*Proceedings of the 2002 Winter Simulation Conference*
*E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds.*

# GAUSSIAN RADIAL BASIS FUNCTIONS FOR SIMULATION METAMODELING

Miyoung Shin

Bioinformatics Team
Electronics and Telecommunication
Research Institute
161 Kajong-Dong, Yusong-Gu
Taejon, 305-350, KOREA

Robert G. Sargent
Amrit L. Goel

Department of Electrical Engineering
and Computer Science
Syracuse University
Syracuse, NY 13244, U.S.A.

## ABSTRACT

This paper presents a novel approach for developing simulation metamodels using Gaussian radial basis functions. This approach is based on some recently developed mathematical results for radial basis functions. It is systematic, explicitly controls the underfitting and overfitting tradeoff, and uses a fast computational algorithm that requires minimal human involvement. This approach is illustrated by developing metamodels for the M/M/1 queueing system.

## 1 INTRODUCTION

Kleijnen and Sargent (2000) recently proposed a general methodology for fitting and validating simulation metamodels and identified four general goals: (i) understanding the problem entity, (ii) predicting values of the output or response variable, (iii) performing optimization, and (iv) aiding verification and validation. This paper primarily addresses goal (ii), the development of suitable predictive metamodels. Many types of metamodels have been proposed over the last twenty-five years including polynomial regression, neural networks, radial basis functions, and splines. Barton (1993) provides a survey and discussion of the key properties of several types of metamodels. Polynomial regression is the most common type of metamodel; however, in general, such models are not suitable for fitting complex surfaces (Barton 1993). In this paper we propose a new approach to metamodeling that employs Gaussian Radial Basis Functions (GRBFs).

The GRBF model can be viewed as a realization of a sequence of two mappings. The first is a nonlinear mapping of the input data via the basis functions and the second is a linear mapping of the basis function outputs via the weights to generate the model output. This feature of having both nonlinearity and linearity in the model, which can be treated separately, makes this a very versatile modeling technique. However, commonly used algorithms for determining GRBF parameters produce irreproducible results due to their ad-hoc nature. Recently, Shin and Goel (1998, 2000) have developed a new algorithm for Radial Basis Function (RBF) modeling that is computationally fast and yields reproducible results in a systematic way. Furthermore, it provides an explicit control on the tradeoff between model underfitting and overfitting.

Following this introduction, Section 2 formally defines and discusses metamodeling using RBFs, Section 3 describes the GRBF model, and Section 4 outlines the SG algorithm for developing GRBF metamodels. In Section 5, metamodels for an M/M/1 queueing system are developed and analyzed. Finally, in Section 6, we present some concluding remarks.

## 2 PROBLEM DEFINITION

The objective here is to construct a GRBF metamodel that approximates an unknown input-output mapping on the basis of given simulation data. The goal, however, is not to provide an exact fit to the data but to develop a metamodel that captures the underlying relationship so that it can be used to predict the output at some future observation of the input. This latter property is called generalization ability, a term taken from psychology.

Now we state the problem analytically. Suppose we are given an $n \times d$ input matrix $\mathbf{X} = (\mathbf{x_1},\mathbf{x_2},\ldots,\mathbf{x_n})^T$ where each of the n input vectors $\mathbf{x}_i$, $i = 1,\ldots,n$, is in a d-dimensional space. Let $\mathbf{y} = (y_1,y_2,\ldots,y_n)^T$ be the target vector whose elements $y_i$'s are the outputs corresponding to the input vectors $\mathbf{x_i}$, $i = 1,2,\ldots,n$. In other words, we are given a simulation data set

$$D = \{(\mathbf{x_i}, y_i) : \mathbf{x}_i \in R^d, y_i \in R, i = 1,...,n\} \quad (1)$$

in which both the values of the inputs and the corresponding outputs are made available. The problem we address is to construct a metamodel for the above mapping from a d-dimensional input space to a one-dimensional target value based on the simulation data D. In particular, we seek

GRBF parameter values that give the smallest fitting error and also provide good generalization ability.

