ALAMO user manual and installation guide v. 2023.1.22

January 22, 2023

For information about this software, contact Nick Sahinidis at niksah@minlp.com.

Contents

1 Introduction

The purpose of ALAMO (Automatic Learning of Algebraic MOdels) is to generate algebraic surrogate models of black-box systems for which a simulator or experimental setup is available.

Consider a system for which the outputs z are an unknown function f of the system inputs x. The software identifies a function f, i.e., a relationship between the inputs and outputs of the system, that best matches data (pairs of x and corresponding z values) that are collected via simulation or experimentation. ALAMO can:

- build an algebraic model of a simulation or experimental black-box system
- use previously collected data for model building
- call a user-specified function (simulator) to collect measurements
- enforce response variable bounds, physical limits, and boundary conditions
- use a preexisting data set for model validation
- output models in simple algebraic form

The problems addressed by the software have long been studied in the fields of statistics, design of experiments, and machine learning. Whereas existing techniques from this literature can be used to fit data to models, the main challenges addressed by the ALAMO software are in determining where to run the simulations or experiments, what models to fit, and how to determine if the model is accurate and as simple as possible. A distinguishing feature of ALAMO is that it provides models that are accurate yet as simple as possible. Moreover, ALAMO is capable of utilizing theory-driven insights alongside data. The ALAMO models can be used to facilitate subsequent system analysis, optimization, and decision making.

1.1 Licensing and software requirements

The code is available for download at http://minlp.com/alamo. The same URL provides information about licensing the software.

1.2 Installation

Install ALAMO and the ALAMO license file in any directory of your choice and add it to your path. On Windows, ALAMO's installer will take care of these steps for you. On Linux and OSX systems, unzip the ALAMO download and place the files it contains in a directory in your path. Users should not open the license file in an editor as this may invalidate the license in certain operating systems. For all operating systems, make sure that ALAMO and the ALAMO license file are readable by all intended users on your machine.

2 Algorithms implemented

ALAMO seeks to identify low-complexity surrogate models using a minimal amount of data for a system that is described by a simulator or experiment. Surrogate models are constructed using a three-step process. In the first step, an initial design of experiments is generated and the simulation is queried at these points. In the second step, an algebraic model is built using this initial training set. The model is built using integer optimization techniques to select the best subset from a collection of potential sets of basis functions that can be used to build up the model. In the third step, an adaptive sampling methodology based on derivative-free optimization techniques is used to identify points where the model is inaccurate. Once these points are added to the training set, execution returns to the second step of the algorithm. The process continues until the third step confirms the accuracy of a previously built model.

Compared to common techniques, such as forward- or backward-regression, that investigate model sensitivities with respect to one basis function at a time, ALAMO's best subset selection techniques ensure that its model-building steps account for the synergistic effects between different basis functions. Before ALAMO, best subset selection techniques were considered too time consuming for application to realistic data sets. While developing ALAMO, nonlinear integer programming techniques were devised that rely on the BARON software to solve these models in realistic computing times for many industrially relevant systems. ALAMO is also unique in that it utilizes derivative-free optimization techniques in its adaptive sampling step. These techniques offer a systematic approach to interrogate models, identify weaknesses, and guide experimental design towards parts of the space requiring more attention. Another distinctive feature of ALAMO is its constrained regression feature, which is capable of enforcing theory-driven requirements on response variables, including response variable bounds, thermodynamic limitations, and boundary conditions. To enforce these requirements over the entire domain of input variables, ALAMO relies on BARON to solve semi-infinite nonconvex optimization problems.

The bibliography at the end of this document offers more details of the methodology implemented in ALAMO and demonstrates the advantages of this methodology in comparison to currently utilized approaches, including classical regression and the lasso.

3 Running ALAMO

3.1 Running the interactive GUI

The ALAMO GUI allows for a convenient spreadsheet-style input of problem data and algorithmic options, and offers facilities for visualizing the data and results, and saving your work. On all operating systems, Java is required in order to use the ALAMO GUI. Additionally, the user must have write privileges in the directory where the GUI is invoked.

On Windows systems, the installer will create an ALAMO icon on your desktop. Double click at this icon in order to launch ALAMO's interactive GUI.

On Linux and OSX, the interactive GUI is available in the download directory and named alamogui.jar. To run this, open a terminal and type the following command at the prompt

```
java -jar alamogui.jar
```

On all systems, the GUI allows you to save the model input data in the form of an ALAMO file that can be read later by the GUI. After running ALAMO, the results can be saved in an

ALAMO listing file that can also be read later by the GUI to reconstruct the problem inputs and results.

Even if you are not planning to use ALAMO from the command line, you should still plan to quickly read the next subsection and following section as they explain how ALAMO works and what is expected to make it run correctly. Additionally, these sections describe material related to algorithmic options that are available in a very similar format through the GUI.

3.2 Running ALAMO from the command line

As an alterative to using the GUI, users can utilize ALAMO form the command line. ALAMO reads model data and algorithmic options from a text file in a simple format. Even though it is not required, it is strongly recommended that all ALAMO input files have the extension '.alm.' If the input file is named 'test.alm' and the ALAMO executable is named 'alamo,' issuing the command

```
alamo test
```

or

```
alamo test.alm
```

results in ALAMO parsing test.alm and solving the problem. In addition to screen displays, ALAMO can also provide results in the listing file 'test.lst' that is generated during the run. The .lst file is always stored in the execute directory, even when the .alm file is in a different path. During execution, ALAMO creates and utilizes a directory for storing various work files. When calling ALAMO, the user may optionally include a second command line argument in order to specify ALAMO's working directory:

```
alamo test.alm myscratchdir
```

where 'myscratchdir' denotes the name of ALAMO's scratch directory. If this argument is not specified, ALAMO will create and utilize a directory named 'almscr' in the execute directory. If the scratch exists, it is erased in the beginning of the run. At the end of the run, ALAMO will delete its scratch directory.

