International Journal of Fuzzy Logic and Intelligent Systems, vol. 5, no. 4, Dec 2005 pp. 286-290

A simple method to compute a periodic solution of the Poisson equation with no boundary conditions

Byung Soo Moon, Jang Soo Lee, Dong Young Lee, Kee-Choon Kwon

Korea Atomic Energy Research Institute

P.O.Box 105, Yuseong, Daejeon 305-600, Korea

Abstract

We consider the Poisson equation where the functions involved are periodic including the solution function. Let $R=[0,1]\times[0,1]\times[0,1]$ be the region of interest and let $\phi(x,y,z)$ be an arbitrary periodic function defined in the region R such that $\phi(x,y,z)$ satisfies $\phi(x+1,y,z)=\phi(x,y+1,z)=\phi(x,y,z+1)=\phi(x,y,z)$ for all x,y,z. We describe a very simple method for solving the equation $\nabla^2 u(x,y,z)=\phi(x,y,z)$ based on the cubic spline interpolation of $u(x,y,z)$; using the requirement that each interval $[0,1]$ is a multiple of the period in the corresponding coordinates, the Laplacian operator applied to the cubic spline interpolation of $u(x,y,z)$ can be replaced by a square matrix. The solution can then be computed simply by multiplying $\phi(x,y,z)$ by the inverse of this matrix. A description on how the storage of nearly a Giga byte for $20\times20\times20$ nodes, equivalent to a 8000×8000 matrix is handled by using the fuzzy rule table method and a description on how the shape preserving property of the Laplacian operator will be affected by this approximation are included.

Key Words : Poisson Equation; Cubic Spline Interpolation; Fuzzy Rule Table; Cubic B-spline; Shape Preservation;

1. Introduction

Consider the Navier-Stokes equation in the following form; Given $u(x,0)=u^0(x)$ with $\nabla\circ u^0=0$, find $u(t)=(u_1(t),u_2(t),u_3(t))$ such that

$$\frac{\partial u_i}{\partial t}+\sum_{j=1}^{3}u_j\frac{\partial u_i}{\partial x_j}=-\nu\triangle u_i-\frac{\partial P}{\partial x_i}+f_i(x,t) \qquad (1)$$

where $\nu>0$ and $\triangle u_i=\nabla^2 u_i$ and $\nabla\circ u=0$.

Finding a spatially periodic numerical solution to the above is one of the seven Millenium problems [1]. Note that if we differentiate (1) both sides with respect to x_i and take the sum for $i=1,2,3$, then we obtain

$$\frac{\partial}{\partial t}\left(\sum_{i=1}^{3}\frac{\partial u_i}{\partial x_i}\right)+\sum_{j=1}^{3}\left(\sum_{i=1}^{3}\frac{\partial u_j}{\partial x_i}\frac{\partial u_i}{\partial x_j}\right)+\sum_{j=1}^{3}u_j\frac{\partial}{\partial x_j}\left(\sum_{i=1}^{3}\frac{\partial u_i}{\partial x_i}\right)$$
$$=\nu\triangle\left(\sum_{i=1}^{3}\frac{\partial u_i}{\partial x_i}\right)-\triangle P+\sum_{i=1}^{3}\frac{\partial f_i}{\partial x_i}$$

Now, we apply the fact that we require the divergence of u to be zero, i.e $\sum_{i=1}^{3}\frac{\partial u_i}{\partial x_i}=0$ so that the first and the third term on the left and the first term on the right are all zeros. Hence, we have

$$\triangle P=\sum_{i=1}^{3}\frac{\partial f_i}{\partial x_i}-\sum_{j=1}^{3}\sum_{i=1}^{3}\frac{\partial u_j}{\partial x_i}\frac{\partial u_i}{\partial x_j} \qquad (2)$$

Thus, in a numerical solution of the Navier-Stokes equation (1) under the divergence free condition, one has to solve the accompanying Poisson equation involving the pressure at each iteration.

