(5) Harada, H. and Yamashita, K.: The Reactor Core Analysis Code CITATION-1000VP for High Temperature Engineering Test Reactor, JAERI-M 89-135 (1989),(in Japanese).
(6) Fowler, T. B., Vondy, D. R., Cunningham, G. W.: Nuclear reactor core analysis code; CITATION, ORNL-TM-2496 (1971).
(7) Yasuda, H., et al.: Construction of VHTRC(Very High Temperature Reactor Critical assembly), JAERI-1305 (1986),(in Japanese).
(8) Akino, F., et al.: Critical Experiment on Initial Loading Core of Very High Temperature Reactor Critical Assembly(VHTRC), J. Atomic Energy Society, 31 p.p.682-690 (1989),(in Japanese).

AUTOMATED DIFFERENTIATION OF COMPUTER MODELS FOR SENSITIVITY ANALYSIS

B.A. WORLEY
Oak Ridge National Laboratory,
Oak Ridge, Tennessee,
United States of America

Abstract

Sensitivity analysis of reactor physics computer models is an established discipline after more than twenty years of active development of generalized perturbations theory based on direct and adjoint methods. Many reactor physics models have been enhanced to solve for sensitivities of model results to model data. The calculated sensitivities are usually normalized first derivatives, although some codes are capable of solving for higher-order sensitivities. The purpose of this paper is to report on the development and application of the GRESS system for automating the implementation of the direct and adjoint techniques into existing FORTRAN computer codes. The GRESS system was developed at ORNL to eliminate the costly man-power intensive effort required to implement the direct and adjoint techniques into already-existing FORTRAN codes. GRESS has been successfully tested for a number of codes over a wide range of applications and presently operates on VAX machines under both VMS and UNIX operating systems.

I. INTRODUCTION

Sensitivity analysis is an important component of any computer code application for modeling physical systems. The role of sensitivity analysis is to provide a quantitative measure of the effect of computer code data and inputs upon key performance indices. Sensitivity analysis also helps limit the scope of the more complicated problem of quantifying uncertainties.

Sensitivity analysis of computer-generated results consists of determining the effect of model data upon the calculated results of interest. Because computer model equations can be differentiated analytically, sensitivities can be precisely defined and calculated in a deterministic fashion using both direct and adjoint methods.[1-8] The deterministic approach is particularly suited to large-scale problems for which direct perturbation of the model data becomes impractical from a cost standpoint. The main drawback to the deterministic approach has been the initial manpower investment to add the computational capability for calculating the necessary derivatives into existing computer models.

This paper presents the theory and application of the Gradient-Enhanced Software System, GRESS,[9] and its role in calculating model derivatives and sensitivities without a prohibitive initial manpower investment. Storage and computational requirements are discussed.

II. DETERMINISTIC SENSITIVITY ANALYSIS

A brief description of general sensitivity theory is given here as an aid to understanding the problem of applying this theory to computer models. The example to be discussed will be that of a general set of nonlinear equations given by

$$y = F(y, c) , \tag{1}$$

where y represents the dependent variables, c represents the user-specified model data or parameter set, and F defines the model equations. The particular form chosen in Eq. (1) is one that can be used to represent equations coded in the FORTRAN programming language. The left side of the equation can represent the stored value of the variable calculated from the functional formula on the right side.

Since the number of components of the vector y calculated in any typical large-scale modeling problem is large, it is useful to define a generic result for such a calculation that is of particular interest to the model user. Typically many results will be needed for analysis but in most cases they form a much smaller set than the elements of y. A typical result will be defined as

$$R = h(y) , \tag{2}$$

where R is a single number that is a function of the solution to Eq. (1). For notational ease, the generic parameter α_i will be used to denote an individual element of c. The total number of parameters in the problem will be assumed to be M so that the index on α_i will run from 1 to M.

