

ALAMO user manual and installation guide v. 2015.9.18

September 18, 2015

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

Contents

1	Introduction	1
1.1	Licensing and software requirements	2
1.2	Installation	2
2	Algorithms implemented	2
3	Using ALAMO	3
3.1	Calling ALAMO from the command line	3
3.2	Example input file	4
3.3	Input file grammar	4
3.4	ALAMO data and options specification statements	5
3.5	Termination conditions and error messages	12
4	ALAMO output	15
4.1	ALAMO screen output	15
5	Bibliography	17

1 Introduction

The purpose of ALAMO (Automated Learning of Algebraic Models for Optimization) 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 (simulation) function 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 software are in determining where to run the simulations or experiments, what models to fit, and how to determine if a model thus produced is accurate and as simple as possible. A distinguishing feature of the software is that it provides models that are as simple as possible and still accurate. 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. ALAMO makes calls to the third party software GAMS. A separate install and license are required for GAMS and at least one of the mixed-integer quadratic programming solvers under GAMS, preferably GAMS/BARON. Use of the ALAMO constrained regression capability requires availability of GAMS/BARON for the solution of general nonconvex optimization problems. More information about GAMS can be found at <http://www.gams.com>. In the absence of a GAMS license, ALAMO attempts to use enumerative approaches that may be more time consuming or impractical for large problems. For constrained regression only, ALAMO makes calls to MATLAB for which a separate install and license are required. If constrained regression is not used, MATLAB is not required by ALAMO.

1.2 Installation

Install ALAMO and the ALAMO license file in any directory of your choice and add it to your path. Do the same for GAMS. Installation of GAMS is optional but recommended. Install MATLAB if the constrained regression capabilities of ALAMO are needed. Octave does not work in the place of MATLAB.

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 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 accuracy of a previously built model.

In contrast to commonly used 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 synergistic effects between different basis functions are accounted for in its model-building step. Before ALAMO, best subset selection techniques were considered too time consuming to apply 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 Using ALAMO

3.1 Calling ALAMO from the command line

ALAMO reads model data and algorithmic options from a text file in a relatively simple format. Even though not required, it is strongly recommended that all ALAMO input files have the extension '.alm.' If the input file is named 'test.bar' and the ALAMO executable is named 'alamo,' issuing the command

```
alamo test
```

or

```
alamo test.alm
```

results in ALAMO parsing the file and solving the problem. In addition to screen displays, ALAMO can optionally provide results in the listing file 'test.lst' that is generated during the run.

