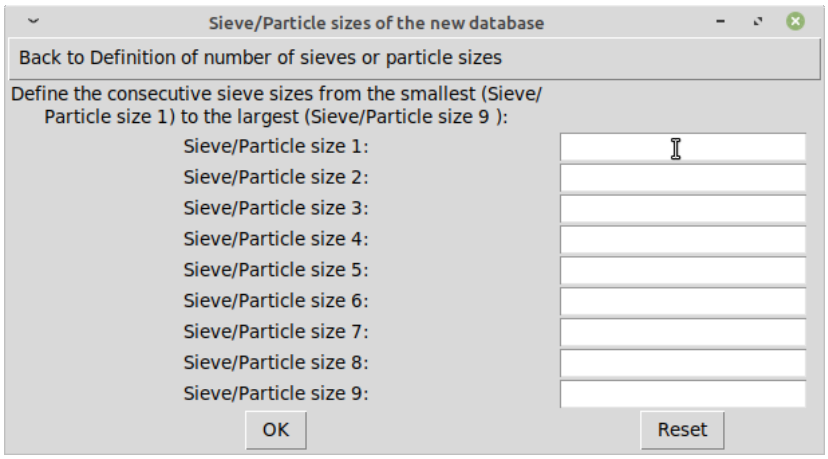
(a) How to define component sizes (b) Choose file with component sizes

Figure 8: Import component sizes

If the user chooses to input the component or sieve sizes manually, firstly, the number of component sizes has to be defined, cf. Figure 9(a). Secondly then, the user is asked to input the compnent sizes, cf. Figure 9(b). The length units are defined in the database settings as explained in the section 'Main window and settings' of the End User Documentation.



(a) Define number of component sizes



(b) Define component sizes

Figure 9: Input of component sizes

After the first material was added, then, the database is created and saved.

Add first material

To add a first material to the newly defined database, the dialog shown in Figure 10 gives three possibilities. The material data can be input manually or imported from a prepared

raw material CSV file. For these two possibilities, the user has to define in what form the particle size distribution of the raw material will be given. Furthermore, it is possible to import the first raw material from another database.

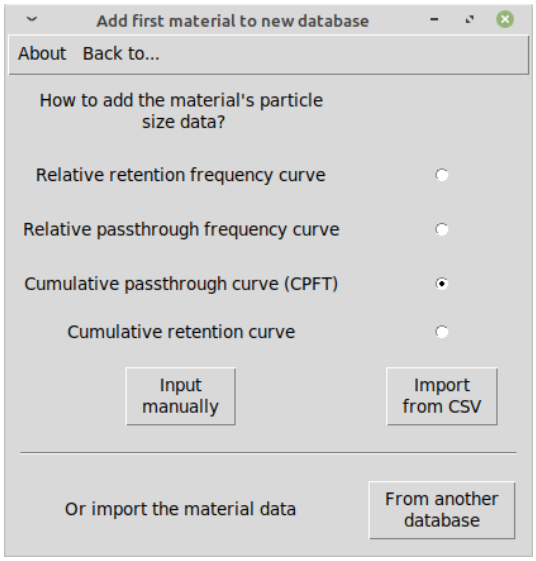
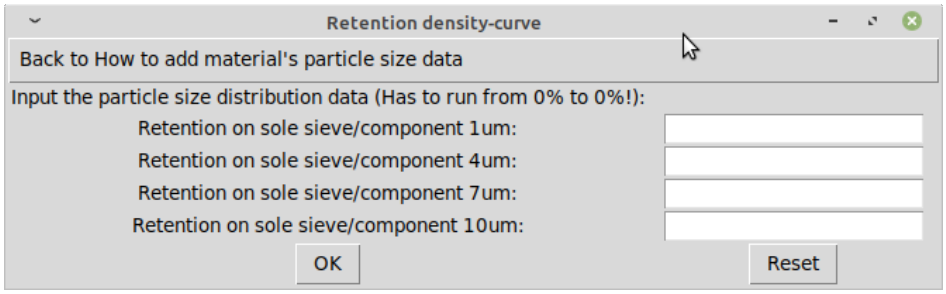


Figure 10: Add first material-dialog

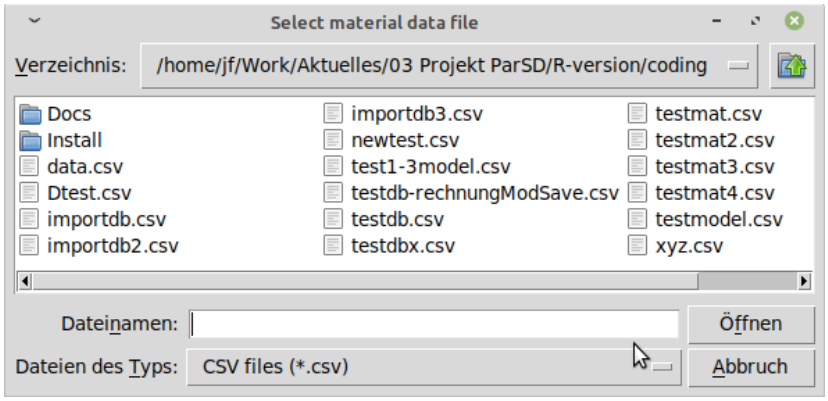
The different types of particle size distributions (relative/cumulative retention/passthrough frequency curves) are explained in the Methodology Documentation and how a prepared raw-material file looks is described in the Technical Documentation. Depending on the type of particle size distribution, the data has to run from 0 % to 0 % for the relative frequency curves or from 0 % to 100 % or vice-versa for the cumulative curves. For the program to work correctly, it is essential that the raw material's minimum particle size is larger than the minimum component size of the database and that the material's maximum diameter is smaller than the maximum component size of the database. Otherwise, the material can obviously not be integrated into the database completely.

If the user inputs the data or imports it from a raw material file, the first step is either to input the data manually (Figure 11(a)) or to choose the raw material file (Figure 11(b)). The data has to be given completely (from 0 % to 0 % for relative frequency curves, ...), but not necessarily for every component size. Values for the missing component sizes are linearly interpolated and the user informed about it (Figure 12(a)). If the particle size data is given for other diameters in the raw material file than the ones defined in the database, the particle size distribution is calculated for the database by linear interpolation. It can be noted that a linear interpolation is directly possible only for a cumulative curve without changing all other values. Therefore, the inputted or imported particle size data is transformed to CPFT-values and then interpolated. After giving the particle size distribution data by inputting or importing, the user is asked to give information about the material (Figure 12(b)).

Predefined fields of the material information are the d(CPFT)-values calculated from the given particle size distribution, the date the raw material was last modified and the unique identifier which is the only mandatory information that has to be given. It is the parameter listed in the dialogs where materials can be selected from a database. All fields can be given or changed. For example, the calculated d(50%) (also referred to as d50-value) could be replaced by the value given in the datasheet of the raw material. For batch analyses in wt%, a true density has to be given. For the database, the units of the given information are not predefined, but have to be consistent within the same database. For the fields price, true density, specific surface area and the d(CPFT)-values, only numbers are recognized—other inputs are deleted/blanked for these fields.



(a) Input manually



(b) Choose raw-material file for import

Figure 11: Input or import raw material particle size data

If the user imports the material data from another database, the type of the particle size distribution data and the material information have not to be given as both are pre-defined in the database which has to be opened (Figure 13(a)) firstly before secondly the material to be imported is selected from the list of materials in the chosen database (Figure 13(b)). Also in this case, it is important that the particle size range of the material is within the range of the component sizes of the database into which it should be imported. Furthermore, also for this import-function, a linear interpolation follows if required to fit the data to the given set of component sizes in the database.

Finally, after adding the first material, the database is created and saved. The user is informed after the successful creation and asked if another material should be added, cf. Figure 14. If declined, it is returned to the main window and if accepted, the 'Add material to an existent database' function is called with the just created database as existent one.

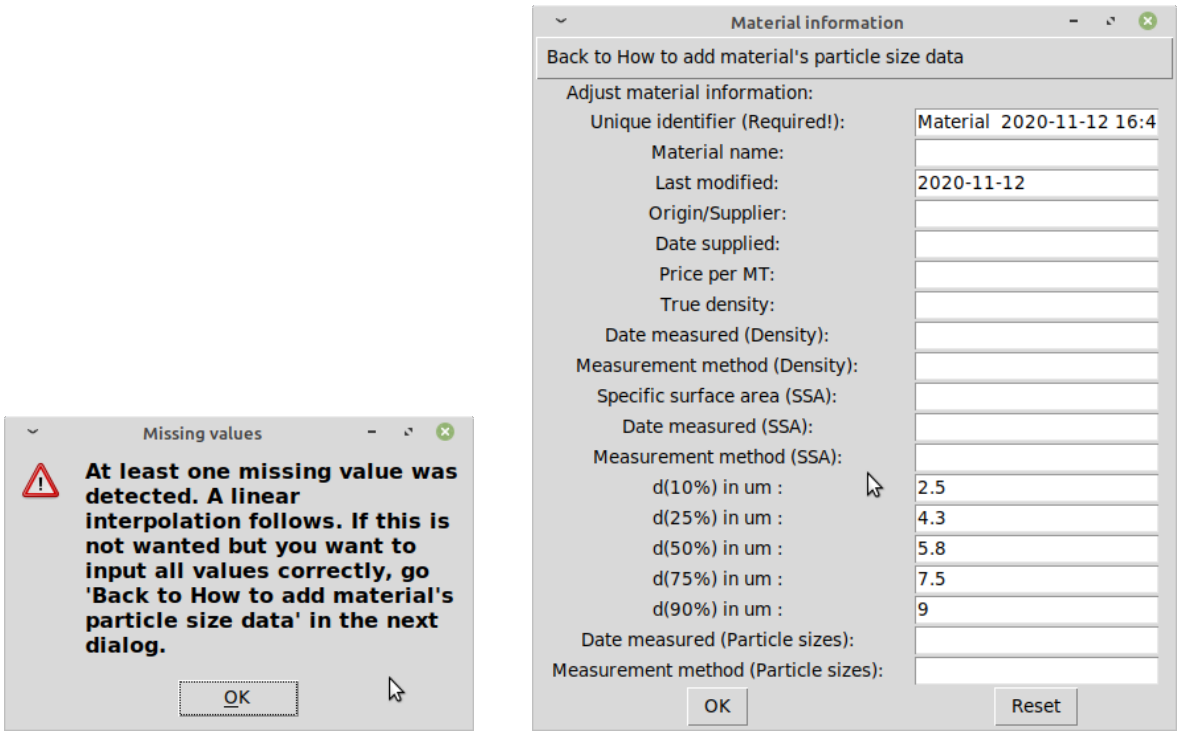
2.2 Open existent database to remove, edit or add materials

To administer a database or the materials within, the database has to be selected (Figure 15).

For adding a material, this is enough while for editing or removing a material either one material or more materials have to or can be selected from the database, respectively (cf. Figure 16).

Remove material or complete database

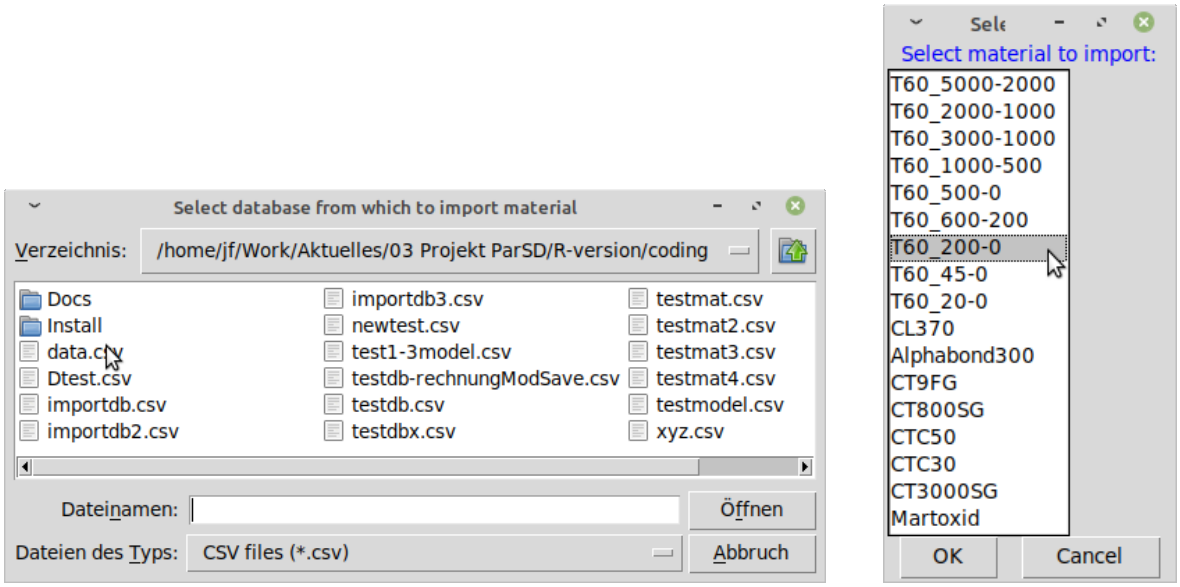
The material(s) selected will be removed from the database without further inquiry. An exception is if all materials were marked: Then it is asked if all materials should be removed together with the database, cf. Figure 17(a). After removal of the selected materials from the database, furthermore, the user is informed about the success and asked if more materials should be deleted (Figure 17(b)).



(a) Interpolation message

(b) Defining material information

Figure 12: Possible linearization step and definition of raw material information



(a) Open database with raw material

(b) Select raw material from database

Figure 13: Import raw material from another database

Edit material

The main dialog (Figure 18(a)) to edit a material shows the saved material information in the upper part. The as-saved material can be transformed to a model and can be saved to be used for the main functions to 'Design a batch' or 'Verify a receipt'. By clicking the

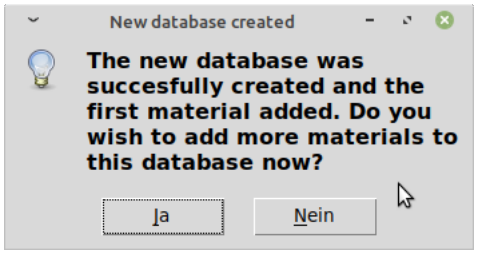


Figure 14: Add another material

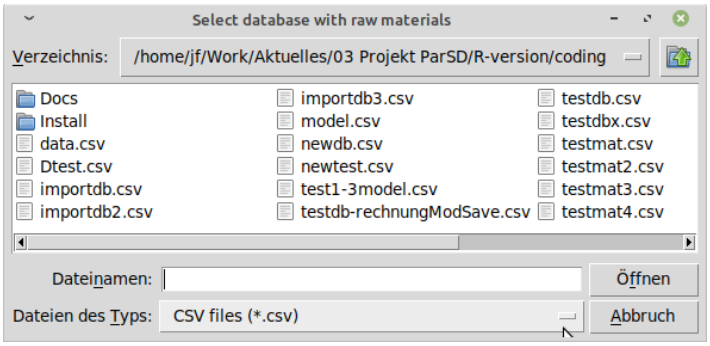
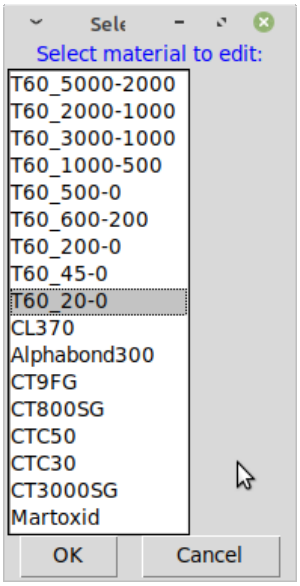
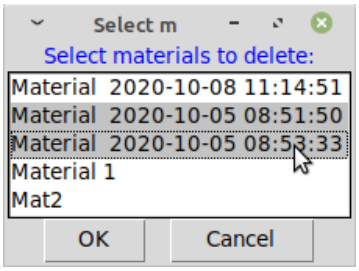


