

**CBP TOOLBOX VERSION 1.1
CODE INTEGRATION ENHANCEMENTS**

Cementitious Barriers Partnership

May 2013

CBP-TR-2013-03, Rev. 1

CBP TOOLBOX VERSION 1.1: CODE INTEGRATION ENHANCEMENTS

F. G. Smith and G. P. Flach
Savannah River National Laboratory
Savannah River Site
Aiken, SC 29808

K. G. Brown
Vanderbilt University, School of Engineering
Consortium for Risk Evaluation with Stakeholder Participation III
Nashville, TN 37235

May 2013
CBP-TR-2013-03, Rev.1

ACKNOWLEDGEMENTS

This report was prepared for the United States Department of Energy under Interagency Agreement No. DE-AI09-09SR22667 and is an account of work performed under that contract. This report was also prepared with the financial support by the U. S. Department of Energy, under Cooperative Agreement Number DE-FC01-06EW07053 entitled 'The Consortium for Risk Evaluation with Stakeholder Participation III' awarded to Vanderbilt University. Reference herein to any specific commercial product, process, or service by trademark, name, manufacturer, or otherwise does not necessarily constitute or imply endorsement, recommendation, or favoring of same by Savannah River Nuclear Solutions, Vanderbilt University or by the United States Government or any agency thereof. The views and opinions of the authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. This report is part of a larger multi-investigator project supported by the U. S. Department of Energy entitled the Cementitious Barriers Partnership. The opinions, findings, conclusions, or recommendations expressed herein are those of the authors and do not necessarily represent the views of the U.S. Department of Energy. This work was also partially supported by the National Institute of Standards and Technology Sustainable, High Performance Infrastructure Materials program.

DISCLAIMER

This work was prepared under an agreement with and funded by the U. S. Government. Neither the U.S. Government or its employees, nor any of its contractors, subcontractors or their employees, makes any express or implied: 1. warranty or assumes any legal liability for the accuracy, completeness, or for the use or results of such use of any information, product, or process disclosed; or 2. representation that such use or results of such use would not infringe privately owned rights; or 3. endorsement or recommendation of any specifically identified commercial product, process, or service. Any views and opinions of authors expressed in this work do not necessarily state or reflect those of the United States Government, or its contractors, or subcontractors.

Printed in the United States of America

**United States Department of Energy
Office of Environmental Management
Washington, DC**

**This document is available on the U.S. DOE Information Bridge and on the
CBP website: <http://cementbarriers.org/>
An electronic copy of this document is also available through links on the following
website: <http://srnl.doe.gov/>**

FOREWORD

The Cementitious Barriers Partnership (CBP) Project is a multi-disciplinary, multi-institutional collaboration supported by the United States Department of Energy (US DOE) Office of Waste Processing. The objective of the CBP project is to develop a set of tools to improve understanding and prediction of the long-term structural, hydraulic, and chemical performance of cementitious barriers used in nuclear applications.

A multi-disciplinary partnership of federal, academic, private sector, and international expertise has been formed to accomplish the project objective. In addition to the US DOE, the CBP partners are the Savannah River National Laboratory (SRNL), Vanderbilt University (VU) / Consortium for Risk Evaluation with Stakeholder Participation (CRESP), Energy Research Center of the Netherlands (ECN), and SIMCO Technologies, Inc. The Nuclear Regulatory Commission (NRC) is providing support under a Memorandum of Understanding. The National Institute of Standards and Technology (NIST) is providing research under an Interagency Agreement. Neither the NRC nor NIST are signatories to the CRADA.

The periods of cementitious performance being evaluated are >100 years for operating facilities and > 1000 years for waste management. The set of simulation tools and data developed under this project will be used to evaluate and predict the behavior of cementitious barriers used in near-surface engineered waste disposal systems, e.g., waste forms, containment structures, entombments, and environmental remediation, including decontamination and decommissioning analysis of structural concrete components of nuclear facilities (spent-fuel pools, dry spent-fuel storage units, and recycling facilities such as fuel fabrication, separations processes). Simulation parameters will be obtained from prior literature and will be experimentally measured under this project, as necessary, to demonstrate application of the simulation tools for three prototype applications (waste form in concrete vault, high-level waste tank grouting, and spent-fuel pool). Test methods and data needs to support use of the simulation tools for future applications will be defined.

The CBP project is a multi-year effort focused on reducing the uncertainties of current methodologies for assessing cementitious barrier performance and increasing the consistency and transparency of the assessment process. The results of this project will enable improved risk-informed, performance-based decision-making and support several of the strategic initiatives in the DOE Office of Environmental Management Engineering & Technology Roadmap. Those strategic initiatives include 1) enhanced tank closure processes; 2) enhanced stabilization technologies; 3) advanced predictive capabilities; 4) enhanced remediation methods; 5) adapted technologies for site-specific and complex-wide D&D applications; 6) improved SNF storage, stabilization and disposal preparation; 7) enhanced storage, monitoring and stabilization systems; and 8) enhanced long-term performance evaluation and monitoring.

Christine A. Langton, PhD
Savannah River National Laboratory

David S. Kosson, PhD
Vanderbilt University / CRESP

TABLE OF CONTENTS

Acknowledgements	ii
Disclaimer	ii
Foreword.....	iii
List of Figures.....	vi
List of Tables	vi
List of Acronyms and Abbreviations	vii
1 Summary.....	1
2 Introduction.....	1
3 CBP Toolbox Enhancements.....	2
3.1 Minor Changes to CBP Toolbox.....	2
3.2 Major Changes to CBP Toolbox	2
3.2.1 Enhanced Graphics to Display Model Results	8
3.2.2 Enhanced Error Messages	17
4 References.....	19
Appendix	
A Listing and Explanation of GNUPLOT Graphics Subroutines	20

LIST OF FIGURES

Figure 1. STADIUM input file after execution of instruction file commands.....	5
Figure 1 (Cont'd). STADIUM input file after execution of instruction file commands.....	6
Figure 1 (Cont'd). STADIUM input file after execution of instruction file commands.....	7
Figure 2. GoldSim plot of nitrate concentration from 2-layer STADIUM model calculation.....	11
Figure 3. STADIUM run controls dashboard.....	11
Figure 4. Two-dimensional x-y plot of nitrate concentration from 2-layer STADIUM model example calculation.....	12
Figure 5. Three-dimensional surface plot of nitrate concentration from 2-layer STADIUM model example calculation.....	12
Figure 6. LeachXS/Orchestra sulfate attack run controls dashboard.....	13
Figure 7. LeachXS/Orchestra sulfate attack View 2D LeachXS/Orchestra Results dashboard.....	13
Figure 8. GoldSim plot of pH profile through concrete barrier from 1-layer LXO sulfate attack model calculation.....	14
Figure 9. Two-dimensional x-y plot of pH profile in concrete from single layer LeachXS/Orchestra sulfate attack model example calculation.....	14
Figure 10. Three-dimensional surface plot of pH profile in concrete from a single layer LeachXS/Orchestra sulfate attack model calculation.....	15
Figure 11. Rotated three-dimensional surface plot of pH profile in concrete from a single layer LeachXS/Orchestra sulfate attack model calculation.....	15
Figure 12. Command line screen from Gnuplot application.....	16
Figure 11. Version 1.0 message reporting error in DLL instruction.....	17
Figure 12. Version 1.1 message reporting error in DLL instruction with improved diagnostics.....	17
Figure 13. Version 1.0 message reporting error reading input file.....	18
Figure 14. Version 1.1 message reporting error reading input file with improved diagnostics.....	18
Figure A-1. Example of GoldSim button settings for running the Gnuplot application.....	22

LIST OF TABLES

Table A-1. Arguments passed to Gnuplot graphics application.....	22
--	----

LIST OF ACRONYMS AND ABBREVIATIONS

CBP	Cementitious Barriers Partnership
CRESP	Consortium for Risk Evaluation with Stakeholder Participation
DLL	Dynamic Link Library
DOE	Department of Energy
ECN	Energy Research Centre of the Netherlands
GTG	GoldSim Technology Group
NIST	National Institute of Standards and Technology
NRC	Nuclear Regulatory Commission
SRNL	Savannah River National Laboratory
STADIUM	Software for Transport and Degradation in Unsaturated Materials
VU	Vanderbilt University

CBP Toolbox Version 1.1: Code Integration Enhancements

F. G. Smith and G. P Flach
Savannah River National Laboratory
Savannah River Site
Aiken, SC 29808

K. G. Brown
Vanderbilt University, School of Engineering
Consortium for Risk Evaluation with Stakeholder Participation III
Nashville, TN 37235

1 SUMMARY

This report describes enhancements made to code integration aspects of the Cementitious Barriers Project (CBP) Toolbox as a result of development work performed at the Savannah River National Laboratory (SRNL) in collaboration with Vanderbilt University (VU) in the first half of fiscal year 2013. Code integration refers to the interfacing to CBP partner codes that are used to analyze the performance of cementitious materials with the CBP Toolbox. The most significant enhancements are:

- 1) Improved graphical display of model results.
- 2) Improved error analysis and reporting.
- 3) Increase in the default maximum model mesh size from 301 to 501 nodes.
- 4) The ability to set the LeachXS/Orchestra simulation times through the GoldSim interface.

