SOURCE IST 2.0: Development and Beta Testing

Duncan H. Barber, Fernando C. Iglesias, Yung Hoang,
(Fuel & Fuel Channel Analysis Department, Ontario Power Generation)
Lawrence W. Dickson, Raymond S. Dickson,
(Fuel Safety Branch, Atomic Energy of Canada Ltd)
Michael J. Richards (Analyse et fiabilité, Hydro-Québec), and
Robert A. Gibb (Nuclear Analysis, New Brunswick Power)

Abstract

SOURCE IST 2.0 is the Industry Standard fission product release code that is being developed by Ontario Power Generation, New Brunswick Power, Hydro-Québec, and Atomic Energy of Canada Ltd. This paper is a report on recent progress on requirement specification, code development, and module verification and validation activities. The theoretical basis for each model in the code is described in a module Software Theory Manual. The development of SOURCE IST 2.0 has required code design decisions about how to implement the software requirements. Development and module testing of the β1 release of SOURCE IST 2.0 (released in July 1999) have led to some interesting insights into fission product release modelling. The beta testing process has allowed code developers and analysts to refine the software requirements for the code. The need to verify physical reference data has guided some decisions on the code and data structure design. Examples of these design decisions are provided.

Module testing, and verification and validation activities are discussed. These activities include code-targeted testing, stress testing, code inspection, comparison of code with requirements, and comparison of code results with independent algebraic, numerical, or semi-algebraic calculations.

The list of isotopes to be modelled by SOURCE IST 2.0 provides an example of a subset of a reference data set. Isotopes are present on the list for a variety of reasons: personnel or public dose, equipment dose (for environmental qualification), fission rate and actinide modelling, or stable (or long-lived) targets for activation processes. To accommodate controlled changes to the isotope list, the isotope list and associated nuclear data are contained in a reference data file.

The questions of multiple computing platforms, and of Year 2000 compliance have been addressed by programming rules for the code. By developing and testing modules on most of the different platforms on which the code is intended to run, cross-platform issues have been identified prior to the final code release.

Overview

The SOURCE IST 2.0 computer code (Reference 1) will provide a tool for calculating the release of fission products from the uranium oxide fuel within the fuel bundles. The code uses a sub-division of the fuel into basis units. Since the fuel volume changes with temperature and porosity, the mass of initial uranium in the basis unit serves as the constant parameter for describing the basis unit. The basis unit may be as a small as an annular ring within an axial segment of a single fuel element, or as large as a fuel element. Five inventory 'bins' are modelled for each basis unit. The role of SOURCE IST 2.0 is to simulate the transfer of fission products from one bin to another. The bins are:

Grain Matrix: the granular, solid, uranium oxide phase and its contained fission products;

Grain Boundary: the gas and vapour in grain boundary bubbles, and the solid or liquid precipitates on the surface of fuel grains within closed porosity;
Fuel Surfaces: solid and liquid precipitates on the surfaces of the fuel;
Gap: the gas and vapour in the pellet-to-sheath gap and in open porosity; and
Released: the material released outside the fuel sheath from the basis unit.

The phenomena to be modelled by SOURCE IST 2.0 were taken from the list of primary phenomena defined for the accidents in the Fission Product Release and Transport Validation Matrix (see Reference 2). Those phenomena that are directly modelled are:

Diffusion	Temperature Transients
Grain Boundary Sweeping / Grain Growth	Grain Boundary Separation
Grain Boundary Coalescence / Tunnel Interlinkage	Fission Product Leaching
Vapour Transport / Columnar Grains	Matrix Stripping
Fission Product Vaporisation / Volatilisation	Fission Yield, Decay and Transmutation

SOURCE IST 2.0 will also account for releases due to the following phenomena, once the fuel performance codes have incorporated models to provide SOURCE with appropriate fuel conditions.

Gap Transport (Failed Elements)	UO2 Zircaloy Interaction
UO2+x Formation	UO2 Dissolution by Molten Zircaloy
U4O9-U3O8 Formation	Fuel Melting
UO2-x Formation	Fuel Cracking (Thermal)

These phenomena have been divided amongst the following modules:

1. Inventory calculation,
2. Grain growth,
3. Grain boundary sweeping,
4. Fission product diffusion
5. Fission product re-distribution calculations due to matrix stripping (fuel volatilisation),
6. Fission product re-distribution calculations during UO_2/Zircaloy interaction,
7. Fission product re-distribution calculations during UO_2 dissolution by molten Zircaloy,
8. Fission product re-distribution calculations due to fuel melting,
9. Fission product re-distribution calculations due to grain boundary separation,
10. Fission product re-distribution calculations due to grain boundary coalescence/tunnel interlinkage,
11. Fission product vaporisation
12. Gap Transport
13. Releases due to rewet, and
14. Leaching releases.