There are many studies on how to solve the Poisson equation numerically [2,3,4,5]. All of these solutions, however, are for the Poisson equation with either a Dirichlet or a Cauchy boundary condition and none with no boundary condition. Furthermore, we need a solution which is more efficient and faster so that it can be used for a solution of the Navier-Stokes equation repeatedly.

In this paper, we describe a very simple and fast algorithm to compute the pressure in (2), check the shape preservation property using the algorithm, and show how the large array can be handled using the fuzzy rule table method.

2. Convolution Matrix for the Laplacian Operator and a Poisson Equation Solver

Consider the Poisson equation $\triangle P(x,y,z)=\phi(x,y,z)$ in a region $R=[0,1]\times[0,1]\times[0,1]$. We divide each of the intervals $[0,1]$ into N subintervals of equal length. Let x_i, y_j, z_k for $i,j,k=-1,0,1,2,\cdots,N,N+1$ be the nodal points in each axis and define the cubic B-splines $B_i(t)'s$ as;

$$\frac{1}{6h^2}\begin{cases}(t-t_{i-2})^3\\ h^3+3h^2(t-t_{i-1})+3h(t-t_{i-1})^2-3(t-t_{i-1})^3\\ h^3+3h^2(t_{i+1}-t)+3h(t_{i+1}-t)^2-3(t_{i+1}-t)^3\\ (t_{i+2}-t)^3\\ 0\end{cases} \qquad (3)$$

on each of the intervals $t\in[t_{i-2},t_{i-1}]$, $t\in[t_{i-1},t_i]$, $t\in[t_i,t_{i+1}]$, and $t\in[t_i,t_{i+1}]$, respectively. Then the

Manuscript received Nov. 11, 2005; revised Nov. 25, 2005.

solution can be approximated by a cubic spline interpolation

$$p(x,y,z)=\sum_{i,j,k} a_{i,j,k}B_i(x)B_j(y)B_k(z) \tag{4}$$

Note that at the nodal points or the grid points, the B-splines take values

$$B_i(t_\alpha)=\frac{1}{6}\begin{cases}1, & \alpha=i-1\\ 4, & \alpha=i\\ 1, & \alpha=i+1\end{cases} \tag{5}$$

and

$$B_i''(t_\alpha)=\frac{1}{h^2}\begin{cases}1, & \alpha=i\\ -2, & \alpha=i\\ 1, & \alpha=i+1\end{cases} \tag{6}$$

Using the above relations, we evaluate

$$\frac{\partial^2 p}{\partial x^2}(x,y,z)=\sum_{i,j,k} a_{i,j,k}B_i(x)B_j(y)B_k(z)$$

at $(x_\alpha, y_\beta, z_\gamma)$ to obtain

$$\frac{\partial^2 p}{\partial x^2}(x_\alpha,y_\beta,z_\gamma)=\frac{1}{36h^2}\times$$

$$\begin{aligned}
&a_{\alpha-1,\beta-1,\gamma-1}+4a_{\alpha-1,\beta-1,\gamma}+a_{\alpha-1,\beta-1,\gamma+1}+\\
&\quad 4a_{\alpha-1,\beta,\gamma-1}+16a_{\alpha-1,\beta,\gamma}+4a_{\alpha-1,\beta,\gamma+1}+\\
&\quad a_{\alpha-1,\beta+1,\gamma-1}+4a_{\alpha-1,\beta+1,\gamma}+a_{\alpha-1,\beta+1,\gamma+1}+\\
&-2a_{\alpha,\beta-1,\gamma-1}-8a_{\alpha,\beta-1,\gamma}-2a_{\alpha,\beta-1,\gamma+1}-\\
&\quad -8a_{\alpha,\beta,\gamma-1}-32a_{\alpha,\beta,\gamma}-8a_{\alpha,\beta,\gamma+1}+\\
&\quad -2a_{\alpha,\beta+1,\gamma-1}-8a_{\alpha,\beta+1,\gamma}-2a_{\alpha,\beta+1,\gamma+1}+\\
&a_{\alpha+1,\beta-1,\gamma-1}+4a_{\alpha+1,\beta-1,\gamma}+a_{\alpha+1,\beta-1,\gamma+1}+\\
&\quad 4a_{\alpha+1,\beta,\gamma-1}+16a_{\alpha+1,\beta,\gamma}+4a_{\alpha+1,\beta,\gamma+1}+\\
&\quad a_{\alpha+1,\beta+1,\gamma-1}+4a_{\alpha+1,\beta+1,\gamma}+a_{\alpha+1,\beta+1,\gamma+1}
\end{aligned} \tag{7}$$