4 Example input file

The following file is referred to as 'e1.alm' and pertains to learning the simple function $z = x^2$. There is one input and one output in the model. The input is restricted between -5 and 5. An initial sampling data set is specified and is comprised of 11 preexisting data points. The user options do not call for adaptive sampling to be used, effectively requesting the best possible model that can be derived from the preexisting data set. Finally, the following functions are permitted in the model: linear, logarithmic, exponential, sine, cosine, and monomials with powers 2 and 3.

```
! Example 1 with data from z = x^2
ninputs 1
noutputs 1
xmin -5
xmax 5

ndata 11

linfcns 1
logfcns 1
expfcns 1
sinfcns 1
cosfcns 1
monomialpower 2 3

BEGIN_DATA
-5      25
-4      16
-3      9
-2      4
-1      1
0       0
1       1
2       4
3       9
4       16
5       25
END_DATA
```

Several additional examples of ALAMO input files accompany the distributed code.

5 Input file grammar

The following rules should be followed when preparing an ALAMO input file:

- The name of the input file should include its exact path location if the file is not present in the execute directory.
- The name of the input file should not exceed 1000 characters in length.
- The input is not case sensitive.
- Most options are entered one per line, in the form of 'keyword' followed by 'value'. Certain vector options are entered in multiple lines, starting with 'BEGIN_<keyword>', followed by the vector input, followed by 'END_<keyword>'.

- Certain options must appear first in the input file. This requirement is discussed explicitly in option descriptions provided below.
- With the exception of arguments involving paths, character-valued options should not contain spaces.
- Blank lines, white space, and lines beginning with *, #, % or ! are skipped. Inline comments that are preceded by #, % or ! are permitted in any line that contains alphanumeric options. Blocks of comment lines are allowed using 'BEGIN_COMMENT', followed by the block of comment lines, followed by 'END_COMMENT'; these comment blocks are entirely ignored by ALAMO.

6 ALAMO data and options specification statements

6.1 Required scalar parameters

The following parameters must be specified in the input file in the order listed below.

Parameter	Description
NINPUTS	Number of model input variables. NINPUTS must be a positive integer and defines the dimension of the vector x.
NOUTPUTS	Number of model output variables. NOUTPUTS must be a positive integer and defines the dimension of the vector z.

6.2 Required vector parameters

The following parameters must be specified in the input file in the order listed below and only after the scalar required parameters have already been specified.

Parameter	Description
XMIN	Row vector specifying minimum values for each of the input variables. This should contain exactly NINPUTS entries that are space delimited.
XMAX	Row vector specifying maximum values for each of the input variables. This should contain exactly NINPUTS entries that are space delimited.

XMIN and XMAX do not necessarily correspond to the minimum and maximum values of the data provided by the user to ALAMO. Instead, they should reflect the physically meaningful lower and upper bounds for the input variables. These values are used by ALAMO for scaling purposes as well as for generating samples when no preexisting data set is provided or, in general, when adaptive sampling is done.

6.3 Optional data specifications

This section describes optional parameters pertaining to the particular problem being solved.

Option	Description	Default
NDATA	Number of data points in a preexisting data set specified by the user. NDATA must be a nonnegative integer.	0
NPREDATA	Number of data points for which ALAMO will provide predictions to the user. At the end of the run, ALAMO will calculate predictions using its best model at each of the NPREDATA data points. NPREDATA must be a nonnegative integer.	0
NSAMPLE	Number of data points to be generated by sampling before any model is built. These points will be used for model building along with the NDATA points specified by the user. NSAMPLE must be a nonnegative integer.	0
NVALSETS	Number of data sets to be used for validation after model generation. Validation of the model will be performed on each data set separately. NVALSETS must be a nonnegative integer.	0
NVALDATA	Number of preexisting data points in each of the NVALSETS data sets. These data points are not used to develop the model but only to compute model errors at the validation data points. NVALDATA must be an array of NVALSETS nonnegative integers. If NVALDATA is provided and NVALSETS has not already been specified in the input file, ALAMO will assume that NVALSETS equals 1.	0 0 0 . . .
NVALSAMPLE	Number of data points to be sampled and added to each of the NVALSETS data sets for validation. These data points are not used to develop the model but only to compute model errors at the validation data points. NVALSAMPLE points are sampled randomly and added to the validation data sets. The sampling facility requires that the user provides a SIMULATOR. Validation can rely exclusively on preexisting data (through the NVALDATA option), exclusively on sampled data (through the NVALSAMPLE option), or on any combination desired by the user. NVALSAMPLE must be an array of NVALSETS nonnegative integers. If NVALSAMPLE is provided and NVALSETS has not already been specified in the input file, ALAMO will assume that NVALSETS equals 1.	0 0 0 . . .
MAXSIM	Maximum number of successive simulator failures allowed before we quit. MAXSIM must be a non-negative integer. If MAXSIM equals 0, ALAMO will continue calling the simulator even in the case of repeated failures.	0

MINPOINTS	At any stage of the adaptive sampling process, convergence is assessed only if the simulator is able to compute the output variables for at least MINPOINTS out of the data points requested by ALAMO. MINPOINTS must be a positive integer.	NINPUTS
MAXPOINTS	The number of data points requested by ALAMO from the simulator during an adaptive sampling iteration. MAXPOINTS must be a positive integer at least as large as MINPOINTS.	NINPUTS+6
XFACTOR	Row vector of scaling factors used to scale the input variables. One per input variable, space separated.	1 1 1 . . .
XSCALING	A 0−1 indicator. If 1 and XFACTORS are not provided in the input file, ALAMO sets XFACTORS equal to the range of each input variable.	0
SCALEZ	A 0−1 indicator. If 1, outputs are scaled when solving mixed-integer optimization problems; otherwise, they are not scaled.	0
XLABELS	Row vector of labels to denote the input variables. One per input variable, space separated. Each label can be no more than 128 characters long. All labels must begin with an alphabetical character (A-Z or a-z) and contain only alphanumerical characters (A-Z, a-z, 0-9) or underscores. No label should start with the string 'alm_'.	X1 X2 X3 . . .
ZLABELS	Row vector of labels to denote the output variables. One per output variable, space separated. Each label can be no more than 128 characters long. All labels must begin with an alphabetical character (A-Z or a-z) and contain only alphanumerical characters (A-Z, a-z, 0-9) or underscores. No label should start with the string 'alm_'.	Z1 Z2 Z3 . . .
MONO	Number of monomial powers to be considered as basis functions. MONO must be a nonnegative integer.	0
MULTI2	Number of powers to be considered for pairwise combinations of basis functions. MULTI2 must be a nonnegative integer.	0
MULTI3	Number of powers to be considered for three variable combinations of basis functions. MULTI3 must be a nonnegative integer.	0
RATIOS	Number of ratio combinations of powers to be considered as basis functions. RATIOS must be a nonnegative integer.	0
EXPFCNS	A 0−1 indicator. Exponential functions are considered as basis functions if 1; otherwise, they are not considered.	0
LINFCNS	A 0−1 indicator. Linear functions are considered as basis functions if 1; otherwise, they are not considered.	1

