A NEURAL NETWORK MODEL OF VOLATILE FISSION PRODUCT RELEASE FROM FUEL ELEMENTS AND FRAGMENTS UNDER SEVERE ACCIDENT CONDITIONS

W.S. Andrews and B.J. Lewis, Department of Chemistry and Chemical Engineering,
Royal Military College of Canada, Kingston, Ontario Canada K7K 5L0

D.S. Cox, Atomic Energy of Canada Limited - Research Company,
Chalk River, Ontario, Canada K0J 1J0

INTRODUCTION

The accidents at Three Mile Island and Chernobyl have underscored the need to define nuclear power reactor source terms for fission products released during severe accident conditions. In the United States, tests have been conducted involving the heating, or annealing, of fuel fragments and short segments of Light Water Reactor (LWR) fuel rods under varying environmental conditions, such as steam and hydrogen[1]. Analysis of early annealing experiments performed at the Oak Ridge National Laboratory (ORNL) has provided a correlation of the cumulative release of volatile fission products with temperature and time in a steam environment. This correlation, called CORSOR-M, is used by the United States Nuclear Regulatory Commission for LWR source term prediction[2]. Its applicability to CANDU pressurized heavy water reactor (PHWR) fuel has yet to be established. Further, it only considers two variables (temperature and time) in one environment (steam).

Corresponding annealing experiments have been conducted at the Chalk River Laboratories (CRL) on fuel fragments and mini-elements of CANDU fuel. In fact, the Hot Cell Experiments 1 and 2 (HCE-1 and HCE-2)[3,4] and the Metallurgical Cell Experiment (MCE-1)[5] include a total of 37 separate tests conducted under a wide variety of sample size, conditions and environments. The resulting large data base has yet to be analysed to the point that an overall comprehensive model could predict experimental results with any degree of confidence.

The physical mechanisms involved in the release of volatile fission products under severe accident conditions are felt to be extremely complex. In the U.S., the FASTGRASS code has been developed to model such phenomena as fission gas bubble nucleation, migration, interlinking and resolution[6], while in Canada, FREEDOM is a mechanistic model for CANDU fuel[7]. Both of these codes, however, are computation-intensive and thus do not run in real time. As a result, work continues on simpler, semi-empirical models which are based on the controlling physical phenomena. Examples of this are recent work by Osborne and Lorenz at ORNL[8] and by Lewis et al. in Canada[9]. This notwithstanding, the comprehensive model of fission product release considering the full spectrum of conditions of the annealing tests and able to run in real time has not appeared yet. Consequently, this paper will outline a novel approach to empirically model the results of 29 CRL tests with the use of an artificial neural network (ANN) in order to predict the cumulative fission product release of volatile fission products (specifically Cs-134).

ARTIFICIAL NEURAL NETWORKS

Neural network development began in the mid 1940's, but went into a hiatus due to a lack of discernable applications and sufficiently powerful computers. The most widely-used paradigm, back propagation, emerged in the mid 1980's from the work of two psychologists, McClelland and Rumelhart, in their efforts to model the functioning of the brain[10]. Neural networks are composed of simple nodes (neurons) which take inputs, sum them, perform a simple mathematical operation on this sum via a transfer function and pass the result on to other nodes. Before the output arrives at a subsequent node, however, a weight is applied to it. As a consequence, nodes receive outputs of preceeding nodes which have been modified by these weights.

Current practice is to use three layers of neurons in back propagation networks. As can be seen in Figure 1, these layers are usually named input (X), hidden (Y) and output (Z). The input layer contains the values of the variables and parameters considered for correlation. In this application, each node would correspond to a value from the list in Table 1. These inputs are normalized, either to the highest value of the particular variable in the input space, or to some other conveniently high value. It is also customary in back propagation networks to connect each input node to each node in the hidden layer. It is also usual to connect a bias node to all hidden and output nodes. This bias node has a set value of unity (1). It offsets the origin of the transfer function and tends to cause the network to converge more quickly.