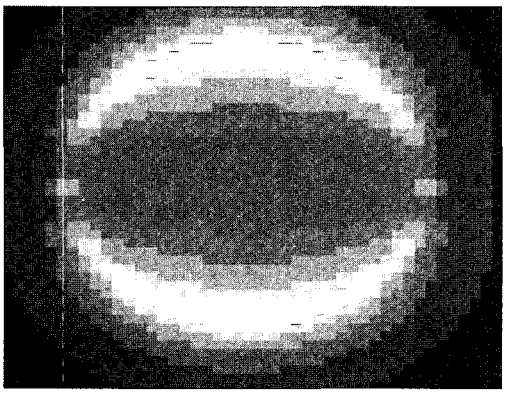

fig.1. Potential field for point sources located on a unit circle

Table 1. Convolution Matrix for $\frac{\partial^2 p}{\partial x^2}\times 36h^2$

$i=\alpha-1$			$i=\alpha$			$i=\alpha+1$		
1	4	1	-2	-8	-2	1	4	1
4	16	4	-8	-32	-8	4	16	4
1	4	1	-2	-8	-2	1	4	1

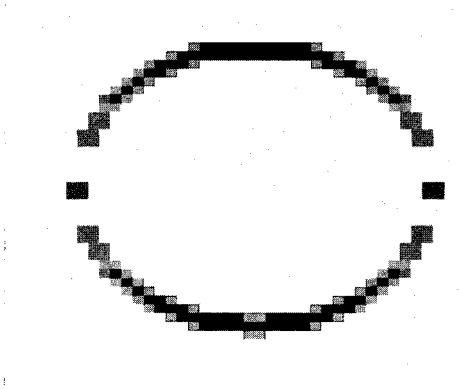

fig.2. Laplacian of the function in fig.1. (Numerical approximation)

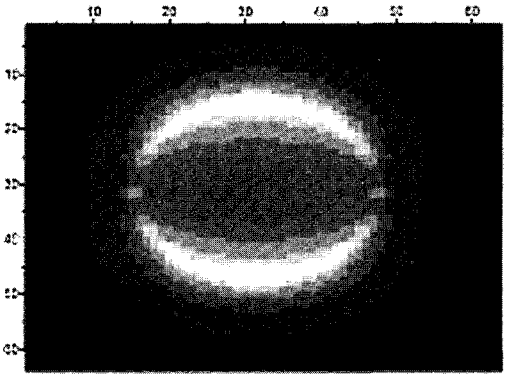

fig.3. Solution of the Poisoon Equation for the function in fig.1.

Thus, the evaluation of $\frac{\partial^2 p}{\partial x^2}(x_i,y_j,z_k)$ can be considered as evaluating the convolution of an $N\times N\times N$ matrix $\{a_{i,j,k}\}$ by a $3\times3\times3$ matrix shown in Table 1.

The other partial derivatives $\frac{\partial^2 p}{\partial y^2}(x_\alpha,y_\beta,z_\gamma)$ and $\frac{\partial^2 p}{\partial z^2}(x_\alpha,y_\beta,z_\gamma)$ can also be evaluated by using the same method. When the three terms are added, the resulting calculation turns out to be equivalent to evaluating a convolution of the coefficient matrix with the 3×3 matrix shown in Table 2.