LOGFCNS	A 0−1 indicator. Logarithmic functions are considered as basis functions if 1; otherwise, they are not considered. Natural logarithms are used.	0
SINFCNS	A 0−1 indicator. Sine functions are considered as basis functions if 1; otherwise, they are not considered. The arguments of these functions are assumed in radians.	0
COSFCNS	A 0−1 indicator. Cosine functions are considered as basis functions if 1; otherwise, they are not considered. The arguments of these functions are assumed in radians.	0
CONSTANT	A 0−1 indicator. A constant will be considered as a basis function if 1; otherwise, it will not be considered.	1
NCUSTOMBAS	Number of user-specified basis functions. NCUSTOMBAS must be a nonnegative integer. If this option is utilized, then a BEGIN_CUSTOMBAS ... END_CUSTOMBAS section must be supplied to provide the algebraic expressions of the user-specified basis functions.	0
GRBFCNS	A 0−1 indicator. Gaussian radial basis functions centered around the set of the user-specified NDATA points are considered as basis functions if 1; otherwise, they are not considered. These functions are deactivated if their textual representation requires more than 128 characters (in the case of too many input variables and/or data points).	0
RBFPARAM	Multiplicative constant used in the Gaussian radial basis functions.	1.0
TRACE	A 0−1 indicator. If set to 1, a trace file is generated at the end of the run, including a succinct summary of the results. First, a header line beginning with a # is printed describing the contents of each line of the trace file. Then, for each output, results are printed in one line for each data set. Data sets are marked by the numbers 0 (observed data set), -1 (user-provided validation data set), and 1, ..., NVALSETS (ALAMO-generated validation sets).	0
TRACEFNAME	Name of trace file. Summaries are appended to an existing trace file.	trace.trc

MODELER	Fitness metric to be used for model building. Possible values are 1 through 8, with the following meaning: 1. BIC: Bayesian information criterion, 2. Cp: Mallow's Cp, 3. AICc: the corrected Akaike's information criterion, 4. HQC: the Hannan-Quinn information criterion, 5. MSE: mean square error, 6. SSEp: the sum of square errors plus a penalty proportional to model size, 7. RIC: the risk information criterion, and 8. MADp: the maximum absolute deviation plus a penalty proportional to model size. The deviation is expressed as absolute percent deviation from measurements that exceed 10^{-3} in magnitude and as an absolute difference for small measurements.	1
BUILDER	A 0−1 indicator. If set to 1, a greedy heuristic builds up a model by adding one variable at a time. This model is used as a starting point for solving an integer programming formulation according to the choice of MODELER. If an optimizer is not available, the heuristic model will be the final model to be returned.	1
BACKSTEPPER	A 0−1 indicator. If set to 1, a greedy heuristic builds down a model by starting from the least squares model and removing one variable at a time.	0
CONVPEN	When MODELER is set to 6 or 8, a penalty consisting of the sum of square errors (MODELER=6) or the maximum absolute error (MODELER=8) and a term penalizing model size is used for model building. The size of the model is weighted by CONVPEN. If CONVPEN=0, this metric reduces to the classical sum of square errors (MODELER=6) or the maximum absolute deviation (MODELER=8).	0.0
SCREENER	Screening method used to reduce the number of potential basis functions before optimization of the selected fitness metric. Possible values are 0, 1, and 2, corresponding to no screening, screening with the lasso, and sure independence screening, respectively.	0
NCVF	Number of folds to be used for cross validation by the lasso screener. ALAMO will use a two-fold validation if fewer than 10 data points are available. NCVF must be a nonnegative integer.	5

SISmult	This parameter must be non-negative and is used to determine the number of basis functions retained by the SIS screener. The number of basis functions retained equals the floor of SSISmult$\frac{n}{\ln(n)}$, where n is the number of measurements available at the current ALAMO iteration.	1
INITIALIZER	Technique to be used for sampling of the NSAMPLE points (or INITIALPOINTS minus NDATA) at the beginning of the algorithm. A nonzero value of NSAMPLE directs ALAMO to use sampling according to the value of INITIALIZER and requires the presence of a user-provided SIMULATOR. Possible INITIALIZER values are: 1 (random); 3 (Faure). INITIALPOINTS is described in Section 9.	1
SAMPLER	Technique to be used for adaptive sampling. A value of MAXITER different than 1 directs ALAMO to use adaptive sampling according to the value of SAMPLER and requires the presence of a user-provided SIMULATOR. Possible SAMPLER values are: 1 (random), 2 (SNOBFIT), 3 (Faure).	1

SIMULATOR	SIMULATOR is the name of the executable that ALAMO can call in order to obtain function evaluations of the black box. The simulator must be able to read file SIMIN and write file SIMOUT. SIMIN is provided by ALAMO. The first line of SIMIN provides the number of requested data points, k, followed by pid, an integer that provides the process id of the current ALAMO process that generated SIMIN. After this first line, there are k additional lines, one for each of the data points where function evaluations are requested. Each point comes in NINPUTS space-separated reals. Following these lines, a single line contains NOUTPUTS space-separated T/F (true/false) flags indicating whether ALAMO is requesting a simulation of each corresponding output variable; the simulator may choose to ignore this information or utilize it in order to avoid simulation of outputs for which this flag is F. In SIMOUT, the simulator must return a number of lines, each containing a point in the input variable space (space-separated NINPUTS reals) where a simulation was performed, along with the corresponding output variable values (space-separated NOUTPUTS reals). ALAMO allows for the number of these points to be different from k and for these points to be different than the points where simulations were requested. If more than k points are provided, only the first k are used. If the simulation fails or is impossible for certain output variables, partial simulation results may be returned and the non-available output variables must be set equal to PRESET. The simulator must be in the directory where ALAMO is launched or in the user's path; alternatively, its complete path must be specified through this option. ALAMO will execute the simulator in a scratch directory it generates during its run; hence, the simulator should not rely on any relative paths in order to access other programs or files. The simulator may utilize pid in order to halt and resume the execution of ALAMO. For instance, in Linux, this can be achieved with the commands 'kill -TSTP pid' and 'kill -CONT pid'; additionally, checkpointing can be used to save all program information in case a system reboot takes place while waiting for the simulator to complete.	mysim
PRESET	A value indicating that the simulator was not able to compute a specific output variable at a specific point. This value must be carefully chosen to be an otherwise not realizable value for the output variables.	-111111.