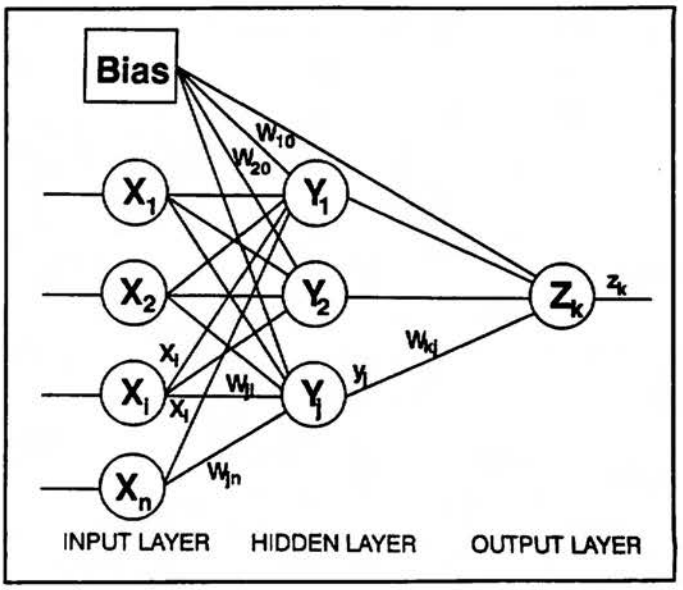

Figure 1. Architecture of a Back Propagation Artificial Neural Network

Table 1. Variables and parameters for input layer used to train neural net model.

	VARIABLE or PARAMETER
1	Time sample is above 1000°C (s)
2	Temperature (°C)
3	Time at Temperature (s)
4	Time in steam (s)
5	Time in air (s)
6	Rate of temperature change (°C)
7	Zircaloy weight (g)
8	Cladding open (y or n)
9	Steam flow rate (mL/min)
10	Air flow rate (mL/min)
11	Inert gas flow rate (mL/min)
12	Peak linear power of fuel in reactor (kW/m)
13	Sample burnup (MWh/kgU)
14	Bare fuel weight (g)

As stated above, each connection to a hidden layer node contains the normalized input value leaving the input layer or bias node, which has a weight applied to it. Thus, the i^{th} input value, x_i, connected to the j^{th} hidden node, has a weight, w_{ji}, applied to it, so that the hidden node receives as an input from the i^{th} node, the value $w_{ji}x_i$. Also, as noted above, each node receiving inputs sums these inputs. Consequently, in Figure 2, I_j is the sum of i inputs, each multiplied by its own connection weight, so that

$$I_j = \sum w_{j1} x_1$$

When optimized, this is the equation of a linear regression, with the intercept being the weight associated with the bias node, w_{jo}.

Non-linearity is introduced into the model by the transfer function, which is applied to the summed inputs, I_j. Several different functions are available (sinusoidal, sigmoidal), but the one found particularly useful in this application is the hyperbolic tangent.

The application of the tanh transfer function yields an output y_j from the j^{th} hidden node, such that

$$y_j = \frac{e^{I_j} - e^{-I_j}}{e^{I_j} + e^{-I_j}}$$

These transfer functions are continuous and continuously differentiable, with the derivative being expressed in terms of the function itself, as shown below for the tanh function:

$$y'_j = (1 + y_j)\,(1 - y_j)$$

In fact, the tanh is the smooth version of the step function from -1 to +1.