Table 2. Convolution Matrix for the Laplacian Operator $\times 36h^2$

$i=\alpha-1$			$i=\alpha$			$i=\alpha+1$		
1	-2	1	4	-8	4	1	-2	1
4	-8	4	16	-32	16	4	-8	4
1	-2	1	4	-8	4	1	-2	1

Now, the Poisson equation can be written as a set of linear equations whose typical form is given in (7). These can in turn be written as a matrix equation form $Qz=b$, where Q is the coefficient matrix and

$z=(a_{1,1,1}, a_{2,1,1}, \cdots, a_{N,1,1}, a_{1,2,1}, a_{2,2,1}, \cdots, a_{N,N,N})$,
$b=(\phi(x_1, y_1, z_1), \phi(x_2, y_1, z_1), \cdots, \phi(x_N, y_1, z_1), \cdots,$
$\phi(x_1, y_2, z_1), \phi(x_N, y_2, z_1), \cdots, \phi(x_N, y_N, z_N))$.

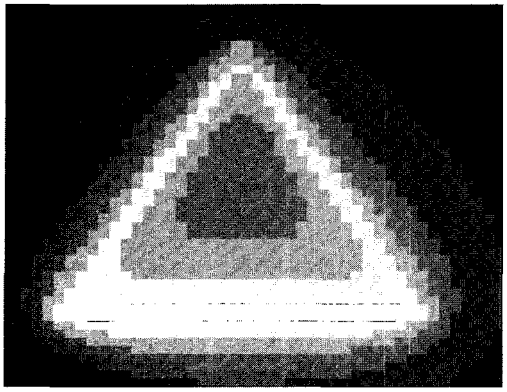

fig.4. The potential energy for the sources located on a triangle

Therefore, once the inverse of the constant matrix Q^{-1} is computed and stored, then the solution of the Poisson equation can be computed repeatedly simply by evaluating $Q^{-1}b$. Note that the arrangement of the elements $\phi(x_i, y_j, z_k)$ in b must be in conformance with the layout of the entries in the matrix Q.

2. Effect on the Shape Preservation Property

As a validity check for our solution, we examine in this section how the shape preservation property [6] will be affected by the approximation based on the cubic spline interpolation. Consider a finitee number of sources located at $P_1=(x_1, y_1)$, $P_2=(x_2, y_2)$, ..., $P_N=(x_N, y_N)$. The Poisson equation becomes

$$\nabla^2\left(\frac{a_1}{\sqrt{(x-x_1)^2+(y-y_1)^2}}+\cdots+\frac{a_n}{\sqrt{(x-x_n)^2+(y-y_n)^2}}\right) = a_1\delta(P_1)+a_2\delta(P_2)+\cdots+a_N\delta(P_N) \quad (8)$$

For simplicity, we take $a_1=a_2=\cdots=a_N=10$ and consider the contour graph of the function before the Laplacian. When the points $P_1, P_2, \cdots, P_N$ are located on a circle, the contour graph of this function after the discretization with a pixel size 64×64, we obtain the graph shown in fig.1, while the Laplacian of this function is shown in fig.3.

For our cubic spline approximation, the Laplacian operator for the functions of two variables can be evaluated by taking the convolution of the coefficient matrix with the 3×3 matrix in Table 3.

Table 3. Convolution Matrix for the Laplacian Operator in the 2 Dimensional Case

	$i=\alpha-1$	$i=\alpha$	$i=\alpha+1$
$j=\beta-1$	1	1	1
$j=\beta$	1	-8	1
$j=\beta+1$	1	1	1

When the Laplacian of the discretized function in fig.1 is computed, we obtain the image shown in fig.2. Now, if we solve the Poisson equation with the function in fig.2 as $\phi(x, y)$, we obtain the solution shown in fig.7. Note that this compares with fig.1 which is an exact analytical solution of the Poisson equation. In the case where we use a triangle instead of a circle, we obtain the results shown in fig.4, fig.5, and fig.6 instead of fig.1, fig.2, and fig.3.