MAXTIME	Maximum total execution time allowed in seconds. This time includes all steps of the algorithm, including time to read problem, preprocess data, solve optimization subproblems, and print results.	1000
MAXITER	Maximum number of ALAMO iterations. Each iteration begins with a model-building step. An adaptive sampling step follows if MAXITER does not equal 1. If MAXITER is set to a number less than or equal to 0, ALAMO will enforce no limit on the number of iterations.	1
DATALIMITTERMS	A 0−1 indicator. If 1, ALAMO will limit the number of terms in the model to be no more than the number of data measurements; otherwise, no limit based on the number of data measurements will be placed. The user may provide an additional limit on the number of terms in the model through the MAXTERMS option.	1
MAXTERMS	Row vector of maximum terms allowed in the modeling of output variables. One per output variable, space separated. A −1 signals that no limit is imposed.	-1 -1 -1 ...
NUMLIMITBASIS	A 0−1 indicator. If 1, ALAMO will eliminate basis functions that are not numerically acceptable (e.g., $\log(x)$ will be eliminated if x may be negative; otherwise, no limit based on the number of data measurements will be placed. The user may provide additional limits on the the type and number of selected basis functions through the options EXCLUDE and GROUPCON.	1
EXCLUDE	Row vector of 0/1 flags that specify which input variables, if any, ALAMO should exclude during the model building process. All input variables must be present in the data but ALAMO will not include basis functions that involve input variables for which EXCLUDE equals 1. This feature does not apply to custom basis functions or RBFs.	0 0 0 ...
IGNORE	Row vector of 0/1 flags that specify which output variables, if any, ALAMO should ignore. All output variables must be present in the data but ALAMO does not model output variables for which IGNORE equals 1.	0 0 0 ...
XISINT	Row vector of 0/1 flags that specify which input variables, if any, ALAMO should treat as integers. For integer inputs, ALAMO's sampling will be restricted to integer values.	0 0 0 ...
ZISINT	Row vector of 0/1 flags that specify which output variables, if any, ALAMO should treat as integers. For integer variables, ALAMO's model will include the rounding of a function to the nearest integer (equivalent to the nint function in Fortran.)	0 0 0 ...

TOLRELMETRIC	Relative convergence tolerance for the chosen fitness metric for the modeling of output variables. One per output variable, space separated. Incremental model building will stop if two consecutive iterations do not improve the chosen metric by at least this amount.	1e-6 1e-6 1e-6 . . .
TOLABSMETRIC	Absolute convergence tolerance for the chosen fitness metric for the modeling of output variables. One per output variable, space separated. Incremental model building will stop if two consecutive iterations do not improve the chosen metric by at least this amount.	1e-6 1e-6 1e-6 . . .
TOLMEANERROR	Row vector of convergence tolerances for mean errors in the modeling of output variables. One per output variable, space separated. Incremental model building will stop if TOLMEANERROR, TOLRELMETRIC, or TOLABSMETRIC is satisfied.	0 0 0 . . .
TOLMAXERROR	Absolute tolerance for the adaptive sampling procedure to terminate during the modeling of output variables. One per output variable, space separated. Adaptive sampling will stop if the current model predictions and measurements do not differ in magnitude by more than this amount.	0.05 0.05 0.05 . . .
TOLSSE	Absolute tolerance on sum of square errors (SSE). ALAMO will terminate if it finds a solution whose SSE is within TOLSSE from the SSE of the full least squares problem.	0
MIPOPTCA	Absolute convergence tolerance for mixed-integer optimization problems. This must be a nonnegative scalar.	0.05
MIPOPTCR	Relative convergence tolerance for mixed-integer optimization problems. This must be a nonnegative scalar.	0.0001
LINEARERROR	A 0−1 indicator. If 1, a linear objective is used when solving mixed-integer optimization problems; otherwise, a squared error will be employed.	0
SIMIN	Name of input file for the simulator. ALAMO generates this file.	input.txt
SIMOUT	Name of output file for the simulator. ALAMO expects the simulator to provide this file after each call.	output.txt
SOLVEMIP	A 0−1 indicator. The BARON optimization solver, which is embedded in ALAMO, will be used to solve ALAMO's MIPs/MIQPs if this option is set to 1; if set to 0, no MIP/MIQP optimizer will be used even if one is available.	0
PRINT_TO_FILE	A 0−1 indicator. Output is directed to the listing file if this option is set to 1; if set to 0, no output is sent to the listing file.	1

PRINT_TO_SCREEN	A 0−1 indicator. Output is directed to the screen if this option is set to 1; if set to 0, no output is sent to the screen.	1
FUNFORM	A positive integer to specify the format for printing basis functions and models found by ALAMO. Fortran intrinsics used in custom basis functions are retained in Fortran format; all other functions are translated based on the value of FUNFORM. Possible values are 1 through 5, with the following meaning: 1. FORTRAN format 2. GAMS format 3. BARON format 4. C format 5. Excel format Note that a large number of digits may be printed in all of these formats. In order to avoid problems reading these strings into GAMS, the GAMS $offdigit option can be used in the user's GAMS file.	5
NTRANS	A nonnegative integer showing how many of the output variables are to be obtained through transformations of input/output variables. The last NTRANS of the output variables are obtained through algebraic transformations of the input variables and/or the first NOUTPUTS-NTRANS output variables; their values should not be provided in any DATA section or calculated by the simulator.	0

The parser is not case sensitive. For example, output variable labels Z1 and z1 are equivalent. For vector inputs, any items provided in excess of those required will be ignored. For example, if more than NINTPUTS XLABELS are provided, the extra labels are ignored.

In deciding whether to deactivate printing to the screen or file, users should consider that model coefficients are printed with two significant digits to the screen and with 23 digits to the listing file.

