International Journal of Fuzzy Logic and Intelligent Systems, vol. 3, no. 1, June 2003 pp. 13-17

A continuous solution of the heat equation based on a fuzzy system

Byung Soo Moon, Inkoo Hwang and Kee-Choon Kwon

Korea Atomic Energy Research Institute
P.O.Box 105, Yuseong, Daejeon 305-600, Korea

Abstract

A continuous solution of the Dirichlet boundary value problem for the heat equation $u_t=a2u_{xx}$ using a fuzzy system is described. We first apply the Crank-Nicolson method to obtain a discrete solution at the grid points for the heat equation. Then we find a continuous function to represent approximately the discrete values at the grid points in the form of a bicubic spline function $S(t,x)=\sum_{i,j}^{N,M} c_{ij}B_i(t)B_j(x)$ that can in turn be represented exactly by a fuzzy system. We show that the computed values at non-grid points using the bicubic spline function is much smaller than the ones obtained by linear interpolations of the values at the grid points. We also show that the fuzzy rule table in the fuzzy system representation of the bicubic spline function can be viewed as a gray scale image. Hence, the fuzzy rules provide a visual representation of the functions of two variables where the contours of different levels for the function are shown in different gray scale levels

Key words : Heat Equation; Fuzzy Systems; Dirichlet Boundary Value Problem; B-splines; Cubic Spline Function; Visual Representation of Functions; Partial Differential Equation

1. Introduction

In this paper, we consider the parabolic partial differential equation

$$\frac{\partial u}{\partial t}=a^2\frac{\partial^2 u}{\partial^2 x} \tag{1}$$

where u is a function of t and x, i.e. u=u(t,x) and u(0,x), u(t,0), u(t,1) are given as the boundary conditions. For simplicity, we restrict our attention to the region $[0,1]\times[0,1]$ and assume that the boundary conditions are given as the Dirichlet type only.

The above heat equation is often solved by the Crank-Nicolson method with accuracy of $O(h_t^2+h_x^2)$ where h_t and h_x are the step lengths in t and x coordinates respectively. The Crank-Nicolson method[1,2] is a finite difference method given by

$$r=a^2\frac{h_t}{h_x^2}$$

$$ru_{i-1,j+1}+2(1+r)u_{i,j+1}-ru_{i+1,j+1}= ru_{i-1,j}+2(1-r)u_{i,j}+ru_{i+1,j} \tag{2}$$

The above relation (1) can be solved simultaneously using the Dirichlet type boundary value conditions.

By applying the LU decomposition to the coefficient matrix, however, one can compute the solution by $O(h)$ operations instead of $O(h^2)$ where N is the number of subintervals.

Assume that we have the solution for some h_t and h_x and that we want to estimate the value of the function u(t,x) at a non-grid point. Here we must employ a linear interpolation using the nearest 4 neighboring grid points. As shown in Table 1, the maximum error at the non-grid points is much larger than the one at the grid points.

Table 1. Comparison of Max. Errors at Grid Points & at Non-Grid Points

a	Example 1 0.1125	Example 2 0.2250	Example 3 0.06366
Grid Points	0.003738	0.000918	0.000116
Non-Grid Pts	0.018825	0.004785	0.004785

Table 1 shows some of the example calculation results where the analytical solutions are of the form $u(t,x)=e^{-at}Sin(n\pi x)$ with $h_t=h_x=\frac{1}{16}$, $a=\frac{\sqrt{a}}{n\pi}$. They show the maximum errors of the solution by the Crank-Nicolson method at the grid points and the maximum errors of the interpolated values at 100×100 equally spaced points. From the table, one can see that the maximum error at the interpolated points are about 5 times larger than those at the grid points.

If we had a fuzzy system for the approximate solution to the partial differential equation, then the interpolation process would become much easier since we would then have a continuous solution. Also, if we use the bicubic B-splines as the input fuzzy sets, the interpolation error should be reduced. Now, consider the following fuzzy rules;

$$\text{If (t,x) is } B_i\times B_j\text{, then f is } \widetilde{f_{i,j}} \tag{3}$$

where B_i, B_j are input fuzzy sets whose supports are centered at t_i and x_j, and $\bar{f}_{ij}$'s are the solutions of (1) which are estimates of f_{ij}. There have been some studies[3,4] on the solution of differential equations based on fuzzy rules like the above. In our earlier work [5], we have also shown that a fuzzy system can be generated as an approximate solution to partial differential equations.