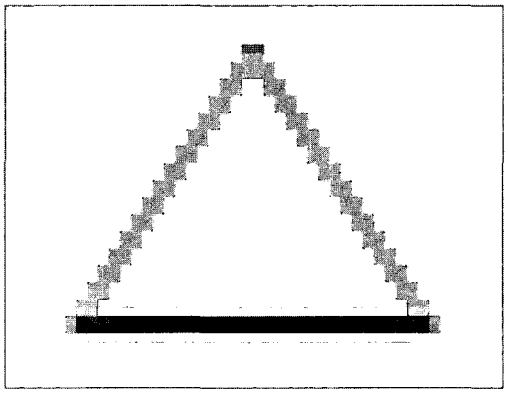

fig.5. Laplacian of the function in fig.4 - Numerical Approximation

To see how the Laplacian preserves the shape, consider an image where the only nonzero pixel is at (k, l), then the Laplacian of the image will have exactly 9 nonzero pixels; the immediate neighboring pixels of the pixel at (k, l). Hence, the boundary of the original image is maintained after an application of the Laplacian operator. Note that the shapes may not be preserved if the Laplacian is applied to a shape before the shape is thinned.

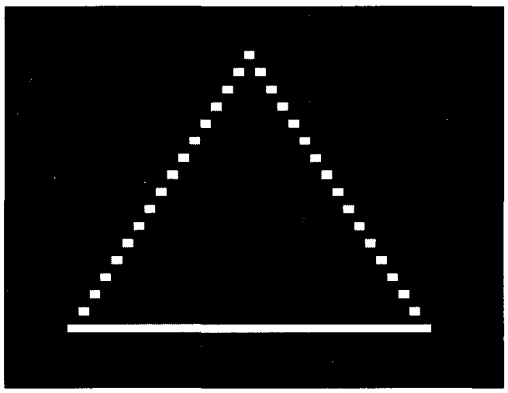

fig.6. Laplacian of the function in fig.4 - Analytical computation

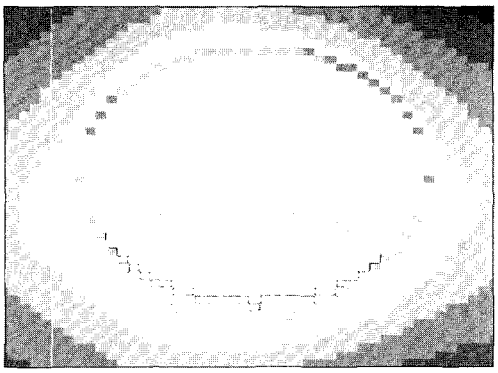

fig.7. Inverse Laplacian by our algorithm applied to the function in fig.2

3. Fuzzy rule table representation of the matrices and error estimates

In section 2, we noted that for a $20\times20\times20$ node discretization, a matrix of size 8000×8000 needs to be stored for the inverse of the convolution matrix. When the full accuracy of E15.8 is used to store each of the entries, the required storage will be 0.96GB and furthermore it will waste a great deal of time for reading in the matrix. To reduce the storage and hence to reduce the I/O time, we may apply the fuzzy rule method where the entries of the matrix are fuzzified and each entry is approximated by a representative value in the corresponding fuzzy set support intervals, i.e. each entry is replaced by the fuzzy set number, equivalently by an integer.

When we compute the maximum and the minimum of the entries of $Q^{-1}=(q_{i,j})$ in the inverse matrix, we find that $p_m=-2.212066E-2$ for the minimum entry while for the maximum, we have $p_M=-2.1034071E-2$. We take one hundreth of the difference $h=\frac{p_M-p_m}{100}$ as the uniform support length so that we get $h=1.086589E-5$. If we compute $\frac{q_{i,j}-p_0}{h}$ and truncate to the nearest integer to obtain $n_{i,j}$, then the matrix of two decimal digit integers $(n_{i,j})$ can be stored instead of Q^{-1}.