If the parameter NDATA is set, then a data section must follow subsequently in the input file with precisely NDATA rows, one for each data point (pair of x and z values) specified in the following form:

```
BEGIN_DATA
⋮
END_DATA
```

If the parameter NPREDATA is set, then a data section must follow subsequently in the input file with precisely NPREDATA rows, each containing precisely NINPUTS values, thus corresponding to a point in the x-space, specified in the following form:

```
BEGIN_XPREDATA
⋮
END_XPREDATA
```

If the parameter NVALDATA is set, a similar data section must be provided using a similar construct:

```
BEGIN_VALDATA
⋮
END_VALDATA
```

If the parameter NCUSTOMBAS is set, user-specified basis functions must be provided using the construct:

```
BEGIN_CUSTOMBAS
⋮
END_CUSTOMBAS
```

where basis functions are provided one per line. The parser is not case sensitive and allows for Fortran functional expression in terms of the XLABELS. The following functions are currently accepted by the parser: addition, subtraction, multiplication, division, power (∗∗ and ˆ), abs, exp, log, log10, sqrt, sinh, cosh, tanh, sin, cos, tan, asin, acos, and atan. Other functions may be expressed in terms of the preceding operators and functions, e.g., $\min(a, b) = (a+b)/2-|a-b|/2$.

If the parameters MONO, MULTI2, MULTI3, or RATIOS are set, the corresponding powers must also be specified as row vectors of corresponding length in the following way:

Parameter	**Description**
MONOMIALPOWER	Row vector of monomial powers considered in basis functions; powers of 0 or 1 are not allowed. If MONO is provided, MONOMIALPOWER must be of length MONO.
MULTI2POWER	Row vector of powers to be considered for pairwise combinations in basis functions. If MULTI2 is provided, MULTI2POWER must be of length MULTI2.
MULTI3POWER	Row vector of powers to be considered for triplet combinations in basis functions. If MULTI3 is provided, MULTI3POWER must be of length MULTI3.
RATIOPOWER	Row vector of powers to be considered for ratios in basis functions. If RATIOS is provided, RATIOPOWER must be of length RATIOS.

The entries of the above vectors must be space separated. As stated above, the user is not obligated to specify the parameters MONO, MULTI2, MULTI3, or RATIOS. If any of the corresponding power options are provided, ALAMO will count them and infer the total number of powers specified by the user.

If the parameter NTRANS is set to a positive entry, NTRANS functions must be provided using the construct:

```
BEGIN_TRANSFORMS
⋮
END_TRANSFORMS
```

where transformation functions are provided one per line. Line k of a TRANSFORMS section provides an algebraic transformation that determines transformed output k which, in turn, corresponds to output variable NOUTPUTS-NTRANS+k. The parser is not case sensitive and allows for Fortran functional expression in terms of the XLABELS and ZLABELS. For the types of functions supported, see the discussion under BEGIN_CUSTOMBAS above.

6.4 Constrained regression

This section describes ALAMO's constrained regression capabilities. There are two distinct constrained regression capabilities implemented currently in ALAMO:

- The ability to enforce constraints, such as bounds, on the response function.
- The ability to enforce combinatorial constraints on the types of basis functions utilized, including constraints on groups of basis functions.

The primary options that control application of constrained regression to ALAMO's response function are:

Parameter	Description
ZMIN	Minimum values for output variables. One per output variable, space separated. If this vector is specified, the corresponding lower bounds on output variables are enforced.
ZMAX	Maximum values for output variables. One per output variable, space separated. If this vector is specified, the corresponding upper bounds on output variables are enforced.
EXTRAPXMIN	Minimum values for safe extrapolation region. One per input variable, space separated. If this vector is specified, ZMIN and ZMAX are enforced over EXTRAPXMIN to (EXTRAP)XMAX; otherwise, they are enforced over XMIN to (EXTRAP)XMAX.

EXTRAPXMAX	Maximum values for safe extrapolation region. One per input variable, space separated. If this vector is specified, ZMIN and ZMAX are enforced over (EXTRAP)XMIN to EXTRAPXMAX; otherwise, they are enforced over (EXTRAP)XMIN to XMAX.
PRINTEXTRAP	A 0/1 flag to signal printing of ALAMO's predictions in the extrapolation region. By default, PRINTEXTRAP is set to 0. If set to 1, ALAMO will report predicted values at points within the region between EXTRAPXMIN to EXTRAPXMAX. The reported points will be generated randomly with approximately the same density as that of input points within XMIN to XMAX that were used by ALAMO for model determination (these points include points in the user-specified preexisting data set and ALAMO-selected simulation set).

Custom constrained regression, i.e., constrained regression for enforcing conditions other than simple bounds, can be done by setting the option CRNCUSTOM:

Option	Description	Default
CRNCUSTOM	Number of custom constraints (other than bounds). CRNCUSTOM must be a nonnegative integer.	0

If CRNCUSTOM is specified, the custom constraints themselves are described through a related section:

```
BEGIN_CUSTOMCON

⋮

END_CUSTOMCON
```

where, in each of CRNCUSTOM lines of this section, one would need to specify the output variable index j associated with a custom constraint, followed by white space, followed by a function $g(x, z)$ expressed in terms of a Fortran expression of input and output variable labels. ALAMO will then enforce the constraint $g \leq 0$ when building a model for output variable j. This feature supports nonlinear expressions in g that include multiplication, division, powers, and exponentials.

The following are algorithmic options that control implementation aspects of the above constrained regression features. These options may be optionally set as follows:

Option	Description	Default
CRTOL	Tolerance within which custom constraints must be satisfied. CRTOL must be a real that is no smaller than 1e-5. Bound and custom constraints will be satisfied within an absolute tolerance equal to CRTOL.	1e-3

CRNINITIAL	Number of random bounding points at which constraints are sampled initially. CRNINITIAL must be a nonnegative integer.	0
CRMAXITER	Maximum allowed constrained regression iterations. Constraints are enforced on additional points during each iteration. CRMAXITER must be a positive integer.	10
CRNVIOL	Number of bounding points added per round per constraint (bound or custom) in each iteration. CRNVIOL must be a positive integer.	2*NINPUTS
CRNTRIALS	Number of random trial bounding points per round of constrained regression. CRNTRIALS must be a positive integer.	100

In addition to imposing constraints on the response surface it produces, ALAMO has the ability to enforce constraints on groups of selected basis functions. This can be accomplished through ALAMO's NGROUPS option:

Option	**Description**	**Default**
NGROUPS	Number of groups that must be constrained. NGROUPS must be a nonnegative integer.	0

If a positive NGROUPS is specified, the groups themselves must be specified through a related section:

```
BEGIN_GROUPS
⋮
END_GROUPS
```

where, in each line of this section, one would need to specify information of the form

```
Group-id Member-type Member-indices <Powers>
```

In this construct, each group is uniquely associated with a Group-id ranging from 1 to NGROUPS. Each line must contain three required parameters (Group-id, Member-type, Member-indices); the fourth parameter (Powers) is required only in the context of basis functions that involve powers. The syntax of this section must obey the following rules:

- Each line pertains to a single group.
- A group may be described over several lines, with each line restricted to describing a single type of component of the group.
- Group-id is a nonnegative integer between 1 and NGROUPS that denotes the numerical id (index) of a group described (at least partly) in a line.