These code interface enhancements have been included in a new release (Version 1.1) of the CBP Toolbox.

2 INTRODUCTION

The modifications to the CBP Toolbox documented in this report involved improvements to the code integration functionality. The current version of the CBP Toolbox (Version 1.0) allows the user to run either a version of the STADIUM® code (SIMCO, 2010) or a version of the LeachXS/Orchestra code (ECN, 2007; Meeussen, 2009). A user interface to these CBP partner codes was developed with the

GoldSim software (Brown and Flach, 2009; Smith et al. 2010a). Simultaneous with the code upgrades described in this report, work has been conducted to enhance the CBP Toolbox modeling capabilities by including a model of cement carbonation developed in LeachXS/Orchestra. A description of the carbonation model is the subject of a separate report.

The connection between GoldSim and the external codes was created by developing a general Dynamic Link Library (DLL) interface. The overall concept behind this development was to use GoldSim as top level modeling software with interfaces to external codes for specific calculations. The DLL that performs the code linking is designed to perform the following functions:

1. Take a list of code inputs such as initial mineral and chemical concentrations from GoldSim,
2. Create an input file for the external application,
3. Run the external code, and
4. Return a list of outputs that are read from files created by the external application back to GoldSim for further processing.

Instructions for creating the input file, running the external code, and reading the output are contained within a user created file that is read and interpreted by the DLL. The design and functionality of the DLL interface has been described in detail by Smith et al. (2010a). This report describes modifications made to the DLL interface and within the GoldSim code to enhance code integration in going from the initial release Version 1.0 to Version 1.1.

3 CBP TOOLBOX ENHANCEMENTS

The enhancements listed below have been incorporated into Version 1.1 of the CBP Toolbox.

3.1 Minor Changes to CBP Toolbox

Two relatively minor changes were made in the GoldSim interface in Version 1.1. These changes, which do not require further explanation, are listed below.

1. An error in the assignment of density in STADIUM calculations for three-layer models was corrected.
2. The fractional porosity parameter used in the LeachXS/Orchestra sulfate attack model was modified to accept values between 0 and 0.9 instead of the more limited range from 0 to 0.45 range allowed in Version 1.0.

3.2 Major Changes to CBP Toolbox

Six changes of more significance were also made in Version 1.1 of the CBP Toolbox to enhance code integration capabilities. These changes are listed below and, in cases where examples are useful, described in more detail in subsections.

1. The maximum allowable mesh size for both STADIUM and LeachXS/Orchestra simulations was increased from 301 to 501. Implementing this change required changing the values of the number of GoldSim inputs and outputs in the DLL interfaces to the external codes. The total number of inputs and outputs must be entered as a fixed number in the six DLL Fortran files named “Params_xxx_yyy.f90” located in the CBP Toolbox Codes/DLLExternalCode_V1.1 folder. In the parameter file names, xxx is either LXO or STADIUM and yyy is either, 1Layer, 2Layers, 3Layers or CO3.

The DLL’s must be recompiled after the number of inputs and outputs in the parameter files are changed. Two batch files that can be used to compile and link the revised “Params_xxx_yyy.f90” files in two steps have been added to folder “Codes/DLLExternalCode_V1.1/G95_GoldSim”. These batch files are named:

g95_make_callFromGoldSim_LXO.bat, and
g95_make_callFromGoldSim_STADIUM.bat

If even larger meshes are required for other applications, CBP Toolbox users may wish to request a modified version of the code from the developers rather than attempt to make this change themselves. However, to facilitate making this change, comments were added to each of the “Params_xxx_yyy.f90” files explaining how the number of outputs is calculated from the number of mesh nodes and other simulation parameters. All of the parameter files must be changed to accommodate a change in the number of nodes or GoldSim will recognize a mismatch between the number of inputs and outputs being sent to the DLL interface and the number of inputs and outputs the interface will accept. In this event, GoldSim will display an error message and will not run.

2. Mesh positions are now extracted from STADIUM simulation output back to GoldSim. As noted in the plotting enhancements described below, concentrations within STADIUM are stored in vector arrays dimensioned with the mesh node number. By passing actual mesh positions back to GoldSim, concentrations can then be associated with the actual position in the material. Because, in the current version of the CBP Toolbox, GoldSim is simply functioning as an interface to the two partner codes and the codes are operating independently, the mesh positions are not used. However, the positions could potentially be used to improve GoldSim plots or to facilitate simulations using results from both partner codes.
3. The example STADIUM DLL instruction files DLL_STADIUM_2Layers.dat and DLL_STADIUM_3Layers.dat provided with the CBP Toolbox were modified to write the mesh size in all locations where it could be used. In the example STADIUM input file shown in Figure 1, input values that can be specified through GoldSim dashboards

and changed through the provided instruction files are highlighted. In this example, a mesh size of 216 nodes was specified. The values highlighted in green were not changed by the example instruction files provided in Version 1.0. These values all pertain to conduction boundary conditions at the outer boundary conditions which will default to zero. Therefore, it is not necessary to specify the outer node to obtain default conditions. However, to avoid introducing errors into simulations by ignoring this boundary, the example instruction files were modified to set these node numbers.

The Version 1.0 DLL instructions were general enough to allow the entry of some number of consecutive input values into consecutive rows or columns of model input with a single command. However, in this case, it was desired to enter the same input value into consecutive columns. A minor change was made to the DLL such that when the number of consecutive entries requested is negative the input value is not incremented and the same input is placed into file positions.

```

COOR
120cm-20cm-mesh.cor
ELEM
120cm-20cm-mesh.ele
RESO
    NUMBER_NUM_PARAM.    14
    integration_pts      2
    tolerance            1.00E-03
    itermax              30
    cartesian_axi       1
    Duration_years      1000
    Init_time_step_sec  86400
    f_sat               3
    Tangential_matrix   0
    damage              1
    physical_cl         0
    CO2_level_%         0
    Max_time_step_sec   4320000
    Step_Adapt_Factor   1.5
    Step_Adapt_Crit     5.00E-03

PREL
    N_PREL_GROUP        2
    N_PREL              18
    temperature         23
    W/B                 0.38
    Binder              405
    aggregates          1659
    Binder_density      2885
    Porosity            0.135
    Permeability        1.80E-21
    oh_diff_coef        1.40E-11
    Isotherm_b          -25.928
    Isotherm_c          0.4285
    Relative_perm       18
    init_hydrat         28
    tref_meas          28
    hydrat_a            0.8
    hydrat_alpha        0.015
    k_thermal           2
    spec_heat           1000
    ex_rate_CO2         1.00E-05

CHIM
    NUMBER_CHEM_PARAM.  3
    m_max              5
    print_level        1
    iter_max           1000

    Nions              11
    Nsolides           9
    
```

Figure 1. STADIUM input file after execution of instruction file commands.

```

Database_file CHM-DB-STADIUM-V3.txt

OH
Na
K
SO4
Ca
Al (OH) 4
Cl
H2SiO4
CO3
NO3
NO2

Portlandite
CaH2SiO4
Ettringite
Monosulfate
C4AH13
Thaumasite
Calcite
Monocarboaluminate
Gypsum

COND
Nb_cycles          1
Sequences_days     0      365

OH          1      0      216      0      0      0
Na          1      0      216      0      0      0
K           1      0      216      0      0      0
SO4         1      0      216      0      0      0
Ca          1      0      216      0      0      0
Al (OH) 4   1      0      216      0      0      0
Cl          1      0      216      0      0      0
H2SiO4     1      0      216      0      0      0
CO3        1      0      216      0      0      0
NO3        1      0      216      0      0      0
NO2        1      0      216      0      0      0
Humidity    2      0      1      0      0      1
            216      0      0      0      1

Potential      1      0      216      0      0      0
Temperature    0

CONV
Nb_cycles          1
Sequences_days     0      365

OH          0
Na          0
K           0
SO4         0
Ca          0
    
```