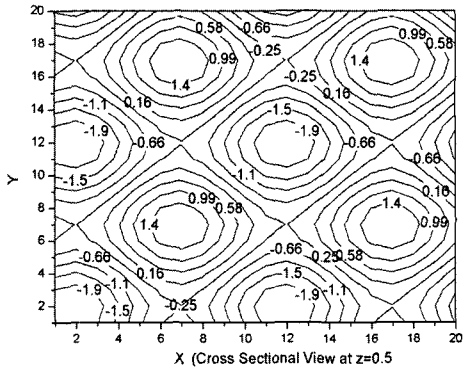

fig.8. Solution of the Poisson equation when $\phi(x,y,z)=(4\pi)^2\left(Sin(4\pi x+\frac{\pi}{8})+Sin(4\pi y+\frac{\pi}{8})+Sin(4\pi z+\frac{\pi}{8})\right)$

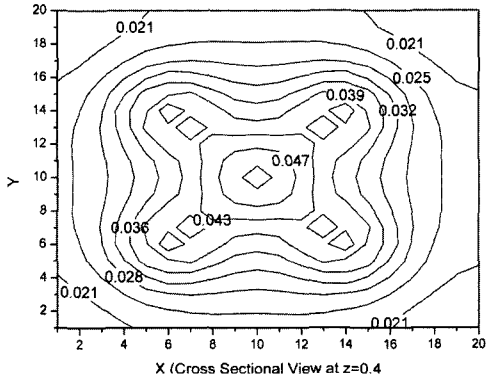

fig.9. Solution of the Poisson equation when $\phi(x,y,z)$ is sum of 5 delta functions

To verify our algorithm, we used an example function $u(x,y,z)=Sin\left(4\pi x+\frac{\pi}{8}\right)+Sin\left(4\pi y+\frac{\pi}{8}\right)+Sin\left(4\pi z+\frac{\pi}{8}\right)$ with $\phi(x,y,z)=(4\pi)^2u(x,y,z)$. First, we used the exact values for the entries of Q^{-1}. We found that the maximum absolute error is 0.2412 and the relative error computed for the cases with $|u(x,y,z)|>0.3$ is 0.4778. Note that the function $u(x,y,z)$ takes values in the interval [-3,3]. Next, we used the fuzzy set approximation to find that the difference is negligible; the maximum absolute error is 0.2699 and the maximum relative error is 0.5001.

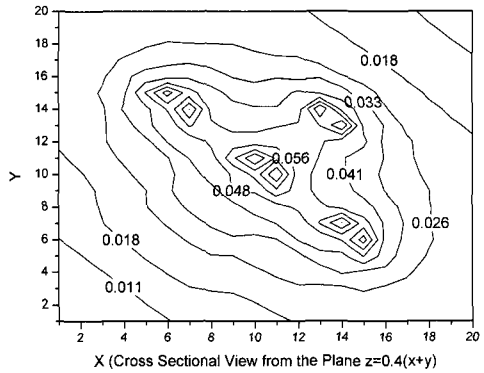

fig.10. Solution of the Poisson equation when $\phi(x,y,z)$ is sum of 5 delta function (Cross section with plane $z=0.4(x+y)$)

The result of another example calculation is shown in fig.9 where $\phi(x,y,z)$ is the sum of $\delta(x-x_i,y-y_i,z-z_i)$ for $i=1,2,\ldots,5$. The five points are located at $(0.3,0.3)$, $(0.3,0.7)$, $(0.5,0.5)$, $(0.7,0.3)$, and $(0.7,0.7)$ all at z=0.5 with the source intensity -320. Fig.9 shows the cross sectional view of the computed result at z=0.4 and fig.10 shows the cross sectional view on the plane $z=0.4(x+y)$.