- Member-type is an attribute for the member(s) of the group described in the specific line and can take anyone of the values LIN, LOG, EXP, SIN, COS, MONO, MULTI2, MULTI3, RATIO, RBF, CUST, and CONST corresponding to different types of basis functions. In addition, Member-type may be set equal to GRP if it is desired to specify a group of groups.
- Member-indices is used to specify the composition of a group in terms of indices of input variables and groups. The keyword CONST should not be followed by any Member-indices (as there is only one constant in the model). The keywords MONO, EXP, LOG, SIN and COS must be followed by exactly one index that has a value between 1 and NINPUTS corresponding to the input variable involved in the basis function; alternatively, a value of -1 may be used to denote that all input variables should be considered (with a specific power in the case of MONO). The keywords MULTI2, MULTI3 and RATIO must be followed by two, three, and two indices, respectively, indicating the input variables involved in the corresponding basis function; alternatively, an index of -1 may be used to denote that all possible input variable combinations in any of these basis functions should be considered. The keyword LIN may be followed by as many as NINPUTS Member-indices and specifies which linear terms of the model are included in the group; alternatively, an index of -1 may be used to denote that all possible linear terms are included in the group. Similarly, the keywords CUST and RBF may be followed by as many as NCUSTOMBAS and NDATA Member-indices and an index of -1 may be used to denote all possible custom basis functions and RBFs in a group. As many as NGROUPS-1 indices may follow the keyword GRP in order to specify which groups form a group.
- When Member-type is one of MONO, MULTI2, MULTI3, or RATIO, in addition to Member-indices, the input line must specify the power(s) involved in the group; if power equals -1111, all powers are considered.
- Membership in a group is non-exclusive; a basis function or group may belong to multiple groups.

Once the number of groups has been specified and each group has been described through the GROUPS construct, group constraints can be specified through the GROUPCON section:

```
BEGIN_GROUPCON
⋮
END_GROUPCON
```

ALAMO permits different group constraints to be imposed on different output variables. Each line of the GROUPCON section is dedicated to a group-output variable combination and has the following information:

```
Group-id Output-id Constraint-type Integer-parameter
```

The rules for completing this section are as follows:

- Each line describes a single group constraint.
- Group-id is a nonnegative integer between 1 and NGROUPS that denotes the numerical id (index) of the primary (and sometimes only) group involved in the constraint.
- Output-id is a nonnegative integer between 1 and NOUTPUTS for which the constraint will be imposed; a value of -1 can be used in this entry to denote that the constraint should be enforced for all output variables.
- Constraint-type is a string descriptor that can take anyone of the following values:
  - NMT: to denote a no-more-than constraint, i.e., require that no more than Integer-parameter members of this group should be selected in the model.
  - ATL: to denote an at-least constraint, i.e., require that at least Integer-parameter members of this group should be selected in the model.
  - REQ: to require that, if the primary group is selected, then the group with id equal to Integer-parameter should also be selected.
  - XCL: to require that, if the primary group is selected, then the group with id equal to Integer-parameter should not be selected.
- A group may appear in more than one constraint. This flexibility coupled with the fact that input variables and groups may belong to multiple groups allows us to enforce sparsity constraints on model attributes within groups, between groups, and groups organized in clusters, trees, or any other structure.

7 ALAMO output

7.1 ALAMO screen output

The screen output below is obtained for problem e1.alm.

```
***************************************************************************
ALAMO version 2023.1.22. Built: WIN-64 Sun Jan 22 22:54:59 EST 2023

If you use this software, please cite:
Cozad, A., N. V. Sahinidis and D. C. Miller,
Automatic Learning of Algebraic Models for Optimization,
AIChE Journal, 60, 2211-2227, 2014.

ALAMO is powered by the BARON software from http://www.minlp.com/
***************************************************************************
Licensee: Nick Sahinidis at TOF, niksah@gmail.com.
***************************************************************************
Reading input data
Checking input consistency and initializing data structures
```

```
Warning: eliminating basis log(X1)

Step 0: Initializing data set
User provided an initial data set of 11 data points
We will sample no more data points at this stage
***************************************************************************
Iteration 1 (Approx. elapsed time 0.0 s)

Step 1: Model building using BIC

Model building for variable Z
----
BIC = -0.100E+31 with Z = X1^2

Calculating quality metrics on observed data set.

Quality metrics for output Z
----------------------------
SSE OLR:            0.00
SSE:                0.00
RMSE:               0.00
R2:                 1.00
R2 adjusted:        1.00
Model size:         1
BIC:                -0.100E+31
Cp:                 -9.00
AICc:               -0.100E+31
HQC:                -0.100E+31
MSE:                0.00
SSEp:               0.00
RIC:                3.89
MADp:               0.00

Total execution time 0.0 s
Times breakdown
    OLR time:         0.0 s in 6 ordinary linear regression problem(s)
    MINLP time:       0.0 s in 0 optimization problem(s)
    Simulation time:  0.0 s to simulate 0 point(s)
    All other time:   0.0 s in 1 iteration(s)

Normal termination
***************************************************************************
```

The software first reports the version, platform, and compilation date of the executable, followed by credits. Then, after reading the input data, a consistency check is run on the problem data and, if passed, the data structures are initialized. In this specific example, a warning is issued that logarithmic basis functions are not considered since the input variable is allowed to take negative values. Subsequently, information is provided for all algorithmic steps. During initialization (Step 0), it is reported that 11 data points are used for sampling and that no simulator is called in addition to using the preexisting data set. In Step 1, the model is built in stages. Earlier in the search, the perfect model $z = x^2$ is identified. Since there is no simulator provided, there is no adaptive sampling and execution terminates here after reporting a detailed breakdown of CPU times for the different algorithmic steps, including the number of calls to the optimizer (0 in this example) and the simulator (0 in this example). There are no calls to an optimizer in this example because the problem is small enough to be solved faster by complete enumeration.