Figure 15: Open existent database



(a) Select one material to be edited



(b) Select one or materials to be removed

Figure 16: Select material(s) for editing or removing from database

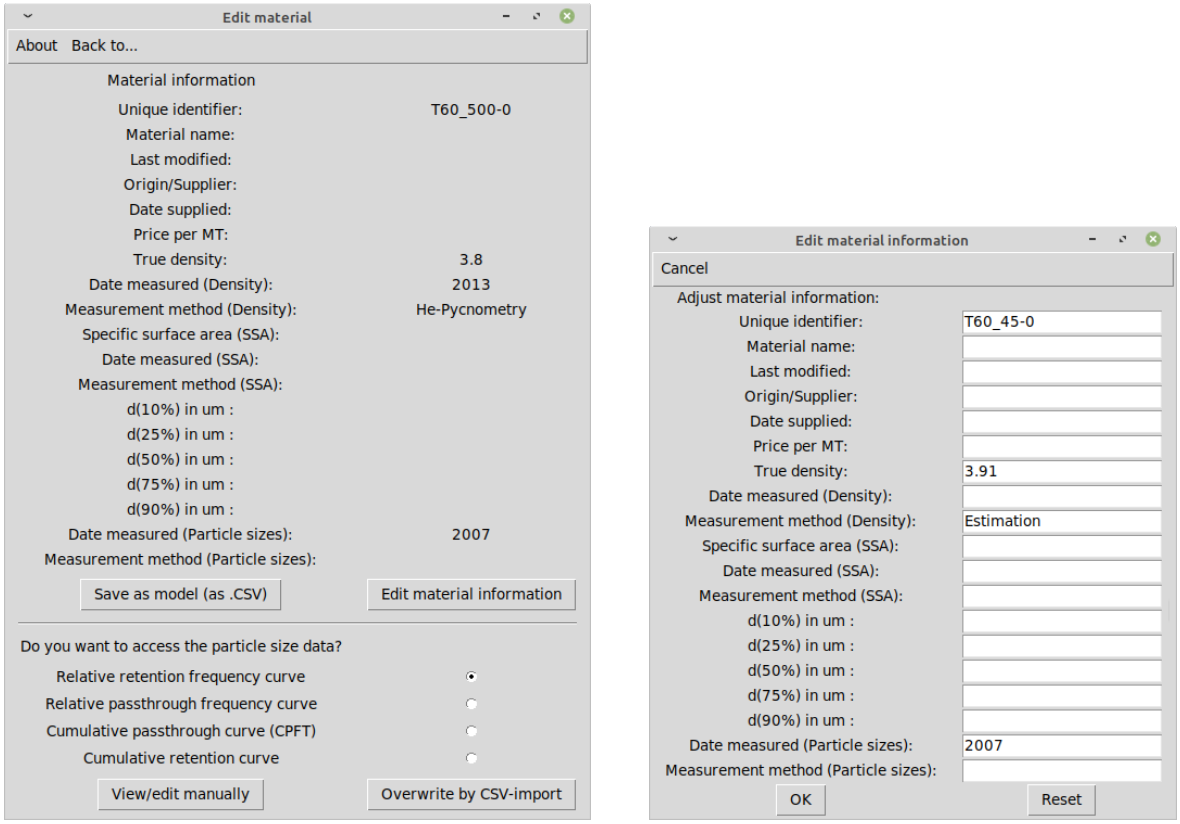
button, a save file-dialog opens in which the .csv-extension should be added to the filename. Furthermore, the displayed material information can be accessed and changed (Figure 18(b)).

The unique identifier is mandatory and may not be deleted, but can be also changed. It will be listed in the dialogs where materials can be selected from a database. All fields can be given or changed. For the database, the units of the given information is not predefined, but it has to be consistent within the same database. For the fields price, true density, specific surface area and the d(CPFT)-values, only numbers are recognized—other inputs are deleted/blanked for these fields.

(a) Inquiry to remove database

(b) Material removeal confirmation

Figure 17: Remove material(s) or delete complete database



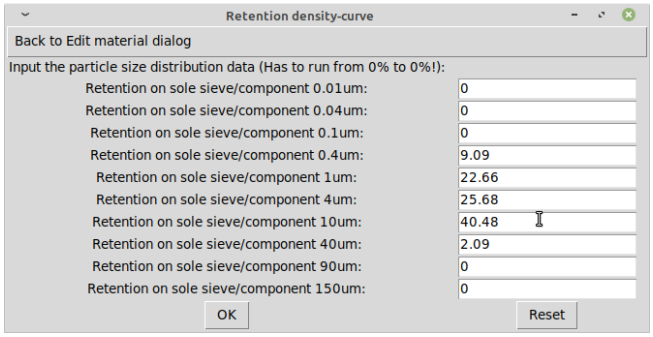
(a) Edit material dialog

(b) Select one or materials to be removed

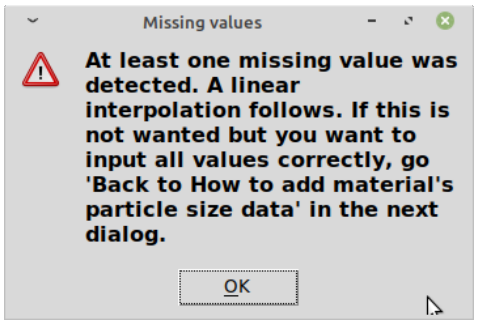
Figure 18: Edit material and edit material info dialogs

In the lower part of the main edit material dialog (Figure 18(a)), the particle size distribution data of the material can be accessed. The particle size distribution of the selected type (relative/cumulative retention/passthrough frequency curve) can be viewed by clicking on 'View/edit manually' which opens dialogs where the particle size distribution could be also changed, cf. Figure 19(a). The alternative is to import a new particle size distribution of the selected type for the material. This overwrites the old data without further inquiry after the user selected the prepared raw-material CSV file in an open file-dialog.

The different types of particle size distributions (relative/cumulative retention/passthrough frequency curves) are explained in the Methodology Documentation and how a prepared raw-material file looks is described in the Technical Documentation. Depending on the type of particle size distribution, the data has to run from 0 % to 0 % for the relative frequency curves or from 0 % to 100 % or vice-versa for the cumulative frequency curves. For the program to



(a) Access particle size data manually

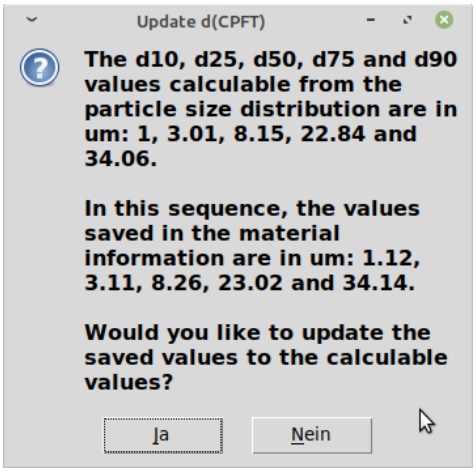


(b) Information about missing values to be interpolated

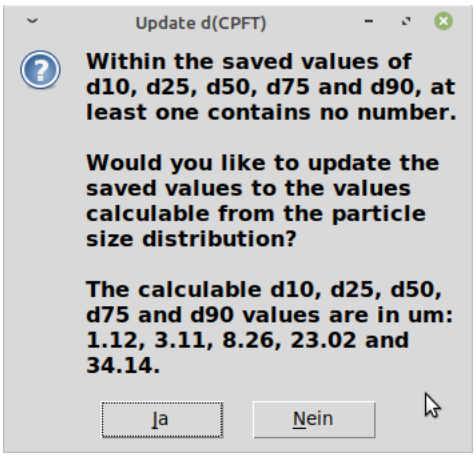
Figure 19: View/edit particle size data manually for chosen particle size distribution type

work correctly, it is essential that the raw material's minimum particle size is larger than the minimum component size of the database and that the material's maximum diameter is smaller than the maximum component size of the database. Otherwise, the material cannot be integrated into the database completely.

The data has to be given completely (from 0 % to 0 % for relative frequency curves, ...), but not necessarily for every component size. Values for the missing component sizes are linearly interpolated and the user informed (Figure 19(b)). If the particle size data is given for other diameters in the raw material file than the ones defined in the database, the particle size distribution is calculated for the database by linear interpolation. It can be noted that a linear interpolation is directly only possible for a cumulative curve without changing all other values. Therefore, the inputted or imported particle size data is transformed to CPFT-values and then interpolated.



(a) No missing values in database



(b) Missing values in database

Figure 20: Updating d(CPFT) values

From the changed particle size distribution, d(CPFT) values are calculated and if they differ from the saved ones, the user is asked if these specific fields of the material information should be updated (Figure 20). There are two possible cases. Firstly, the information was completely filled out or there were empty fields which could for example happen if a user decided to use the datasheet value for d50 and the other fields were empty.

Add material

To add a material to the chosen database, the dialog shown in Figure 21 gives three possibilities. The material data can be inputted manually or imported from a prepared CSV file. For these two possibilities, the user has to define in what form the particle size distribution of the raw material will be given. Furthermore, it is possible to import the raw material from another database.

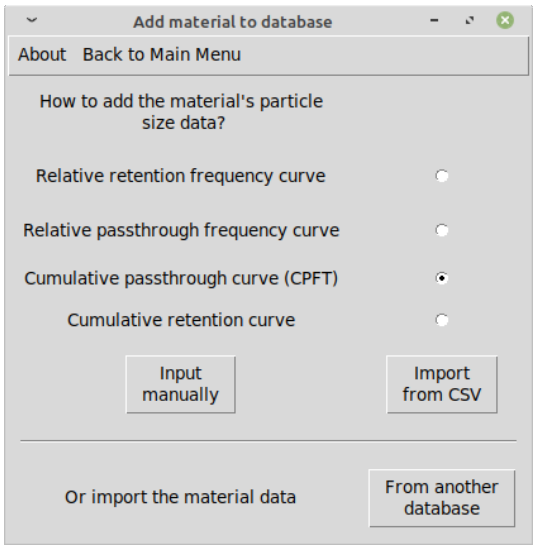
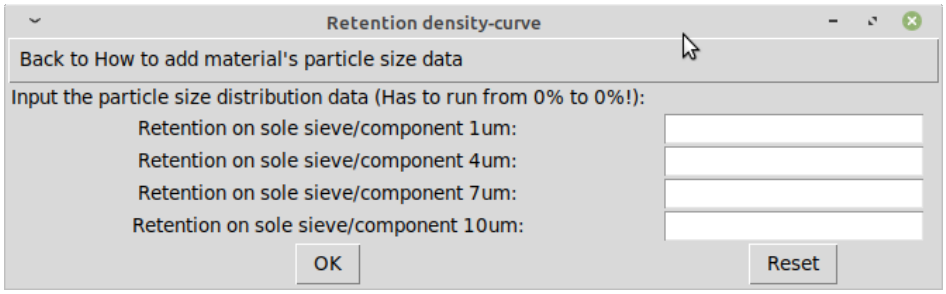


Figure 21: Add material-dialog

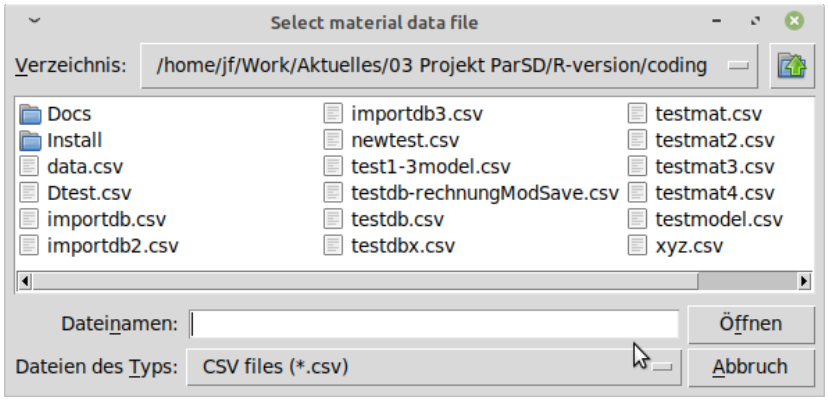
The different types of particle size distributions (relative/cumulative retention/passthrough frequency curves) are explained in the Methodology Documentation and how a prepared raw-material file looks is described in the Technical Documentation. Depending on the type of particle size distribution, the data has to run from 0 % to 0 % for the relative frequency curves or from 0 % to 100 % or vice-versa for the cumulative frequency curves. For the program to work correctly, it is essential that the raw material's minimum particle size is larger than the minimum component size of the database and that the material's maximum diameter is smaller than the maximum component size of the database. Otherwise, the material cannot be integrated into the database completely.

If the user inputs the data or imports it from a raw material file, the first step is either to input the data manually (Figure 22(a)) or to choose the raw material file (Figure 22(b)). The data has to be given completely (from 0 % to 0 % for relative frequency curves, ...), but not necessarily for every component size. Values for the missing component sizes are linearly interpolated and the user informed about it (Figure 23(a)). If the particle size data is given for other diameters in the raw material file than the ones defined in the database, the particle size distribution is calculated for the database by linear interpolation. It can be noted that a linear interpolation is only possible for a cumulative curve without changing all other values. Therefore, the inputted or imported particle size data is transformed to CPFT-values and then interpolated. After giving the particle size distribution data by inputting or importing, the user is asked to give information about the material (Figure 23(b)).

Predefined fields of the material information are the d(CPFT)-values calculated from the given particle size distribution, the date the raw material was last modified and the unique identifier which is the only mandatory information that has to be given. It is the parameter listed in the dialogs where materials can be selected from a database. All fields can be given or changed. For example, the calculated d(50%) (also referred to as d50-value) could be replaced by the value given in the datasheet of the raw material. For batch analyses in wt%,



(a) Input manually



(b) Choose raw-material file for import

Figure 22: Input or import raw material particle size data

a true density has to be given. For the database, the units of the given information are not predefined, but have to be consistent within the same database. For the fields price, true density, specific surface area and the d(CPFT)-values, only numbers are recognized—other inputs are deleted/blanked for these fields.

If the user imports the material data from another database, the type of the particle size distribution data and the material information have not to be given as both are pre-defined in the database which has to be opened (Figure 24(a)) firstly before secondly the material to be imported is selected from the list of materials in the chosen database (Figure 24(b)). Also in this case it is important that the particle size range of the material is within the range of the component sizes of the database into which it should be imported. Furthermore, also for this import-function, a linear interpolation follows if required to fit the data to the given set of component sizes in the database.

Finally, the database is saved. The user is informed and asked if another material should be added, cf. Figure 25. If declined, it is returned to the main window and if accepted, the function 'Add material to existent database' is called again with the chosen database.