3.2 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: logarithmic, exponential, sine, cosine, and monomials with powers 1, 2, and 3.

```
! Example 1 with data from z = x^2
```

```
ninputs 1
```

```
noutputs 1
```

```
xmin -5
```

```
xmax 5
```

```
initialpoints 11
```

```
ndata 11
```

```
logfcns 1
```

```
expfcns 1
```

```
sincfcns 1
```

```
cosfcns 1
```

```
mono 3
```

```
monomialpower 1 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.

3.3 Input file grammar

The following rules should be adhered to while 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.
- Input is not case sensitive.
- Blank lines, white space, and lines beginning with *, #, % or ! are skipped.
- 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.
- Character-valued options, such as paths and file names, that contain spaces and forward slashes (/) must be provided in quotes.

3.4 ALAMO data and options specification statements

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 .
INITIALPOINTS	Number of data points in the initial sample set. INITIALPOINTS must be a nonnegative integer.

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.

Optional data specifications

This section contains optional data relating to the particular problem to be solved.

Option	Description	Default
NDATA	Number of data points in a preexisting data set specified by the user. NDATA must be a nonnegative integer that is no more than INITIALPOINTS.	0
NVALDATA	Number of data points in a preexisting validation data set specified by the user. This data set is not used to develop the model but only to compute model errors at the validation data points. NVALDATA must be a nonnegative integer.	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
XFACTOR	Row vector of scaling factors used to scale the input variables. One per input variable, space separated.	1 1 1 ...
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.	X1 X2 X3 ...
ZLABELS	Row vector of labels to denote the output variables. One per input variable, space separated. Each label can be no more than 128 characters long.	Z1 Z2 Z3 ...
MONO	Number of monomial powers to be considered as basis functions. MONO must be a nonnegative integer.	0
MULTI2	Number of pairwise combinations of powers to be considered as basis functions. MULTI2 must be a nonnegative integer.	0
MULTI3	Number of three variable combinations of powers to be considered as 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
EXRFCNS	A 0–1 indicator. Exponential functions are considered as basis functions if 1; otherwise, they are not considered.	0
LOGFCNS	A 0–1 indicator. Logarithmic functions are considered as basis functions if 1; otherwise, they are not considered.	0

SINFCNS	A 0–1 indicator. Sine functions are considered as basis functions if 1; otherwise, they are not considered.	0
COSFCNS	A 0–1 indicator. Cosine functions are considered as basis functions if 1; otherwise, they are not considered.	0
RBF	A 0–1 indicator. 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 Gaussian and 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
SCRATCH	ALAMO creates this directory and uses it to store temporary files and results. If the directory exists, it is erased in the beginning of the run.	almscr
MODELER	Fitness metric to be used for model building. Possible values are 1, 2, 3, 4, 5, and 6 corresponding, respectively, to Bayesian information criterion, Mallow’s Cp, the corrected Akaike’s information criterion, the Hannan-Quinn information criterion, mean square error, and a convex penalty consisting of the sum of square errors and a term penalizing model size.	1
CONVPEN	When MODELER is set to 6, a convex penalty consisting of the sum of square errors and a term penalizing model size is used for model building. In this case, the size of the model is weighted by CONVPEN.	10.0
REGULARIZER	Regularization method used to reduce the number of potential basis functions before optimization of the selected fitness metric. Possible values are 0 and 1, corresponding to no regularization and regularization with the lasso, respectively.	0
SAMPLER	Technique to be used for adaptive sampling. Setting this option to 0 turns off the adaptive sampling step, thus forcing ALAMO to resort to model building using the INITIALPOINTS provided by the user. A value of 1 directs ALAMO to use a random sampler that requires the presence of a user-provided SIMULATOR. A value of 2 invokes the SNOBFIT code for sampling the SIMULATOR; SNOBFIT requires a MATLAB license to run.	0

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 capable of reading the number of requested data points, k , followed by k lines for each of the data points where function evaluations are requested. The simulator must return a number of lines, each containing a point in the input variable space where a simulation was performed, along with the corresponding output variable values. ALAMO allows for the number of these points to be different than 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 user's path or 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 to access other programs or files.	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 sub-problems, and print results.	1000
MAXTERMS	Row vector of maximum terms allowed in the modeling of output variables. One per input variable, space separated. A -1 signals that no limit is imposed.	-1 -1 -1 ...
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 ...
TOLMEANERROR	Row vector of convergence tolerances for mean errors in the modeling of output variables. One per output variable, space separated.	1e-3 1e-3 1e-3 ...
TOLRELMETRIC	Convergence tolerance for the chosen fitness metric for the modeling of output variables. This must be a nonnegative scalar.	1e-4
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
GAMS	Complete path of GAMS executable (or name if GAMS is in the user path). On some systems, GAMS does not run if there is white space in the current path.	gams
GAMSSOLVER	Name of preferred GAMS solver for solving ALAMO's mixed-integer quadratic subproblems. Special facilities have been implemented in ALAMO and BARON that make BARON the preferred selection for this option. However, any mixed-integer quadratic programming solver available under GAMS can be used.	BARON
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

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