Figure (Cont'd). STADIUM input file after execution of instruction file commands.

```

Al (OH) 4      0
Cl             0
H2SiO4        0
CO3           0
NO3           0
NO2           0
Humidity      0
Potential     0
Temperature   2      0      1      365      0      5      15
                216      365      0      5      15

INIT
  external_file 0

  OH           400      670.08
  Na           282.1    4420
  K            138      120
  SO4          8        130.7
  Ca           0.5      0.41
  Al (OH) 4    0.1      0.14
  Cl           5        9
  H2SiO4       0        9.7
  CO3          0        2.9
  NO3          0        2000
  NO2          0        1575
  Rel_Humidity 1        1
  Potential    0        0
  Temperature  23      23

  Portlandite  13.6     41.9
  CaH2SiO4    37.9     103.3
  Ettringite   0        28.6
  Monosulfate  19.4     0
  C4AH13      14.8     0
  Thaumasite  0        0
  Calcite      0        4.8
  Monocarboaluminate 0      11
  Gypsum       0        0

IMPR
  number_print_times 8
  print_times
  10
  20
  50
  100
  200
  500
  1000
  2000
  print_before_chm 0
  level_1_2      1
  imp_flux_0_1_2 0

STOP

```

Figure (Cont'd). STADIUM input file after execution of instruction file commands.

4. The GoldSim simulation time is now used to set the LeachXS/Orchestra simulation time. In Version 1.0, the LeachXS/Orchestra run time was not related to the GoldSim run time, this lead to some confusion when users attempted to control LXO simulations. To correct this issue, one data statement and two new functions were created within the GoldSim interface. The data statement fixes the leach period at 90 days. This variable should be entered as user input through a GoldSim dashboard but this refinement was not made in Version 1.1. The new functions convert the 90 day leach period into hours and use the simulation time specified in GoldSim to calculate the number of LeachXS/Orchestra sweeps. The equation used to calculate the number of sweeps is:

$$Sweeps = 1 + trunc \left\{ \left(\frac{Simulation\ Duration}{Leach\ Period} \right) \left(\frac{1\ Sweep}{10\ hours} \right) \right\}$$

The bracketed calculation is truncated and one added to make sure the LXO simulation goes up to and beyond the specified simulation duration. The dimensional consistency of this equation is not clear. However, when this calculated value is entered into the LXO input as the sweeps variable and the leach period in hours is entered as the leach period, the LXO calculation runs for the duration of the simulation time set through the GoldSim simulation control. The two calculated values are written into the LeachXS/Orchestra leachxs_parameters.txt file using the DLL interface.

5. A Fortran application using GnuPlot was written and included with the Toolbox for enhanced plotting of STADIUM results. The use of this new part of the CBT Toolbox is described in Section 2.2.1 where a few examples are shown. An outline of the Fortran code is provided in Appendix A.
6. Error trapping and reporting within the DLL interface were improved. These changes required modification of Fortran codes filework.f90 and DLLEExternalCode_g95.f90 and recompiling the DLL codes. This modification to the CBP Toolbox is described in Section 2.2.2.

3.2.1 Enhanced Graphics to Display Model Results

Version 1.0 of the CBP Toolbox used GoldSim plotting options to display results from model calculations. A typical plot graphing the nitrite concentration at 100 years from a two-layer STADIUM model calculation is shown in Figure 2. As can be seen from Figure 3, which shows the corresponding “STADIUM Run Controls” dashboard, the calculation is for a material consisting of 120 cm of SRS salt waste with a 20 cm section of SRS Vault 2 concrete using 216 nodes in the model calculation. While STADIUM can save results at multiple times during the simulation, the CBP Toolbox was written assuming that the user would pass results at only one

time (presumably, but not necessarily, the last time in the calculation) back to GoldSim for further analysis. Within GoldSim, concentrations are stored in a vector array dimensioned by the maximum allowed nodes in the mesh. Therefore, the GoldSim plot of the data shows nitrate concentration as a function of computational node position. The computational nodes are not typically evenly spaced so the GoldSim plot distorts the spatial concentration distribution to some degree. It is also often difficult (or, as the example illustrates, virtually impossible) to read the node number along the x-axis on the GoldSim plot without editing the plot. The GoldSim plots are only available following completion of a simulation when GoldSim is in result mode and are lost when the CBP Toolbox is returned to edit mode to perform another set of calculations.

To enhance the display of model results, a Fortran application was developed using the Gnuplot software supplied with the CBP Toolbox to plot results from STADIUM and LeachXS/Orchestra calculations. The Fortran program reads data directly from output files created by the partner codes. Therefore, if there are either pre-existing STADIUM or LeachXS/Orchestra output files in the Runs/realization_0 directory, the revised Toolbox can display plots of these results without performing new model calculations. Two types of plots are available, a two-dimensional plot (x-y plot) of model results from the last calculation time and a three-dimensional plot (surface plot) of model results at each saved time.

Figure 4 shows an example of an x-y plot displaying the nitrate concentration from the same STADIUM example calculation used to obtain Figure 2. The plot now shows the nitrate concentration as a function of position in the model mesh giving a more accurate picture of the concentration profile through the material. The saltstone material is in the region from 0 to 120 cm while the vault concrete is in the region from 120 to 140 cm. Figure 5 shows an example surface plot graphing the nitrate concentration as a function of both distance and time. For this example calculation, STADIUM results were saved at 10 year intervals between 0 and 100 years.

The Gnuplot code assumes that the naming convention commonly used by SIMCO for STADIUM input files has been used so that input files for two and three layer models are:

stad09d-cbp-task7-2layers.inp for a two layer model and
stad09d-cbp-task7-3layers.inp for a three layer model.

Having separate file names for the input to the same model has the advantage of making two templates available for the user. However, using this convention, STADIUM creates separate output files correspondingly named:

stad09d-cbp-task7-2layers.xls for a two layer model and
stad09d-cbp-task7-3layers.xls for a three layer model.

In keeping with this convention, two sets of buttons for output plotting are provided, as shown in Figure 3. The top set of buttons apply for a two-layer model while the bottom set of buttons apply for the three layer model and will be active when this option is selected. However, the only difference between the two options is the name of the output file that the plotting routine opens. It would be more efficient to adopt a standard STADIUM input file name and thereby eliminate the redundant plotting features.

The plotting functions implemented for STADIUM results have also been applied to plot LeachXS/Orchestra output. Figure 6 is a screen capture of the revised dashboard used to control sulfate attack modeling. The previous “View Results” button has been replaced with two buttons that can be used to select either the two-dimensional x-y plots or three-dimensional surface plots. Figure 7 shows a screen capture of the GoldSim dashboard that appears when the “View 2D Results” button is selected. A similar screen appears when 3D results are selected.

Figure 8 shows a plot of the pH profile through the concrete barrier at the end of a 50 year simulation with the L XO one-layer sulfate attack model. Figure 8 was created with the Version 1.0 GoldSim plotting function. Figure 9 shows an x-y plot of the same pH profile made with the Gnuplot application. Figure 10 shows a surface plot of the pH results from this same trial calculation as a function of time and depth. A useful feature of the surface plots is that if the user clicks on the plot and holds the mouse down the plot can be rotated to any viewing angle. The Gnuplot application has been written so that the last screen view is printed when a copy of the plot is saved. This feature is demonstrated by Figure 12 where the view has been rotated about 90° counterclockwise about the z axis. This feature gives the user flexibility in displaying three-dimensional results when the default settings are not satisfactory.