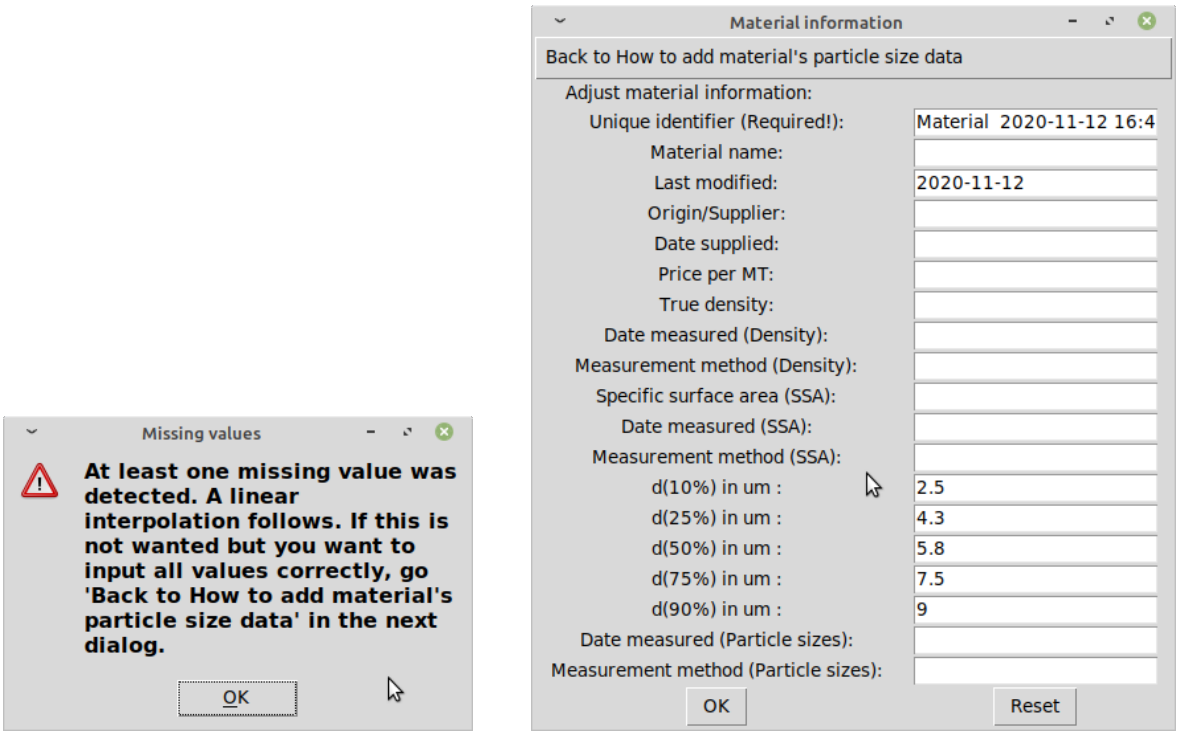
3 Main functions

The three main functions 'Design a batch', 'Verify a recipe' and 'Calculate model parameters' follow generally the same operational steps (cf. flowcharts in the Technical Documentation) which form also the subsections of the current section:

⊤ Raw materials

↓ Batch

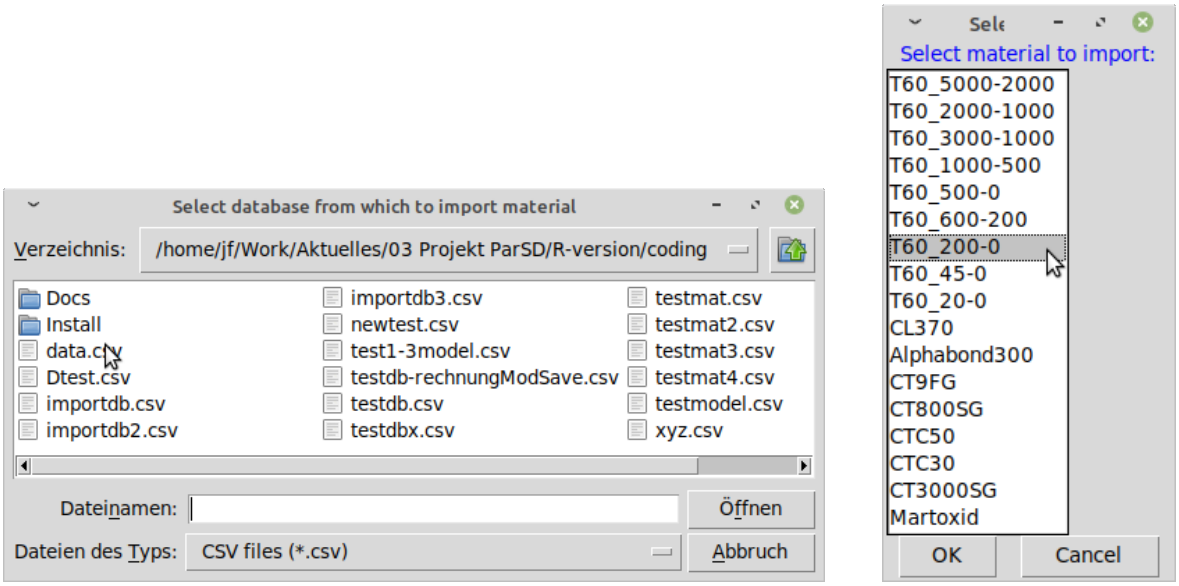
↓ Model



(a) Interpolation message

(b) Defining material information

Figure 23: Possible linearization step and definition of raw material information



(a) Open database with raw material

(b) Select raw material from database

Figure 24: Import raw material from another database

↓ Calculations

⊥ Results

For the 'Design a batch' function, the model is fully defined, for the 'Calculate model parameters' function the batch. The batch or model by its parameters, respectively, are

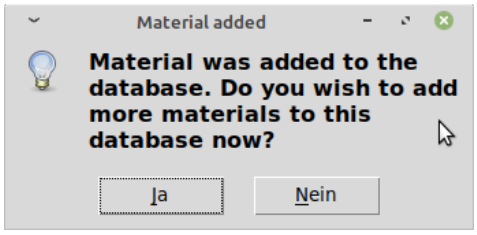


Figure 25: Add another material

calculated by a numerical optimization. For the 'Verify a recipe' function, both batch and model are fully defined and the result (a comparison in this case) is calculated analytically.

3.1 Select raw materials from a database or recipe

The first step for all three main functions is to select the raw materials which will be composed subsequently to a batch. The user can decide to select the raw materials from a database or a recipe file, cf. Figure 26. Both file types are CSV files and have the structure described in the section 'Main menu and database functions' of the Technical Documentation.

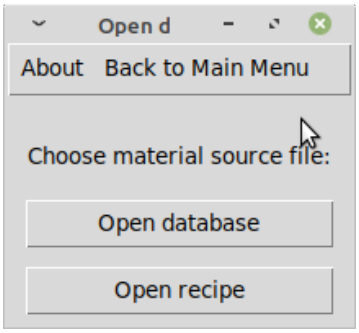


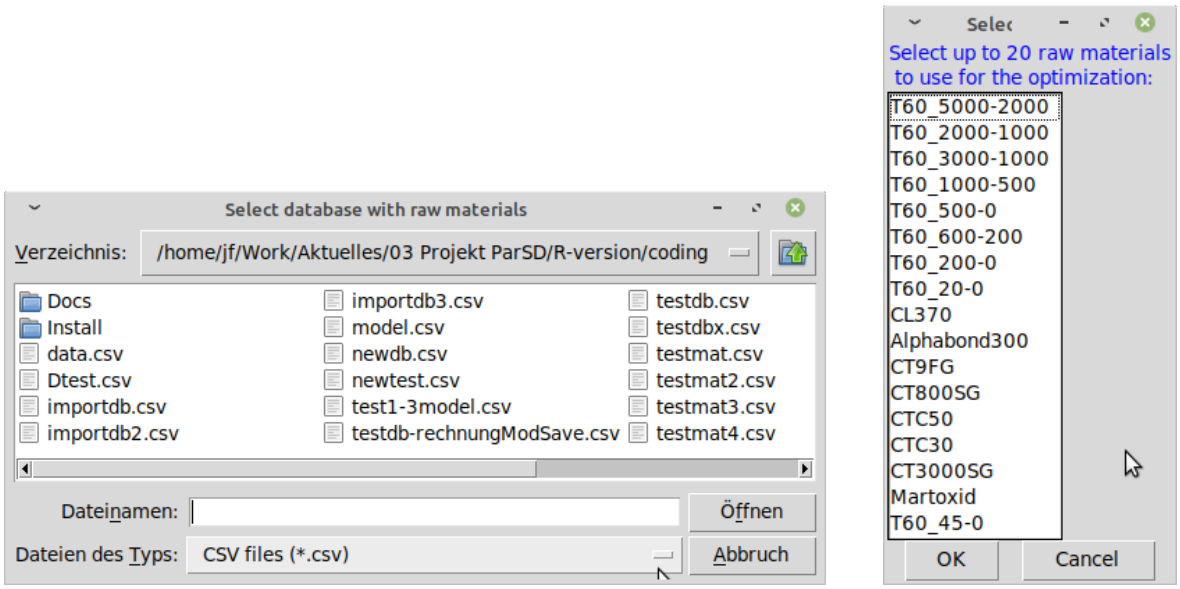
Figure 26: Select material source

The difference lies within the following procedure. After selecting a database CSV-file (Figure 27(a)), the user just selects the raw materials to be included in the batch from the list of raw materials contained in the database (Figure 27(b)). The database is meant to contain all required raw materials. The software can handle only batches composed of up to 20 raw materials, which is also checked and an error (Figure 28) returned if too many materials were selected.

In contrast, if a recipe CSV-file is selected (Figure 29(a)), it is expected that this file contains most of the required raw materials but also only these. Therefore, all materials in the recipe file are pre-marked for selection and the user can de-select materials firstly (Figure 29(b)). Secondly, it is possible to add materials from another source file (Figure 30). If the user decides to add materials from another source, this source has to be chosen and then the materials to add are selected from the materials list similar to the 'Open database' procedure (cf. Figure 27). Finally, also for the number of raw materials chosen from a recipe and possibly from an additional source, the maximum number altogether is 20 which is checked and an error returned (Figure 28) if exceeded.

3.2 Set up batch properties

Depending on the main function either the batch is fully defined ('Verify a recipe' and 'Calculate model parameters' main functions) or the batch is the target to be optimized and there can only bounds be defined ('Design a batch' main function).



(a) Choose database file

(b) Select raw materials from database

Figure 27: Select materials from database

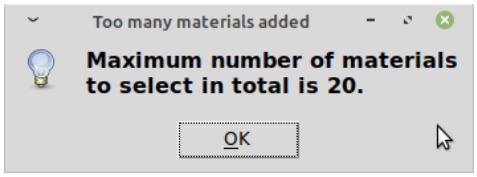
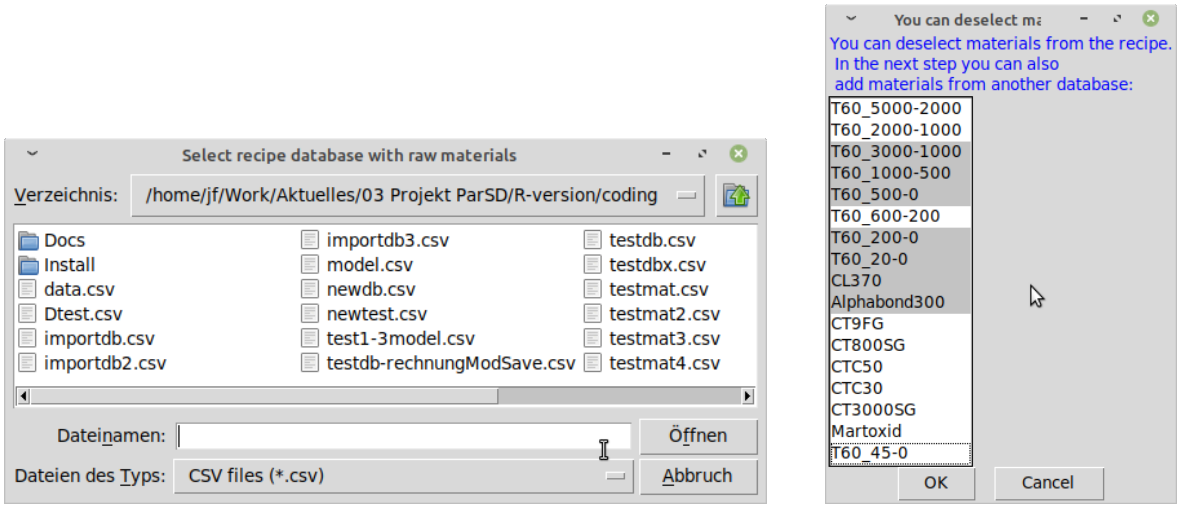


Figure 28: Error message if maximum number of raw materials is exceeded



(a) Choose recipe file

(b) De-select raw materials from recipe

Figure 29: Select materials from recipe-database

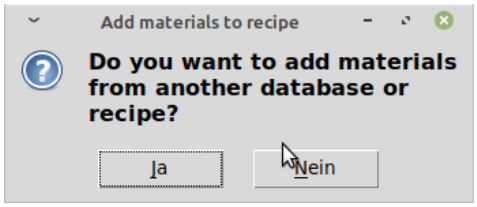
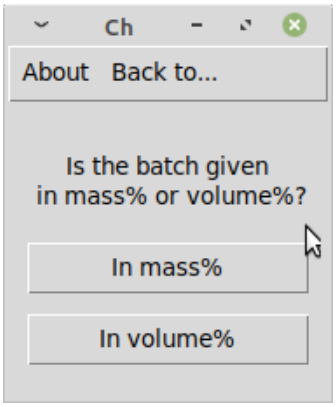


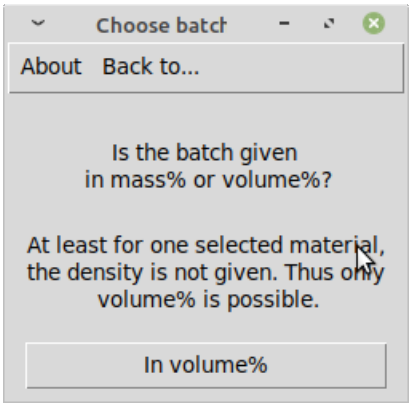
Figure 30: Add materials from another source file

Batch definition ('Verify a recipe' & 'Calculate model parameters' main functions)

Depending on the saved properties of the selected raw materials—namely if the true density for all chosen materials is given—the batch can be defined either in vol% and wt% or only in vol%. The software checks this circumstance beforehand (cf. Figure 31).



(a) Window in case of given densities



(b) Window in case not all densities are saved

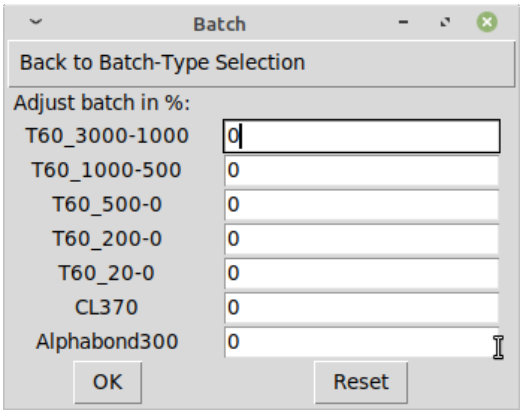
Figure 31: Different dialogs depending on whether true densities are given for all raw materials

The reason that calculations in wt% (including batches in wt%) are only possible if the true densities for the raw materials are given is that the particle size distributions saved in the database or recipe files are in vol% and furthermore are considered to be of a true density independent on the particle size for the single raw materials. A composition by volume is thus always possible, but the weight fractions differ by the respective densities. Hence, to calculate in wt%, the true densities for all raw materials to be included in the batch are required. If not given during addition of a raw material to a database, the true density can always be added by using the 'Edit material in an existent database' function accessible from the Main window menu of the ParSD software.

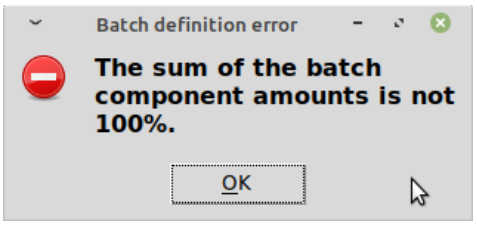
After choosing for the batch type (vol% or wt%), the batch has to be composed (Figure 32(a)) and the software checks if the batch adds up to 100 % (Figure 32(b)). Caution: In the present version of the software it is not chekced, if for all materials a value was given at all although this is required by the software. If you accidentally added more materials than required, you might define their share as 0 %.

Bounds definition ('Design a batch' main function)

Depending on the saved properties of the selected raw materials—namely if the true density for all chosen materials is given—the batch bounds can be defined either in vol% and wt% or only in vol%. The software checks this circumstance beforehand (cf. Figure 33). Moreover, there is a shortcut if no bounds are to be defined: In this case automatically all lower bounds



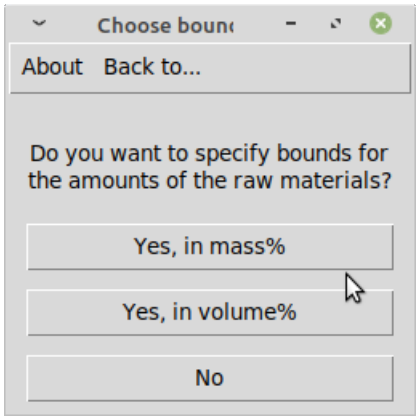
(a) Input batch composition



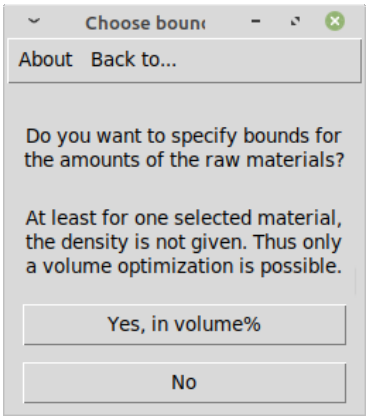
(b) Error message if batch $\neq 100\,\%$

Figure 32: Batch composition

are set to 0 % and all upper bounds to 100 % defining the co-domains for the contents.