The basic problem in any sensitivity study is to find the rate of change in the result R arising from changes in any model parameters. For the generic parameter α_i, then, the quantity of interest is the numerical value of $dR/d\alpha_i$ given analytically by

$$\frac{dR}{d\alpha_i} = \frac{\partial h}{\partial y}\frac{dy}{d\alpha_i} . \tag{3}$$

Since the functional dependence of R on y through $h(y)$ is defined analytically by the model user, only $dy/d\alpha_i$ needs to be generated in order to evaluate Eq. (3). The procedure needed to get $dy/d\alpha_i$ is to differentiate Eq. (1) as follows:

$$\frac{dh}{d\alpha_i} = \frac{\partial F}{\partial y}\frac{dy}{d\alpha_i} + \frac{\partial F}{\partial c}\frac{dc}{d\alpha_i} . \tag{4}$$

Rearranging Eq. (4) yields the following set of coupled equations to solve for $dy/d\alpha_i$,

$$\left(I - \frac{\partial F}{\partial y}\right)\frac{dy}{d\alpha_i} = \frac{\partial F}{\partial c}\frac{dc}{d\alpha_i} , \tag{5}$$

or in more compact form,

$$Ay_i' = s_i , i = 1, \ldots, M , \tag{6}$$

where I is the identity matrix and A, y_i', and s_i are given by

$$A = 1 - \frac{\partial F}{\partial y} , \tag{7}$$

$$y_i' = \frac{dy}{d\alpha_i} , \tag{8}$$

and

$$s_i = \frac{\partial F}{\partial c}\frac{dc}{d\alpha_i} . \tag{9}$$

If Eq. (6) were solved directly for y_i', the result could be used in Eq. (3) to evaluate $dR/d\alpha_i$. This method of sensitivity analysis is called the "direct" approach and is a classical methodology that has received a great deal of attention in the literature.[1,5] Since Eq. (6) must be solved each time a new α_i is defined, the direct approach is most suitable for problems with relatively few input parameters of interest, for problems in which the solution of Eq. (6) is very inexpensive compared to the solution of the model itself, or for analytical problems in which the inverse of A can be explicitly determined.

For large-scale models with a large database in which the ultimate objective is still the evaluation of $dR/d\alpha_i$ for many α_i, the intermediary step of solving for $dy/d\alpha_i$ and its inherent computational inefficiency can be avoided. For such problems, the "adjoint" approach is far more applicable. In this methodology, use is made of the fact that Eq. (6) is linear in y_i, and appropriate adjoint equations can therefore be developed specifically to evaluate Eq. (3).

Defining the matrix adjoint of A as A^* and using the usual definition of this adjoint give the identity,

$$u^{tr}Av = v^{tr}A^*u , \tag{10}$$

where u and v are arbitrary vectors and A^* is defined as

$$A^* = A^{tr} . \tag{11}$$

Here the tr superscript represents the transpose of the vector or matrix.

If specific vectors for the problem at hand are chosen for u and v, the problem-specific adjoint equation can be set up as follows:

$$A^* y^* = s^* , \tag{12}$$

where

$$A^* = A^{tr} = \left(I - \frac{\partial F}{\partial y} \right)^{tr} . \tag{13}$$

Choosing s^* as

$$s^* = (dh/dy)^{tr} , \tag{14}$$

Equation (3) can now be evaluated as follows:

$$\frac{dR}{d\alpha_i} = y^{*tr} \frac{\partial F}{\partial c} \frac{dc}{d\alpha_i} , i = 1, \ldots, M , \tag{15}$$

where y^* is now the solution to

$$\left(I - \frac{\partial F}{\partial y} \right)^{tr} y^* = \left(\frac{dh}{dy} \right)^{tr} . \tag{16}$$

The simplicity of the adjoint approach lies in the fact that Eq. (16) needs to be solved only once to get any and all sensitivities in the problem. This is a result of Eq. (16) being independent of the definition of α_i. The particular choice of α_i is only reflected in the evaluation of Eq. (15), which involves simple vector products. In essence, the adjoint approach reduces the computational effort needed to evaluate $dR/d\alpha_i$ from solving many coupled linear equations to the evaluation of several vector products. For large-scale systems with many thousands or even millions of parameters, this represents orders of magnitude in computational efficiency.

It should be noted here that both the direct and adjoint equations [i.e., Eqs. (6) and (16)] are in any case far easier to solve than the original model [Eq. (1)]. Both Eqs. (6) and (16) are linear while Eq. (1) is nonlinear. The direct and adjoint approaches, however, require the results of the original model equations to be available in order to set up Eqs. (6) and (16), since the A matrix and the vectors s, and s^* depend on y.