When the plotting is activated a command line screen appears displaying the message “Hit enter to close plot”. When enter is pressed the plot disappears and another message appears on the command line screen as shown in Figure 12 which instructs the user to enter the character “j” to save a copy of the plot in jpeg format or to enter any other character to close the plotting application. When “j” is entered, a date and time stamped jpeg file is saved in the Runs/realization_0 folder. For example, this file would have a name such as:

date_28-03-2013_time_16-17-57.jpeg

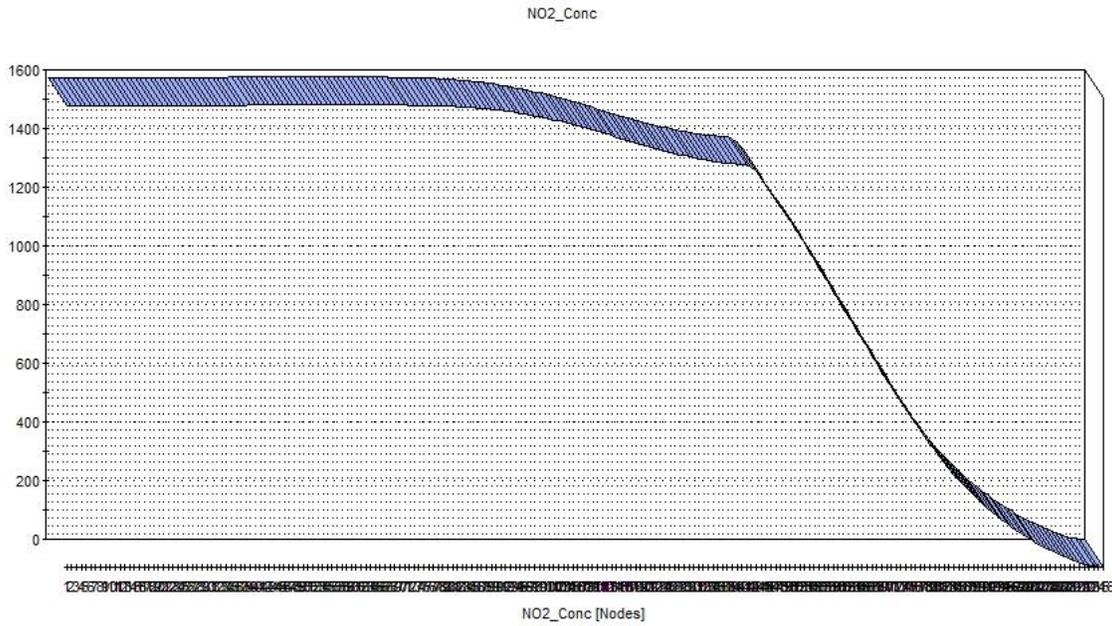


Figure 2. GoldSim plot of nitrate concentration from 2-layer STADIUM model calculation.

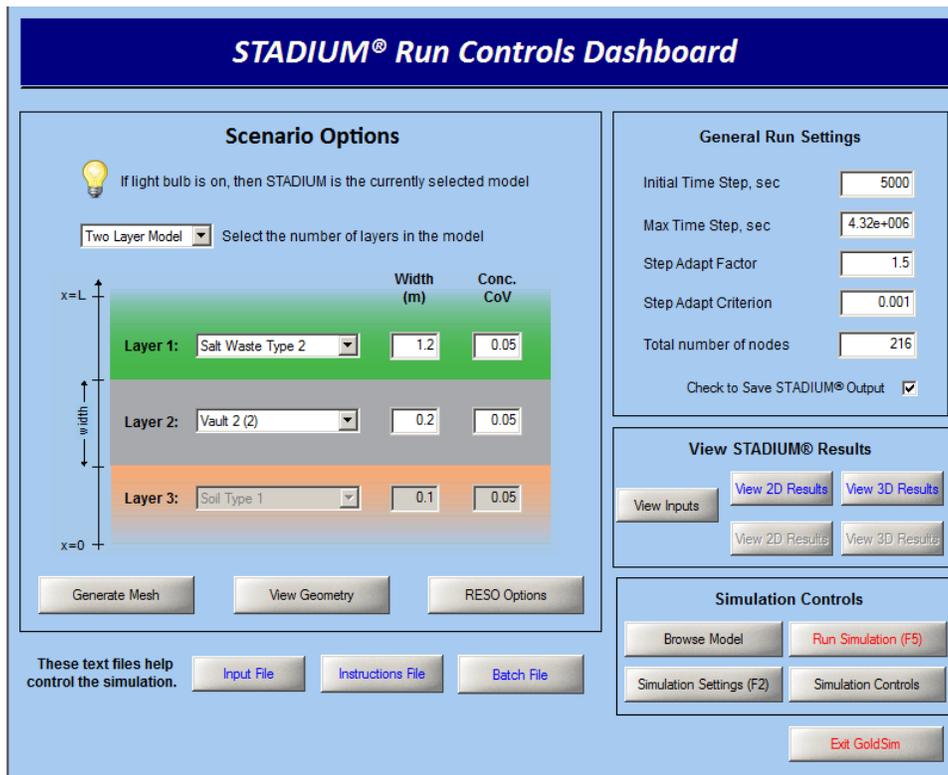


Figure 3. STADIUM run controls dashboard.

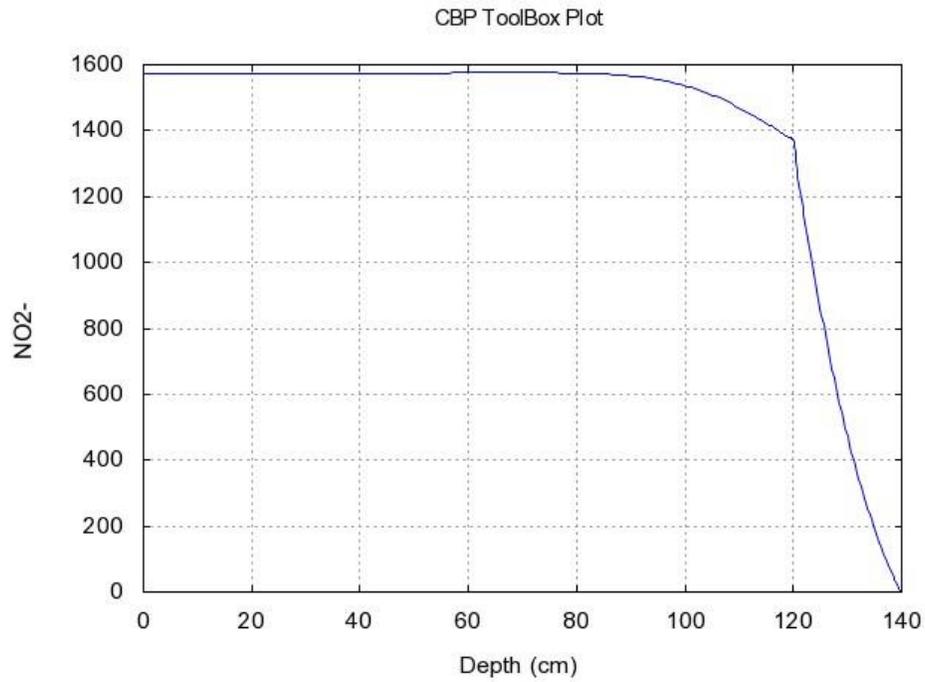


Figure 4. Two-dimensional x-y plot of nitrate concentration from 2-layer STADIUM model example calculation.

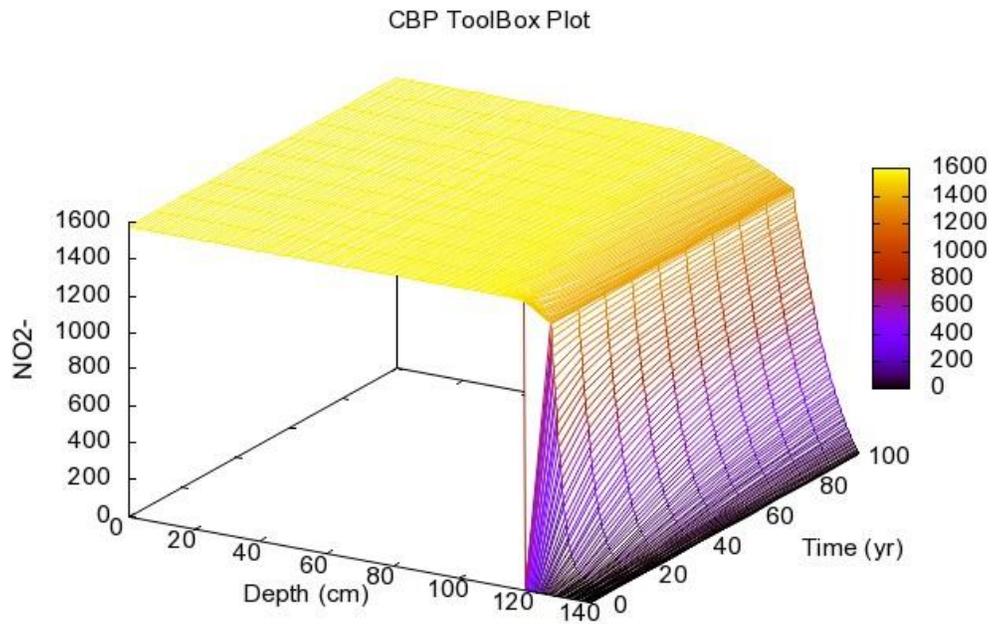


Figure 5. Three-dimensional surface plot of nitrate concentration from 2-layer STADIUM model example calculation.