Using the same notations as in [5], if we apply $S(t,x)=\sum_{i,j}^{N,M} C_{ij}B_i(t)B_j(x)$ to the heat equation $u_t=a^2u_{xx}$ where $B_i(t)$, $B_j(x)$ are the cubic B-splines, then we get

$$\lambda=\frac{2h_t}{h_x^2}a^2$$

$$\begin{aligned}&(1+\lambda)C_{k-1,l-1}+(4-2\lambda)C_{k-1,l}+(1+\lambda)C_{k-1,l+1}\\&+4\lambda C_{k,l-1}-8\lambda C_{k,l}+(\lambda-1)C_{k+1,l-1}\\&-(2\lambda+4)C_{k+1,l}+(\lambda-1)C_{k+1,l+1}=0\end{aligned} \quad (4)$$

Solving (4) implicitly, we will obtain an approximate solution $C_{k,l}$ at the grid points (t_k,x_l) with accuracy $O(h_t^2+h_x^2)$. However, the solution requires $O(N^2)$ calculations, compared with $O(N)$ for the Crank- Nicolson method. Thus, we try to use the solution $u_{i,j}$ of (2) by the Crank-Nicolson method and compute the spline coefficients $C_{k,l}$ in (4) using the $u_{i,j}$'s. In the following, we describe how this can be done.

2. A bicubic spline representation of the solution by the Crank-Nicolson method

We will assume in the following that the solution $u(t,x)$ of the heat equation is a three times continuously differentiable function. This is acceptable not only because we deal with discrete solutions only while the analytical solution is unknown for pratical problems, but also because the heat equation describes a natural phenominon which must be a smooth function. Thus, the following theorem can be used to define a bicubic spline approximation to $u(t,x)$. Let $B_i(t)$, $B_j(x)$ be the cubic B-splines defined on equally spaced subintervals of $[0,1]$ with length h. Then the following is satisfied, a proof of which is found in [6].

Theorem 1. Let $f(t,x)$ be three times continuously differentiable on $[0,1]\times[0,1]$ and let $S(t,x)$ be the bicubic spline function $\sum_{i,j}^{N,M} f_{ij}B_i(t)B_j(x)$ with $f_{i,j}=f(t_i,x_j)$, then we have $f(t,x)-S(t,x)=O(h^2)$ for all $t,x\in[0,1]$.

Let $u_{i,j}$ be the solution by the Crank-Nicolson method and consider a nongrid point $(t_0,x_0)\in[0,1]\times[0,1]$. To find an approximate solution (t_0,x_0), we normally use the linear interpolation, i.e. we take the values $u_{i,j}$ at the four grid points nearest to (t,x) and compute a linear combination of them. This should yield an $O(h)$ error even though the Crank-Nicolson method converges with $O(h^2)$.

In order to improve the interpolation error, we will try an interpolation using the bicubic spline function. Note that if we had a solution $C_{k,l}$ for the system of equations (4), then we would have $S(t_0,x_0)-u(t_0,x_0) \;=O(h^2)$ by Theorem 1. Thus, we will try to compute an approximate solution $C_{k,l}$ from the solution $u_{i,j}$ of (2). Note that at a grid point (t_k,x_l), we have

$$\begin{aligned}S(t_k,x_l)=&\sum_{i,j=1}^{N,M} C_{i,j}B_i(t_k)B_j(x_l)\\&\frac{1}{36}\{C_{k-1,l-1}+4C_{k-1,l}+C_{k-1,l+1}\\&+4C_{k,l-1}+16C_{k,l}+4C_{k,l+1}\\&+C_{k+1,l-1}+4C_{k+1,l}+C_{k+1,l+1}\}\end{aligned} \quad (5)$$

where we have used the values of the cubic B-splines at grid points shown in Table 2.