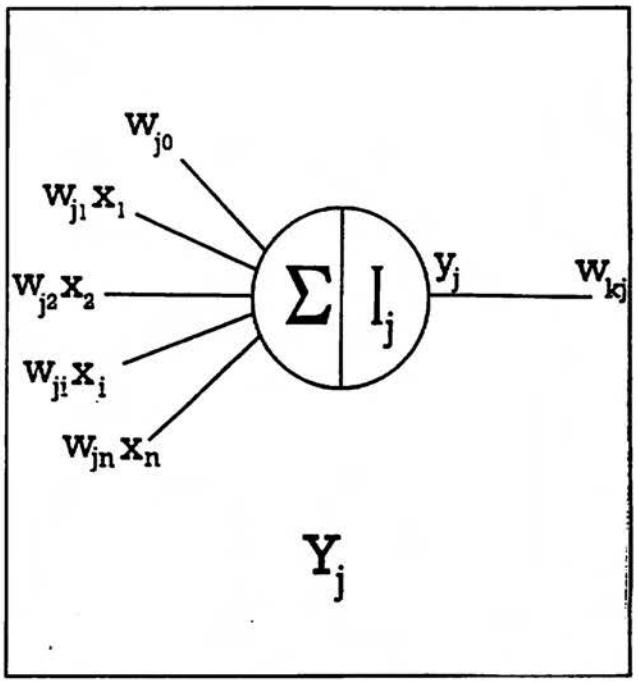

Figure 2. Architecture of the j^{th} neuron in the Hidden Layer

This same process is repeated between the hidden layer and the output layer, with the transfer function again applied to the summed inputs to produce the output. When the network is initialized, the values of the weights are randomly assigned. The "knowledge" or "artificial intelligence" within the network, however, resides in the distribution of these weights which must be adjusted to be able to produce an output as close as possible to the desired output. This process of adjusting the weights is called supervised learning and is conducted during the training phase of the network development.

To effect training, the network is not only presented with a full array of inputs, but also with known (measured) outputs for each input set. When the initial pass through the network for a given set of data is complete, an output, z_k, is determined. This value is then compared to the desired output value, d_k, to arrive at a global error E:

$$E = \frac{1}{2}\,(d_k - z_k)^2$$

The global error is then propagated backwards through the network to adjust the individual connecting weights. This back and forth iterative process is continued until the global error is minimized. At this point, another set of input data (vector) is introduced into the network and the process is repeated. Generally, the connecting weights are adjusted after an "epoch" of up to about 500 different inputs (this can be adjusted to facilitate learning, but updates are rarely done after each input set, in order to prevent oscillations in weight values).

Back propagation networks usually use modifications of the "delta" learning rule during training. When the global error is propagated backwards, the local error at the output from each hidden node, e_j, is determined by

$$e_j = -\frac{dE}{dI_j}$$

which can also be expressed as

$$e_j = (1+y_j)\,(1-y_j)\sum_k e_k w_{kj}$$

where, in this specific application, with one output node, k=1. The corresponding expression for the local error at the output, e_k, is

$$e_k = (1+z_k)\,(1-z_k)\,(d_k - z_k)$$

The adjustment to each connecting weight, Δw_{ji}, needed to minimize the local error (and hence the global error) is determined by the gradient descent rule. Thus

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}}$$

where η is the learning coefficient and usually has values between 0.1 and 0.3.

This learning rule can be modified by adding a momentum term to take advantage of the previous adjustment $(\Delta w_{ji})_{old}$

$$(\Delta w_{ji})_{new} = \eta e_j x_i + \alpha\,(\Delta w_{ji})_{old}$$

where α is often referred to as the momentum coefficient, with $\alpha < 1$ and is often set as $\alpha = (1 - \eta)$.

For training, input vectors should be used in random order, and may have to be used a number of times. Training is considered complete when the global error has stabilized and been minimized. At this point, the network is considered to have converged onto the optimal or best solution possible.

Once training is complete, the network is tested against data which had not been seen during training. Like the training set, the test set should represent, to the greatest extent possible, the whole range of the input space. Also, like the training set, the test set must have known output values available for comparison with the network predicted values. Finally, the network should be validated by predicting results for a data set representative of a likely application.