In order to solve either the direct or adjoint sensitivity analysis, then, the model user must first generate the matrices $\partial F/\partial y$ and $\partial F/\partial c$ from the original nonlinear computer model. For large-scale problems, this generally requires a great deal of painstaking human effort. First, the model equations must be extracted from the computer coding. They must then be differentiated with respect to all parameters of interest, and finally direct or adjoint sets of equations must be set up for computational solution. Successful automation of this procedure greatly reduces the human effort involved, potentially by orders of magnitude. The advantages of automation of sensitivity model development is, therefore, great indeed. The next two sections discuss the GRESS automated system that uses the rules of calculus to add capability to existing FORTRAN computer models for solving the direct or adjoint equations.

III. GRESS CHAIN OPTION
(An Automated System for Solving the Direct Sensitivity Problem)

The basic principle of GRESS is to read the model source program and search for model equations. These are identified uniquely by the appearance in the FORTRAN source program of the "=" symbol. Since all FORTRAN "equations" so identified occur in the form of Eq. (1) (i.e., with a single dependent variable on the left side of such an expression), GRESS can search for and analyze each equation in terms of its functional dependence on y and c. The basic computer calculus operations of the GRESS CHAIN option is then used to compute the successive elements of $\partial F/\partial c$ and $\partial F/\partial y$ as each expression is encountered. The differentiation is carried out analytically using calculus software for all permissible FORTRAN functions and operators and the results are computed and stored numerically using the local (current) values of the independent and dependent variables. The CHAIN option takes advantage of the fact that in solving Eq. (5), the matrix $(I - \partial F/\partial y)$ is lower triangular and the y vector can be computed by forward substitution. The important point is that the components of y are solved successively as each equation is differentiated and that the $(I - \partial F/\partial y)$ matrix does not have to be stored. (The adjoint problem requires the storage of this matrix, as will be discussed in the next section.)

GRESS only recognizes real-variable store operations as valid equations (i.e., the left side variable in a FORTRAN equation must be real), since continuous derivatives are to be calculated. Also, the left-hand side of an equation is treated as a separate component of y each time it is executed (including each execution in a DO LOOP). The calculation of $\partial F/\partial y$ and $\partial F/\partial c$ in effect means that GRESS can be used to calculate the derivative of any real variable in the model with respect to any other real variable in the model. All derivatives are available for both internal and/or external use.

The application of GRESS to an existing FORTRAN model consists of an automated precompilation in which the automated code translation necessary to compute derivatives is performed using computer calculus. This step consists primarily of a rearrangement of the program data structure and a substitution of calls to GRESS interpretive software in place of all arithmetic lines of coding. All arithmetic operations of the original model are precompiled into a pseudomachine code (the GRESS P-code) for use during program execution. The two output files of this step are the enhanced model and the binary P-code file. These two files and a set of GRESS software subroutines supporting the enhanced model are compiled and run as a normal FORTRAN program to produce both the reference model results and gradient information. The gradients and reference results are used to calculate the sensitivities.

IV. GRESS ADGEN OPTION
(An Automated System for Solving the Adjoint Sensitivity Problem)

The adjoint problem is defined by Eqs. (12-16). As previously mentioned, the calculation of the adjoint solution vector y^* from Eq. (16) is not a function of the selection of input parameter α_i and thus need only be performed once to determine the derivatives of a response of interest with respect to any parameter of interest. The matrix $\partial F/\partial c$ must also be determined, but it, too, is independent of the parameter of interest. The only parameter-dependent operation required to calculate the derivative $dR/d\alpha_i$ is the simple

matrix multiplication operation $(y^{*tr})(\partial F/\partial c)(dc/d\alpha_i)$ in which the vector $dc/d\alpha_i$ is a function of α_i. The option in GRESS to automate the calculation of derivatives based upon the solution of the adjoint equations is referred to as the ADGEN (ADjoint GENerator) option. Recall that GRESS solves Eq. (5), taking advantage of the fact that the matrix $(I - \partial F/\partial y)$ is lower triangular and the solution by forward substitution requires only that the vector $dy/d\alpha$ be stored. However, to solve the adjoint problem, all derivatives that constitute the n×n matrix $(I - \partial F/\partial y)^{tr}$ must be stored, where n = total number of equations, counting each time an equation is solved in a DO LOOP as a separate equation; the left-hand side of each equation in a DO LOOP is treated as a separate element of y. Although only the non-zero elements are saved, the storage of the matrix $(I - \partial F/\partial y)^{tr}$ may require a substantial amount of storage capability. The storage difficulties are counterbalanced by features of Eqs. (15) and (16) that make the ADGEN calculation of y^* both practical and cost efficient. Note that the matrix $(I - \partial F/\partial y)^{tr}$ is upper triangular and that the column vector $(dh/dy)^{tr}$ is a simple user-defined vector (for most cases a vector with a single non-zero entry of unity). Thus Eq. (16) is easily solved by back substitution and the values of y^* can be successively stored in the space allocated for the $(dh/dy)^{tr}$ vector. The calculation of $dR/d\alpha_i$ from Eq. (15) must be performed for each α_i, but this requires only trivial matrix multiplications and very little computer cost.