From a modeling perspective, we seek a parsimonious model that also yields a small fitting error. However, a model with a very small number of terms will suffer from underfitting because it fails to capture the input-output mapping and hence provides a poor fit. On the other hand, a model that has too many terms would fit the data too well to be able to generalize on unseen data and would suffer from overfitting. This phenomenon is well known in modeling as the bias and variance dilemma. Figure 1 is an idealized graphical depiction of this phenomenon, which illustrates that simple models tend to have high bias and low variance, while complex models have low bias and high variance. The objective is to develop a metamodel that provides an acceptable trade-off between bias and variance. Conceptually, this is similar to seeking a model with just enough complexity that represents an acceptable trade-off between fitting and generalization errors.

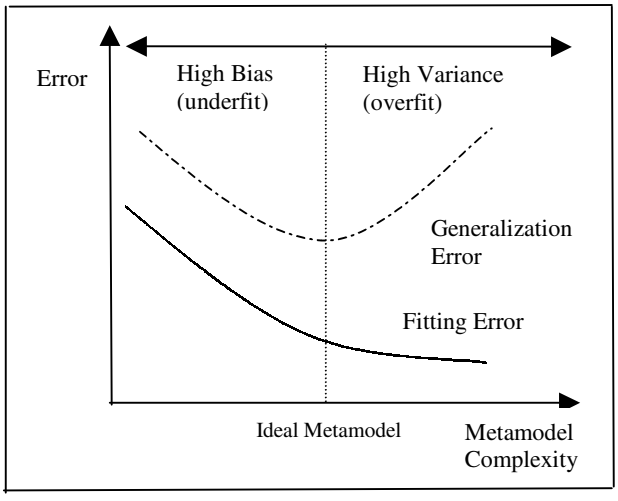


Figure 1: Idealized Depiction of Bias-Variance and Fitting-Generalization Errors

### 2.1 Metamodel Evaluation

The performance of a metamodel can be evaluated according to the following criteria (Mehrotra, Mohan, and Ranka 1997).

- **Quality of results**: gauged in terms of an error function, e.g., mean squared error as discussed below.
- **Generalizability**: measured on some validation data different from the data used for developing the metamodel; also measured in terms of an error function.
- **Computational resources**: generally measured in terms of metamodeling resources and resources for model implementation such as CPU time and memory requirements.
- **Model complexity**: a measure of the size of the metamodel. For a GRBF metamodel the complexity is proportional to the number of terms in the model.

### 2.2 Error Functions

A metamodel produces estimates output $\hat{y}_i$ 's that could deviate from the observed (simulated) values $y_i$'s. Therefore some measure is required to assess the degree of this deviation. Such measures are usually termed error functions. In general, any differentiable function that is minimized upon setting $\hat{y}_i = y_i$ could be used as an error function. Each measure has its strengths and weaknesses (Mehrotra, Mohan, and Ranka 1997) and hence some are preferred over others in a given situation. In this paper we employ the mean squared error (MSE) defined as follows:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 . \qquad (2)$$

## 3 GRBF MODEL

The RBF model is a special type of neural network consisting of three layers. The first layer (input layer) distributes input vectors to each of the receptive field units in the second layer (hidden layer) without any multiplicative factors. The hidden layer has m receptive field units (or hidden units), each of which represents a nonlinear transfer function called a basis function. The hidden units play a role in simultaneously receiving the input vector and nonlinearly transforming the input vector into an m-dimensional vector. The outputs from the m-hidden units are then linearly combined with weights to produce the network output at the output layer. Thus, the typical RBF model is described by specifying the number of basis functions (i.e., hidden units), basis function parameters, and the weights of the basis function outputs to produce the network output.

Formally, for a mapping $f : R^d \rightarrow R$, the RBF network model can be described mathematically as

$$f(\mathbf{x}) = \sum_{j=1}^{m} w_j \varphi_j(\mathbf{x}) = \sum_{j=1}^{m} w_j \varphi(\left\|\mathbf{x} - \boldsymbol{\mu}_j\right\| / \sigma_j) , \qquad (3)$$

where $\mathbf{x} \in R^d$ is the input vector, $\boldsymbol{\mu}_j \in R^d$ is the $j$th basis function center, $\|\cdot\|$ denotes the Euclidean distance, $w_j$'s are weights, and $\sigma_j$'s are basis function widths. The basis function $\phi(\cdot)$ here plays the role of transfer functions in traditional neural networks. Yet, the basis functions have the unique feature that their responses to the input vectors are not only radially symmetric but also monotonically de-

creasing or increasing with distance from the center. In practice, several forms of $\phi$ are used for RBF models. Amongst these, the Gaussian is probably the most popular basis function because it has attractive mathematical properties of universal and best approximation and its hill-like shape is easy to control with the parameter $\sigma$. For a Gaussian RBF (GRBF), equation (3) becomes

$$f(\mathbf{x}) = \sum_{j=1}^{m} w_j \exp(-\|\mathbf{x} - \boldsymbol{\mu}_j\|^2 / 2\sigma_j^2) . \quad (4)$$