(a) Window in case of given densities

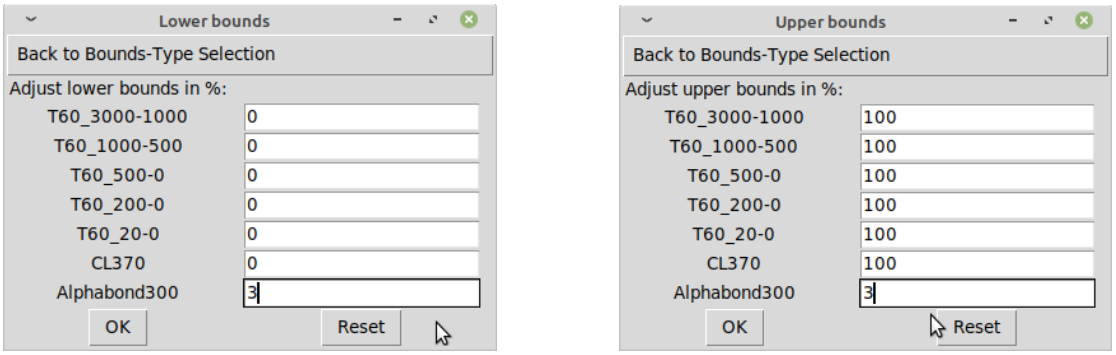


(b) Window in case not all densities are saved

Figure 33: Different dialogs depending on whether true densities are given for all raw materials

The reason that calculations in wt% (including batches and their bounds in wt%) are only possible if the true densities for the raw materials are given is that the particle size distributions saved in the database or recipe files are in vol% and furthermore are considered to be of a true density independent on the particle size for the single raw materials. A composition by volume is thus always possible, but the weight fractions differ by the respective densities. Hence, to calculate in wt%, the true densities for all raw materials to be included in the batch are required. If not given during adding a raw material to a database, the true density can always be added by using the 'Edit material in an existent database' function accessible from the Main window menu of the ParSD software.

After choosing a bounds type of either vol% or wt%, the bounds are inputted (Figure 34). The predefined values are 0 for the lower and 100 for the upper bounds. It is possible to adjust lower and upper bounds to the same value which means that this raw material is to be contained with a constant amount. For example, a user might want to fix the content of a special additive or of two components giving a reaction bond. The software only checks that not all materials were fully defined meaning that for all materials equal lower and upper bounds were given (Figure 35) which would equal a batch definition and would lead to errors in the further calculations. Caution: In the present version of the software it is not chekced,



(a) Lower bounds (b) Upper bounds

Figure 34: Input bounds

if the lower bounds are lower or equal to the upper bounds which is mandatory.

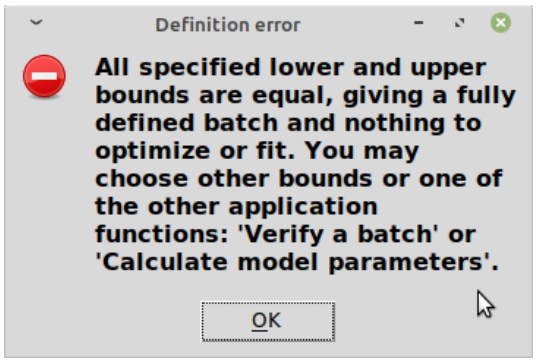


Figure 35: Error message if the batch was fully defined

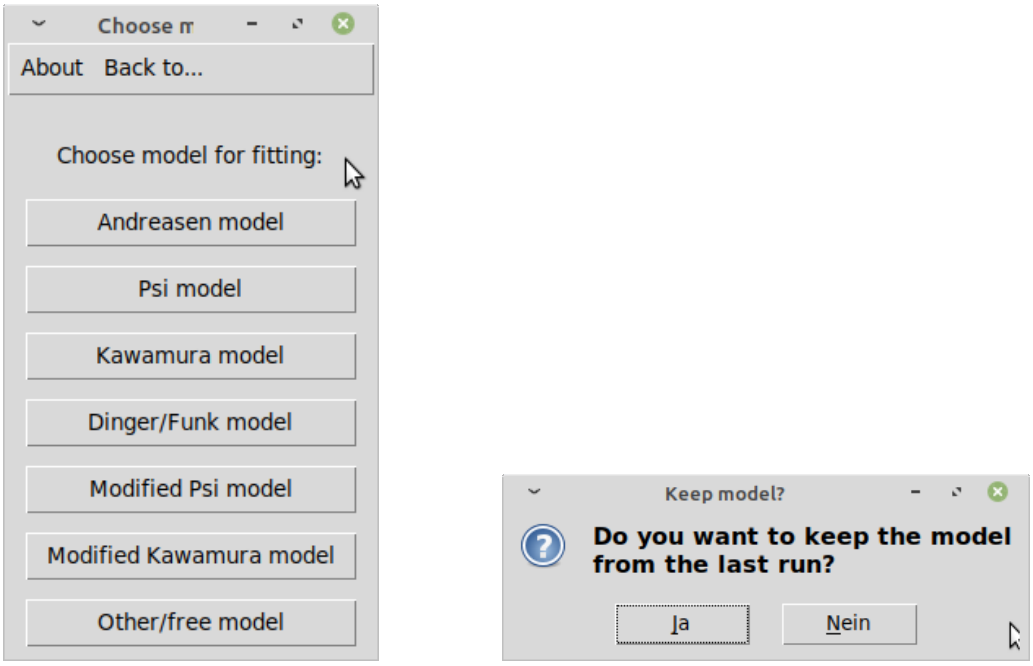
3.3 Set up model

Depending on the main function, either the model is fully defined ('Verify a recipe' and 'Design a batch' main functions) or the model by its parameters is the target to be optimized and there can only bounds be defined for these parameters ('Calculate model parameters' main function).

Model definition ('Verify a recipe' & 'Design a batch' main functions)

After the information to the raw material composition are completed, the model has to be set up (Figure 36(a)). If in the storage is already an instant of a model, the user is asked whether to keep the model (Figure 36(b)). For example this might be the case if an optimization failed in the subsequent process step and the user wants to place differing bounds for the raw materials. Then the user might return to the 'Batch bounds definition' from the 'Optimization error' dialog and after defining new bounds, the user can skip defining the model entirely new by keeping the model.

Possible models to be set up are the models described in the Methodology Documentation and a 'Free/other model'. For the models from the literature, by clicking on them the user is asked to input the required parameters to define the model completely (Figure 37(a)). The preset values which can be also returned to by clicking 'Reset' are for the maximum particle size one that is calculated from the given raw materials (taking the component sizes of the database into consideration as well as the percentage of oversized grains specified in the

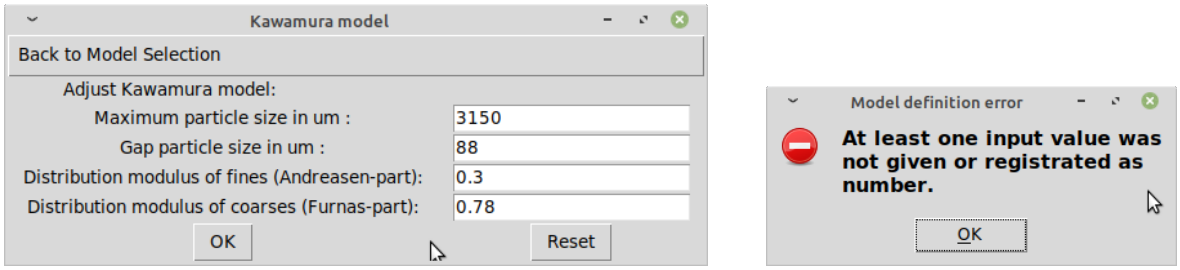


(a) Choose model

(b) Keep model?

Figure 36: Choose or keep model

Settings accessible from the Main Window Menu). The other parameters' preset values are the standard values from the Settings accessible from the Main Window Menu for the model in question. If in these settings no minimum particle size was chosen for models including such, the preset value is the smallest component size from the database. It is checked if all parameters were defined (Figure 37(b)), but not if the input values are within reasonable ranges.



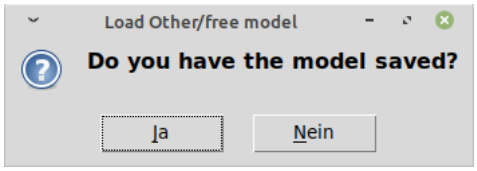
(a) Define model parameters of chosen model

(b) Model definition error

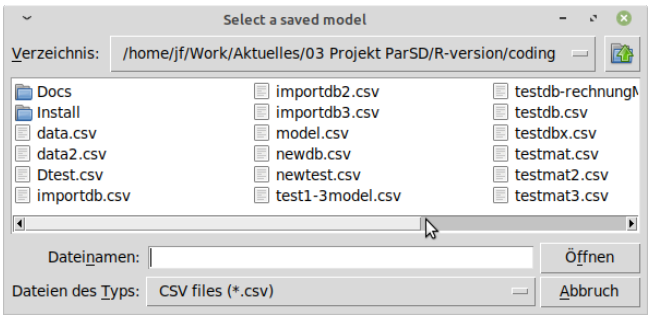
Figure 37: Define literature model by its parameters

The 'Free/other model' gives the user the possibility to define a model by its CPFT(d)-values. Thus, the software can be also used for models different from the ones chosen from the literature. It is firstly asked if such a free model is saved and only has to be opened, cf. Figures 38(a) and 38(b). If the free model was not prepared in a file, but will be given manually, the user is asked to specify a name for the model (Figure 38(c)) and then to input the CPFT(d) values for all component sizes d (Figure 38(d)). Afterwards, it is possible to save the specified free model (Figures 38(e) and 38(f)) as a CSV file by adding the extension to the filename in the save-dialog.

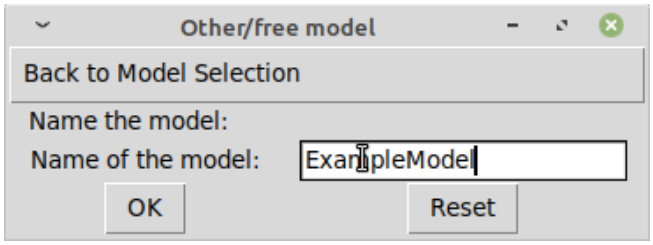
A prepared model CSV file (Example in Figure 39 for a raw material database like the one presented in the Technical Documentation) is required to contain two columns of which



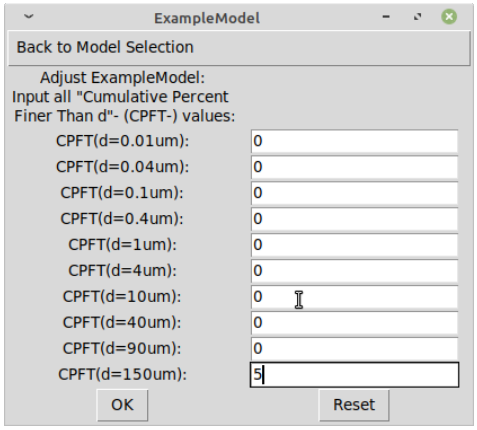
(a) Other model saved?



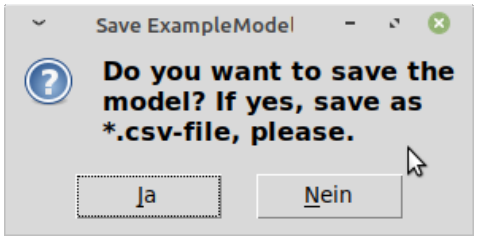
(b) Open other model



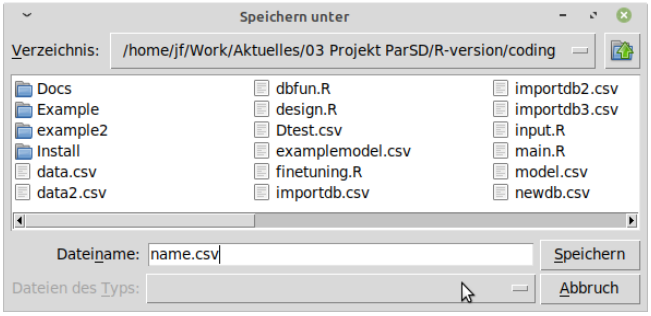
(c) Name model



(d) Input CPFT(d)



(e) Save free model?



(f) Save as CSV file

Figure 38: Define free/other model

the first has to be named 'Diameter' and the name of the second specifies the name of the model. In the Diameter-column then all component sizes of the database which contains the batch raw materials have to be listed in ascending order. In the column named after the model-name, for all fields a CPFT value has to be given. Empty fields are not automatically filled and there are also no interpolations done.

However, there is a work-around implemented. Within the 'Edit material in an existent database' function, the possibility is given to save a material as a model. This gives the possibility to define the model for differing diameters/component sizes but also to define it not as CPFT(d) but as another retention or passthrough curve, too. For more information please refer to the description of this database function in the End User Documentation.

	A	B
1	Diameter	T60_3000-1000-model
2	0,01	0
3	0,04	0
4	0,1	0
5	0,4	0
6	1	0
7	4	0
8	10	0
9	40	0
10	90	0
11	150	0
12	315	0
13	630	0,07
14	1000	2,83
15	1250	9,43
16	2000	42,32
17	2500	67,85
18	3150	96,69
19	4000	100
20	5000	100

Figure 39: Prepared model CSV file

Parameter bounds definition ('Calculate model parameters' main function)

For the model of which the parameters should be calculated to fit a defined batch composition optimally, the bounds are input in a separate window (Figure 40). For the maximum particle size's lower and upper bound, the preset value is the maximum particle size calculable from the given raw materials taking the database's component sizes and the overgraining Setting into consideration. The optimization routine is much simplified if the diameters required for the models are given and do not have to be fitted, too. The fittings are then also less susceptible to errors.

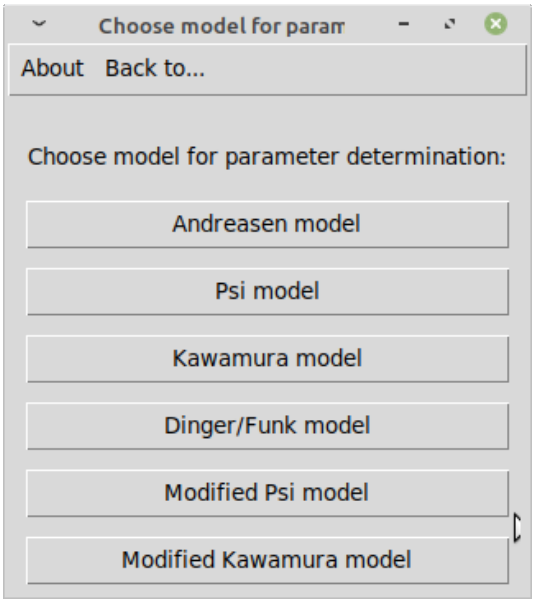
For the inputted bounds, it is checked that lower bounds are smaller or equal to the upper bounds, that not all lower and upper bounds are equal giving a fully defined model and it is checked if the maximum particle size range includes the calculable preset value (Figure 41). This value has not to be giving the best fit, but it can be expected that the optimum maximum particle size is near to it.

3.4 Calculations

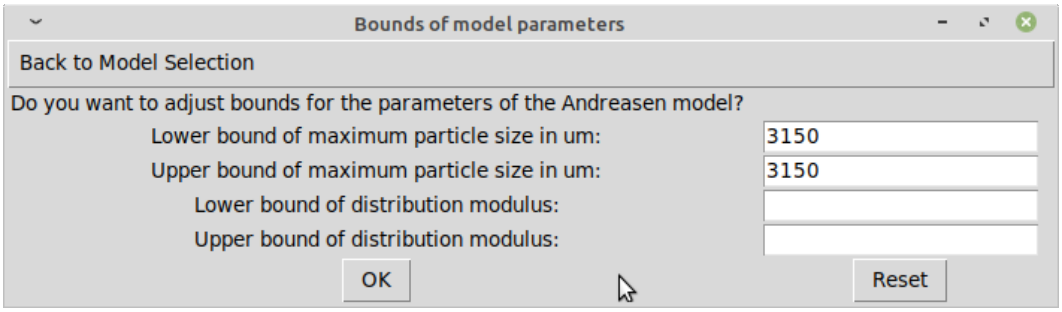
For the 'Verify a recipe' main function, there is no End User interaction as in the software, the batch and model curves are calculated analytically. For the other two main functions, numerical optimizations are done. There is user interaction in the form of a progress output, but also due to possible optimization errors. The possible optimization output differs in dependence on the main function.