The ADGEN option calculates the normal model results as well as the derivatives making up the $\partial F/\partial y$ and $\partial F/\partial c$ matrices. Again, the major difference from the direct approach using the CHAIN option is that the ADGEN option requires that the matrix $(I - \partial F/\partial y)^{tr}$ be stored and includes a post-processor solver routine to calculate the adjoint solution.

V. APPLICATIONS OF GRESS

The distinguishing feature in solving Eq. (5) using the GRESS CHAIN option is that only the elements of a single row of $\partial F/\partial y$ and $\partial F/\partial \alpha_i [= (\partial F/\partial c)(\partial c/\partial \alpha_i)]$ need be saved in computer memory at any one time. This advantage is important if one wishes to solve for the derivatives of many LHS elements (responses) with respect to a data element α_i. The disadvantage is that to calculate derivatives with respect to other data elements, Eq. (5) must be solved for each additional α_i of interest. The computational burden is approximately proportional to the number of α_i. In our experience to date, the computational time for calculating derivatives with respect to m chosen elements of c, denoted by T_m, is

$$T_m \approx T_{REF}(\beta_0 + \beta_1 m)$$

where T_{REF} is the execution time of the reference model before derivative enhancement and β_0 and β_1 are constants falling between 1.0 and 30.0 and 0.1 and 1.4 respectively.

Another problem sometimes occurring in practice is that the elements of $dy/d\alpha_i$ must be stored in memory as Eq. (5) is solved for each row. Therefore, the number of α_i with respect to which derivatives are calculated in a single execution of the enhanced model may be limited by system memory resources using the CHAIN option.

As mentioned in Section IV, solution of the adjoint equations, Eqs. (15-16), using the ADGEN option reduces the computation effort for calculating derivatives of a single response with respect to many parameters compared to repeatedly solving Eq. (5) for each α_i, as is done in the CHAIN option. The solution to Eq. (16) is straightforward due to the upper triangular structure of $(I - \partial F/\partial y)^{tr}$, but requires storage of the nonzero elements of $(I - \partial F/\partial y)^{tr}$. ADGEN circumvents the necessity to store this matrix in memory by using an efficient scheme for solving Eqs. (15-16) based upon retrieval of portions of $(I - \partial F/\partial y)^{tr}$ from off-line storage and segmenting the calculation of derivatives. A method to reduce the size of the adjoint matrix to be stored by explicitely recognizing the input parameters of interest has been developed and implemented (called FORWARD REDUCTION[9]). Storage size of the adjoint matrix can then be further reduced based on the output parameters of interest (called BACK REDUCTION)[9].

The computer models that have been enhanced using GRESS are listed in Table 1. The first column of the table contains the code name. The second column provides a brief description of the system being modelled. For some of the models the approximate number of lines of coding (excluding comment cards) are given in parenthesis. The next two columns indicate the year that the code was enhanced and the version of GRESS used. The next, fifth, column indicates whether the direct (C=CHAIN) or adjoint (A=ADGEN) option was used. The sixth and seventh columns give the number of input and output parameters for which sensitivities were printed. Note that in the CHAIN option, derivatives are calculated for all dependent variables with respect to user-selected input parameters. In the ADGEN option, derivations of user-selected responses (results) are calculated with respect to all variables. The purpose of showing the numbers of input and output variables is to provide an idea of the range of input and output variables of interest in typical sensitivity analysis applications. The eighth column gives an approximate measure of the increase in execution time and storage requirements for the GRESS-enhanced version of each code. Except where noted, the factors represent the ratio of the enhanced-code execution time (or memory storage requirement) to the reference code execution time (or memory storage requirement). For the ADGEN options, the storage requirement of the adjoint matrix that must be provided for (either in memory or in off-line storage) is shown . The last column gives an abbreviated reference of the publication describing the application of GRESS for each computer code and the primary author.