Development

Software Requirement Specifications

Two types of Software Requirement Specifications (SRS) are prepared. These are the high-level SOURCE 2.0 specification that describes the entire code, and the module SRS which document the detailed requirements for individual modules. Prior to adoption of SOURCE 2.0 as an

Industry Standard Tool, a high-level Software Requirement Specification (SRS) had been prepared by Ontario Hydro Nuclear (now Ontario Power Generation), Hydro-Québec, and New Brunswick Power. This SRS was reviewed by the IST working group on Fission Product Release & Transport. Revisions were made to the SOURCE 2.0 SRS to incorporate new requirements of the IST partners. One new requirement of New Brunswick Power and Atomic Energy of Canada is the ability to model CANFLEX fuel, which has more fuel elements than other modelled bundle designs, and has fuel elements of different diameters in the same bundle.

The high-level SOURCE IST 2.0 SRS identifies the phenomena to be modelled by SOURCE IST 2.0. A development plan was developed to divide the different phenomena amongst software modules. Modular development results in a set of modules, each of which contains only closely-coupled processes. Disparate calculations are performed by separate modules. The generation, decay, neutronic transformation and diffusion of fission products within the fuel grains are closely-coupled phenomena that are performed in a single module (Fission Product Diffusion). The careful preparation of module software requirements ensures that processes are not accounted for in multiple modules. For example, the generation, decay and neutronic transformation of fission products within the grains is not included in the inventory calculation module. Instead this module uses values calculated by the fission product diffusion module.

With the exception of fission yield, decay and neutronic transmutation, the phenomena to be modelled by SOURCE IST 2.0 are listed in the Fission Product Release and Transport Validation Matrix. The Validation Matrix contains descriptions of the phenomena and references to literature dealing with these phenomena. These descriptions and references served as a basis for preparing the SRS for each module.

Software Theory Manual

The code and module Software Requirement Specifications guide the author of a Software Theory Manual for the module in selecting the desired features to be modelled. The Software Theory Manual describes the basis for modelling of the physical/chemical phenomena included in a module. It does not normally discuss the numerical analysis, or computing algorithms in the solution methodology used to evaluate the model. During the preparation of the Software Theory Manual, additional input data requirements for the module may be identified. It is necessary to determine where the data may be available, whether a data item is a fixed parameter, whether it will be local or global, and whether other modules might share the data. Revision of the Software Requirement Specification to reflect the results of these determinations is required.

Software Design Description

The Software Design Description documents the design of the software. It contains:

- a high-level description of the code for the module,
- a low-level description of each SUBROUTINE or FUNCTION, (including derivations required to get from the mathematical formulation of the problem in the Software Theory Manual to the coding present in the module),
- a list of local reference data in the module (with values, units, and a literature reference),
- a list of global reference data provided to the module,
- a complete listing of the FORTRAN code for the module, and
- a listing of a test driver used to test the module.

Programmer's Manual

The SOURCE 2.0 development team adopted a set of Programming Rules before starting development. These rules require or recommend certain programming practices that are aimed at providing a legible and maintainable code that can be run on a variety of target platforms. Each participating organisation has reviewed these rules. These rules are the basis for much of the Programmer's Manual. Since SOURCE IST 2.0 development is undertaken at multiple sites and the code is developed and is to be run on a number of different platforms, platform breadth is a major consideration. The prescribed programming language is FORTRAN-77 with a few defined extensions permitted. The extensions are almost all FORTRAN-90 features and are available in the FORTRAN compiler at each site. Thus FORTRAN-90 represents a migration path for SOURCE IST 2.0 when each site is able to support it.