Batch optimization ('Design a batch' main function)

With a defined model and a list of raw materials (possibly with set bounds), the batch composition can be calculated by fitting the raw material contents in the way that the resulting



(a) Select model



(b) Specify bounds for model parameters

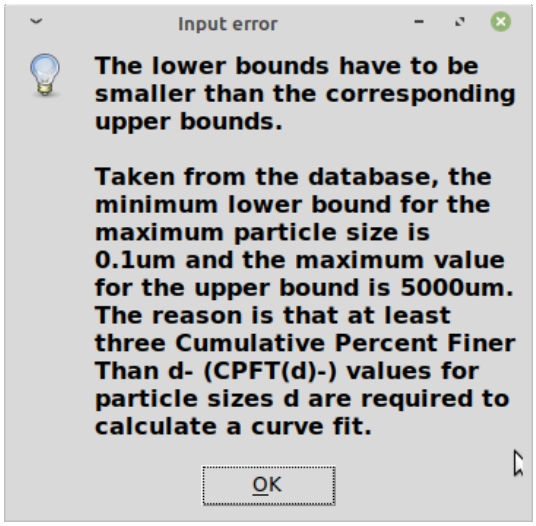
Figure 40: Choose model and specify parameter bounds

batch composition fits the model curve with a minimum squared deviation summed up over all component sizes.

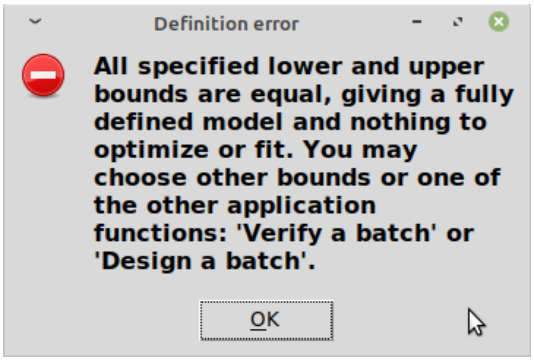
This batch optimization process is parted into two or three steps depending on if it is a volume or weight optimization, respectively. For the volume optimization, it is iterated that the batch adds up to 100 vol% and then the optimal content of every raw material is numerically calculated. If bounds in wt% have to be considered, in-between the two steps iterations take place to put the raw materials in question within the set wt%-bounds. The progress of the optimization is displayed (Figure 42) excluding the last step.

During the optimization, errors can occur. It is possible that due to rounding issues or 'bonded' materials (materials having exactly the same particle size distribution), a solution cannot be calculated. To ensure that the program doesn't run infinitely, on the one hand in the Settings the user specifies a maximum number of iterations (for the vol% and wt% iterations together). If this limit is reached, the user is informed (Figure 43(a)). Possible solutions are:

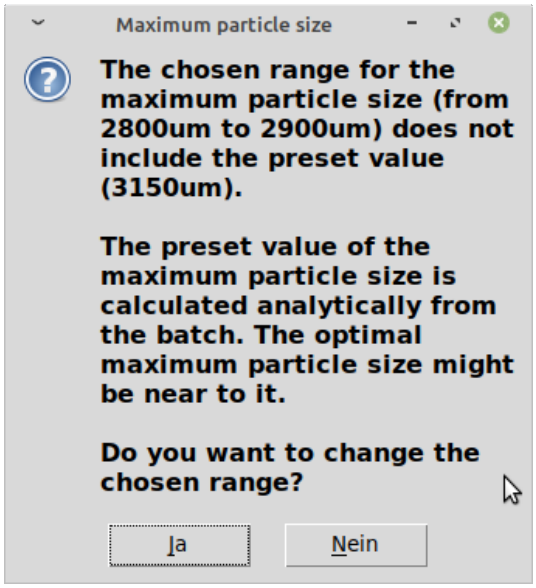
- Setting the iteration limit to a higher value (in Main Window Menu → Settings)
- Changing the accuracy setting (in Main Window Menu → Settings) might help if it is a rounding issue
- Changing bounds might help because this alters the starting values of the optimization process and can influence the result



(a) Bounds error



(b) Definition error



(c) Maximum particle size inquiry

Figure 41: Possible error messages or inquiries

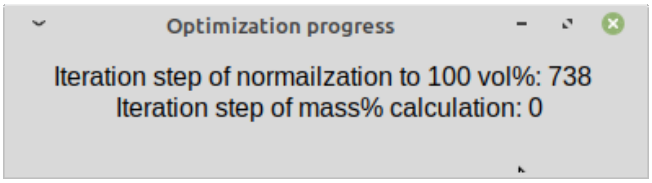
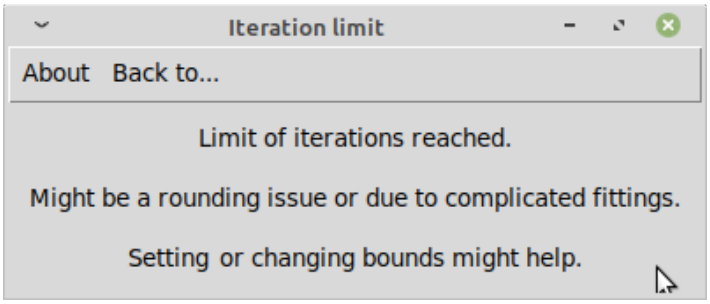


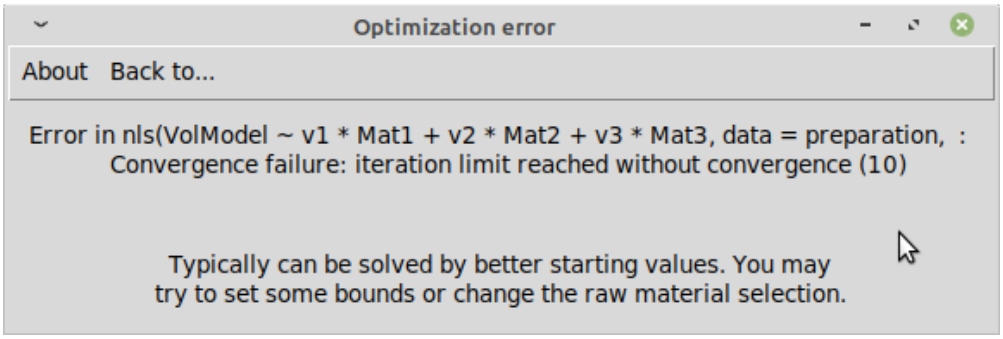
Figure 42: Batch optimization progress

- Changing the raw material selection might help if e.g. 'bonds' exist

If the last optimization step, the calculation of the optimal contents for all mateials which is done by the nls()-function, fails, another error message is displayed (Figure 43(b)). It shows the returned error of the nls()-function and suggests the latter two points listed above to solve the problem as either the starting values are insufficient or the problem is not solvable by a unique set of raw material contents as is for example the case if 'bonds' exist.



(a) Iteration limit reached



(b) Optimization error

Figure 43: Optimization issues

Model optimization ('Calculate model parameters' main function)

With a defined batch composition and a list of model parameters (possibly with set bounds), the parameter values can be calculated by fitting the particle size distributions of the batch and the calculable model to each other minimizing the squared deviation summed up over all component sizes.

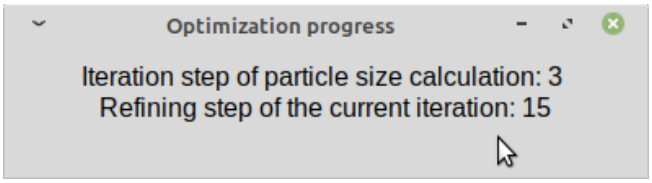
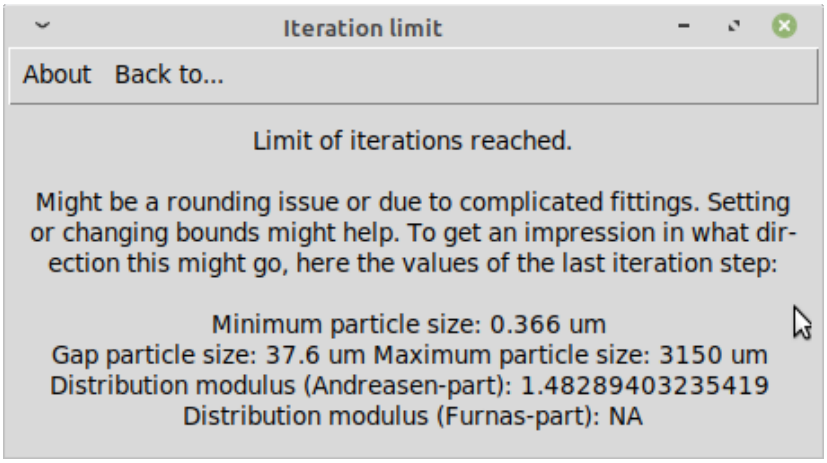


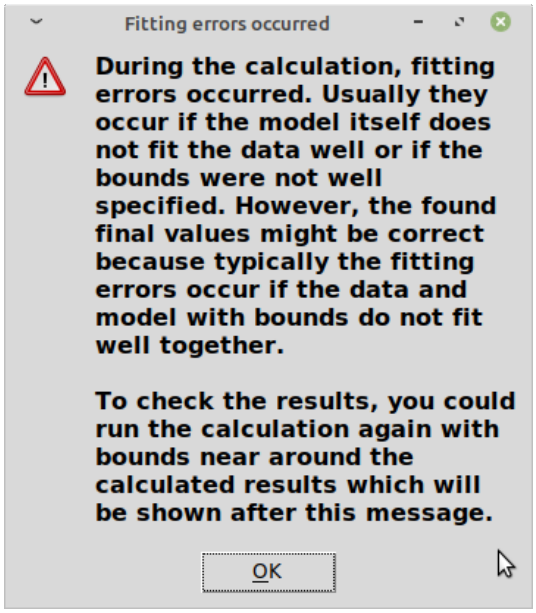
Figure 44: Model optimization progress

This batch optimization process is parted into two steps. The first step is that for the particle sizes—which are not given as constant—a simultaneous optimization of all particle sizes in question is done (referred to as refining steps). For all combinations, the second step, the calculation of the optimal parameters is done. Followingly, the combination of particle sizes returning the lowest of the minimum squared deviations is taken as the result of the first run. Around the particle size values returning this hopefully global minimum now, the ranges for the particle size are slightly narrowed and the whole process starts again. It is proceeded till the lower and upper borders of these ranges for the particle sizes to return minimum values are constant within the set accuracy for the grain sizes (in the Main Window Menu Settings). The progress of the optimization is displayed (Figure 44) showing the overall run in the first line and the refining step in the second line.

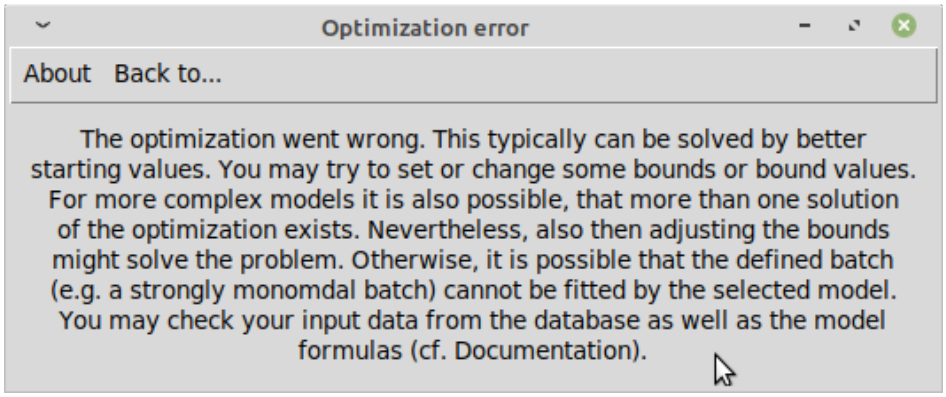
During the optimization, errors can occur. It is possible that due to rounding issues, 'bonded' materials (materials having exactly the same particle size distribution), due to



(a) Iteration limit reached (Main run steps)



(b) Fitting errors (Refining steps)



(c) Optimization error (nls()-functions)

Figure 45: Optimization issues

existing local minima besides the global one or due to the possibility that more than one solution exists (especially for the more complex models with a higher number of parameters), a reasonable single solution cannot be calculated. To ensure that the program doesn't run

infinitely, on the one hand in the Settings the user specifies a maximum number of iterations of the main runs. If this limit is reached, the user is informed (Figure 45(a)). Possible solutions are:

- Setting the iteration limit to a higher value (in Main Window Menu → Settings)
- Changing the accuracy and grain size accuracy settings (in Main Window Menu → Settings) might help if it is a rounding issue
- Changing bounds might help because this alters the starting values of the optimization process and can influence the result. Moreover, running into a local minimum instead of a global one is less probable with reasonable bounds for e.g. the particle sizes. As this is probably the best solution to the issue, for the last iteration run, the parameter values which returned the minimum sum of squared deviations for the fit are also shown in the error message so that the user has a hint in what direction it might go.
- Changing the raw material selection might help if e.g. 'bonds' exist.

It is probable that not for all refining steps the nls()-function finds a solution because some combinations of particle sizes will be just not reasonable. The user will be informed if this happened (Figure 45(b)). Nevertheless, for the reasonable combinations a minimum might be reached and even if this error or caution message appears, the final result should be fine. However, it is also possible—especially for complex models—that the optimizations (nls()-function calls) for all refining steps fail. For this iteration run, then, no minimum exists and the optimization is aborted showing the message in Figure 45(c). To solve this issue, it is recommended to change the bounds or the selected raw materials.

3.5 Results

The display of results differs for the three main functions. In case of the 'Design a batch' and 'Calculate model parameters' main functions, where the respective parameters are fitted by a set accuracy, a fine-tuning of the obtained parameter values is possible.

Results of 'Verify a recipe' main function

The results are presented in a window (Figure 46) with an extended menu bar, followed by the results table containing the batch properties and in the lower part can be found a detailed comparison of the model and batch particle size distributions.

The menu item 'Save' gives the user the possibilities to save the...

- recipe, which is a subset database containing the materials used for the present calculation. Has to be saved as CSV file in the Save-As window by adding the file extension to the filename as shown in Figure 47(a).
- batch, which is the table shown in the results window giving the contents of the materials. Has to be saved as CSV file in the Save-As window by adding the file extension to the filename as shown in Figure 47(a).
- model (comparison), which is the lower part of the results-window including information to the model (type and parameters). Has to be saved as CSV file in the Save-As window by adding the file extension to the filename as shown in Figure 47(a)).
- graph, as adjusted and displayed by the 'Open/close plot' button. Has to be saved as PNG file in the Save-As window by adding the file extension to the filename (cf. Figure 47(a)).
- complete results (recipe, batch, model (comparison) & plot). In this case, a folder will be created into which the four files will be put. The user is asked to give the parent folder into which the new results folder should be placed (Figure 47(b)) and then to give a name for the results folder (Figure 47(c)). If a folder with this name already exists, the user is informed but can decide to overwrite the contents (Figure 47(d)).