A new version of GRESS (Version 2.0) is under development that replaces the P-code by using symbolic differentiation. This version will have the capability to enhance selected subroutines. For the PRESTO-II and AIRDOSE models, a comparison of the figures in column seven of Table 1 reveals that the successive versions of GRESS have substantially reduced the execution and storage requirements.

Table 1. Computer Codes Enhanced Using GRESS

Code[a]	Description[b]	Date of GRESS Enhancement	GRESS Version[c]	GRESS Option[d]	Input Parameters[e]	Output Parameters[e]	Run-Time Cost Factor [Storage][f]	Reference[g]
SWENT	3-D Finite Difference Transient Geohydrology Model (15,000)	1985	D	C	7	900	30 [3X]	NSE,94,1986 Oblow
ORIGEN2	Radioactive Decay Model	1985	D	C	7	140,000	25	NSE,94,1986 Worley
UCBNE10.2 (NEUCB)	Migration of Radioactive Nuclides in Ground Model	1985	D	C				Nucl. Chem. Waste Man. – Fin
TEMP (BRINETEMP)	Finite-Line Heat Transfer Model	1986	D	C	10	140		ORNL/TM-9975 Worley
WAPPA-C (WAPPA-B)	Waste Package Performance Assessment Model	1986	D	C				ORNL/TM-9976 Worley
CRESS	Commercial and Residential Energy Use and Emissions Model	1986	D	C	10	800	3	ORNL/TM-10304 Trowbridge
CFEST	3-D Transient Finite Element Hydrology Model (16,000)	1988	0.0	C(A)	38	20,000	23(73)	Letter Report Horwedel
VSL3DNQ	Chemical Reactions in High-Speed Flows (12,000)	1988	0.0	C	6	2		Letter Report Wright
PRESTO-II	Shallow-Land Disposal of Radioactive Waste Model (6,900)	1988 (to present)	0.0,1.0,2.0	A(C)	69,000	2	52,33,10 [144,88,11]	Various Conf. Papers Worley, Horwedel

Table 1. Computer Codes Enhanced Using GRESS (Cont'd.)

Code	Description	Date of GRESS Enhancement	GRESS Version	GRESS Option	Input Parameters	Output Parameters	Run Time Cost Factor [Storage]	Reference
AIRDOS-EPA	Radionuclide Transport in Air and Human Intake Model	1989 (to present)	0.0,1.0,2.0	A(C)	1,271	2	10,29.6 [5.71 .01]	ORNL/TM-11373 Horwedel
DO	Dissolved Oxygen at Dam Sites Model	1989	0.0	C	14	27,700		ORNL/TM-10953 Railsback
D2PC	Chemical Hazard Prediction Model	1989	0.0	C	3	9		Letter Report Worley
E9E	Tumor Growth Model	1989	1.0	C(A)				
EQ3	Geochemical Model	1990	1.0,2.0	A(C)	31,000	9	27	ORNL/TM-11407 Horwedel
HELP	Hydrological Evaluation of Landfill Performance Model	1990	1.0,2.0	A(C)				Letter Report Horwedel
DRUFAN	Fluid Flow Dynamics for Nuclear Safety	1990	1.0,2.0	A(C)				Letter Report Horwedel

[a] Code names shown in parenthesis are similar codes also enhanced using GRESS.
[b] Number in parenthesis indicates approximate number of lines of source codes, excluding comments.
[c] The current version being distributed is Version 1.0. Version 2.0 is under development.
[d] Indication of the method used for sensitivity analysis for the particular code application of interest. C refers to the GRESS CHAIN option (direct method) and A refers to the GRESS ADGEN option (adjoint method). The option in parenthesis indicates the GRESS option used for derivation verification and alternate sensitivity calculations.
[e] Refers to the parameters of interest for a particular application of the code. In the GRESS CHAIN option, derivatives of *all* variables with respect to a single parameter are calculated. In the GRESS ADGEN option, derivatives of a single response with respect to *all* variables are calculated.
[f] The cost factor is the ratio of the execution time of the GRESS-enhanced code to the execution time of the reference code. For the CHAIN option the ratio is for the calculation of derivatives with respect to one input parameter. Number in parenthesis refers to the cost factor for the total number of input parameters of interest. Numbers in brackets followed by an x (i.e., $3x$) refer to the factor of storage required as a multiple of the reference code storage for the CHAIN option. Numbers in brackets without an x refer to the adjoint matrix storage requirements in Megabytes for the ADGEN option.
[g] Brief reference of report describing the application of GRESS to the subject code and sensitivity analysis results of interest. First author's name of publication is shown.