## 4 SG ALGORITHM FOR GRBF METAMODEL

The GRBF model is completely defined by the parameters $P = (m, \boldsymbol{\sigma}, \boldsymbol{\mu}, \mathbf{w})$. Here m is the number of basis functions with widths $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, ..., \sigma_m)$, basis function centers $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_m)$, and weights from the basis functions to the output are $\mathbf{w} = (w_1, w_2, ..., w_m)$. Therefore the GRBF design problem is that of determining its 3m parameters, namely, m centers, m widths and m weights. It is quite common in many applications to use a global width $\sigma = \sigma_i$, i = 1,…,m, and we do so in this paper. Then the number of parameters to be determined reduces to (2m+1).

Next we summarize the SG algorithm that is based on a new mathematical framework for radial basis functions developed in Shin and Goel (1998, 2000). In this algorithm the modeling problem is formulated as a three-step process as illustrated in Figure 2. In step 1, the algorithm selects m for a given $\sigma$ and a specified value of $\delta$, a parameter used to control the degree of fit as discussed below. For a selected $(m,\sigma)$ pair, the centers are determined in step 2. This completes the hidden layer design, i.e., the nonlinear parameters of the GRBF model. In step 3, the linear parameters (weights) are determined by the least squares method.

Now we briefly describe $\delta$ which is discussed at length in Shin and Goel (1998) along with its mathematical underpinnings. In step 1, for given $\sigma$, we wish to know how many basis functions are needed to provide adequate representation of the input space when we know that 100% coverage would require m to be equal to n, the number of data points. This corresponds to $\delta = 100\%$. Shin and Goel (1998, 2000) and Shin and Park (2000) show that GRBF models with values of $\delta$ less than 100% provide adequate coverage and yield m values much less than n which results in simpler, yet good models. A heuristic suggestion for the range of $\sigma$ values is $0.01 \le \sigma \le \sqrt{d/2}$ where $d$ is the number of input variables (Shin and Goel 1998, 2000).

For fixed width $\sigma$, larger values of $\delta$, in general, require larger m. However, the interdependencies among $\delta$, $\sigma$, and m and the other model parameters are completely dictated by the data for which the metamodel is to be developed. These interdependencies determine the exact shapes of the fitting and generalization error curves in Figure 1. In other words, determining the right tradeoff between the two errors in Figure 1 is equivalent to determining an appropriate combination of $\delta$ and $\sigma$ values that are employed by the SG algorithm to find the most appropriate model. Determining the right tradeoff is an important objective of developing a good GRBF metamodel. A very important feature of the SG algorithm is that it provides the mathematical and computational machinery for evaluating and controlling this trade-off.

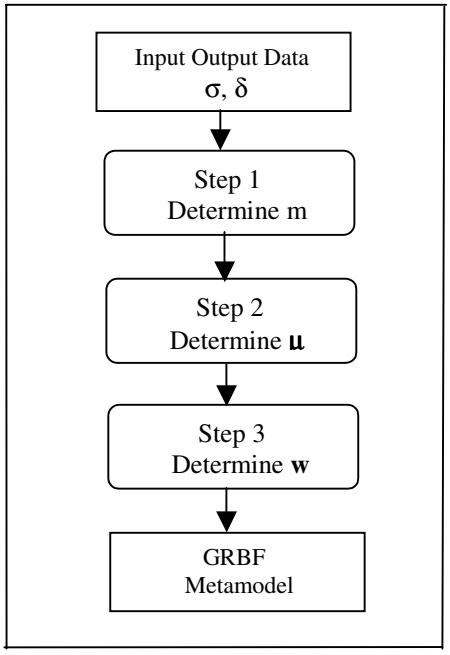


Figure 2: Steps in GRBF Metamodeling

## 5 SINGLE SERVER QUEUEING METAMODEL

In this section we illustrate the GRBF metamodeling process for the M/M/1 queue with FIFO queue discipline. The performance measure (output variable) of interest is the expected steady-state sojourn time, *W*, in the system (i.e., waiting time plus service time). Let $\lambda$ be the arrival rate and $\eta$ be the service rate. The true response *W* is well known to be