Essentially, then, a trained neural network is an n-dimensional correlation, and provides a result similar to a non-linear regression. The link with the physical phenomena it is modelling is through the choice of variables or parameters. Any relationship among these inputs is established by the learning rule itself, and not by any real or postulated physical relationships.

Neural networks have a number of advantages over other types of models or correlations:

1. With the "knowledge" or "intelligence" distributed throughout the network, a reasonable response is possible when the input space contains incomplete, noisy or previously unseen values.

2. A careful analysis of the weights throughout the network permits the various parameters or variables in the input space to be ranked in order of influence on the output.

3. A trained neural network model operates in real time, making it suitable for being embedded in much more complex computer codes, such as modelling the progression of a severe reactor accident.

EXPERIMENTAL

The data base used to construct the model comprised 9 tests of HCE-1, 12 of HCE-2 and 8 of MCE-1, with the tests conducted between 1350 and 2100°C and in steam, air or argon/hydrogen atmospheres. Each annealing test involved placing a fragment or mini-element in an induction furnace, as shown in Figure 3, and introducing the appropriate environment (steam, air or argon/hydrogen) into the furnace. The release of fission products from the sample is determined by measuring the change in fission product activity by using gamma ray spectrometry. As well as environment, other factors varied included temperature, time at temperature, heating ramp rate, sample size, amount of zircaloy cladding and sample burnup. The fission products measured included cesium (Cs-134 and Cs-137), ruthenium (Ru-103) and iodine (I-131), although the model reported on in this paper was developed for Cs-134, as the cumulative release values showed minimal randomness and the cesium behaviour was felt to be representative of volatile fission products in general.

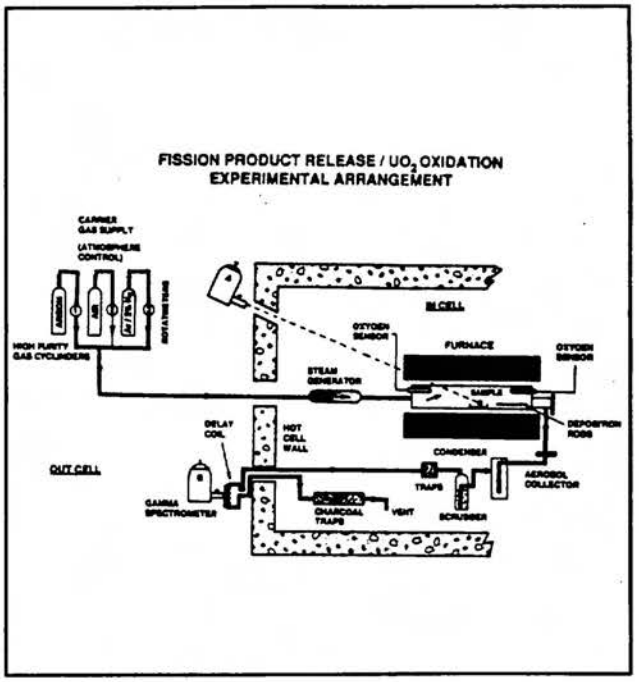

Figure 3. Schematic of equipment used in HCE-2.

ANALYSIS

The neural net used to model the CRL tests was based on *NeuralWorks Professional II/Plus* by NeuralWare and featured 14 different inputs, as listed in Table 1. Related to Figure 1, the input space would extend from x_1 to x_{14}. A single hidden layer was used with differing numbers of nodes (from 2 to 15). All the architectures returned comparable results, except the networks with only 2 hidden nodes, which provided poor correlations. The output layer contained a single node and represented the cumulative fractional release of Cs-134.