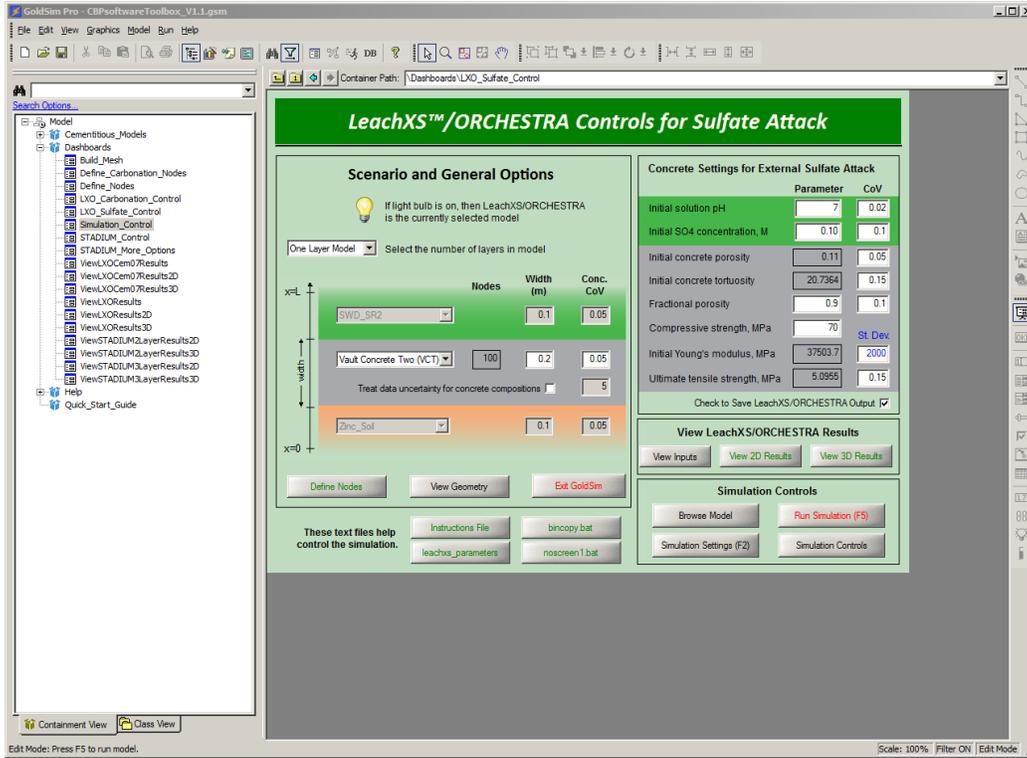


Figure 6. LeachXS/Orchestra sulfate attack run controls dashboard.

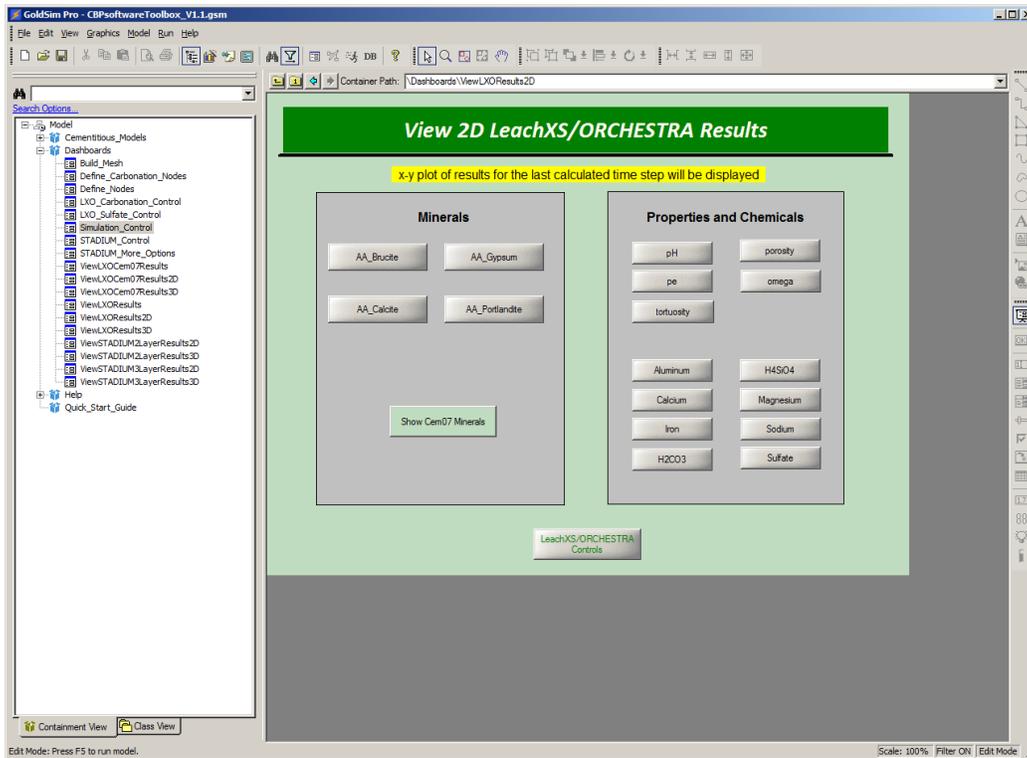


Figure 7. LeachXS/Orchestra sulfate attack View 2D LeachXS/Orchestra Results dashboard.

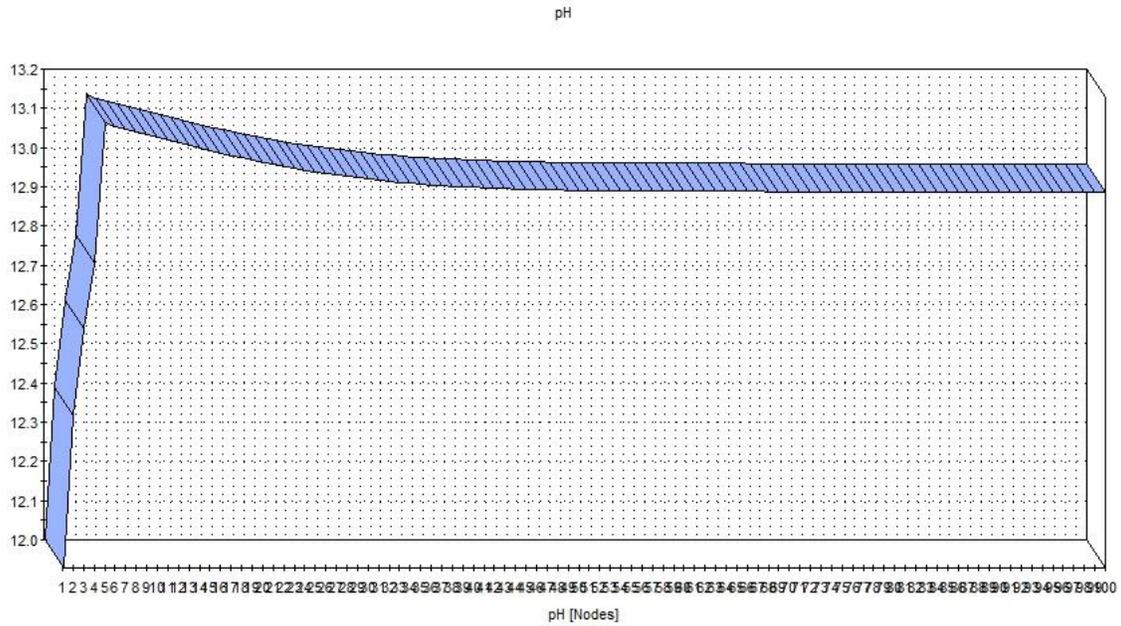


Figure 8. GoldSim plot of pH profile through concrete barrier from 1-layer LXO sulfate attack model calculation.