$$W = 1 / (\eta - \lambda) . \quad (5)$$

We prepared two data sets for this example with $\lambda$ = 1.0 and $\eta$ in the range 1.1 to 10.0. For the first data set, DS1, $\eta$ was set at twenty equally spaced values in the above range. The corresponding *W* values were computed from equation (5). The objective here is to develop a metamodel for data set DS1. Note that the generalization error in Figure 1 cannot be computed, as it requires knowledge of the true model. In practice, error on an independent data set, called test error, is employed as an indicator of

generalization error. For this purpose, a second data set, DS2, was prepared for 21 values of η; 19 equally spaced values between 1.25 and 9.75 and two additional values at 1.0 and 10.0. Again, the corresponding values of *W* for λ = 1.0 were computed from equation (5).

### 5.1 Metamodeling Process

We now illustrate the GRBF metamodeling process for the above system. First, we choose δ to be 99.5% and select a set of seven σ values as (0.05, 0.10, 0.15, 0.20, 0.30, 0.40, 0.50). Then the SG algorithm yields exactly one metamodel for each σ. Each model obtained by this algorithm has the property that it not only provides at least 99.5% coverage of the input space, but also has the smallest value of m. In other words, each metamodel is the most parsimonious for given σ while satisfying the δ requirement. For each model, the fitted MSE is computed based on DS1 and the test MSE is computed from DS2. A summary of the seven resulting metamodels (A to G) is given in Table 1.

Table 1: GRBF Metamodels

| Metamodel | σ | No. of Parameters | Fitting MSE | Test MSE |
|---|---|---|---|---|
| A | 0.50 | 7 | 2.72 | 2.66 |
| B | 0.40 | 9 | 1.96 | 1.87 |
| C | 0.30 | 11 | 1.45 | 1.43 |
| D | 0.20 | 15 | 0.76 | 0.96 |
| E | 0.15 | 17 | 0.65 | 0.90 |
| F | 0.10 | 25 | 0.15 | 0.61 |
| G | 0.05 | 41 | 0.00 | 0.70 |

Next, we study metamodels A to G in Table 1 in the context of idealized complexity and error relationships depicted in Figure 1. We note that the number of basis functions increases from 3 to 20, for metamodels A to G, while the number of corresponding model parameters increases from 7 to 41. The fitting MSE decreases from 2.72 to 0.00 with increasing model complexity. However, test MSE first decreases from metamodels A to F and then shows a slight increase for metamodel G. These fitting and test error behaviors with respect to model complexity are consistent with the idealized behavior indicated in Figure 1.

It should, however, be noted that for the GRBF model, both m and σ play an important role in determining the quality of the metamodel. Thus, it is also necessary to look at both the fitting and test MSE's with respect to m and σ. These error curves are plotted in Figure 3 in the (m,σ) plane. The errors with respect to m alone are shown in Figure 4. From these Figures we note that fitting MSE curve monotonically decreases from left to right (metamodel A to metamodel G) and the test MSE first decreases and then increases. This seems to indicate that metamodels A to D are too simple and represent underfitting while metamodel G is too complex and represents overfitting.