Much of the effort needed to train a neural network must be invested in creating the data base to provide the input vectors. Each test contained values for temperature and cumulative fractional release measured at intervals of 100 to 300 s. Most tests exhibited a characteristic response of an initial plateau on the Time/Fractional Release curve (Figure 4) displaying an initial low release rate, a fairly steep climb due to an increased release rate as the grain boundary fission product inventories were depleted and a final plateau, indicating a much lower release rate due to diffusion (if there is any release at all). Most of the data provided, then, was confined to the initial and final plateaus, with little available from the high release rate portion. Further, more tests were conducted at 1600°C than at any other temperature, although the isothermal test temperatures ranged from 1350 to 2100°C. In order for the model to be able to interpolate with any degree of confidence, the input space had to be as balanced as possible. Without this, inadvertent biases would be introduced and trained into the network. In other words, the network would tend to provide better predictions for conditions approximating the preponderance of training data and provide poorer predictions for other areas in the input space. To redress this imbalance, the data available for each test were expanded significantly by linear interpolation, so that the available number of training vectors was increased from 1371 to 4049. Any inaccuracies introduced by this approach were felt to be well within the actual noise of the data itself.

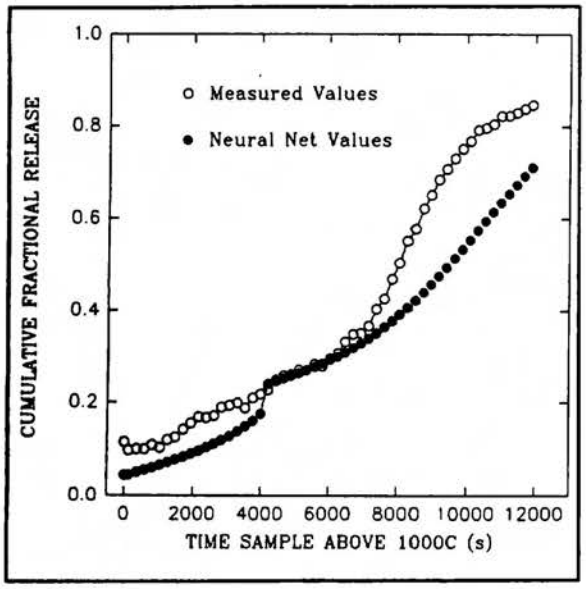

Figure 4. Cumulative release of Cs-134 during Test HCE-2 CM6, as measured by CRL and predicted by neural net model.

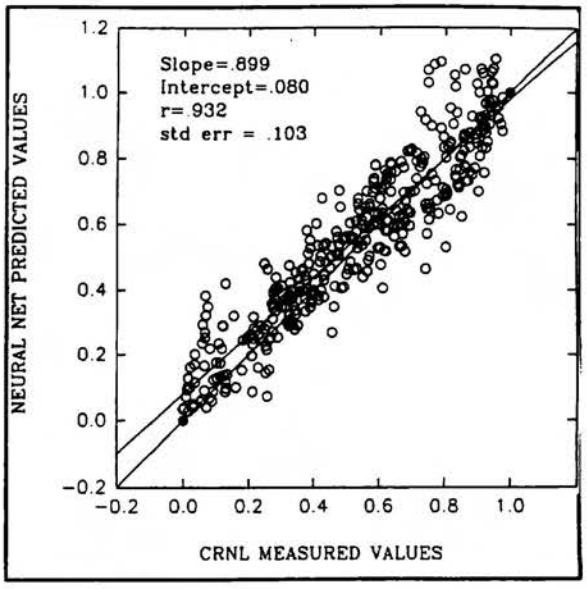

Figure 5. Scatter plot of cumulative fractional release values of Cs-134 predicted by neural net model vs values measured by CRL. Values are for test set.

The expanded data base was separated into two portions, with 90% of the data provided for the training set and the remaining 10% for the test set. In order to achieve balance, data from some tests were repeated, so that the total input space contained 12,516 vectors. Further, the results from a complete test, HCE2-CM6, were withheld from both the above sets to provide a validation by exposing the trained network to conditions it had not been trained on.

Model effectiveness was gauged in three ways:

1. Network predicted cumulative fractional releases (outputs) were plotted against the corresponding values actually determined by CRL. A perfect correlation would have all points fall along the diagonal with a slope of 1 and an intercept of 0. The corresponding values from the linear regression through the data were compared to the ideal values.