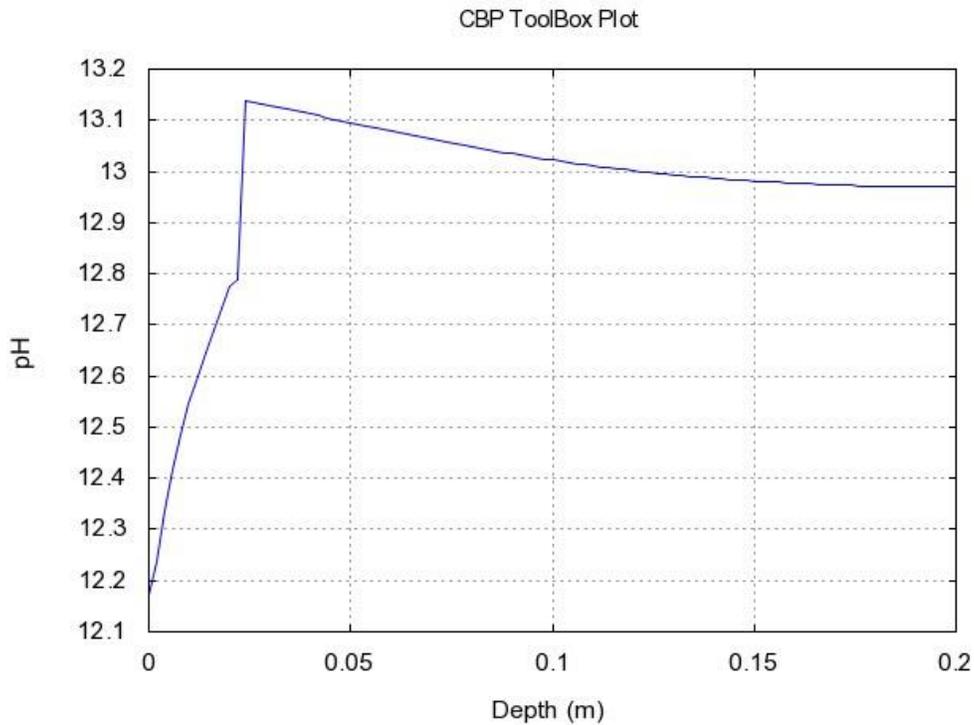


Figure 9. Two-dimensional x-y plot of pH profile in concrete from single layer LeachXS/Orchestra sulfate attack model example calculation.

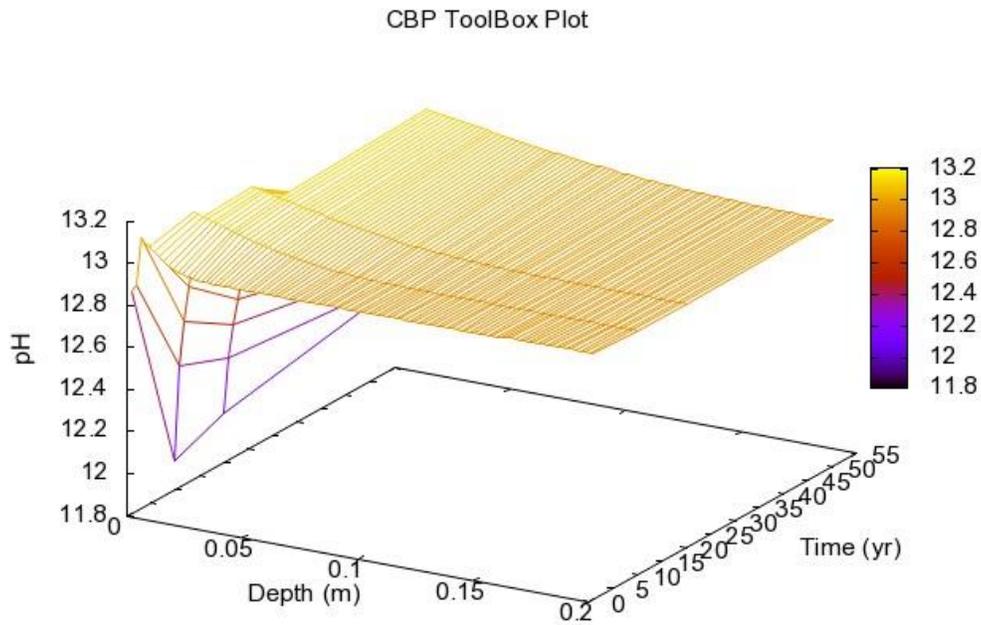


Figure 10. Three-dimensional surface plot of pH profile in concrete from a single layer LeachXS/Orchestra sulfate attack model calculation.

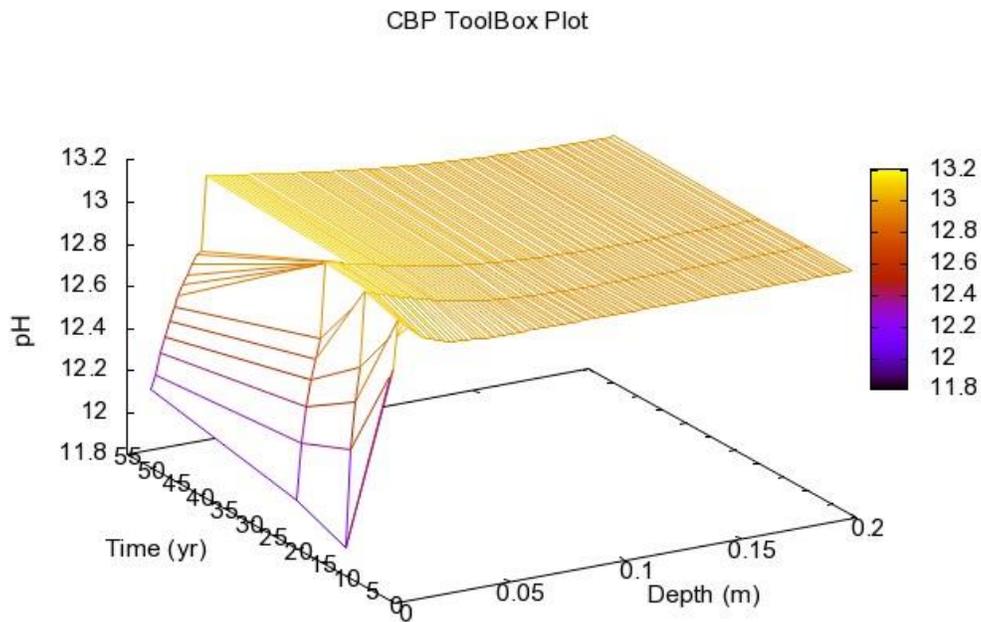


Figure 11. Rotated three-dimensional surface plot of pH profile in concrete from a single layer LeachXS/Orchestra sulfate attack model calculation.

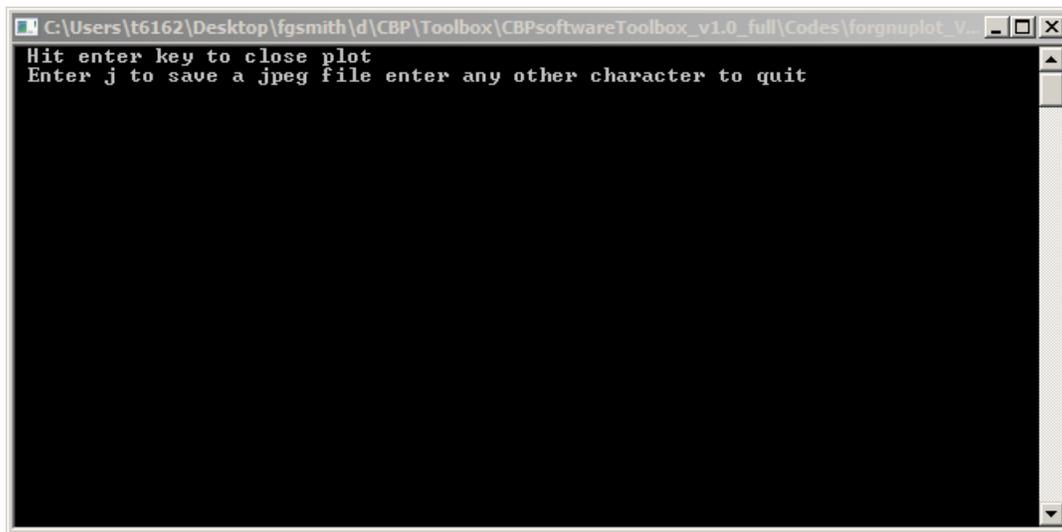


Figure 12. Command line screen from Gnuplot application.