4. Conclusion

We have shown that there is a very simple method to solve the Poisson equation when the Laplacian of the solution function $\phi(x, y, z)$ is periodic; the vector $(\phi(x_1, y_1, z_1), \phi(x_2, y_2, z_2), \cdots, \phi(x_N, y_N, z_N))$ is multiplied by the inverse of the convolution matrix related to the cubic spline interpolation. This method, however, is not applicable to general boundary value problems. It will be useful for a solution to the Navier-Stokes equation when periodic solutions are required.

One of the drawbacks of this method is that the matrix takes up a very large storage and requires a considerable amount of time for reading it in from the disk. As an example, we need a matrix of size 8000×8000 for a discretization of size $20 \times 20 \times 20$ nodes. When the 'E15.8' format is used for storing each of the entries, it will require a storage of 0.96GB. By using the fuzzy set method, however, we reduced this size to 128MB, about one eighth of the original storage. The I/O time for the initial read-in from disk also reduces by the same factor.

Acknowledgements

This work has been carried out under the nuclear research and development program supported by the Ministry of Science and Technology of Korea.

References

[1] Charles L. Fefferman, Existence and Smoothness of the Navier-Stokes Equation, Millenium Problems, *Clay Mathematical Institute* (2000) http://www.claymath.org/millenium.

[2] P. Labute and M. Santavy, "Numerical Solution of the Non-Linear Poisson-Boltzmann Equation", *J. Chemical Computing Group Inc.* (Spring 1999).

[3] Byung Soo Moon, et al., "Solution of Dirichlet Boundary Problem for the Poisson Equation Based on a Fuzzy System", *Computational Intelligent Systems for Applied Research, Proc. of 5th International FLINS Conference* (2002) 58-65.

[4] A. Shmilovici and O. Maimon, "On the Solution of Differential Equations with Fuzzy Spline Wavelets", *Fuzzy Sets and Systems*, 96(1999) 77-99.

[5] A. Shmilovici and O. Maimon, "The fuzzy rule-base solution of differential equations", *Information Sciences* 92 (1996) 233-254.

[6] H. Haidar, et al., "Characterizing the shape of anatomical structures with Poisson's equation", Proc. 7th Int. Conf. on Medical Image Computing and Computer Assisted Intervention (MIC-CAI 2004), Saint-Malo, France (Sept 26-29, 2004).

Byung Soo Moon
received his MS and Ph.D. in Mathematics from University of Illinois, 1974. From 1974 to 1978 he worked with Sargent & Lundy Engineers (Chicago). Since 1978 he is working at Korea Atomic Energy Research Institute. The areas of his research interests are Fuzzy systems, Pattern Recognition, Neural Nets.
Phone : 042-868-2980
Fax : 042-868-8916
E-mail : bsmoon@kaeri.re.kr

Jang Soo Lee
received his BS degree in Computer Science from Kyungpook National University in 1982, MS and PhD in Computer Science from KAIST in 1984, 2002. Since 1984. he is working at Korea Atomic Energy Research Institute. The major area of his interests is in software engineering.
Phone : 042-868-8325
E-mail : jslee@kaeri.re.kr

Dong-Young Lee
received the B.S. and M.S. degree in Electrical Engineering from Kyungpook National University in 1984 and 1987 and PhD from Choongnam National University in Feb. 2006. He has been a researcher of the Korea Atomic Energy Research Institute since 1987. His research interests include instrumentation and control system, reliability analysis, and safety assessment.
Phone : 042-868-2927
E-mail : dylee2@kaeri.re.kr

Kee-Choon Kwon
received his BS in Electronics Engineering from Kyungbook National Univ. in 1980, MS and PhD in Computer Science from KAIST in 1989, 1999. The areas of his interests are in applicatopn of artificial intelligence to nuclear power plant, software verification and validation.
Phone : 042-868-2926
Fax : 042-868-8916
E-mail : kckwon@kaeri.re.kr