2. The correlation coefficient, r, (or the square root of r^2, which is sometimes referred to as Pearson product moment correlation coefficient or the correlation of determination, and is a measure of the closeness of fit of a scatter graph to its regression line with $r^2=1$ being a perfect fit) is the covariance divided by the product of the sample standard deviations[11].

3. The standard error (standard deviation) of the mean, which is computed by dividing the sample standard deviation by the square root of the sample size.

As noted already, networks of differing numbers of hidden nodes had comparable r values for the test set. An example of the scatter plot for a network having 4 hidden nodes in one hidden layer can be seen in Figure 5. The solid dots show a perfect correlation, which can be compared with the actual linear regression through the points. Most of the 396 test vectors provide points very close or on the regression lines. Overall, the slope of .899 and intercept of .080 are fairly close to the optimal values of 1 and 0 respectively. The r value of .932 shows that the regression itself accounts for 93.2% of the dispersion in the data, with the remainder attributable to the data itself.

The validation of the network involved using the vectors of a complete test, HCE2-CM6, with the results contained in Figure 6. For this particular test, the network provides a good linear correlation (r is very good at .98). At low fractional releases, the error is as high as 60%, although in this region there is considerable uncertainty in the CRL data itself. In the middle third of the test, the predicted values are virtually identical to the measured values. At the upper end, the discrepancy is 16%, for an overall average standard error of 11.7%. Returning to Figure 5, however, it can be seen that some tests are underpredicted while others are overpredicted, while the vast majority of vectors are well predicted.

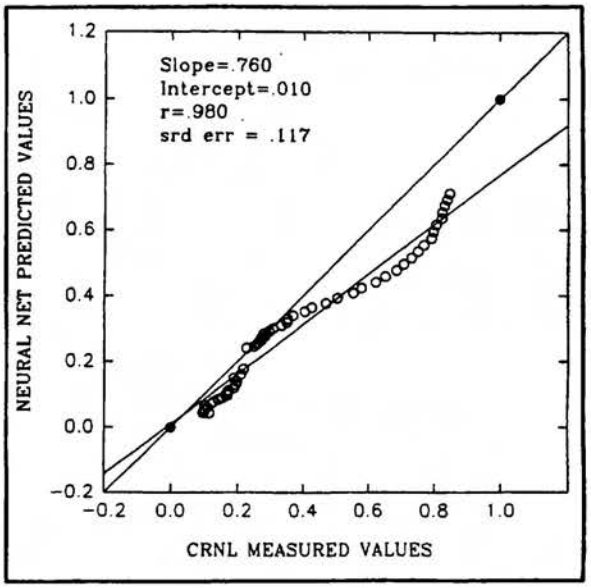

Figure 6. Scatter plot of cumulative fractional release values of Cs-134 predicted by neural net model vs values measured by CRL. Values are for validation set, test HCE-2 CM6.

Figure 4 shows the measured and predicted cumulative fractional release plotted against time that the sample is above 1000°C. Two points are of note here. The neural network model provides a smoothing of the data, with the exception of the discontinuity at about 4000 s. The second point is that the model is able to reproduce the non-linearity of the relationship between fractional release and time, due to varying release rates. In this particular test, though, the model values diverge from the measured at less than 4000s and beyond about 7500 s. The first 14 points (26% of the data set) have discrepancies of about 40%, while the remainder of the population is significantly lower.

The overall relative closeness of the model predictions to the values measured by CRL indicates that a trained neural network model has been able to establish a good correlation between a number of disparate parameters representative of possible severe reactor accident conditions and the cumulative fractional release of fission product cesium. This also suggests that the network model can be analyzed further to provide insights into the relative importance of the various input parameters to fission product release and release rates. Further work will also involve comparing network model predictions of release with CORSOR-M and will investigate the performance of a trained model as a source term subroutine in a more comprehensive accident progression code.