3.2.2 Enhanced Error Messages

Error messages in Version 1.1 of the CBP Toolbox DLL interface have been enhanced to more clearly explain problems encountered during execution of the DLL instruction files. One example is shown below. Figure 11 is the error message produced in Version 1.0 if the user attempts to retrieve results from a STADIUM calculation for a time that was not included in the output. The error simply states that a “specified row, label or value” was not found. The improved error message in Version 1.1 is shown in Figure 12. The message now provides more specific information that “row value 100.0” was not found while trying to execute a GET command from instruction file “DLL_STADIUM_2Layers.dat”. The additional information will allow the user to quickly identify the problem in the DLL instruction file. In this case, results for year 100 were not output from the STADIUM simulation but the user has requested that this data be extracted from the output file. Similar error messages will explicitly identify row or column positions, values, or labels that cannot be located and identify whether this occurred during the execution of a PUT or GET command.

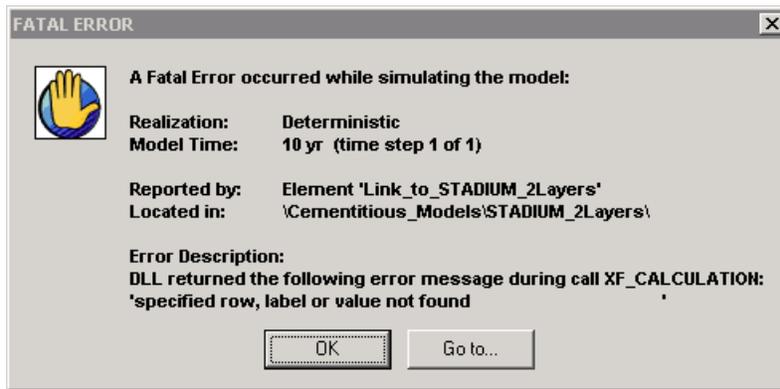


Figure 11. Version 1.0 message reporting error in DLL instruction.

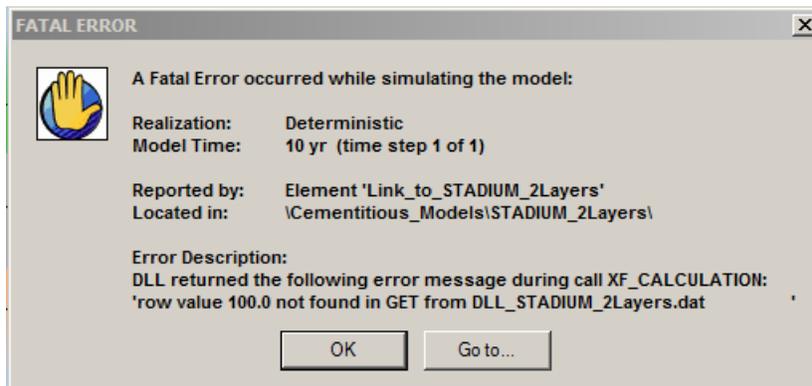


Figure 12. Version 1.1 message reporting error in DLL instruction with improved diagnostics.

A second example of improved error diagnostics is illustrated in Figures 13 and 14. In this case, the name of the STADIUM input file has been entered incorrectly. The Version 1.0 error message is shown in Figure 13 where the non-specific message ***** READ ERROR ***** is returned. The Version 1.1 error message returns the message that the named file was not found.

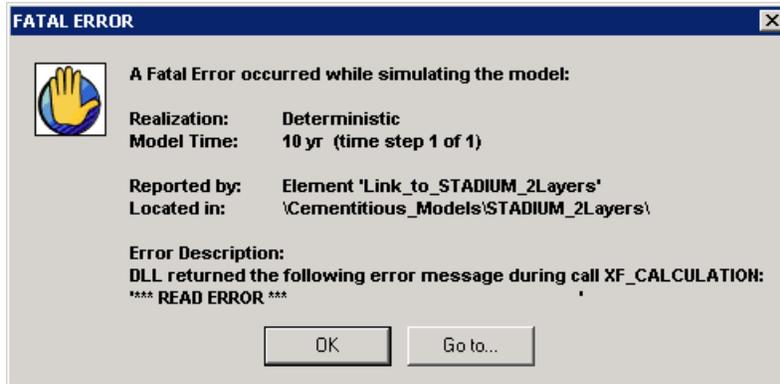


Figure 13. Version 1.0 message reporting error reading input file.

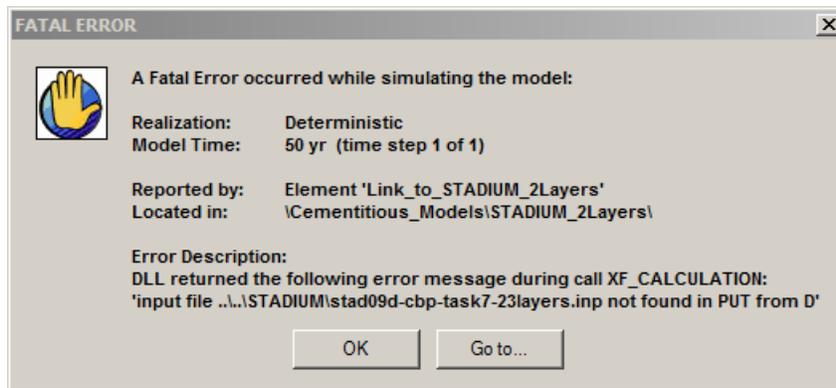


Figure 14. Version 1.1 message reporting error reading input file with improved diagnostics.

4 REFERENCES

Brown, KG & Flach, GP 2009, *CBP Software Summaries for LeachXS™/ORCHESTRA, STADIUM®, THAMES, and GoldSim*, CBP-TR-2009-003, Rev. 0, in *Description of the Software and Integrating Platform*, Vanderbilt University/CRESP and Savannah River National Laboratory; Cementitious Barriers Partnership, Nashville, TN and Aiken, SC. Available from: <http://cementbarriers.org/reports.html>.

Crawford, R., *Gnuplot 4.4 An Interactive Plotting Program*, November 25, 2009.

ECN 2007, *LeachXS User Manual*, Energy Research Center of The Netherlands, Denmark. Available from: <http://www.leaching.org> (September 1, 2009).

Meeussen, JCL 2003, *ORCHESTRA: An Object-Oriented Framework for Implementing Chemical Equilibrium Models*, *Environmental Science & Technology*, vol. 37, no. 6, pp. 1175-1182.

SIMCO 2010, *CBP Task 7 Demonstration of STADIUM® for the Performance Assessment of Concrete LAW Storage Structures*, CBP-TR-2010-007-C3, Rev. 0, SIMCO Technologies Inc.; Cementitious Barriers Partnership, Quebec, Canada. Available from: <http://cementbarriers.org/reports.html>.

Smith III, FG, Flach, G & Brown, KG 2010a, *CBP Code Integration GoldSim DLL Interface*, CBP-TR-2010-009-2, Rev. 0, Savannah River National Laboratory and Vanderbilt University/CRESP; Cementitious Barriers Partnership, Aiken, SC and Nashville, TN. Available from: <http://cementbarriers.org/reports.html>.

Smith III, FG, Flach, G, Brown, KG & Sarkar, S 2010b, *CBP Phase I Code Integration*, CBP-TR-2011-009-1, Rev. 0, Savannah River National Laboratory and Vanderbilt University/CRESP; Cementitious Barriers Partnership, Aiken, SC and Nashville, TN. Available from: <http://cementbarriers.org/reports.html>.