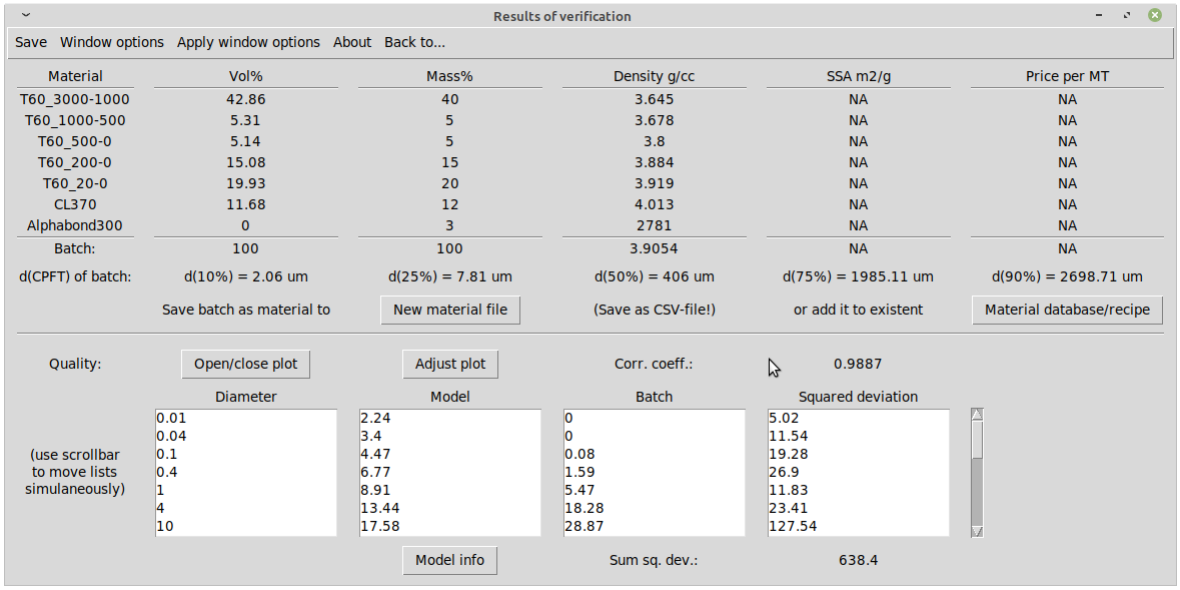
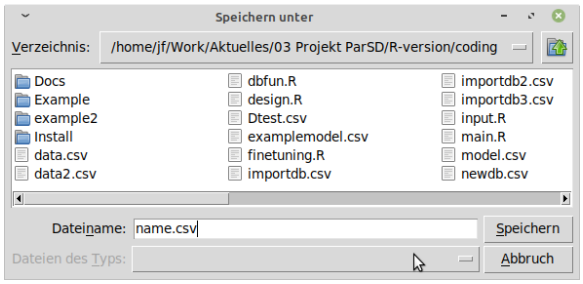
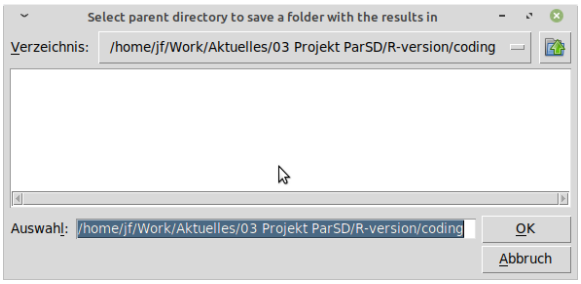


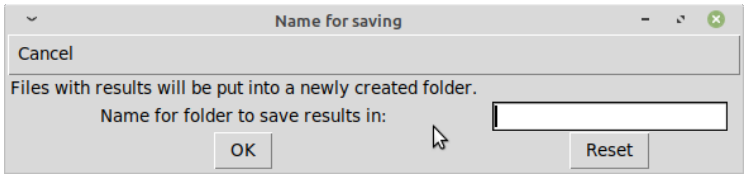
Figure 46: Results window



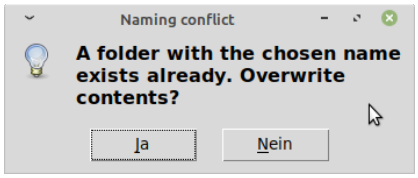
(a) Save-as dialog



(b) Select parent directory



(c) Name results-folder



(d) Select parent directory

Figure 47: Save results

The menu item 'Apply window options' is a button which activates the settings decided for in the menu item 'Window options'. The window options accessible are the listbox height and the distance between the window elements. The listboxes are the boxes in the lower part of the results window showing the comparison of the model and batch particle size distributions. The distance between the window elements also refers to the vertical distance between the lines. The reason is that for batches containing a lot of materials (up to 20), it is possible that the results window could not fit on the screen for a large distance between the window elements.

Coming to the results table (cf. Figure 46), there the materials with their contents in vol% and wt% in the batch are listed together with information from the database if saved there. The last line shows the batch properties. For the batch, furthermore, the describing d(CPFT) values are given to summarize the properties of the batch particle size distribution. The batch can moreover be saved as a new material file (database with one material only)

or it can be added to an existing recipe or database. In the first case, the user is shown the save-as dialog (Figure 47(a)) where the user decides for a path and name for the new CSV (database) file. In the second case, the user selects a database from the filesystem. In both cases, afterwards, the user is asked to give the batch-material information (Figure 48). The fields are pre-filled, but can be edited.

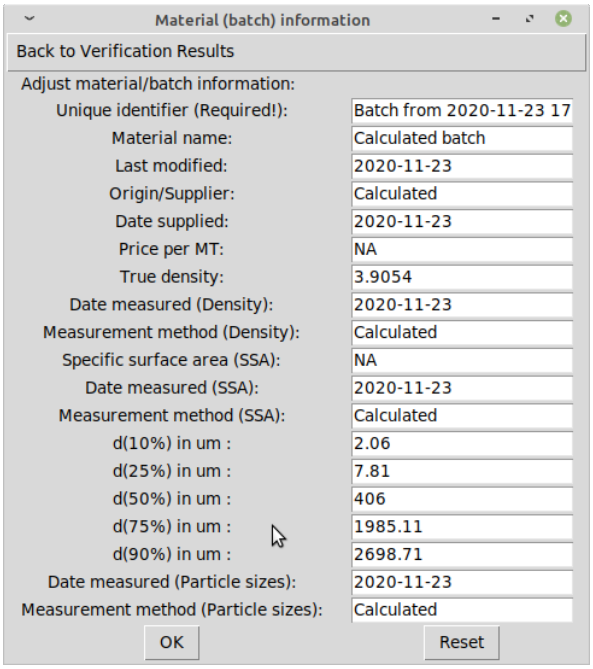


Figure 48: Give batch-material information to save batch as material

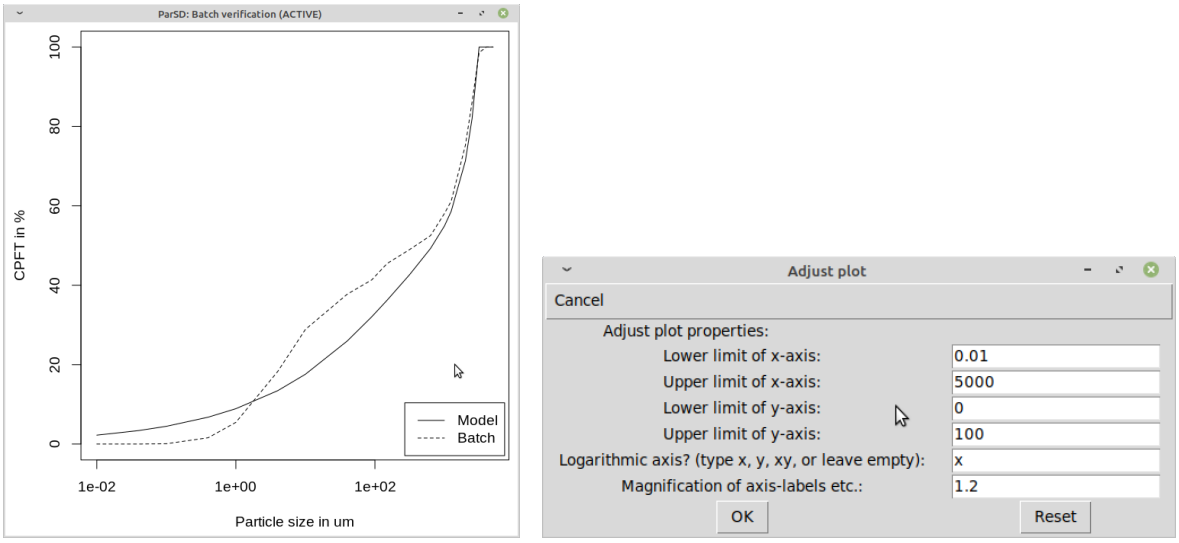
The reason that the Save batch as material-function was included is that by this also relationships between raw materials can be accounted for. An example: Two raw materials should react completely with each other, so their ratio is fixed. If also their together amount is fixed, it can just be defined in the batch (by equal lower and upper bounds for both raw materials), but if their together share of the batch is to be optimized, the user can firstly define their relationship in a batch of two materials, save this batch as 'reaction-phase-material' and work in the further steps with this batch material.

In the lower part of the results window (cf. Figure 46), the quality of the fit is evaluated. The user can evaluate it visually by plotting the results (the CPFT curves) and by adjusting the plot to make details visible (Figure 49). The axis ranges and the magnification of the axis-labels can be adjusted and it can be decided if there should be logarithmic axes.

Additionally, The quality of the fit is described by the two numbers correlation coefficient and sum of squared deviations. Firstly, a correlation coefficient (≤ 1) for the correlation between the batch-CPFT(d) values and model-CPFT(d) values of '1' would describe a perfect fit. The second value is the sum of the squared deviations of the batch-CPFT(d) from the model-CPFT(d) for all component sizes d. The single values are shown in the listboxes by row. In the first listbox the component size d, in the second the model-CPFT(d), in the third the batch-CPFT(d) and in the fourth listbox the squared deviation is shown. Furthermore, the model information (type and parameters) can be presented (Figure 50) by clicking on the button 'Model info'.

Results and fine-tuning of 'Design a batch' main function

For the 'Design a batch' main function, the display of the results and the fine-tuning are parted because the user can decide to fine-tune the batch in vol% or wt%.



(a) Graph (b) Adjust plot

Figure 49: Plot results

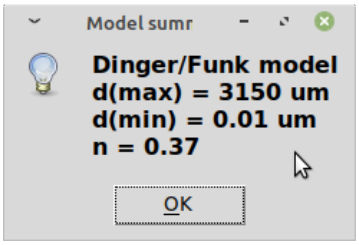


Figure 50: Model info

a) Results of 'Design a batch' main function

The results are presented in a window (Figure 51) with an extended menu bar, followed by the results table containing the batch properties and in the lower part can be found a comparison of the model and batch particle size distributions.

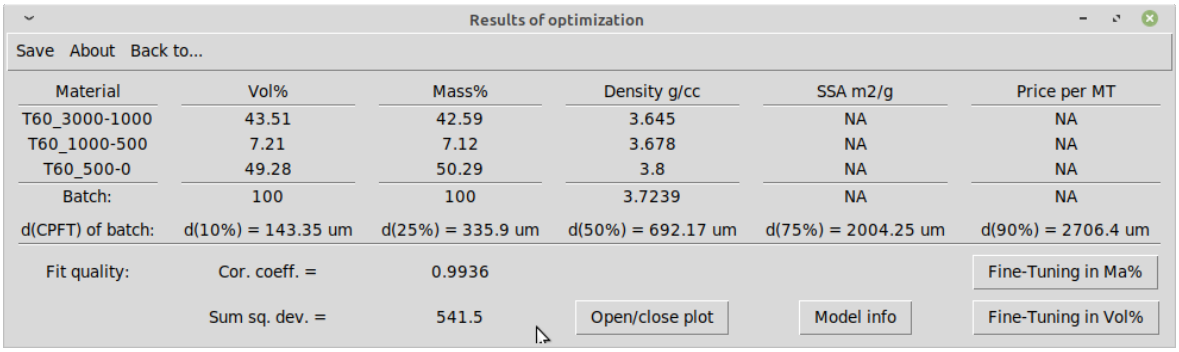
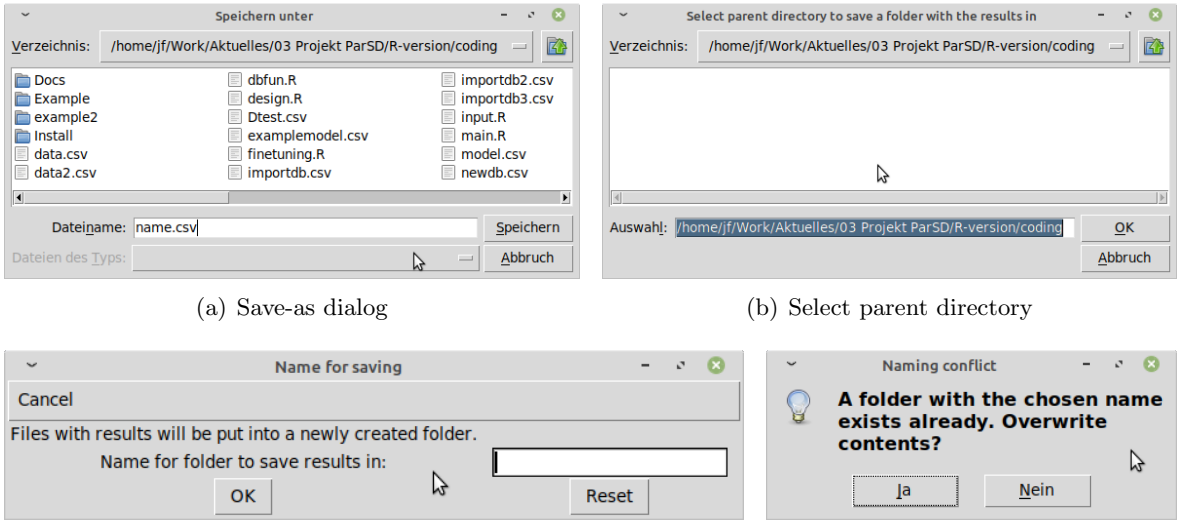


Figure 51: Results window

The menu item 'Save' gives the user the possibilities to save the...

- recipe, which is a subset database containing the materials used for the present calculation. Has to be saved as CSV file in the Save-As window by adding the file extension to the filename as shown in Figure 52(a).

- batch, which is the table shown in the results window giving the contents of the materials. Has to be saved as CSV file in the Save-As window by adding the file extension to the filename as shown in Figure 52(a).
- model (comparison), which is the lower part of the results-window including information to the model (type and parameters). Has to be saved as CSV file in the Save-As window by adding the file extension to the filename as shown in Figure 52(a).
- graph, as adjusted and displayed by the 'Open/close plot' button. Has to be saved as PNG file in the Save-As window by adding the file extension to the filename (cf. Figure 52(a)).
- complete results (recipe, batch, model (comparison) & plot). In this case, a folder will be created into which the four files will be put. The user is asked to give the parent folder into which the new results folder should be placed (Figure 52(b)) and then to give a name for the results folder (Figure 52(c)). If a folder with this name already exists, the user is informed but can decide to overwrite the contents (Figure 52(d)).



(a) Save-as dialog

(b) Select parent directory

(c) Name results-folder

(d) Select parent directory

Figure 52: Save results

Coming to the results table (cf. Figure 51), there the materials with their contents in vol% and wt% in the batch are listed together with information from the database if saved there. The last line shows the calculable batch properties. For the batch, furthermore, the describing d(CPFT) values are given to summarize the properties of the batch particle size distribution.

In the lower part of the results window (Figure 51), the quality of the fit is evaluated. The user can evaluate it visually by plotting the CPFT curves (Figure 53). The quality of the fit is moreover described by two numbers. Firstly, by the correlation coefficient (≤ 1) for the correlation between the batch-CPFT(d) values and model-CPFT(d) values for which '1' would describe a perfect fit. The second value is the sum of the squared deviations of the batch-CPFT(d) from the model-CPFT(d) for all component sizes d. Furthermore, the model information (type and parameters) can be presented (Figure 54) by clicking on the button 'Model info'.

Further-on, the two fine-tuning buttons lead the user to results dialogs with (1) more information presented and (2) with the possibility to re-adjust either the vol% or wt% values and re-calculate the fit for the new parameters. By this for example differing accuracies for different raw material types (aggregates, additives, ...) can be adjusted.