CONCLUSIONS

A back propagation neural network model with a modified delta learning rule has been trained to predict the cumulative fractional release of the volatile fission product cesium from CANDU fuel fragments and mini-elements under a variety of simulated severe accident conditions. The model was able to reproduce the non-linearities inherent in the relationship between fractional release and time, and provided a smoothing of the data. Finally, the model was able to predict the general trend of the release kinetics for a validation set which was not used for training, and to predict the cumulative fractional release to within a standard error of 12%.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the use of the experimental data performed by the Fission Product Release Group of the Fuel Engineering Branch at the Chalk River Laboratories. The experimental work was funded by Atomic Energy of Canada Limited, Ontario Hydro, Hydro Quebec, and New Brunswick Electric Power Corporation under the CANDU Owners Group agreement. The work performed at the Royal Military College was funded by the Academic Research Program of the Department of National Defence of Canada under allocation 3705-882.

REFERENCES

1. M.F. Osborne and R.A. Lorenz, "Results of ORNL VI Series Fission Product Release Tests", Proceedings of the 20th Water Reactor Safety Information Meeting, Bethesda, Maryland, 21-23 Oct, 1992.

2. M.R. Kuhlman, D.J. Lehmicke and R.O. Meyer, "CORSOR User's Manual", NUREG/CR-4173, USNRC, Mar 1985.

3. D.S. Cox, Z. Liu, P.H. Elder, C.E.L. Hunt and V.I. Arimescu, "Fission-Product Release Kinetics from CANDU and LWR Fuel During High-Temperature Steam Oxidation Experiments", presented at the IAEA Technical Meeting on Fission Gas Release and Fuel Rod Chemistry Related to Extended Burnup, Pembroke, Ontario, 28 Apr to 1 May, 1992.

4. D.S. Cox, Z. Liu, R.S. Dickson and P.H. Elder, "Fission-Product Releases During Post-Irradiation Annealing of High Burnup CANDU Fuel", presented at the Third International Conference on CANDU Fuel, Pembroke, Ontario, 4-8 Oct, 1992.

5. B.J. Lewis, F.C. Iglesias, C.E.L. Hunt and D.S. Cox, "Release Kinetics of Volatile Fission Products Under Severe Accident Conditions", Nucl. Technol., Vol 99, 330 (1992).

6. J. Rest, "Modeling the Behavior of Xe, I, Cs, Te, Ba and Sr in Solid and Liquefied Fuel During Severe Accidents", J. Nucl. Mater., Vol 150, 203 (1987).

7. L.D. MacDonald, D.B. Duncan, B.J. Lewis and F.C. Iglesias, "FREEDOM: A Transient Fission-Product Release Model for Radioactive and Stable Species", Proceedings of Technical Committee Meeting on Water Reactor Fuel Element Computer Modelling in Steady-State, Transient and Accident Conditions, Preston, England, 18-22 Sep, 1988, AECL-9810, International Atomic Energy Agency, (1988)

8. M.F. Osborne and R.A. Lorenz, "ORNL Studies of Fission Product Release Under LWR Severe Accident Conditions", Nuclear Safety, Vol. 33, No. 3, 334 (1992)

9. B.J. Lewis, D.S. Cox and F.C. Iglesias, "A Kinetic Model for Fission-Product Release and Fuel Oxidation Behaviour for Zircaloy-Clad Fuel Elements Under Reactor Accident Conditions", Proceedings of INC 93.

10. D.E. Rummelhart and J.L. McLelland, "Parallel Distributed Processing: Explorations in the Microstructure of Cognition Volume 1: Foundations, MIT Press, Cambridge, 1986.

11. R.L. Mason, R.F. Gunst and J.L. Hess, "Statistical Design and Analysis of Experiments with Applications to Engineering and Science", John Wiley & Sons, New York, (1989).