APPENDIX A
LISTING AND EXPLANATION OF GNUPLOT
GRAPHICS SUBROUTINES

The CBP Toolbox includes a copy of the freeware Gnuplot 4.4 (2009). Gnuplot is an interactive plotting program that has been under development since 1986 and contains many features that allow plotting of a wide variety of graphs to display data. The Gnuplot graphics code used in the CBP Toolbox to plot results from the STADIUM and LeachXS/Orchestra models was written in Fortran 90 and is modularized into the two files and 10 subroutines or functions listed below.

program plotsetup.f90

```
subroutine count_Stadium
subroutine read_Stadium
subroutine count_LXO
subroutine read_LXO
```

module gnuforplot.f90

```
subroutine xyplot
subroutine surface
subroutine run_gnuplot
function my_date_and_time
function output_terminal
subroutine get_unit
```

The Gnuplot application (`gnuforplot.exe` in directory `Codes/forgnuplot_V1.1`) is executed by clicking on one of the GoldSim buttons shown in Figure 3 or Figure 6. Figure A-1 shows the properties of a typical button. The button that produces an x-y plot of nitrate concentration from a 2-layer STADIUM calculation is used as an example. Program `plotsetup` is the main program in the plotting application that collects the data to be plotted and calls the plotting subroutine. The first action the program takes is to read the six integer values in the argument list. Table A-1 lists these arguments, their function, and the values that can be entered. The last argument is only needed to identify the name of the STADIUM output file. Program `plotsetup` then reads the data from the indicated source file and calls either the `xyplot` or `surface` plotting routine in module `gnuforplot`. The operation of individual subroutines and functions is explained below.

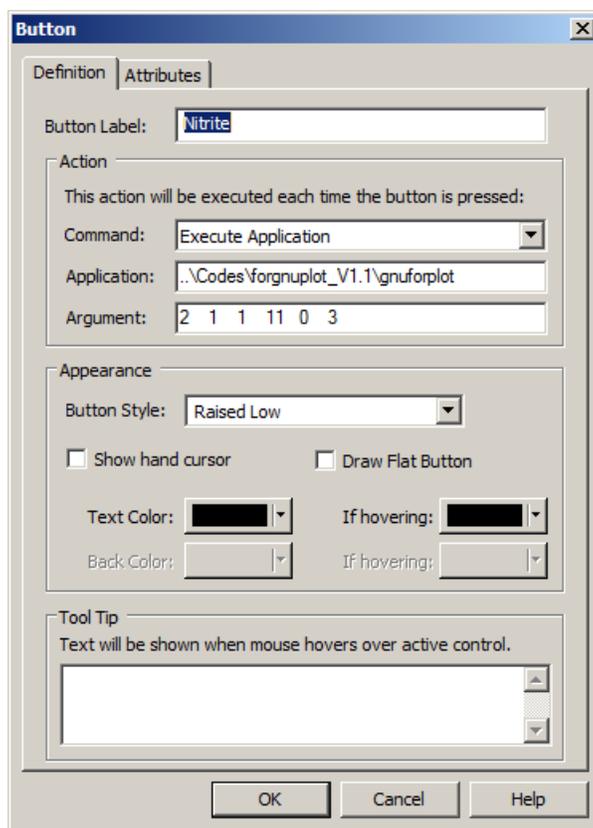


Figure A-1. Example of GoldSim button settings for running the Gnuplot application.

Table A-1. Arguments passed to Gnuplot graphics application.

Argument Number	Function	Value	Result
1	Plot Type	1	x-y
		2	surface
2	Model	1	Stadium
		2	LXO
3	Material	1	Chemical
		2	Mineral
4	Stadium Species	1-11	Chemicals
		1-9	Minerals
5	LXO Species	1-8	Chemicals
		1-30	Minerals
		1-5	Properties
6	Layers	2	Stadium 2-layer
		3	Stadium 3-layer

GNUPLOT SUBROUTINES

count_Stadium (ninp, nodes, times) and
count_LXO (ninp, nodes, times)

These subroutines count the number of nodes and the number of time steps in the STADIUM and LXO model output files.

ninp.....Integer input argument identifying the unit number to read (fixed value = 22)
nodesNumber of nodes used in the model calculation
timesNumber of times values are output from the model calculation

read_Stadium (ninp, nodes, times, nfield, xyz, ntot) and
read_LXO (ninp, nodes, times, nfield, xyz, ntot)

Irrespective of the plot type specified, these subroutines read all output values from the STADIUM and LXO output files using the data dimensions determined in the `count_Stadium` and `count_LXO` subroutines. Separate arrays are created for STADIUM and LXO with the thought that in a future release the ability of plot data from both models on the same graph will be added.

ninp.....Integer input argument identifying the unit number to read (fixed value = 22)
nodesNumber of nodes used in the model calculation
timesNumber of times values are output from the model calculation
nfield.....Position of plot variable in a line of model output
xyzThree-dimensional array dimensioned (nodes, times, plot values)
ntot.....Number of variable entries in a line of output

module gnuforplot

This main module of the Gnuplot graphics routines defines data arrays containing lists of STADIUM and L XO chemicals and minerals available for plotting and contains the subroutines and functions described below.

xyplot (xyz, model, material, species, style, pause, color, terminal, filename, input, linewidth)

xyzThree-dimensional array of dimension (nodes, times, plot values)
modelInput argument number 1 identifying model
materialInput argument number 2 identifying material
species.....Input argument number 3 identifying species
styleGnuplot parameter that can be used to plot either points or lines. The value has been set to plot lines.
pauseGnuplot parameter used to pause execution after plot is displayed
colorGnuplot parameter used to control plot color. The value has been set to 'blue'.
terminalOptional Gnuplot parameter to identify display terminal. The value has been set to 'windows'
filenameOptional Gnuplot parameter to identify input file name (unused)
inputOptional Gnuplot parameter to identify (unused)
linewidth.... Gnuplot parameter to control the width of the plotted line. The value has been set to 2.

Subroutine `xyplot` controls the plotting of two-dimensional x-y plots. Gnuplot contains many parameters that can be used to control the display. Only a few of these have been implemented in this application.

surface (xyz, model, material, species, pause, palette, terminal, filename, pm3d, contour, persist, input)

xyzThree-dimensional array of dimension (nodes, times, plot values)
modelInput argument number 1 identifying model
materialInput argument number 2 identifying material
species.....Input argument number 3 identifying species
pauseGnuplot parameter used to pause execution after plot is displayed
palette.....Gnuplot parameter used to set graphics color pallet. The value has been set to 'RGB'.
terminalOptional Gnuplot parameter to identify display terminal. The value has been set to 'windows'
filenameOptional Gnuplot parameter to identify input file name (unused)
pm3d.....Optional Gnuplot parameter to control surface display (unused)

contour.....Optional Gnuplot parameter to control surface contouring (unused)
persist.....Optional Gnuplot parameter to plot display (unused)
inputOptional Gnuplot parameter to identify (unused)

Subroutine `surface` controls the plotting of three-dimensional plots. Gnuplot contains many parameters that can be used to control the display. Only a few of these have been implemented in this application.

run_gnuplot (command_file_name)

command_file_name.....Name of file containing Gnuplot commands

Subroutine `run_gnuplot` is called to run Gnuplot. Commands to Gnuplot are written to a temporary file in subroutines `xyplot` and `surface` that is read by Gnuplot. The command file name is passed into this subroutine.

write_xydata (xyz,plot_time,data_file_name)

xyzThree-dimensional array of dimension (nodes, times, plot values)
plot_time.....Simulation time that x-y data was extracted
data_file_name.....Name of file containing data for plotting

Subroutine `write_xydata` is called to extract data from the `xyz` array at the final time step for x-y plotting. The data is written to file `data_file_name` where it will be read by Gnuplot and the time is written to variable `plot_time` to be used in the plot title.

write_surfdata (xyz,data_file_name)

xyzThree-dimensional array of dimension (nodes, times, plot values)
data_file_name.....Name of file containing data for plotting

Subroutine `write_xydata` is called to extract data from the `xyz` array and write it to file `data_file_name` where it will be read by Gnuplot.

my_date_and_time ()

Function `my_date_and_time` is called to create a string from the system date and time. The string is used to give a unique name to jpeg plot files saved by the user.

output_terminal (terminal)

terminalOptional Gnuplot parameter to identify display terminal

Subroutine `output_terminal` uses the `terminal` parameter specified in the calls to `xyplt` and `surface` to define display terminal settings. The `terminal` parameter has been set to 'windows' for this application.

get_unit (iunit)

iunitFortran unit number

Subroutine `get_unit` checks for a free Fortran unit number that is used to create the data file where Gnuplot commands are written.



U.S. DEPARTMENT OF
ENERGY



VANDERBILT
UNIVERSITY



NIST
National Institute of
Standards and Technology



SIMCO
Technologies inc.