8 Termination conditions and error messages

Errors in the input file are reported on the screen and/or the listing file in the form of "warnings" and "errors." ALAMO attempts to continue execution despite warnings. If the errors are severe, the program execution is stopped and the line where the fatal error occurred is displayed. The input file should be checked even if the warnings are not severe, as the problem might have been parsed in a way other than it was intended to be. Detailed error messages are provided in that case.

If execution terminates normally, ALAMO prints 'Normal termination.' If there is an error, the message on the screen or file is 'ALAMO terminated with termination code,' followed by one of the following error codes, all of which are self-explanatory:

1. ALAMO must be called with one or two command line arguments.
2. ALAMO input file name must be no longer than 1000 characters.
3. ALAMO input file not found.
4. ALAMO input file cannot be opened.
5. Keyword not recognized in input file.
6. Keyword too long in input file.
7. Incomplete input file.
8. Input value in error in input file.
9. Number of input variables (NINPUT) must be specified before specifying XMIN values.
10. Number of input variables (NINPUT) must be specified before specifying XMAX values.
11. Number of input variables (NINPUT) must be specified before specifying XFACTOR values.
12. Number of input variables (NINPUT) must be specified before specifying XLABELS.
13. Number of output variables (NOUTPUT) must be specified before specifying ZLABELS.

14. MONOMIALPOWER values have been set already. Multiple declarations are not allowed.
15. Number of input variables (NINPUT) must be specified before the DATA section of the input file.
16. Number of output variables (NOUTPUT) must be specified before the DATA section of the input file.
17. Number of data points (NDATA) must be specified before the DATA section of the input file.
18. Number of input variables (NINPUT) must be specified before the XDATA section of the input file.
19. Number of data points (NDATA) must be specified before the XDATA section of the input file.
20. Number of output variables (NOUTPUT) must be specified before the ZDATA section of the input file.
21. Number of data points (NDATA) must be specified before the ZDATA section of the input file.
22. Input data file missing required keyword(s).
23. END_DATA missing or incomplete DATA section.
24. END_XDATA missing or incomplete XDATA section.
25. END_ZDATA missing or incomplete ZDATA section.
26. Only one of XDATA and DATA sections is allowed.
27. Only one of ZDATA and DATA sections is allowed.
28. MULTI2POWER values have been set already. Multiple declarations are not allowed.
29. MULTI3POWER values have been set already. Multiple declarations are not allowed.
30. Unable to open output file.
31. Maximum number of iterations reached.
32. RATIOPOWER values have been set already. Multiple declarations are not allowed.
33. Error while trying to use the optimizer to solve the MIP for best subset.
34. Error while attempting to access the ALAMO execution directory.
35. Error while attempting to access the ALAMO scratch directory.
36. Error while attempting to access the external simulator.
37. Error while attempting to write the external simulator input file.

38. Error while attempting to read the external simulator output file.
39. Scaling by zero is not allowed.
40. XMAX−XMIN for all input variables must be positive.
41. XDATA must be in the range [XMIN, XMAX].
42. Simulator should not return NaN for input variable values.
43. Simulator should not return NaN for output variable values. For any variable that the simulator cannot compute, return the value of PRESET.
44. Input file is missing XMIN values.
45. Input file is missing XMAX values.
46. MONOMIALPOWERS must be specified if MONO is used.
47. MULTI2POWER must be specified if MULTI2 is used.
48. MULTI3POWER must be specified if MULTI3 is used.
49. RATIOPOWER must be specified if RATIOS is used.
50. DATA section must be specified when NDATA is nonzero.
51. Insufficient memory to allocate data structures.
52. Number of validation data points (NVALDATA) must be specified before the VALDATA section of the input file.
53. VALDATA section must be specified when NVALDATA is nonzero.
54. VALDATA section must be specified when NVALSECTIONS is nonzero.
55. Premature end of input file.
56. Number of custom constraints (CRNCUSTOM) must be specified before specifying CUSTOMCON section.
57. END_ZMIN missing or incomplete ZMIN section.
58. END_ZMAX missing or incomplete ZMAX section.
59. Number of input variables (NINPUT) must be specified before specifying EXTRAPXMIN values.
60. Number of input variables (NINPUT) must be specified before specifying EXTRAPXMAX values.
61. END_CUSTOMCON missing or incomplete CUSTOMCON section.
62. Number of output variables (NOUTPUT) must be specified before specifying ZMIN values.

63. Unable to open trace file TRACEFNAME.
64. No keyword may be specified more than once.
65. Variable index is out of range.
66. Error while trying to run SNOBFIT.
67. Error while trying to solve ordinary least squares regression subproblem with the optimizer.
68. Maximum CPU time (MAXTIME) exceeded.
69. Error while trying to write in the ALAMO scratch directory.
70. Number of output variables (NOUTPUT) must be specified before specifying TOLMEANERROR values.
71. A least squares subproblem failed during enumeration and no optimizer is available.
72. Licensing error. A valid license is required in order to run this software.
73. Error while trying to use the optimizer to solve the constrained regression model.
74. Error while trying to copy file to disk.
75. CUSTOMCON section must be specified when CRNCUSTOM is nonzero.
76. All output variables ignored by user. No point in calling ALAMO.
77. END_CUSTOMBAS missing or incomplete CUSTOMBAS section.
78. Number of custom basis functions (NCUSTOMBAS) must be specified before the CUSTOMBAS section of the input file.
79. Syntax error in custom basis function.
80. All variable labels must begin with an alphabetical character (A-Z or a-z).
81. Variable labels may only contain alphanumerical characters (A-Z, a-z, 0-9) or underscores.
82. All variable labels must be distinct.
83. All CRCUSTOMIND values must range from 1 to NOUTPUTS.
84. Each custom constraint must be expressed in terms of the labels of input variables and a single output variable.
85. Each line of the input file must contain no more than 10000 characters. Longer data records may be split into multiple lines using & at the end of a line to signify continuation of the record in the next line.
86. Syntax error in input file.
87. Inline comments must be preceded by ! or #.