Verification Activities

Stress Testing

Stress testing is the subject of another paper at this conference (Reference 3). It has been used to test modules over a range of one input variable at a time. The results shown in Figures 1, 2 and 3 were obtained when testing the grain boundary sweeping module. This module calculates the fraction of the solid fuel volume that is swept during grain growth. The calculated swept fraction of a grain with an initial diameter and length of 2 μm is subjected to different amounts of equiaxed grain growth is shown in Figure 1. The behaviour, as expected, is smooth and continuous, and is within the expected range 0 to 100% swept volume. The results also compare favourably with the expected analytic solution. The results for a similar test for columnar growth are shown in Figure 2. The calculated swept fraction (f_{sw}) is plotted vs. columnar grain growth during a time step (ΔL_{col}). Whilst the results are continuous, discontinuities in the first derivative ($df_{sw}/d\Delta L_{col}$) are evident. A close-up of the region surrounding the jump is shown in Figure 3. The module is switching amongst three expressions for the swept fraction. The first expression, which corresponds to the rising curve, is
$f_{sw} = \Delta L_{col}/(L+\Delta L_{col})$ and applies when $L+\Delta L_{col} \leq$ 2.0E-4 m (200μm). The second expression $f_{sw} = \Delta L_{col}/$(2.0E-4 m) corresponds to a straight line from the origin to the point (1,2.0E-4 m) and applies when $L+\Delta L_{col} \geq$ 2.0E-4 m (200μm). When $L+\Delta L_{col}$ = 2.0E-4 m the two expressions are equal, which provides continuity at the changeover point. The final expression is the upper limit that $f_{sw} \leq 1$, which means that . $f_{sw} = 1$, for $\Delta L_{col} \geq$ 2.0E-4 m. When ΔL_{col} = 2.0E-4 m, either of the last expressions applies, again providing continuity in the swept fraction. In this case it can be concluded that the module results are a correct implementation of the model, that the model is continuous in swept fraction as columnar grain growth increases, but that the model is not continuous in the first derivative ($df_{sw}/d\Delta L_{col}$).

Comparison with Analytic Solution

An analytic solution to the model that has been implemented may exist for a particular test case. In many cases, these solutions do not cover the entire problem domain, or impose restrictive conditions on the problem to be solved. Consider the steady-state solution for the concentration profile of the first member of a decay chain for the diffusion of a radioactive fission product from a spherical fuel grain (Reference 4):

$$C(r) = B/\lambda \{1 - R \sinh(\alpha r)/[r \sinh(\alpha R)]\} \quad (1)$$

where C(r) is the concentration (atoms/m^3),
r is radial position between 0 and R (m),
B is the production (birth) rate from fission (atoms/(m^3.s)),

λ is the decay constant of the nuclide (1/s),
R is the radius of the spherical grain (m),
α is the "lumped" constant $(\lambda/D)^{1/2}$ (1/m) where D is the diffusion coefficient (m^2/s),
and sinh is the hyperbolic sine function.

Comparisons of this analytic result to the results of module calculations are helpful under certain conditions. Indeed if these conditions are met, agreement between the module results and the correct evaluation of this analytic solution is a necessary, but not sufficient, condition for the module to be correct. In this case, the analytic expression contains no information about the transient response of the solution. Therefore there is no basis to evaluate the transient response of the module. The conditions placed on the test are that the birth rate (B) is constant over the time duration of interest, that the time duration is long enough for the steady-state solution to be reached (alternatively that the half-life is short enough for a steady-state to be reached), and that the diffusion coefficient (D) is constant. It is unlikely that any integrated test would reproduce all of these conditions. One strength of module testing is that the module can be isolated from the rest of the code and the external conditions can be set to arbitrary test values that are consistent with a stylised test case.

A further difficulty with the above comparison is that the analytic expression cannot be evaluated over the entire domain of the input variables and parameters if it is coded as written. On a computing platform with finite range, the expression for $\sinh(\alpha R)$ and $\sinh(\alpha r)$ are subject to overflow for large values of αR or of αr. Unfortunately such values correspond to the small values of D close to the expected values of D within the temperature range of interest. Fortunately, various approximations or limits can be substituted into equation (1) that allow it to be evaluated over the entire domain of interest. These expressions are shown in Table 1. In many cases, the terms containing $e^{-\alpha(R-r)}$ will underflow to zero over much of the grain radius. The resulting profile is flat at the centre of the grain and has a thin shell in which the concentration drops to zero. For verification purposes, evaluation of equation (1) or the approximations involving $\sinh(\alpha R)$ or $\sinh(\alpha r)$ for $\alpha R < 1$ or $\alpha r < 1$ in a spreadsheet is not recommended as no spreadsheets that the first author has access to implements sinh(x) correctly for these small arguments. One widely-used commercial spreadsheet does implement sinh(x), but its results are increasing in error for $|x| < 1.0$ as $|x|$ approaches zero.