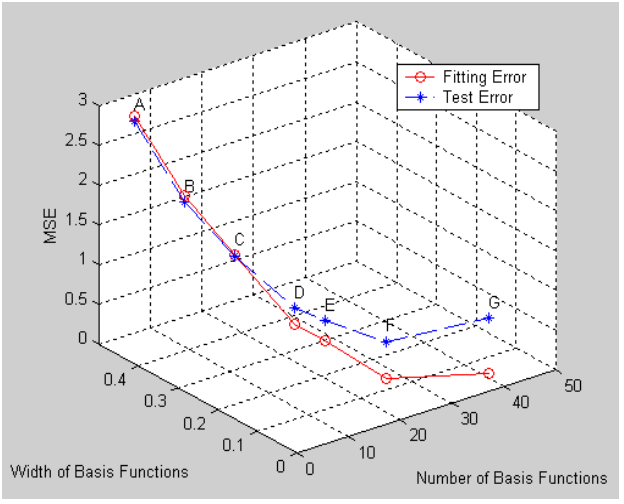


Figure 3: Fitting and Test Error Plots Based on Data in Table 1

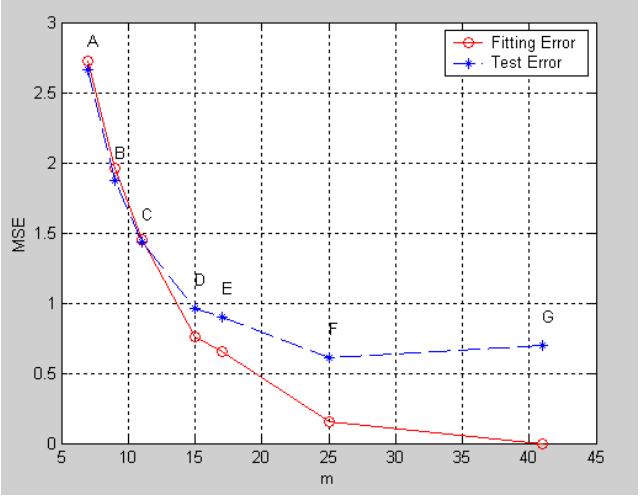


Figure 4: Error Curves versus m

### 5.2 Metamodel Selection

The real question we face is which metamodel is the best. There is no unique answer to this question since the choice of a metamodel is governed by many application specific considerations. Among other things, choosing a model involves trade-offs between model complexity, fitting MSE, and test MSE. From the MSE perspective alone metamodel F may be the best. However, if model complexity is of primary concern, metamodel E seems to be a more appropriate choice.

In the above discussion we have argued about the underfitting and overfitting tradeoff in metamodeling using model complexity, fitting error, and test error. Now we graphically evaluate four representative models from Table 1 with respect to the true underlying model of equation (5). We select metamodel B to show underfit, metamodel G to

show overfit and metamodels E and F as representatives of good compromise metamodels.

Plots of mean time in system from equation (5), and from metamodels B and G are given in Figure 5. Note that the plot of metamodel B is simple but its fit is unsatisfactory, i.e., it represents underfit. Metamodel G is an overfitted model with maximum possible complexity and it seems to capture the underlying relationship of the true model. This is possibly due to the output relationship of the true model being deterministic and smooth. Finally, in Figure 6, metamodels E and F seem to be good representatives of the true model. Yet, they have different error rates and metamodel F is much more complex than metamodel E as seen from the data in Table 1.

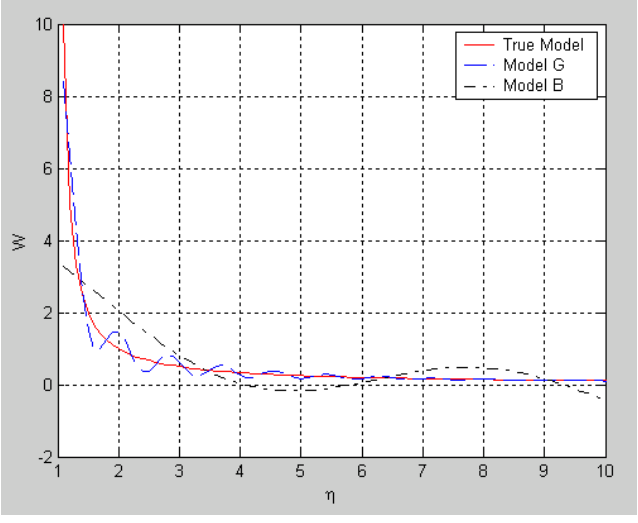


Figure 5: Plots of B, G, and True Models

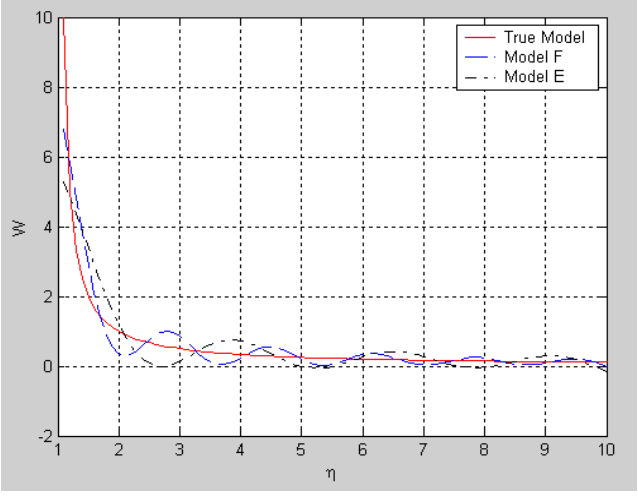


Figure 6: Plots of E, F, and True Models

In summary, the analyses presented above demonstrate that the approach developed here can be employed to objectively distinguish among the metamodels in terms of their fitting and test errors and model complexity.