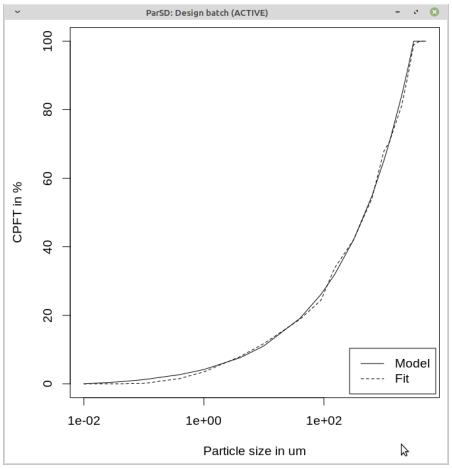


Figure 53: Plot results

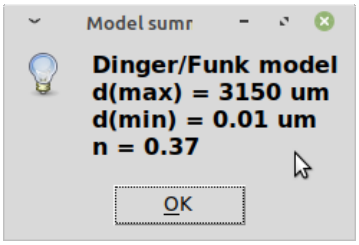


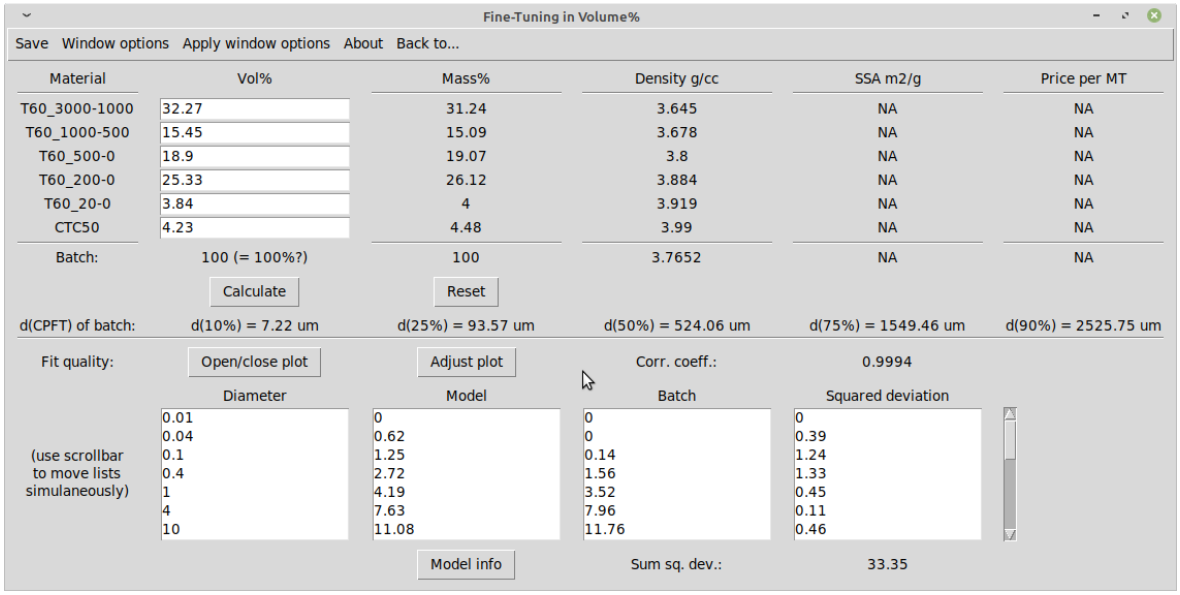
Figure 54: Model info

b) Fine-tuning of results of 'Design a batch' main function

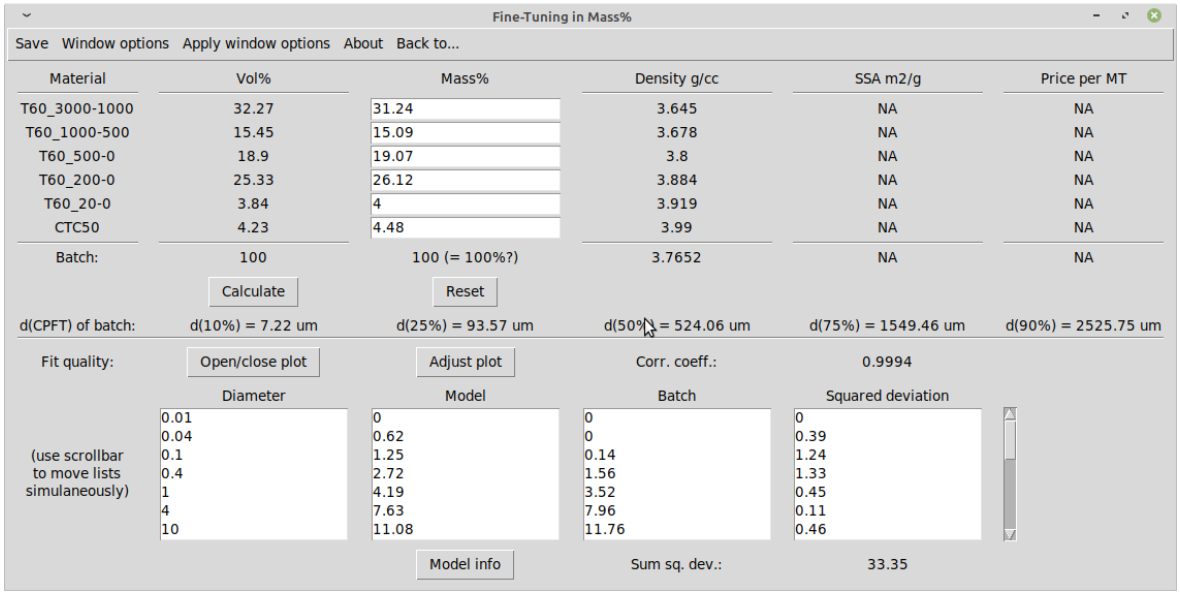
The fine-tuning windows for either the vol% or wt% version differ only by the column which is editable, cf. Figure 55. The windows have an extended menu bar, followed by the results table containing the batch properties and in the lower part can be found a detailed comparison of the model and batch particle size distributions characterizing the quality of the fit.

The menu item 'Save' gives the user the possibilities to save the fine-tuned...

- recipe, which is a subset database containing the materials used for the present calculation. Has to be saved as CSV file in the Save-As window by adding the file extension to the filename as shown in Figure 56(a).
- batch, which is the table shown in the results window giving the contents of the materials. Has to be saved as CSV file in the Save-As window by adding the file extension to the filename as shown in Figure 56(a).
- model (comparison), which is the lower part of the results-window including information to the model (type and parameters). Has to be saved as CSV file in the Save-As window by adding the file extension to the filename as shown in Figure 56(a).
- graph, as adjusted and displayed by the 'Open/close plot' button. Has to be saved as PNG file in the Save-As window by adding the file extension to the filename (cf. Figure 56(a)).
- complete results (recipe, batch, model (comparison) & plot). In this case, a folder will be created into which the four files will be put. The user is asked to give the parent folder into which the new results folder should be placed (Figure 56(b)) and then to give a name for the results folder (Figure 56(c)). If a folder with this name already exists, the user is informed but can decide to overwrite the contents (Figure 56(d)).



(a) Vol% fine-tuning

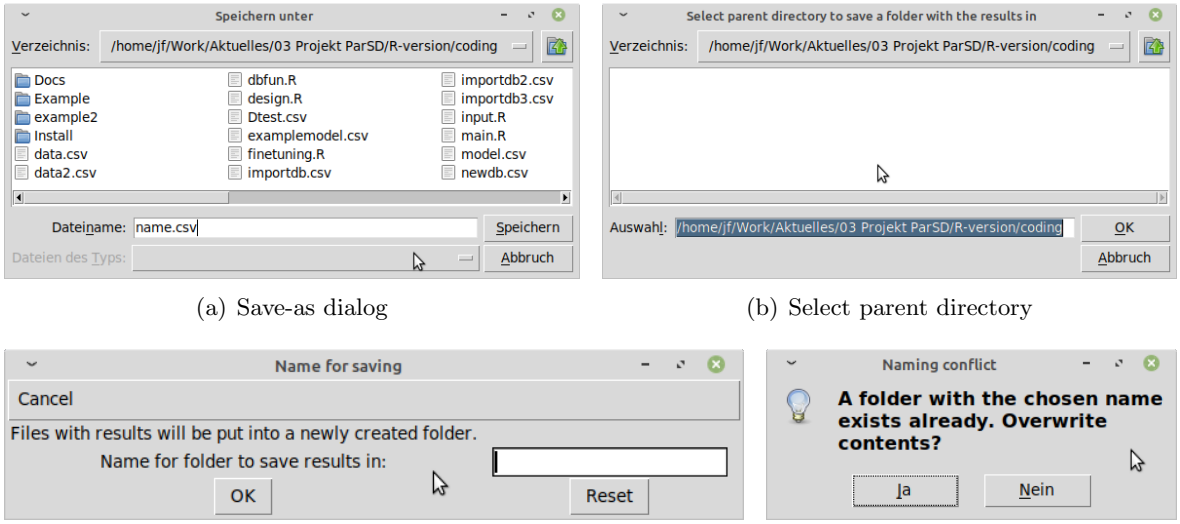


(b) Wt% fine-tuning

Figure 55: Fine-tuning dialogs for fine-tuning in vol% or wt%

The menu item 'Apply window options' is a button which activates the settings decided for in the menu item 'Window options'. The window options accessible are the listbox height and the distance between the window elements. The listboxes are the boxes in the lower part of the results window showing the comparison of the model and batch particle size distributions. The distance between the window elements also refers to the vertical distance between the lines. Reason is that for batches containing a lot of materials (up to 20), it is possible that the results window could not fit on the screen for a large distance between the window elements.

Coming to the table in the upper part (cf. Figure 55), there the materials with their contents in the batch are listed together with information from the database if saved there. For vol% fine-tuning, the vol%-column is editable and for wt% fine-tuning the wt%-column. The last line of the table shows the batch properties containing also the calculable (true) batch density, its specific surface area and the costs. By editing (fine-tuning) the contents



(a) Save-as dialog

(b) Select parent directory

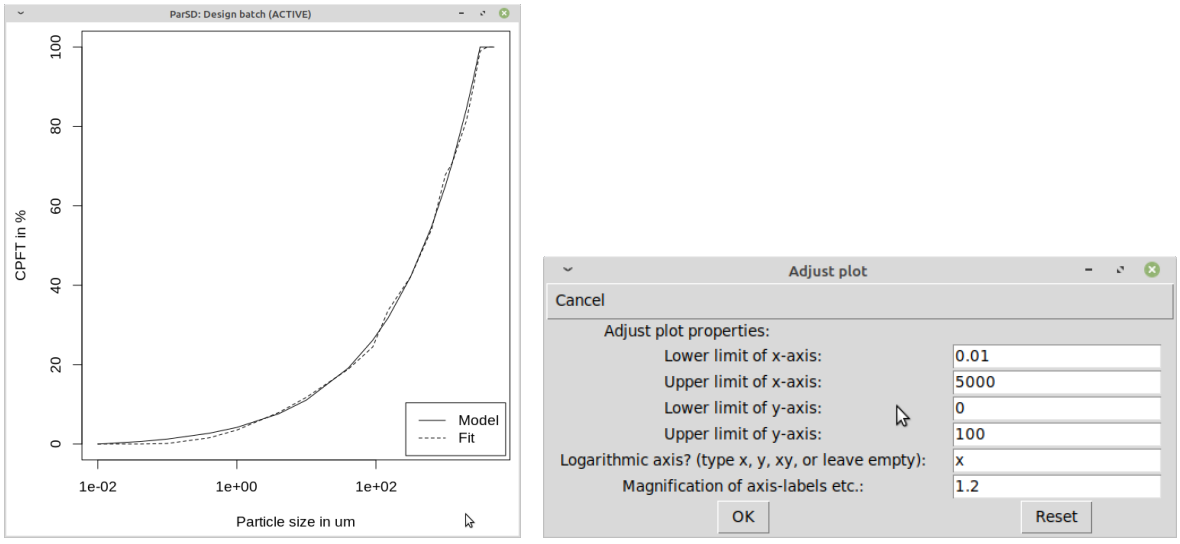
(c) Name results-folder

(d) Select parent directory

Figure 56: Save fine-tuned results

and then clicking 'Calculate', the complete set-up (batch and fit) is re-calculated for the new values. By the button 'Reset' it is always possible to get again the original non-fine-tuned results. It can be noted that by resetting, the unrounded values are shown. Caution: It is not checked if the edited values sum up to 100 %. In the line below the buttons, for the batch the describing d(CPFT) values are given to summarize the properties of the batch particle size distribution.

In the lower part of the fine-tuning windows (Figure 55), the quality of the fit is evaluated. The user can evaluate it visually by plotting the results (the CPFT curves) and by adjusting the plot to make details visible (Figure 57). The axis ranges and the magnification of the axis-labels can be adjusted and it can be decided if there should be logarithmic axes.



(a) Graph

(b) Adjust plot

Figure 57: Plot fine-tuned results

The quality of the fit is described by the two numbers correlation coefficient and sum of squared deviations. Firstly, a correlation coefficient (≤ 1) for the correlation between the

batch-CPFT(d) values and model-CPFT(d) values of '1' would describe a perfect fit. The second value is the sum of the squared deviations of the batch-CPFT(d) from the model-CPFT(d) for all component sizes d. The single values are shown in the listboxes by row. In the first listbox the component size d, in the second the model-CPFT(d), in the third the batch-CPFT(d) and in the fourth listbox the squared deviation is shown. Furthermore, the model information (type and parameters) can be presented (Figure 58) by clicking on the button 'Model info'.

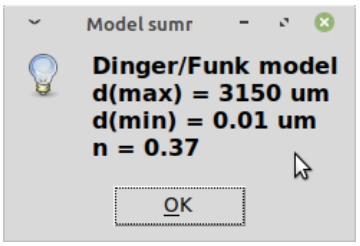


Figure 58: Model info

Results of 'Calculate model parameters' main function and their fine-tuning

The results and fine-tuning window (Figure 59) has an extended menu bar, followed by the parameter table of the model and in the lower part can be found a detailed comparison of the model and batch particle size distributions characterizing the quality of the fit.

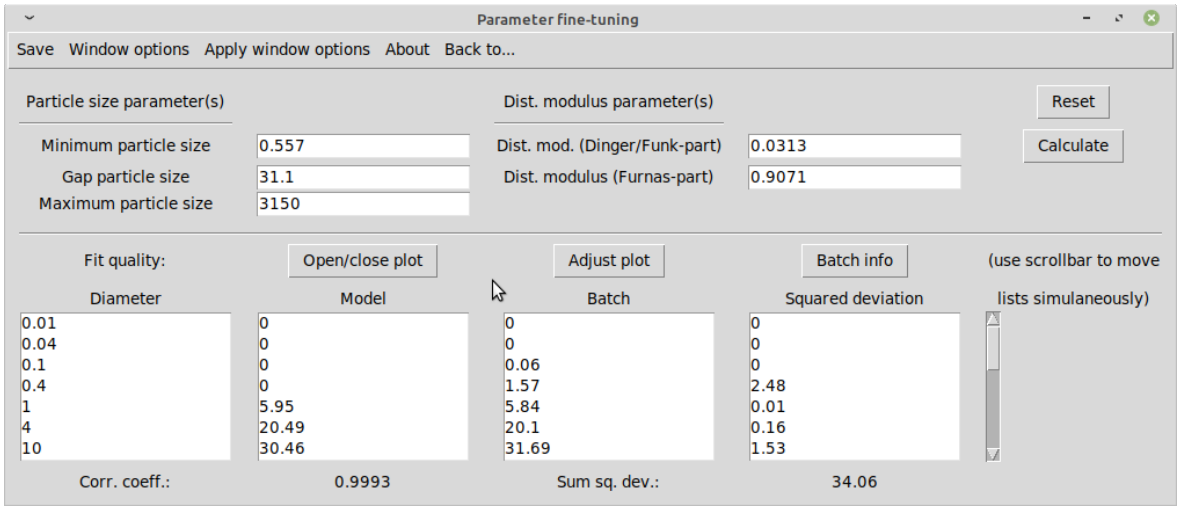
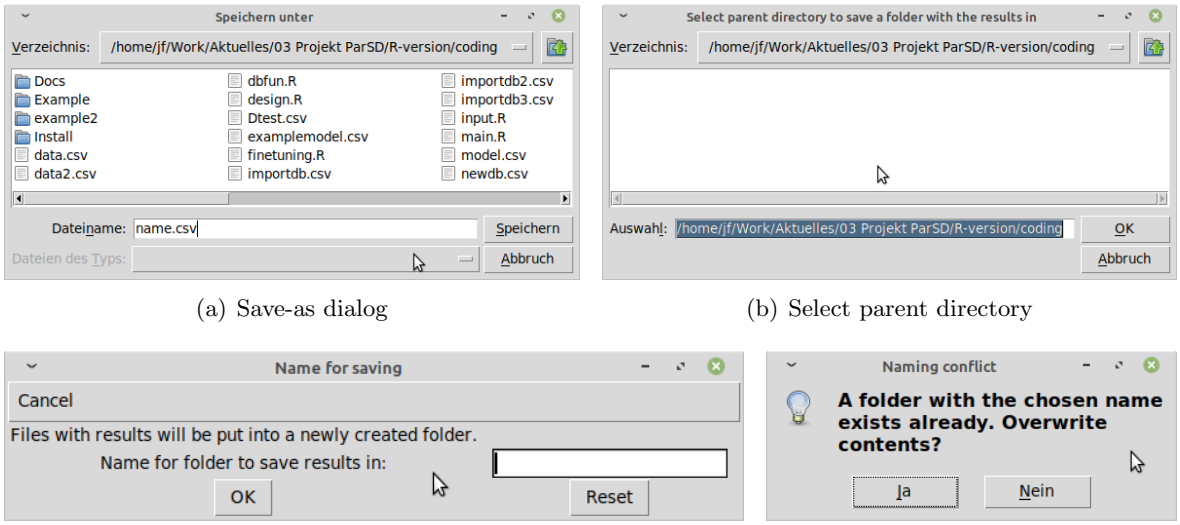


Figure 59: Results and fine-tuning dialog

The menu item 'Save' gives the user the possibilities to save the fine-tuned...