```
BEGIN_VALDATA
```

```
:
```

END_VALDATA

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. This vector must be of length MONO.
MULTI2POWER	Row vector of powers to be considered for pairwise combinations in basis functions. This vector must be of length MULTI2.
MULTI3POWER	Row vector of powers to be considered for triplet combinations in basis functions. This vector must be of length MULTI3.
RATIOPOWER	Row vector of powers to be considered for ratios in basis functions. This vector must be of length RATIOS.

Options and parameters for constrained regression

This section contains optional data relating to constrained regression. The primary options that control application of constrained regression are:

Option	Description	Default
CONREG	A 0–1 indicator to specify whether constraint regression is used or not. By default, no constraint regression is used. If CONREG is set to 1, bounds on output variables are enforced.	0
CRNCUSTOM	Number of custom constraints (other than bounds). CRN-CUSTOM must be a nonnegative integer.	0

If CONREG is set equal to 1, some of the following vectors need to be specified.

Parameter	Description
ZMIN	Minimum values for output variables. One per output variable, space separated. This vector must be specified if CONREG equals 1.
ZMAX	Maximum values for output variables. One per output variable, space separated. This vector must be specified if CONREG equals 1.
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 EXTRAPXMAX.

EXTRAPXMAX	Maximum values for safe extrapolation region. One per input variable, space separated. If this vector is specified, ZMIN and ZMAX are enforced over EXTRAPXMIN to EXTRAPXMAX.
------------	---

Custom constrained regression, i.e., constrained regression for enforcing conditions other than simple bounds, can be done by setting CRNCUSTOM equal to the number of custom constrained regression conditions to be enforced. Once this number has been specified, the custom constraints themselves are specified 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 input and output variable labels. ALAMO will then enforce the constraint $g \leq 0$ when building a model for output variable j .

Finally, some algorithmic options that control implementation aspects of constrained regression may be optionally set:

Option	Description	Default
CRNINITIAL	Number of random bounding points at which constraints are sampled initially. CRNINITIAL must be a nonnegative integer.	0
CRMAXITER	Maximum allowed constrained regressions 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 bound in each iteration. CRNVIOL must be a positive integer.	100
CRNTRIALS	Number of random trial bounding points per round of constrained regression. CRNTRIALS must be a positive integer.	100

Compatibility with previous versions of ALAMO

Starting with ALAMO v. 2013.10.0, ALAMO's input format was changed. Compatibility with input requirements of earlier versions was 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
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.

3.5 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 exactly one command line argument.
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. Number of monomial powers (MONO) must be specified before specifying MONOMIALPOWER values.
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. Number of multi2 powers (MULTI2) must be specified before specifying MULTI2POWER values.
29. Number of multi3 powers (MULTI3) must be specified before specifying MULTI3POWER values.
30. Unable to open output file.
31. Maximum number of iterations reached.
32. Number of ratio powers (RATIOS) must be specified before specifying RATIOPOWER values.

33. Error while trying to use GAMS 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 run ordinary least squares regression.
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 TOLMEAN-ERROR 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 GAMS 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.

4 ALAMO output

4.1 ALAMO screen output

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

```
*****  
ALAMO version 2015.9.7. Built: LNX-64 Mon Sep 7 18:12:41 EDT 2015
```

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 The Optimization Firm, LLC., 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 0.30E-02 s)

Step 1: Model building using BIC

Model building for variable Z1

$Z1 = 1.0 * X1^{2.0}$

Calculating quality metrics on observed data set.

Quality metrics for output Z1

RMSE:	0.00
R2:	1.00
Model size:	1
BIC:	2.40
Cp:	-9.00
AICc:	-Infinity
HQC:	-Infinity
MSE:	0.00
Convex penalty:	10.0

Total execution time 0.40E-02 s

Times breakdown

OLR time:	0.0 s in 17 ordinary linear regression problem(s)
MIP time:	0.0 s in 0 quadratic integer problem(s)
Simulation time:	0.0 s to simulate 0 point(s)
All other time:	0.40E-02 s in 1 iteration(s)

Normal termination

The software first reports the version, platform, and compile 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 as 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. The initial model is $z = 0$ and, after the first mixed-integer quadratic program is solved, 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.

5 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.