Table 2. Nonzero Values of $S(t_k,x_l)\times 36$

	j=l−1	j=l	j=l+1
i=k−1	1	4	1
i=k	4	16	4
i=k+1	1	4	1

Now, if we require $S(t_k,x_l)=u_{k,l}$ for all k,l, then we have

$$\begin{aligned}&C_{k-1,l-1}+4C_{k-1,l}+C_{k-1,l+1}+\\&4C_{k,l-1}+16C_{k,l}+4C_{k,l+1}+\\&C_{k+1,l-1}+4C_{k+1,l}+C_{k+1,l+1}=36u_{k,l}\end{aligned}$$

In view of Theorem 1, we simply take $C_{k,0}=u_{k,0}$, $C_{k,N}=u_{k,N}$, $C_{0,N}=u_{0,N}$, and $C_{N,l}=u_{N,l}$, for $k,l=0,1,2,\cdots,N$. For interior points with $k,l\geq 2$, the above can be written as

$$\begin{aligned}16C_{k,l}\simeq&36u_{k,l}-C_{k-1,l-1}-4C_{k-1,l}-4C_{k,l+1}\\&-C_{k-1,l+1}-4C_{k,l-1}\\&-u_{k+1,l-1}-4u_{k+1,l}-u_{k+1,l+1}\end{aligned} \quad (6)$$

The above can be used iteratively, for $k=1,2,\cdots,N$ as k varies from $1,2,\cdots,N$. Using these $C_{i,j}$'s, we computed the maximum and average errors for the solution at 100×100 points in $[0,1]\times[0,1]$ interval and some of the results are shown in the following tables. Note that the error by the bicubic spline functions are smaller than the ones by the linear interpolations.

Table 3. Compaerison of the Maximum and Average Errors for $u(t,x)=e^{-0.5t}Sin(2\pi x)$

$\frac{1}{h}$	Linear		Spline	
	Max	Average	Max	Average
16	.098136	.008894	.128193	.005591
32	.006217	.002047	.002995	.000894
64	.001576	.000508	.000722	.000216
128	.000397	.000123	.000203	.000055

Table 4. Compaerison of the Maximum and Average Errors for $u(t,x)=e^{-t}Sin(\pi x)$

$\frac{1}{h}$	Linear		Spline	
	Max	Average	Max	Average
16	.099098	.002929	.102803	.005591
32	.001377	.000371	.000664	.000174
64	.000353	.000093	.000162	.000046
128	.000108	.000021	.000148	.000025

Table 5. Compaerison of the Maximum and Average Errors for $u(t,x)=e^{-t}Sin(\pi x+0.1)$

$\frac{1}{h}$	Linear		Spline	
	Max	Average	Max	Average
16	.116732	.013429	.108064	.002400
32	.067797	.012250	.000665	.000178
64	.078078	.012508	.000169	.000049
128	.083334	.012666	.000153	.000034

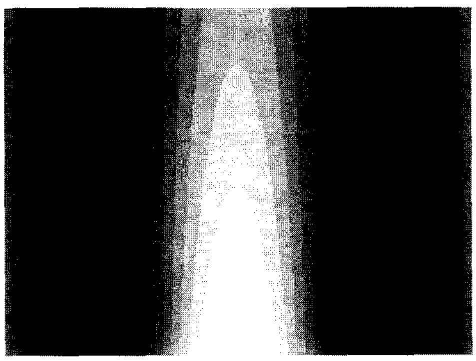

Fig. 1. Fuzzy Rules for $e^{-t}Sin(3\pi x)$

3. Fuzzy system and gray scale image representation of the solution

In this section, we consider a fuzzy system representation of the solution obtained in the previous section. It is shown[7] that a bicubic spline function of the form $S(t,x)=\sum_{i,j}^{N,M} C_{ij}B_i(t)B_j(x)$ can be represented exactly by a fuzzy system. The cubic B-splines $B_i(t)$ and $B_j(x)$ are used as input fuzzy sets to fuzzify the input variables t and x respectively. To generate the fuzzy combination rules, we first sort the coeeficient array $C_{i,j}$ in an ascending order so that the smallest comes first and the largest becomes the last. After deleting the duplicate ones, we define traingular fuzzy sets centered at these points and take them as output fuzzy sets with the ordinal number of the sorted array as the fuzzy set number.