Reference Data Sets

SOURCE IST 2.0 centralises the location of all of the physical data required by more than one module in Reference Data Sets. Global Reference Data may be treated in one of two ways. A physical constant that is not expected to be re-evaluated frequently is hard-coded in a support subroutine and accessible to the rest of the code via a COMMON BLOCK. Modules that use the COMMON BLOCK are not allowed to modify these constants. Data that may change more frequently (for example, the selection of isotopes that SOURCE IST 2.0 will model) are recorded in a Reference Data File.

Reference Data Files (like the FORTRAN code files) are under revision control. The need to verify the reference data was recognised in designing the data files that contain them and in selecting the data that are recorded. For ease of verification, half-lives, rather than decay constants, are stored in the reference data file. Since half-lives are tabulated in the literature in various units, the value and time units of a half-life are stored in the isotope data file. When the data are read, a FUNCTION takes the half-life and the time unit and calculates the decay constant in reciprocal seconds. This value is stored for internal use by SOURCE IST 2.0. Other calculations that are done when the data are read are the sum of modelled removal cross-sections

for an isotope, and the products of appropriate decay branching ratios and decay constants to provide a decay production constant for a product isotope.

Chronometric Considerations

SOURCE IST 2.0 is being developed for use in the year 2000 and into the next millennium. As such the Year 2000 issue was addressed by design. A Programming Rule that addressed the Y2K question was adopted before there were any Y2K problems with the code. Prior to the OPG 1999 June 30 deadline for Y2K compliance, a previous version of the code was inspected and certified. The latest developmental version of SOURCE IST 2.0 was subjected to the same inspection protocol. No problems were identified with either version. Each site is required to provide a FORTRAN SUBROUTINE with a defined CALL sequence. The CALL sequence is a subset of the FORTRAN-90 CALL sequence for the intrinsic SUBROUTINE DATE_AND_TIME. Thus sites that use a Y2K compliant FORTRAN-90 compiler will not have to provide an additional SUBROUTINE. Y2K compliance of the released code will be certified to the requirements of OPG before the first formal release is declared to be in-service.

Summary

SOURCE IST 2.0 is being developed in accordance with a structured process that incorporates early testing of software components. The early testing and error correction result in less time being taken with integrated testing of the assembled code than would be required if all modules were tested only after integration within the main code. Modular construction allows individual modules or groups of modules to be isolated for thorough testing.

Comparison with analytic solutions or semi-analytic solutions provides a necessary test of the correct functioning of modules. Even when analytic solutions exist, numerical evaluation of the analytic expression may be required.

SOURCE IST 2.0 coding practices control the use of reference data within the code. This control localises the changes required if a fundamental physical constant is re-evaluated, if new data requirements are identified, or if data are no longer needed. It also avoids the problems that may be associated with the presence of slightly different values of a constant being used in different parts of the code.

The existing FORTRAN coding of SOURCE IST 2.0 is year 2000 compliant.

References:

1) A.C Brito, F.C. Iglesias, Y. Liu, M.A. Petrilli, M.J. Richards, R.A. Gibb, and P.J. Reid, "SOURCE 2.0: A Computer Program to Calculate Fission Product Release from Multiple Fuel Elements for Accident Scenarios", Proceedings of the 4th International Conference of CANDU Fuel, 1995 October 1-4.
2) E.O. Moeck, J.C. Luxat, M.A. Petrilli, and P.D. Thomson, "Status of Validation Matrices for CANDU Power Plants", Proceedings of the 18th Canadian Nuclear Society Conference, Toronto, Ontario, 1997 June 8-11.
3) D. Evens and R. Rock, "Verification and Uncertainty Analysis of Fuel Codes Using Distributed Computing", 6th International Conference on CANDU Fuel, 1999 September.
4) A.H. Booth, "A Suggested Method for Calculating the Diffusion of Radioactive Rare Gas Fission Products from UO2 Fuel Elements and a Discussion of Proposed In-Reactor Experiments That May Be Used to Test its Validity", Atomic Energy of Canada Report, AECL-700, 1957.

Table 1

Evaluation of the Grain Concentration Profile

Conditions	$\varepsilon \geq \alpha r$	$\varepsilon < \alpha r < L1$	$\alpha r \geq L1$
$\varepsilon < \alpha R \leq L1$	$\frac{B}{\lambda} \frac{\sinh(\alpha R) - \alpha R}{\sinh(\alpha R)}$	Direct evaluation of Equation (1)	0 for r = R Not applicable for r >R
$L1 < \alpha R < L2$	$\frac{B}{\lambda}\left(1 - 2\alpha R\ e^{-\alpha R}\right)$	$\frac{B}{\lambda} \frac{\left(r - 2R\ e^{-\alpha R} \sinh(\alpha r)\right)}{r}$	$\frac{B}{\lambda} \frac{\left(r - R\ e^{-\alpha(R-r)}\right)}{r}$
$\alpha R \geq L2$		$\frac{B}{\lambda} \frac{\left(r - R\left[e^{-\alpha(R-r)} - e^{-\alpha(R+r)}\right]\right)}{r}$	

ε is the unit round-off, the difference between 1 and the next largest floating-point number.