VI. CONCLUSIONS

An approach to sensitivity analysis of large-scale computer models based on an automated system for implementing the direct and adjoint method is now available. GRESS calculates model derivatives and sensitivities and has been successfully applied to many large-scale computer models. The availability of GRESS greatly reduces the man-effort required to add sensitivity capability to existing FORTRAN models.

REFERENCES

1. R. Tomovic and M. Vukobratovic, *General Sensitivity Theory*, Elsevier North-Holland, Inc., New York (1972).
2. W. M. Stacey, *Variational Methods in Nuclear Reactor Physics*, Academic Press, New York (1974).
3. E. Greenspan, "Developments in Perturbation Theory," *Advances in Nuclear Science and Technology*, Vol. 9, Academic Press, New York (1976).
4. E. M. Oblow, "Sensitivity Theory for Reactor Thermal-Hydraulics Problems," *Nucl. Sci. Eng.*, 68, 322 (1978).
5. P. M. Frank, *Introduction to System Sensitivity Theory*, Academic Press, New York (1978).
6. D. G. Cacuci, *J. Math. Phys.* (NY), 22, 2794 (1981).
7. D. G. Cacuci, *J. Math. Phys.* (NY), 22, 2803 (1981).
8. C. R. Weisbin, et al., "Sensitivity and Uncertainty Analysis of Reactor Performance Parameters," *Advances in Nuclear Science and Technology*, 14, Plenum Press, New York (1982).
9. "GRESS 1.0, Gradient Enhanced Software System," distributed by Radiation Shielding Information Center, PSR-231.

IMPACT OF DIFFERENT LIBRARIES ON THE PERFORMANCE CALCULATION OF A MODUL-TYPE PEBBLE BED HTR

U. OHLIG, H. BROCKMANN, K.A. HAAS, E. TEUCHERT
Institut für Sicherheitsforschung und Reaktortechnik,
Forschungszentrum Jülich GmbH,
Jülich, Federal Republic of Germany

Abstract

A new multigroup library for the GAM-THERMOS spectrum codes has been compiled from the sources ENDF/B-V and JEF-1. The progress in comparison to the 20 years old standard library has been studied for one specific reactor design of the Modular High Temperature Reactor. The study covers various aspects of the performance of the reactor both for the initial core and for the equilibrium cycle.

For the multiplication factor k_{eff} the difference amounts to $\Delta k_{eff} = 0.0164$ in the startup reactor, which is mainly due to changes in the cross sections of ^{235}U. At the turn to the equilibrium cycle the difference reduces to $\Delta k_{eff} = 0.0017$ as due to various opposite tendencies in the data of the many involved nuclides. The change in the mass balance of the fissile materials is about 5%. The impact on the temperature coefficients is in the order of 4%, and the influence on other safety related properties of the reactor is lower than about 1 or 2 percent, confirming the confidence in formerly received results.

1. Introduction

In 1989 a new multigroup library has been compiled from the ENDF/B-V [1] and JEF-1 [2] data files. It is intended for the VSOP computer code system [3] which is based on the ZUT-GAM-THERMOS spectrum codes [4,5,6].

This VSOP code system is applied to the physics calculation of a reactor from its startup to the equilibrium cycle, to control and safety assessment and to thermal evaluations under operational and accidental situations. Spectrum calculation is performed for many subregions of the reactor and can repeatedly be applied when time proceeds. Neutron diffusion calculation and thermal hydraulics is performed 2-dimensionally in r-z-geometry.

Up to now an older nuclear library has been used which is based on the ENDF/B-II data files [7]. It has been supplemented with data of various other sources as available about the year 1970. Calculations agreed well with existing reactors, therefore the introduction of a new library is considered to be a kind of venture.