## 6 CONCLUDING REMARKS

We introduced a new approach for simulation metamodeling based on GRBFs. A systematic methodology using the SG algorithm for model development and evaluation was described. We illustrated the use of this new approach by developing a set of metamodels for a deterministic simulation, namely, the expected time in an M/M/1 system, and discussed the trade-offs among these metamodels. We believe GRBFs to be a promising alternative to the commonly used polynomial regression for predictive metamodels. This approach also could be used to investigate some of the metamodeling issues discussed in Sargent (1991) and Kleijnen and Sargent (2000).

## REFERENCES

Barton, R. R. 1993. New Tools for Simulation Metamodels. IMSE Working Paper 93-110, Department of Industrial and Management Systems Engineering, The Pennsylvania State University, University Park, Pennsylvania.

Kleijnen, J. P. C., and R. G. Sargent. 2000. A Methodology for Fitting and Validating Metamodels in Simulation. *European Journal of Operational Research* 120: 14-29.

Mehrotra, K., C. Mohan, and S. Ranka. 1992. *Elements of Artificial Neural Networks*. The MIT Press, Cambridge, MA.

Sargent, R. G. 1991. Research Issues in Metamodeling. In: *Proceedings of the 1991 Winter Simulation Conference*, eds. B. L. Nelson, W. D. Kelton, and G. M. Clark, 888-893. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Shin, M., and A. L. Goel. 1998. Radial Basis Function Model Development and Analysis Using the SG Algorithm. Technical Report 98-5, Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, New York 13244.

Shin, M., and A. L. Goel. 2000. Empirical Data Modeling in Software Engineering using Radial Basis Functions, *IEEE Transactions on Software Engineering* 26: 567-576.

Shin, M., and C. Park. 2000. A Radial Basis Function Approach to Pattern Recognition and Its Applications, *ETRI Journal*, 22, 2: 1-10.

## AUTHOR BIOGRAPHIES

**MIYOUNG SHIN** received the B.S. and M.S. degrees in computer science from Yonsei University in Seoul, Korea, in 1991 and 1993, respectively, and the PhD from Syracuse University, New York, USA, in 1998. For her outstanding Ph.D. work, she was awarded the All-University Doctoral

Prize from Syracuse University, where she was also a post-doctoral research fellow in 1998 and 2001. Currently she is working as a senior member of technical staff at the Electronics and Telecommunication Research Institute in Korea. Her research interests include data mining algorithms, pattern recognition, and radial basis function modeling and evaluation in a variety of areas including medical diagnosis, software engineering, and bioinformatics. Her email address is <shinmy@etri.re.kr>.

**ROBERT G. SARGENT** is a Professor Emeritus of Syracuse University. He received his education at The University of Michigan. Dr. Sargent has served his profession in numerous ways and has been awarded the TIMS (now INFORMS) College on Simulation Distinguished Service Award for long-standing exceptional service to the simulation community. His current research interests include the methodology areas of both modeling and discrete event simulation, model validation, and performance evaluation. Professor Sargent has published extensively and is listed in Who's Who in America. His email and web addresses are <rsargent@syr.edu> and <www.cis.syr.edu/srg/rsargent/>.

**AMRIT L. GOEL** received his PhD from The University of Wisconsin, Madison and is currently a Professor of Computer Engineering at Syracuse University. His current interests are in software engineering and data mining algorithms including radial basis functions. He has published extensively and has advised PhD students on a wide range of topics including quality control and reliability, hypothesis testing, databases, parallel architectures and data mining algorithms. He is coauthor of Object Oriented Software Testing, (Springer Verlag, 2000). He was elected a fellow of the IEEE for contributions to the reliability of computer software. His email address is <goel@ecs.syr.edu>.