- recipe, which is a subset database containing the materials used for the present calculation. Has to be saved as CSV file in the Save-As window by adding the file extension to the filename as shown in Figure 60(a).
- batch, which is the table shown in the results window giving the contents of the materials. Has to be saved as CSV file in the Save-As window by adding the file extension to the filename as shown in Figure 60(a).
- model (comparison), which is the lower part of the results-window including information to the model (type and parameters). Has to be saved as CSV file in the Save-As window by adding the file extension to the filename as shown in Figure 60(a).

- graph, as adjusted and displayed by the 'Open/close plot' button. Has to be saved as PNG file in the Save-As window by adding the file extension to the filename (cf. Figure 60(a)).
- complete results (recipe, batch, model (comparison) & plot). In this case, a folder will be created into which the four files will be put. The user is, thus, asked to give the parent folder into which the new results folder should be placed (Figure 60(b)) and then to give a name for the results folder (Figure 60(c)). If a folder with this name already exists, the user is informed but can decide to overwrite the contents (Figure 60(d)).



(a) Save-as dialog

(b) Select parent directory

(c) Name results-folder

(d) Select parent directory

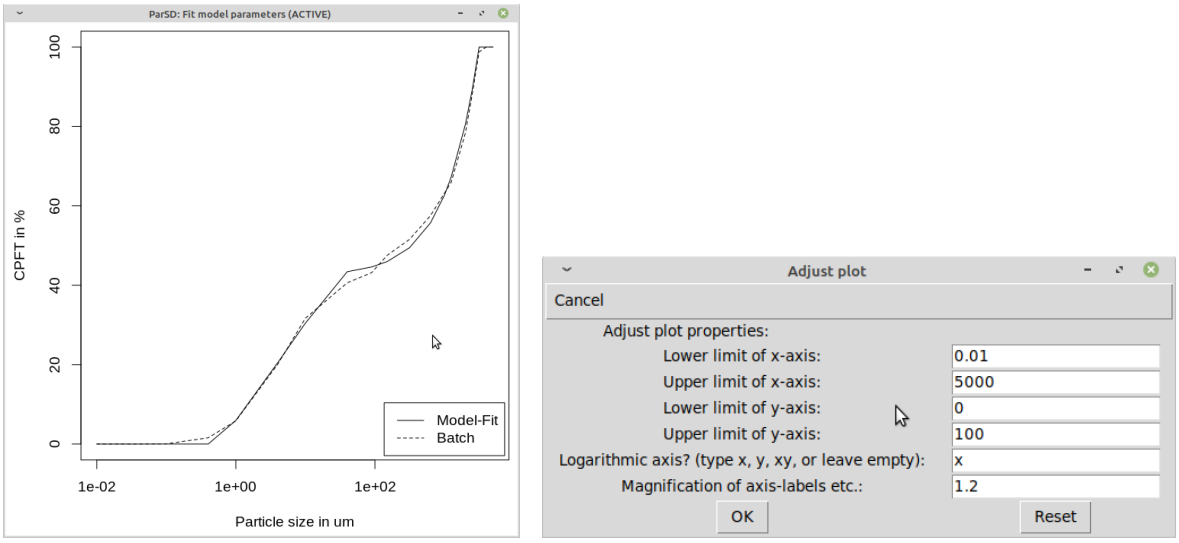
Figure 60: Save results

The menu item 'Apply window options' is a button which activates the settings decided for in the menu item 'Window options'. The window options accessible are the listbox height and the distance between the window elements. The listboxes are the boxes in the lower part of the results window showing the comparison of the model and batch particle size distributions. The distance between the window elements also refers to the vertical distance between the lines.

Coming to the editable parameter table in the upper part (cf. Figure 59), there the optimized parameters of the chosen model are presented. They can be edited (fine-tuned) and by clicking 'Calculate', the complete set-up (batch and fit) is re-calculated for the new values. By the button 'Reset' it is always possible to get again the original non-fine-tuned parameter results.

In the lower part of the fine-tuning window (Figure 59), the quality of the fit is evaluated. The user can evaluate it visually by plotting the fine-tuned results (the CPFT curves) and by adjusting the plot to make details visible (Figure 61). The axis ranges and the magnification of the axis-labels can be adjusted and it can be decided if there should be logarithmic axes.

The quality of the fit is described by the two numbers correlation coefficient and sum of squared deviations. Firstly, a correlation coefficient (≤ 1) for the correlation between the batch-CPFT(d) values and model-CPFT(d) values of '1' would describe a perfect fit. The second value is the sum of the squared deviations of the batch-CPFT(d) from the model-CPFT(d) for all component sizes d. The single values are shown in the listboxes by row. In the first listbox the component size d, in the second the model-CPFT(d), in the third the batch-CPFT(d) and in the fourth listbox the squared deviation is shown. Furthermore, the batch information can be displayed (Figure 62) by clicking on the button 'Batch info'. It



(a) Graph (b) Adjust plot

Figure 61: Plot results

shows a table where the materials with their contents in the batch are listed together with information from the database if saved there. The last line of the table shows the batch properties containing also the calculable (true) batch density, its specific surface area and the costs. Below, for the batch the describing d(CPFT) values are given to summarize the properties of the batch particle size distribution.

Batch information

Back to Parameter Fine-tuning

Material	Vol%	Mass%	Density g/cc	SSA m2/g	Price per MT
T60_3000-1000	37.6	35	3.645	NA	NA
T60_1000-500	5.32	5	3.678	NA	NA
T60_500-0	10.3	10	3.8	NA	NA
T60_200-0	10.08	10	3.884	NA	NA
T60_20-0	24.98	25	3.919	NA	NA
CL370	11.71	12	4.013	NA	NA
Alphabond300	0	3	2781	NA	NA
Batch:	100	100	3.9157	NA	NA
d(CPFT) of batch:	d(10%) = 1.87 um	d(25%) = 6.54 um	d(50%) = 251.14 um	d(75%) = 1799.25 um	d(90%) = 2625.17 um

Figure 62: Batch info

GNU Free Documentation License

Version 1.3, 3 November 2008

<https://fsf.org/>

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "**Document**", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "**you**". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "**Modified Version**" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "**Secondary Section**" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "**Invariant Sections**" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under

this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "**Cover Texts**" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "**Transparent**" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "**Opaque**".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "**Title Page**" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "**publisher**" means any person or entity that distributes copies of the Document to the public.

A section "**Entitled XYZ**" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "**Acknowledgements**", "**Dedications**", "**Endorsements**", or "**History**".) To "**Preserve the Title**" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and

any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation 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. See `https://www.gnu.org/licenses/`.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Bibliography

[AA30] A.H.M. Andreasen and J. Andersen. "Ueber die Beziehung zwischen Kornabstufung und Zwischenraum in Produkten aus losen Körnern (On the relationship of grading and interstices in products of loose grains)". In: *Colloid & Polymer Science* 50 (3 1930), pp. 217–228.

[DF92] D.R. Dinger and J.E. Funk. "Particle Packing III – Discrete versus continuous particle sizes". In: *Interceram* 41.5 (1992), pp. 176–179.

[DF93] D.R. Dinger and J.E. Funk. "Particle Packing IV – Computer Modelling of Particle Packing Phenomena". In: *Interceram* 42.3 (1993), pp. 150–152.

[DGW81a] J.E. Dennis, Jr., D.M. Gay, and R.E. Welsch. "Algorithm 573 NL2SOL—An adaptive nonlinear least-squares algorithm [E4]". In: *ACM Transactions on Mathematical Software* 7.3 (1981), pp. 369–83.

[DGW81b] J.E. Dennis, Jr., D.M. Gay, and R.E. Welsch. "An adaptive nonlinear least-squares algorithm". In: *ACM Transactions on Mathematical Software* 7.3 (1981), pp. 348–68.

[FA14] J. Fruhstorfer and C.G. Aneziris. "The Influence of the Coarse Fraction on the Porosity of Refractory Castables". In: *Journal of Ceramic Science and Technology* 5.2 (2014), pp. 155–166.

[FA17] J. Fruhstorfer and C.G. Aneziris. "Influence of particle size distributions on the density and density gradients in uniaxial compacts". In: *Ceramics International* 43 (2017), pp. 13175–84.

[FD92] J.E. Funk and D.R. Dinger. "Particle Packing Part I – Fundamentals of particle packing monodisperse spheres". In: *Interceram* 41.1 (1992), pp. 10–14.

[FD94] J.E. Funk and D.R. Dinger. *Predictive process control of crowded particulate suspensions*. Kluwer Academic Publishers, 1994.

[FHA19] J. Fruhstorfer, J. Hubálková, and C.G. Aneziris. "Particle packings minimizing density gradients of coarse-grained compacts". In: *Journal of the European Ceramic Society* 39.10 (2019), pp. 3264 –3276. ISSN: 0955-2219.

[Fru16] J. Fruhstorfer et al. "Corrosion of carbon ree and bonded refractories for application in steel ingot casting". In: *steel research international* 87.8 (2016), pp. 1014–23.

[Fru18] J. Fruhstorfer et al. "How the coarse fraction influences the microstructure and the effective thermal conductivity of alumina castables—An experimental and numerical study". In: *Journal of the European Ceramic Society* 38.1 (2018), pp. 303–12.

[Fru19a] J. Fruhstorfer. "Continuous gap-graded particle packing designs". In: *Materials Today Communications* 20.100550 (2019).

[Fru19b] J. Fruhstorfer. "Design of particle size distributions for dense refractory castables and compacts used in high-temperature applications". PhD thesis. TU Bergakademie Freiberg, 2019.

[Fru21] J. Fruhstorfer. "ParSD — Tool to design and analyze particle size distributions". In: *SoftwareX* 100753 (2021).

[Fru22] J. Fruhstorfer et al. “Influence of the fine fraction and sintering on selected isotropic and anisotropic bulk properties of uniaxial compacts”. In: *Interceram* (2022). in press.

[Ful05] W.B. Fuller. “A treatise on concrete, plain and reinforced: materials, construction, and design of concrete and reinforced concrete”. In: ed. by F.W. Taylor and S.E. Thompson. John Wiley & Sons, New York, 1905. Chap. XI Proportioning Concrete, pp. 183–216.

[Fur31] C.C. Furnas. “Grading Aggregates I - Mathematical Relations for Beds of Broken Solids of Maximum Density”. In: *Industrial and Engineering Chemistry* 23.9 (1931), pp. 1052–1058.

[Hof92] R.L. Hoffman. “Factors affecting the viscosity of unimodal and multimodal colloidal dispersions”. In: *Journal of Rheology* 36.5 (1992), pp. 947–964.

[KAO71] J. Kawamura, E. Aoki, and K. Okuzawa. “Calculation of packing volume of multicomponent particle systems”. In: *Chemical Engineering* 35.7 (1971), pp. 777–83.

[Kaw73] J. Kawamura et al. “Particle-size distribution dependence of properties for magnesia refractories”. In: *Yogyo-Kyokai-Shi* 81.8 (1973), pp. 313–21.

[Kaw77] J. Kawamura et al. “Effect of particle-size distribution on characteristics of castable refractory”. In: *Yogyo-Kyokai-Shi* 85.3 (1977), pp. 101–10.

[KHK73] J. Kawamura, K. Hara, and K. Kawasaki. “Survey for model of particle-size distribution and calculation of packing volume of multi-component particle-size system in refractories batches”. In: *Yogyo-Kyokai-Shi* 81.9 (1973), pp. 391–8.

[Lar99] F. De Larrard. “Concrete Mixture Proportioning: A scientific approach”. In: Modern Concrete Technology. E & FN SPON, an imprint of Routledge, London, 1999.

[LS94] F. De Larrard and T. Sedran. “Optimization of ultra-high-performance concrete by the use of a packing model”. In: *Cement and Concrete Research* 24.6 (1994), pp. 997–1009.

[LZK04] W.E. Lee, S. Zhang, and M. Karakus. “Refractories: Controlled microstructure composites for extreme environments”. In: *Journal of Materials Science* 39 (2004), pp. 6675–6685.

[Mat] Elkem Silicon Materials. *EMMA (Elkem Materials-Mixture Analyzer) software*. http://www.materials.elkem.com/.

[MJ13] M.N. Mangulkar and S.S. Jamkar. “Review of particle packing theories used for concrete mix proportioning”. In: *International Journal of Scientific & Engineering Research* 4.5 (2013), pp. 143–8.

[Myh08] B. Myhre. “Let’s make a castable! Part I”. In: *Refractories Applications and News* 13.3 (2008), pp. 16–24.

[Myh94] B. Myhre. “The effect of particle-size distribution on flow of refractory castables”. In: *The American Ceramic Society 30th Annual Refractories Symposium*. March 25. St. Louis, Missouri, 1994, pp. 3–17.

[Por] *Port library documentation*. http://www.netlib.org/port/.

[Ree95] J.S. Reed. *Principles of ceramic processing*. 2nd edition. John Wiley & Sons, Inc., 1995.

[Rou97] G. Routschka. “Feuerfeste Werkstoffe (Refractory materials)”. In: 2nd edition. Vulkan-Verlag Essen, 1997, pp. 8, 17, 21.

[SSD06] A.P. Silva, A.M. Segadaes, and T.C. Devezas. “Particle distribution design in a self-flow alumina refractory castable without cement”. In: *Advances in Science and Technology* 45 (2006), pp. 2260–2265.

[TR23] A.N. Talbot and F.E. Richart. “The strength of concrete—Its relation to the cement aggregates and water”. In: *University of Illinois Engineering Experiment Station* 137 (1923).

[TU98] F. Tomsu and J. Ulbricht. “Entwicklungstrends der stofflichen Zusammensetzung hochbeanspruchter Feuerbetone (Development trends of the material composition of highly stressed refractory concrete)”. In: *Keramische Zeitschrift* 50.9 (1998), pp. 724–731.

[UB99] J. Ulbricht and K. Burkhardt. “Über den Körnungsaufbau grobkeramischer feuerfester Werkstoffe (On the grading of coarse ceramic refractory materials)”. In: *Ceramics Forum International* 76.4 (1999), pp. D9–12.

[WH30] A.E.R. Westman and H.R. Hugill. “The packing of particles”. In: *Journal of the American Ceramic Society* 13.10 (1930), pp. 767–79.

[WIW89] K. Watanabe, M. Ishikawa, and M. Wakamatsu. “Rheology of castable refractories”. In: *Taikabutsu Overseas* 9.1 (1989), pp. 41–53.

[YWH00] S.-P. Yu, M.-C. Wang, and M.-H. Hon. “Properties of Alumina-Zirconia-Graphite Refractories with Particle Size according to the Andreasen Model”. In: *Journal of the Ceramic Society of Japan* 108.8 (2000), pp. 721–727.

[Zie21] T. Zienert et al. “Synthesis of Niobium-Alumina Composite Aggregates and Their Application in Coarse-Grained Refractory Ceramic-Metal Castables”. In: *Materials* 14.6453 (2021).