L1 is the limit on x at which sinh(x) can be evaluated as $e^x/2$, the value of x at which adding e^{-x} to e^x has no effect on the result.

L2 is the limit at which exp(x) overflows.

ε, L1 and L2 are platform-dependant.

$\sinh(x) = (e^x - e^{-x})/2$

Underflow of the exponential function to zero is taken not to generate any numerical exceptions (other than the underflow condition).

Figure 1:
GSWEEP SWEPT FRACTION SCAN:
Pure Equiaxed Growth: square cylinder

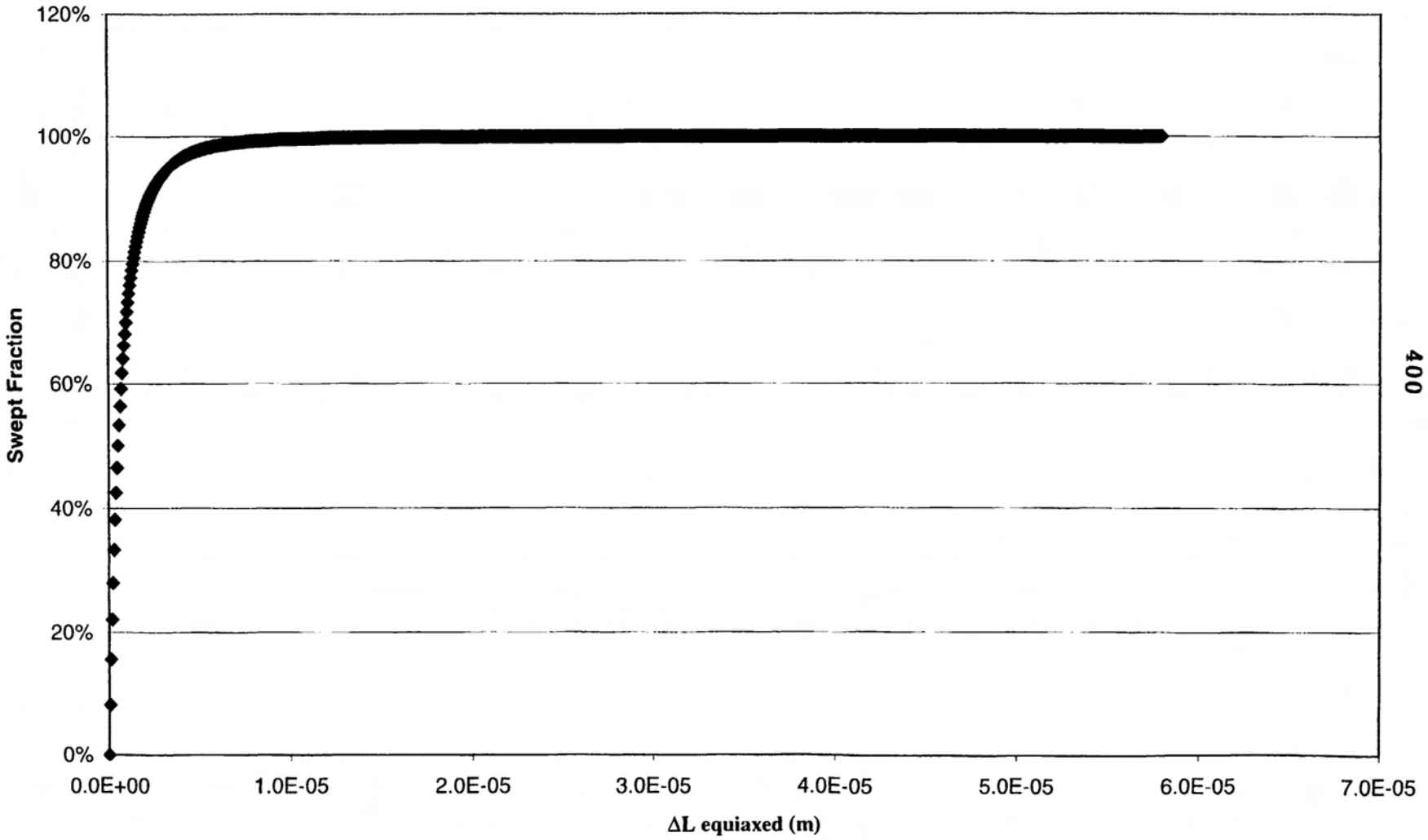

Figure 2:
GSWEEP SWEPT FRACTION SCAN:
Pure Columnar Growth

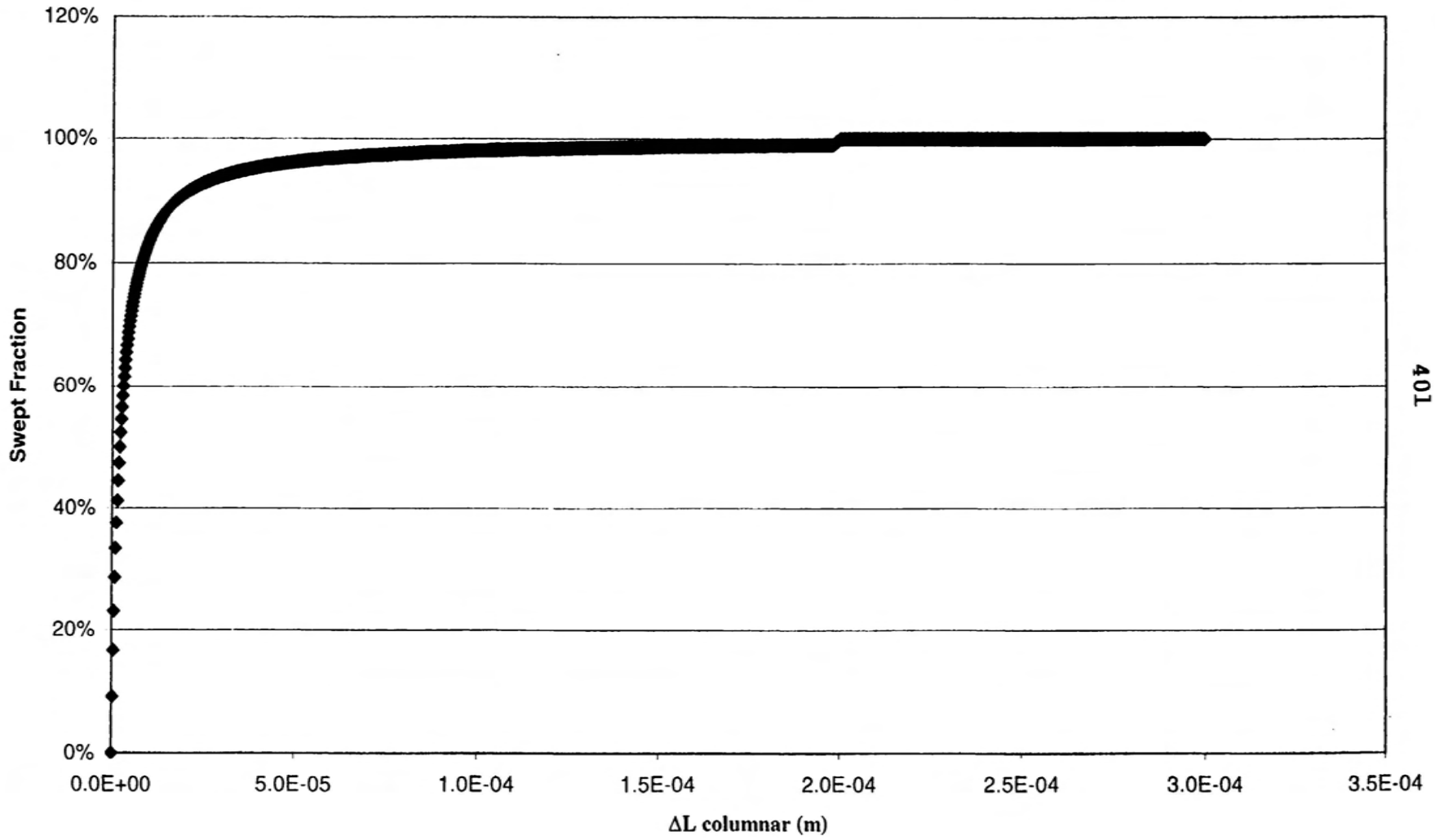

Figure 3:
GSWEEP SWEPT FRACTION SCAN:
Pure Columnar Growth