Now, if the center area method is used as the defuzzification method, then the resulting fuzzy system will generate identical output as the spline function $S(t,x)$.

Table 6. Rule table for $u(t,x)=e^{-t}Sin(\pi x)$

1	12	22	27	28	2	22	12	1
1	10	20	25	26	25	20	10	1
1	9	18	23	24	23	18	9	1
1	7	16	20	21	20	16	7	1
1	6	14	18	19	18	14	6	1
1	5	11	16	17	16	11	5	1
1	4	10	14	15	14	10	4	1
1	3	8	12	13	12	8	3	1
1	2	7	10	11	10	7	2	1

Table 7. Rule table for $u(t,x)=e^{-t}Sin(2\pi x)$

22	35	39	35	22	6	2	6	22
22	36	40	35	20	5	1	6	22
22	33	38	32	21	8	3	9	22
22	31	37	30	21	11	4	12	22
22	28	34	27	21	14	7	15	22
22	26	32	25	21	16	9	17	22
22	24	29	23	21	18	13	19	22
22	26	31	24	21	17	10	18	22
22	22	22	22	22	22	22	22	22

Table 6 and 7 are the examples of the fuzzy rules generated by the above method. The first rows correspond to the function $u(0,x)$ and the first columns correspond to $u(t,0)$. To reduce the number of output fuzzy sets, we replaced some of the consecutive coefficients in the sorted array of $C_{i,j}$'s by the average of them when they are very close to each other, i.e. when $|C_{i,j}-C_{k,l}|\le 0.001$.

The center of supports for the 28 output fuzzy sets in Table 6 are as follows;

0.0000, 0.1408, 0.1547, 0.1805, 0.2001, 0.2316, 0.2594, 0.2858, 0.2976, 0.3357, 0.3709, 0.3827, 0.4042, 0.4318, 0.4757, 0.4831, 0.5229, 0.5546, 0.6053, 0.6205, 0.6707, 0.70712, 0.7185, 0.7777, 0.8059, 0.8723, 0.9239, 1.0000,

and the center of supports for the 40 output fuzzy sets in Table 7 are as follows.

-1.0352, -1.0000, -0.8749, -0.7819, -0.7502, -0.7093,
-0.6875, -0.6304, -0.6072, -0.5800, -0.5643, -0.5414,
-0.5309, -0.4960, -0.4763, -0.4381, -0.4238, -0.3893,
-0.3657, -0.0206, -0.0134, 0.0000, 0.3661, 0.3919,
0.4214, 0.4369, 0.4767, 0.5001, 0.5294, 0.5418,
0.5738, 0.6068, 0.6353, 0.6863, 0.7095, 0.7578,
0.7805, 0.8734, 1.0000, 1.0330

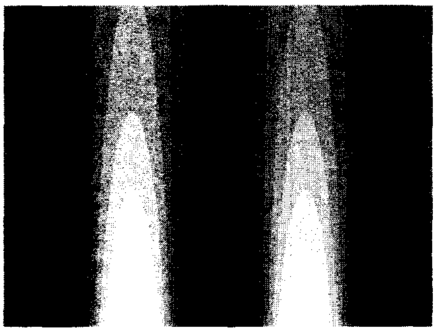

Fig. 2. Fuzzy Rules for $e^{-2t}Sin(5\pi x)$

In view of Theorem 1, the set of grid points (i,j)'s in the rule table with the same output fuzzy set number reflects a 'near' contour curve on the surface $z=u(t,x)$. Thus, if one considers the table of output fuzzy set numbers as a digitized gray scale image, then we have images shown in fig.1 through fig.3.

Fig.1 shows the fuzzy rule table for the solution of the heat equation where its analytical solution is $u(t,x)=e^{-t}Sin(3\pi x)$. We divided the interval $[0,1]\times[0,1]$ into 320×320 subintervals and computed the solution by the Crank-Nicolson method. We then computed the bicubic spline function by the method described in section 2. The fuzzy rule table is then computed allowing maximum of 256 fuzzy sets by the method described above. When the fuzzy rule table is drawn as a gray scale image, we obtain an image shown in fig.1. The contours in fig.1 represent points where the function values are approximately the same.