88. Inconsistent use of NDATA and INITIALPOINTS.
89. A least squares subproblem failed during model buildup and no optimizer is available.
90. Number of output variables (NOUTPUT) must be specified before specifying MAXTERMS values.
91. Number of output variables (NOUTPUT) must be specified before specifying TOLRELMETRIC values.
92. Number of output variables (NOUTPUT) must be specified before specifying TOLABSMETRIC values.
93. END_TRANSFORMS missing or incomplete TRANSFORMS section.
94. Number of transformed output variables (NTRANS) must be specified before the TRANSFORMS section of the input file.
95. Syntax error in output transformation function.
96. Number of transformed output variables (NTRANS) cannot exceed total number of outputs (NOUTPUTS).
97. Number of transformed output variables (NTRANS) must be specified after specifying total number of outputs (NOUTPUTS).
98. Number of transformed output variables (NTRANS) must be specified before providing output data section (DATA or ZDATA).
99. Number of output variables (NOUTPUT) must be specified before specifying ZISINT values.
100. Number of prediction points (NPREDATA) must be specified before the PREDATA section of the input file.
101. END_XPREDATA missing or incomplete XPREDATA section.
102. Number of input variables (NINPUT) must be specified before the XPREDATA section of the input file.
103. XPREDATA section must be specified when NPREDATA is nonzero.
104. A GROUPS section is allowed only if NGROUPS is positive.
105. A GROUPS section is allowed only after NINPUTS has been defined.
106. A GROUPS section is allowed only after NOUTPUTS has been defined.
107. A GROUPS section is allowed only after NDATA has been defined.
108. Group-ids must be integers between 1 and NGROUPS.
109. Member-type must be one of LIN, LOG, EXP, SIN, COS, MONO, MULTI2, MULTI3, RATIO, RBF, CUST, and CONST.

110. All powers in group definitions must appear in user-specified basis functions.
111. NGROUPS has been specified but a smaller number of groups has been described in the GROUPS section or the GROUPS section is entirely missing.
112. A GROUPCON section is allowed only if NGROUPS is positive.
113. A GROUPCON section is allowed only after NOUTPUTS has been defined.
114. Member-indices for input variables must be integers between 1 and NINPUTS.
115. Member-indices for radial basis functions must be integers between 1 and number of data points (NDATA).
116. Member-indices for custom basis functions must be integers between 1 and NCUSTOMBAS.
117. Member-indices for groups must be integers between 1 and NGROUPS.
118. Output variable indices must be integers between 1 and NOUTPUTS.
119. Constraint-type must be one of NMT, ATL, REQ, and XCL.
120. Integer-parameters for REQ and XCL group constraints must be integers between 1 and NGROUPS.
121. Number of input variables (NINPUT) must be specified before specifying EXCLUDE values.
122. Number of output variables (NOUTPUT) must be specified before specifying IGNORE values.
123. Unable to find the external simulator.
124. Simulator failed MAXSIM times.
125. TRANSFORMS section must be specified when NTRANS is nonzero.
126. Evaluation error with transformation function. Try a different transformation.
127. Error while trying to write file to disk.
128. Number of output variables (NOUTPUT) must be specified before specifying ZMAX values.
129. Powers for polynomial basis functions have been set already. Multiple declarations are not allowed.
130. Number of input variables (NINPUT) must be specified before specifying XISINT values.
131. Number of output variables (NOUTPUT) must be specified before specifying TOLMAXERROR values.
132. A MI(N)LP solver is required to proceed with the current set of options but one is not availablefound.
133. Variable labels may not start with the string ‘alm_’.
134. Constrained regression is currently not supported with trigonometric basis functions.

9 Compatibility with previous versions of ALAMO

Starting with ALAMO v. 2013.10.0, the input format was changed. Input requirements of earlier versions were maintained with two exceptions:

- Previous versions required that ALAMO options be specified in a separate file than preexisting data. All ALAMO input must now be entered in a single file.
- Preexisting data can now be entered in a format that combines input and output measurements in a column wise fashion.

For compatibility with early versions of ALAMO, the following keywords are also acceptable in ALAMO v. 2013.10.0 and beyond:

Parameter	Description
INITIALPOINTS	Number of data points in the initial sample set. This parameter represents the sum of NDATA and NSAMPLE. INITIALPOINTS must be a nonnegative integer. If declared, INITIALPOINTS must be greater than or equal to NDATA. If INITIALPOINTS is declared, NSAMPLE will be ignored and set equal to the difference of INITIALPOINTS and NDATA; otherwise, INITIALPOINTS will be set equal to the sum of NDATA and NSAMPLE.
NVARS	This is equivalent to NINPUTS.
BEGIN_XDATA	Can be used in conjunction with BEGIN_ZDATA to pass x-values separately from z-values. Only one of BEGIN_XDATA and BEGIN_DATA is permitted.
BEGIN_ZDATA	Can be used in conjunction with BEGIN_XDATA to pass x-values separately from z-values. Only one of BEGIN_ZDATA and BEGIN_DATA is permitted.
CONREG	It used to serve as an indication that constrained regression will be invoked. This is now ignored and the need for constrained regression is inferred from other options.

Starting with ALAMO v. 2019.7.30, the keyword REGULARIZER was replaced by the more appropriate SCREENER. The keyword REGULARIZER is still acceptable and the possible values of SCREENER are backwards compatible with those of REGULARIZER.

10 Bibliography

The following is a partial list of ALAMO-related publications that describe the algorithms implemented in the software, the theory behind them, and some related applications.

1. A. Cozad, N. V. Sahinidis, and D. C. Miller. Learning surrogate models for simulation-based optimization. *AIChE Journal*, 60, 2211–2227, 2014.

2. A. Cozad, N. V. Sahinidis, and D. C. Miller. A combined first-principles and data-driven approach to model building. *Computers & Chemical Engineering*, 73, 116–127, 2015.

3. Z. T. Wilson and N. V. Sahinidis. The ALAMO approach to machine learning. *Computers & Chemical Engineering*, 106, 785-795, 2017.

4. K. Lindqvist, Z. T. Wilson, E. Næss and N. V. Sahinidis. A machine learning approach to correlation development applied to fin-tube bundle heat exchangers *Energies*, 11(12), 3450, 2018.

5. Z. Wilson and N. V. Sahinidis. Automated learning of chemical reaction networks. *Computers & Chemical Engineering*, 127, 88-98, 2019.