When the same process is applied to the case of $e^{-2t}Sin(5\pi x)$, we obtain the gray scale image shown in fig.2. Hence, the image shown in fig.2 represents the output fuzzy sets in the rule table of size 320×320 and the contour lines indicate the points where the function values are approximately the same. Note that this method can be used as a graphic representation of arbitrary functions of two varaibles, i.e. graphic representation of surfaces. To demonstrate this, we computed the fuzzy system representation of the function $z=f(r,\theta)=e^{-r}Sin(4\pi\theta)$ so that there are 320×320 fuzzy rules with a maximum of 256 output fuzzy sets. The resulting fuzzy rule table is drawn as fig.3.

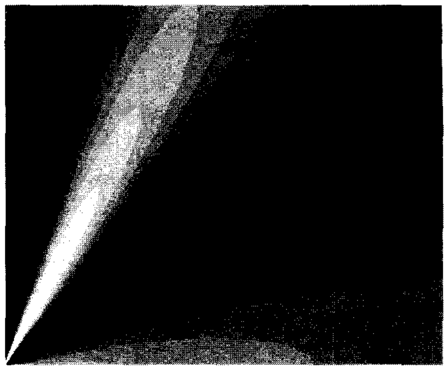

Fig. 3. Fuzzy Rules for $e^{-r}Sin(4\pi\theta)$

There is one remark for the size of fuzzy rule table for a general use of this gray scale representation of functions of two variables. Note that we only need the resolution that the contoue lines can be distinguished and as seen through above examples, the pixel size of 320×320 is about the largest that one would need. Note also that since we are using gray scale images, we will not need more than 256 output fuzzy sets.

4. Conclusion

We have shown that a bicubic spline function can be defined so that it is an approximate continuous representation of the solution of the heat equation. The fuzzy system representation of this function provides not only an efficient way to compute the solution at non-grid points but also it provides a much better approximation at those points.

We have also demonstrated that when the output fuzzy set numbers in the fuzzy rule table of the solution are considered as the gray scales of a digitized gray scale image, then the image shows the contour lines very clearly. Thus, we have shown that the fuzzy rule table of a function of two variables can be identified as a contour image of the function. The maximum size of the fuzzy rule table for the gray scale representation will be about 320×320. This size is quite large when an approximate representation of the function is the only concern for its representation.

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] R.L. Burden, J.D. Faires and A.C. Reynolds, *Numerical Analysis*, Prindle, Weber & Schmidt, pp. 508-519, 1978.

[2] C.F. Gerald and P.O. Wheatley, Applied *Numerical Analysis*, Addison-Wesley Pub. Co. pp. 544-558, 1989.

[3] A. Shmilovici and O. Maimon, "On the solution of differential equations with fuzzy spline wavelets", *Fuzzy Sets and Systems*, Vol. 96, pp. 77-99, 1999.
[4] A. Shmilovici and O. Maimon, "The fuzzy rule-base solution of differential equations", *Information Sciences*, Vol. 92, pp. 233-254, 1996.
[5] B.S. 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, pp. 58-65, 2002.
[6] B. S. Moon, et al., "A fuzzy system representation of functions of two variables and its application to gray scale images", *J. Korea Fuzzy Logic and Intelligent Systems Soc.*, Vol. 11 No.7 pp. 569- 573, 2001.
[7] Byung Soo Moon, "A practical algorithm for representing polynomials of two variables by fuzzy systems with accuracy $O(h^4)$", Fuzzy Sets and Systems Vol.119, No.2, 135-141, 2001.
[8] B. Kosko, "Fuzzy systems as universal approximators", *IEEE Trans. on Computers*, Vol. 43, No.11, pp. 1329-1333, 1994.

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 Networks.
Tel : 042-868-2980, Fax : 042-868-8916,
E-mail : bsmoon@kaeri.re.kr

Inkoo Hwang
received his BS and MS degree in Electric Engineering from Inha University 1986, 1990. From 1986. he is working at Korea AtomicEnergy Research Institute. The areas of his interests are in Signal Processing and sensor development.

Tel : 042-868-2925,
Fax : +82-42-868-8916
Email : ikhwang@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